

**DESIGN AND IMPLEMENTATION OF THE
CRYPTO-ASSISTANT: AN ECLIPSE PLUGIN
FOR USABLE PASSWORD-BASED COLUMN
LEVEL ENCRYPTION BASED ON HIBERNATE
AND JASYPT**

By

Ricardo Rodriguez Garcia

A Thesis Submitted in Partial Fulfilment

of the Requirements for the Degree of

Master of Science (MSc)

in

Computer Science

Faculty of Business and IT

University of Ontario Institute of Technology

Oshawa, Ontario, Canada

March 2013

Copyright © Ricardo Rodríguez Garcia, 2013

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Ricardo Rodríguez García.

CONTENTS

AUTHOR'S DECLARATION	2
CONTENTS	3
ABSTRACT	7
KEYWORDS	8
ACKNOWLEDGMENTS	9
LIST OF TABLES	10
LIST OF FIGURES	11
CHAPTER 1 - INTRODUCTION	12
1.1 PREAMBLE	12
1.2 RESEARCH OBJECTIVES	13
1.3 RESEARCH QUESTIONS	14
1.4 ORGANIZATION OF THESIS	15
2 CHAPTER 2 - LITERATURE REVIEW	17
2.1 INTRODUCTION	17
2.2 COMPUTER SECURITY	17
2.2.1 SECURITY	17
2.2.2 COMPUTER SECURITY	17
2.2.3 SECURITY GOALS	18
2.2.4 SECURITY ASSURANCE AND RISK MANAGEMENT	20
2.3 WHY DO DEVELOPERS MAKE SECURITY ERRORS?	23
2.3.1 SOFTWARE ERRORS.	27
2.3.2 SECURITY ERRORS	29
2.4 TOOLS FOR SECURITY	31
2.4.1 SOFTWARE TOOLS FOR SECURITY	32

2.4.2	SECURITY TOOLS AND INSECURITY	35
2.4.3	SECURITY TOOLS AND USABILITY	36
2.4.3.1	Security Usability Fundamentals	37
2.4.3.2	Making Security Usable	38
2.4.3.3	How Users Make Decisions	41
2.4.3.4	How Users Really Make Decisions	41
2.5	SUMMARY	42
3	<u>CHAPTER 3 – CRYPTO-ASSISTANT</u>	44
3.1	INTRODUCTION	44
3.1.1	MOTIVATION	44
3.2	PROBLEM DEFINITION	45
3.2.1	INITIAL HYPOTHESIS	48
3.2.2	COMMUNICATION STRATEGY	49
3.2.3	WARNINGS AND USER REACTIONS	50
3.2.4	COMPONENT SELECTION	52
3.2.5	DESIGN CONSIDERATIONS	53
3.2.5.1	Recommended Encryption algorithms	55
3.2.5.2	Key management	57
3.2.5.3	Database encryption strategies	58
3.2.5.3.1	Inside the DBMS.	58
3.2.5.3.2	Off-loading encryption outside of the DBMS.	59
3.2.5.3.3	Application level encryption.	59
3.3	DEVELOPMENT	60
3.3.1	ARCHITECTURE	60
3.3.1.1	Eclipse	61
3.3.1.2	Hibernate & Hibernate Tools	62
3.3.1.3	Jasypt Java Simplified Encryption Library	63
3.4	USAGE	64
3.5	INSTALLATION	69
3.6	USABILITY EVALUATION	72

3.6.1	COGNITIVE WALK-THROUGH	72
3.7	SUMMARY	73
4	CHAPTER 4 - PILOT USER STUDY	75
4.1	INTRODUCTION	75
4.2	PURPOSE/BACKGROUND INFORMATION	75
4.3	EXPERIMENTAL DESIGN	76
4.3.1	RECRUITMENT PROCESS	77
4.3.1.1	Initial contact	77
4.3.1.2	Eligibility	77
4.3.2	THE STUDY	78
4.3.2.1	Workshop	78
4.3.2.2	Experiment	78
4.3.3	DATA COLLECTION AND EVALUATION	79
4.3.4	ETHICS	80
4.4	RESULTS	80
4.5	ANALYSIS	82
4.5.1	LESSONS LEARNED AND EXPERIMENTAL LIMITATIONS	83
4.5.2	IMPLICATIONS OF THE RESULTS OBTAINED FROM THE PILOT USER STUDY	85
4.6	SUMMARY	87
5	CHAPTER 5 - DISCUSSION AND CONCLUSION	88
5.1	INTRODUCTION	88
5.2	DISCUSSION	88
5.3	FUTURE RESEARCH	90
5.4	CONCLUSION	90
6	BIBLIOGRAPHY	92
7	APPENDIX A - DATA COLLECTION	99
7.1	SCREENING QUESTIONNAIRE	99
7.2	EXIT QUESTIONNAIRE	109

7.3	PILOT STUDY LOGS	114
8	<u>APPENDIX B - EXPERIMENT MATERIAL</u>	119
8.1	DATA OPT-OUT & REMOVAL FORM	119
8.2	CONSENT FORM	120
8.3	OPT OUT SURVEY	122
8.4	PRE-SCREENING CONSENT FORM	123
8.5	WORKSHOP SLIDES	125
8.6	EXPERIMENT SLIDES	138
8.7	EMAIL CORRESPONDENCE	142
8.8	RECRUITMENT POSTER	150
8.9	REFERENCE CHEAT SHEET	151
9	<u>APPENDIX C - RESEARCH ETHICS BOARD DOCUMENTATION</u>	152
9.1	APPLICATION FOR ETHICAL REVIEW OF RESEARCH INVOLVING HUMAN PARTICIPANTS	153
9.2	CHANGE RENEWAL REQUEST	181
9.3	CHANGE REQUEST APPROVAL	184
10	<u>APPENDIX D CODE DOCUMENTATION</u>	185

ABSTRACT

The lack of encryption of data at rest or in motion is one of the top 10 database vulnerabilities according to team SHATTER [72]. In the quest to improve the security landscape, we identify an opportunity area: two tools Hibernate and Jasypt that work together to provide password-based database encryption. The goal is to encourage developers to think about security and incorporate security related tasks early in the development process through the improvement of their programming system or integrated development environment (IDE). To this end, we modified the Hibernate Tools plugin for the popular Eclipse IDE, to integrate it with Hibernate and Jasypt with the purpose of mitigating the impact of the lack of security knowledge and training. We call this prototype the Crypto-Assistant. We designed an experiment to simulate a situation where the developers had to deal with time constraints, functional requirements, and lack of familiarity with the technology and the code they are modifying. We provide a report on the observations drawn from this preliminary evaluation. We anticipate that, in the near future, the prototype will be released to the public domain and encourage IDE developers to create more tools like Crypto-Assistant to help developers create more secure applications.

KEYWORDS

Security, usability, software tool, action research, encryption, qualitative research, secure software development.

ACKNOWLEDGMENTS

I would like to thank my supervisors Miguel Vargas Martin and Julie Thorpe for supporting me during the realization of this thesis work.

We thank the Natural Sciences and Engineering Research Council of Canada and the University of Ontario Institute of Technology for the financial support of this work.

Thanks to my family and friends who have always given me their support and without which this work would not have been possible.

LIST OF TABLES

Table 1 Actions in programming activity	28
---	----

LIST OF FIGURES

Figure 2-1 Eclipse IDE one of the most popular programming systems.	25
Figure 2-2 Faults are propagated into executable code and become runtime failures.	26
Figure 2-3 Windows Vista user account control dialog.	40
Figure 3-1 How encryption works	45
Figure 3-2 Use of Jasypt and Hibernate for encryption without Crypto-Assistant support.	47
Figure 3-3 Use of Jasypt and Hibernate for encryption with Crypto-Assistant support.	48
Figure 3-5 Crypto-Assistant - High level architecture.	61
Figure 3-6 Property Encryption page added by the Crypto-Assistant.	66
Figure 3-7 Mapping files wizard, preview screen showing the use of Jasypt Hibernate types for encryption.	68
Figure 3-8 Install new software dialog.	69
Figure 3-9 Add repository dialog.	70
Figure 3-10 Installing Crypto-Assistant.	71
Figure 3-11 Restart eclipse dialog.	71

CHAPTER 1 – INTRODUCTION

1.1 Preamble

Information and computer security have acquired relevance lately mainly due in part to attacks against high profile businesses such as Sony, Apple, and Amazon. Tools have reduced the difficulty and skills necessary to launch an attack. On the other hand the difficulty for developers to build secure software has increased with more and more vulnerabilities discovered every day. “Security is a chain; it's only as secure as the weakest link.” [58]. An attacker could compromise the whole system by exploiting the weakest link. In contrast, developers’ main goals imply meeting functionality and time to market requirements; security is a secondary goal [82] that might be desired, but not required [88] depending on the criticality and perceived risk of the application developed.

This thesis work aims to provide some insight to the question: how can we help developers to produce secure software? Based on the premise that the actions of developers are influenced by the tools they use, by providing them the tools that focus on security, we may be able to change developers’ perspectives and behaviours with the aim of increasing the security of information systems.

To make a better use of resources, we focused on a particular problem: the lack of encryption of sensitive information while using Hibernate. Hibernate Object/Relational Mapping (ORM) tool, facilitates the storage and retrieval of Java objects. One of the easiest ways to achieve the encryption of sensitive information is to use custom Hibernate data types provided by the Jasypt (Java simplified encryption) library. A more detailed description of these tools is provided in Chapter 3. Nevertheless, the process to encrypt sensitive data with Hibernate and Jasypt is still fairly complex, not particularly intuitive, and prone to errors. This was the main motivation for us to build a prototype tool to help developers in this task. The prototype consists of a series of modifications to the Hibernate Tools plugin for Eclipse. Hibernate Tools include an Eclipse plugin with the

aim to increase the usability of Hibernate. When we refer to usability we mean “the ease of use and learnability of a human-made object”. The usability increase is achieved through a set of editors and wizards that reduce the learning curve associated with its use. Usability is one aspect that contributes significantly to the use, or not, of security tools; the prototype focuses on improving this property. The integration of Hibernate Tools and Jasypt, reduces the complexity of using encryption to protect sensitive information. The prototype used a security warning as a mechanism to communicate the risk of compromising sensitive data due to a lack of encryption. This was done to encourage developers to classify and protect sensitive data at the early stages of development; when is cheaper and easier to fix any possible issues derived from the lack of security considerations. The ultimate goal was to make the process as easy and intuitive as possible, and reduce the learning curve and potential number of errors in which developers could incur.

The evaluation of the prototype included a user study to simulate the usual conditions that a developer has to deal with: little familiarity with the application code, vague requirements, and time constraints. We designed a programming task that involved the use of the prototype and asked the participants to perform the migration of a web application data layer from JDBC (Java database connectivity) API (Application programming interface) to Hibernate and at the same time improve the application in a given amount of time. A full description of the study is given in Chapter 4. We deliberately hid the fact that we were focusing on security and observed if the changes introduced in the programming system influenced the software artefacts produced. We collected logs, artefacts and questionnaires from participants to gain an insight on their perception of the tools used and tasks performed to understand their behaviour.

1.2 Research objectives

This research work has several objectives:

1. The main goal of this research is to create a tool to help developers build more secure and reliable applications. To achieve this goal, we developed a prototype based on the code of the Hibernate Tools plugin for Eclipse. By choosing the popular

Eclipse platform, we aim to maximize the audience that could benefit from the work done during the realization of this research.

2. The second goal is to help developers in the application of the “Build Security In” [9] concept. This concept indicates that to develop secure applications, security must be present in every phase of the software development life cycle, from the inception through its development and even after during the maintenance phase; it would be much easier to apply this concept with the help of integrated development tools and processes that promote awareness and train developers to act in response to security risks. A well-integrated programming environment that promotes a continuous process of improvement and focus on security principles and practices will result in the ongoing production of more dependable, trustworthy and survivable software systems.

3. Another goal is to support security researchers and tool developers by sharing the experience gained while developing the Crypto-Assistant prototype. It is anticipated that this research could be used as a reference by other security researchers who wish to improve or create tools for security that encourage developers to integrate and carry out security related activities during the development process.

1.3 Research questions

To achieve the goal of producing a tool to help in the development of secure software, the research was motivated by the following research questions:

1. What research has been done to answer what a secure system is?
2. What research has been done to determine the cause of software security errors?
3. What kinds of tools have been created to help developers to produce secure systems?
4. What problems have these tools solved, and which security errors remain that would benefit from a tool?
5. What kind of tool could be developed to aid developers in the encrypting of sensitive data?
6. How to evaluate the usability and security of Crypto-Assistant, a tool to help developers encrypt sensitive data?

7. How well did the Crypto-Assistant work in term of helping developers encrypt their data?

To answer these questions, first it was necessary to define what security is and how to define a secure system. To understand the causes of security errors we begin by defining error to understand how security errors are different from common errors and examine what factors contribute to the introduction of security errors in software. Several examples are presented to show the diversity of ideas and nature of security tools.

One of the main problems of security tools is that they are complex and difficult to use by someone without considerable knowledge about the tool and security. The lack of usability prevents users from benefiting from the security that tools intend to provide. There is an intrinsic relationship between security and usability. A security mechanism can become detrimental to security if it is hard to use and a system that is too usable has to make sacrifices in terms of security. A door is a good example; for usability we can keep it unlocked for easy access, but, if we require security then we lock it restricting its use to only users with the correct key, if we require more security we can add locks or chains for the door but we would require more keys and effort to use it; if we need to use it very frequently, then one might just leave it open, disabling the security mechanism in exchange for better usability. The last two questions regarding Crypto-Assistant are left to be addressed in the Chapters 3 and 4.

1.4 Organization of thesis

The organization of this thesis is as follows. We begin by introducing the concept of security and what a secure system is, then exploring some of the causes of security flaws in software. First we learn about errors in general, following up with code errors and security errors and the possible causes of them.

A general overview of tools for software security is provided to show the diversity and nature of these tools and as a reference point to compare the Crypto-Assistant prototype presented.

Then we will present the Crypto-Assistant prototype starting by the problem it aims to address, along with a usability evaluation in terms of learnability. Chapter 4 is dedicated

to describing the prototype evaluation with a small group of three users. Its content ranges from the presentation of the initial hypothesis, to the experimental design and the rationale behind the design decisions made. Finally, we present the observations from that test and discuss the possible causes that lead to them and the implications of the results.

We conclude this work, presenting some ideas for future research, with the hope that other researchers will continue studying Crypto-Assistant and possibly develop other tools to help software developers create secure code.

2 CHAPTER 2 – LITERATURE REVIEW

2.1 Introduction

In this chapter, the information collected as a background for the prototype developed is presented. The next sections show an overview of the different ideas that contributed to the realization of this research work. This chapter begins with the research questions that led to uncovering the theory, followed by a discussion about the formulation of the hypothesis that led to the design of the prototype evaluated in this research.

2.2 Computer security

In this section, we introduce the concept of security and some definitions of computer security to help the reader understand the content of this chapter. Along this text we will be using computer security and cyber security indistinctively.

2.2.1 Security

Security is defined as the degree of protection to safeguard assets against danger, damage, loss and crime. In military terms, the Department of Defence of the United States of America defines security as “A condition that results from the establishment and maintenance of protective measures that ensure a state of inviolability from hostile acts or influences.” [62]. As a form of protection, we can define security as the structures and processes that improve security as a condition.

2.2.2 Computer security

The NIST (National Institute of Standards and Technology) defines computer security [42] as “Measures and controls that ensure confidentiality, integrity, and availability of

information system assets including hardware, software, firmware, and information being processed, stored, and communicated.” From a business point of view the ISO(International Organization for Standardization)/IEC (International Electrotechnical Commission 27002 standard [39] defines information security as “the protection of information from a wide range of threats in order to ensure business continuity, minimize business risk, and maximize return on investments and business opportunities”.

Bishop [6] analyses different components that are necessary to attain computer security:

1. Security requirements: they refer to the goals of security; what do we want to protect? Against what we want to protect them?
2. Security policies: Requirements dictate that some actions and system states be allowed and others disallowed. A security policy is a specific statement of what is and what is not allowed.
3. Security mechanisms: Enforce the policies; their goal is to ensure that the system never enters a disallowed state. The mechanisms may be technical or operational (sometimes called procedural).
4. Security assurances: The problem of measuring how well requirements conform to needs, policy conforms to requirements, and mechanisms implement the policy fall in the realm of assurance.

When we talk about computer security, what we are trying to protect usually is the information stored in a computer; after all, computers are only tools that help us to process and access information. We protect this information by preserving desired qualities while avoiding or mitigating undesired ones.

2.2.3 Security goals

There is a general consensus that, the main group of desirable qualities or objectives of information security are confidentiality, integrity and availability.

This is known as the security CIA (Confidentiality, Integrity, Availability) triad [76][12][75]; however, there are some works that extend these security objectives and

add others like accountability and assurance; the following concepts are extracted from Gary Stoneburner work [68].

Confidentiality is the requirement that private or confidential information should not be disclosed to unauthorized individuals. Confidentiality protection applies to data in storage, during processing, and while in transit. It comes from the need to use and store sensitive information, for example, defence plans, personal and financial data, trade secrets or intellectual property. When sensitive information is handled, there is a need to restrict access to those resources only to individuals that have been granted appropriate permissions and have genuine business reason to access and use that information. Confidentiality also applies to the existence of data since revealing the mere existence may reveal information that must be protected. Access control mechanisms support confidentiality by providing the means to achieve it; one of such mechanism is cryptography [5], which scrambles data to make it unusable without the appropriate encryption key; this adds another protection layer to the equation because an attacker would be required to have access to the data and the encryption key to be able to decrypt it.

Integrity refers to the ability to ensure that data is an accurate and unchanged representation of the original information. Its goal is that of preventing improper or unauthorized change. Integrity has two facets: data integrity (the content of the information), and origin integrity (the source of the data, often called authentication). The source of the information is important for users to trust the accuracy and credibility of certain data. A mechanism to ensure integrity falls into two classes: prevention mechanisms, and detection mechanisms. Prevention mechanisms try to maintain the integrity of the data by blocking any unauthorized attempts to change the data or any attempts to change the data in unauthorized ways. The former occurs when a user tries to change data which she has no authority to change. The latter occurs when a user authorized to make certain changes in the data tries to change the data in other ways that are not authorized. Detection mechanisms do not try to prevent the modification of data but instead to identify if it is trustworthy, making sure that it meets certain conditions.

These mechanisms can report the cause of the integrity violation or simply report that there is an integrity problem.

Availability refers to the ability to use the resources when desired. This means that the resources are available when they are needed. The most available systems are accessible at all times and have safeguards against power outages, natural disasters, hardware failures and system upgrades. Attempts to block availability, called denial of service attacks, can be very difficult to detect, because the analyst must determine if the unusual access patterns are attributable to deliberate manipulation of resources or the environment. A deliberate attempt to make a resource unavailable may simply look like, or be, an atypical event. In some environments, it may not even appear atypical.

Accountability is the requirement that actions of an entity may be traced uniquely to that entity. Accountability is a fundamental requirement of security policies because directly supports non-repudiation, deterrence, fault isolation, intrusion detection and prevention, and after-action recovery and legal action.

Assurance (that the other four objectives have been adequately met)

We need assurance to be confident that the security instruments, both technical and operational, work as intended to protect the system and the information it processes. The other four security objectives (integrity, availability, confidentiality, and accountability) have been adequately met by a specific implementation when:

Required functionality is present, and correctly implemented.

There is sufficient protection against unintentional errors by users or software.

There is sufficient resistance to intentional penetration or bypass.

Assurance is essential; without it, the other objectives are not met. No methodology can provide absolute assurance that a system is secure, but different methods provide different levels of confidence. The methods for evaluating assurance depend not only on the system, but also on the environment in which the evaluation occurs and on the process used to specify, design, implement, and test the system.

2.2.4 Security assurance and risk management

Risk management is the process of identifying, assessing, and taking steps to eliminate or reduce the risk to an acceptable level [69]. Risk management is an ongoing process, risk assessments should be conducted throughout the system development life cycle, from pre-system acquisition, through system manufacturing and deployment, and during its operations and support.

Before claiming that a system is secure, it is important to identify the threats to the system in question. Enumerating the threats to a system helps system architects develop realistic and meaningful security requirements [51]. Systems security engineering involves identifying security risks, requirements and prevention or recovery strategies. Without identifying threats, it is impossible to provide assurance for the system and justify security measures taken. Proper identification of threats and appropriate selection of countermeasures reduces the vulnerability of the system.

Threat modelling uses the perspective of an aggressor to help a designer to anticipate the goals of an attacker and answers questions about *what* the system is designed to protect, and from *whom*. Threat modelling consists of three high-level steps:

1. Characterizing the system.
2. Identifying assets and access points.
3. Identifying threats.

Once threats are identified, it is necessary to create a threat profile of the system, describing all the potential attacks that need to be mitigated against or accepted as low risk. A risk assessment is performed to map each threat either into a mitigation mechanism or an assumption that it is not worth worrying about it. At this point, the security requirements for the system can be defined.

The threats selected for mitigation must be addressed by some countermeasure. Security requirements are driven by security threats. Security requirements can adopt a negative form and state what must not be allowed to happen.

Assurance gives the user confidence that a system works as intended, without flaws or surprises, even in the presence of malice. According to Snow [64], assurances are

confidence-building activities whose goal is to demonstrate that: “The system's security policy is internally consistent and reflects the requirements of the organization,

1. There are sufficient security functions to support the security policy,
2. The system functions to meet a desired set of properties and *only* those properties,
3. The functions are implemented correctly, and
4. The assurances *hold up* through the manufacturing, delivery and life cycle of the system.” [64]

Assurance is provided through structured design, processes, documentation, and testing, with greater assurance provided by more processes, documentation, and testing.

Securing systems involve trade-offs; finding an ideal balance is a challenge. It is often impossible to mitigate every threat, and even if this could be done, it would almost certainly take place at the cost of decreased usability. It is important to keep in mind that the cost of security should not exceed the cost of the expected risk.

To provide assurance about the security of a system, once the system has been analysed, threats identified and safeguards put in place, the effectiveness of the safeguards must be tested. The goal is to evaluate how well they perform under stress or when used in ways beyond the normal specification. Security acceptance testing not only exercises the product for its expected behaviour given the expected environment and input sequences, but also tests the product with swings in the environment outside the specified bounds and with improper inputs that do not match the interface specification. Tests must include proper inputs, but in an improper sequence. One must anticipate malicious behaviour and design to counter it, and then test the countermeasures for effectiveness. The expectation is that the product will behave safely, even if not properly, under any of these stresses. If it does not, it should be redesigned and the cycle repeated.

Security testing is the process of determining how effectively an entity being assessed meets specific security objectives. Three types of assessment methods can be used to accomplish this—testing, examination, and interviewing.

Testing is the process of exercising one or more assessment objects under specified conditions to compare actual and expected behaviours.

Examination is the process of checking, inspecting, reviewing, observing, studying, or analysing one or more entities to facilitate understanding, achieve clarification, or obtain evidence.

Interviewing is the process of conducting discussions with individuals or groups within an organization to facilitate understanding, achieve clarification, or identify the location of evidence. Assessment results are used to support the determination of security control effectiveness over time.

Despite all the efforts of security researchers and practitioners, it is impossible to guarantee 100% security. However, it is possible to achieve a 100% risk acceptance. Failure to take these elements into consideration can lead to a situation where no risk is judged acceptable, and thus no acceptable system can be designed [51].

2.3 Why do developers make security errors?

All humans are fallible; to make mistakes is part of our nature. The mistakes we make are reflected in the products or artefacts produced by the actions we carry out. In software development, this is reflected in the quality and bug density in the applications produced. To better understand and clarify this issue, we will introduce some terms taken from [36] and [45] :

Mistake – a human action that produces an incorrect result.

Fault [or Defect] – an incorrect step, process, or data definition in a program.

Failure – the inability of a system or component to perform its required function within the specified performance requirement.

Error – the difference between a computed, observed or measured value or condition and the true, specified, or theoretically correct value or condition.

Specification – a document that specifies in a complete, precise, verifiable manner, the requirements, design, behaviour, or other characteristic of a system or component, and often the procedures for determining whether these provisions have been satisfied

Correctness – the degree to which a system or component is free from faults in its specification, design, and implementation.

The degree to which software, documentation, or other items meet specified requirements, and user needs as well as expectations, whether specified or not

Programming system – is a set of components such as “editors, debuggers, compilers, and documentation, each with (1) a user interface; (2) some set of information, such as program code or runtime data, which the programmer views and manipulates via the user interface; and (3) a notation in which the information is represented” [45]. Figure 2-1 shows the user interface of Eclipse IDE one of the most popular programming systems.

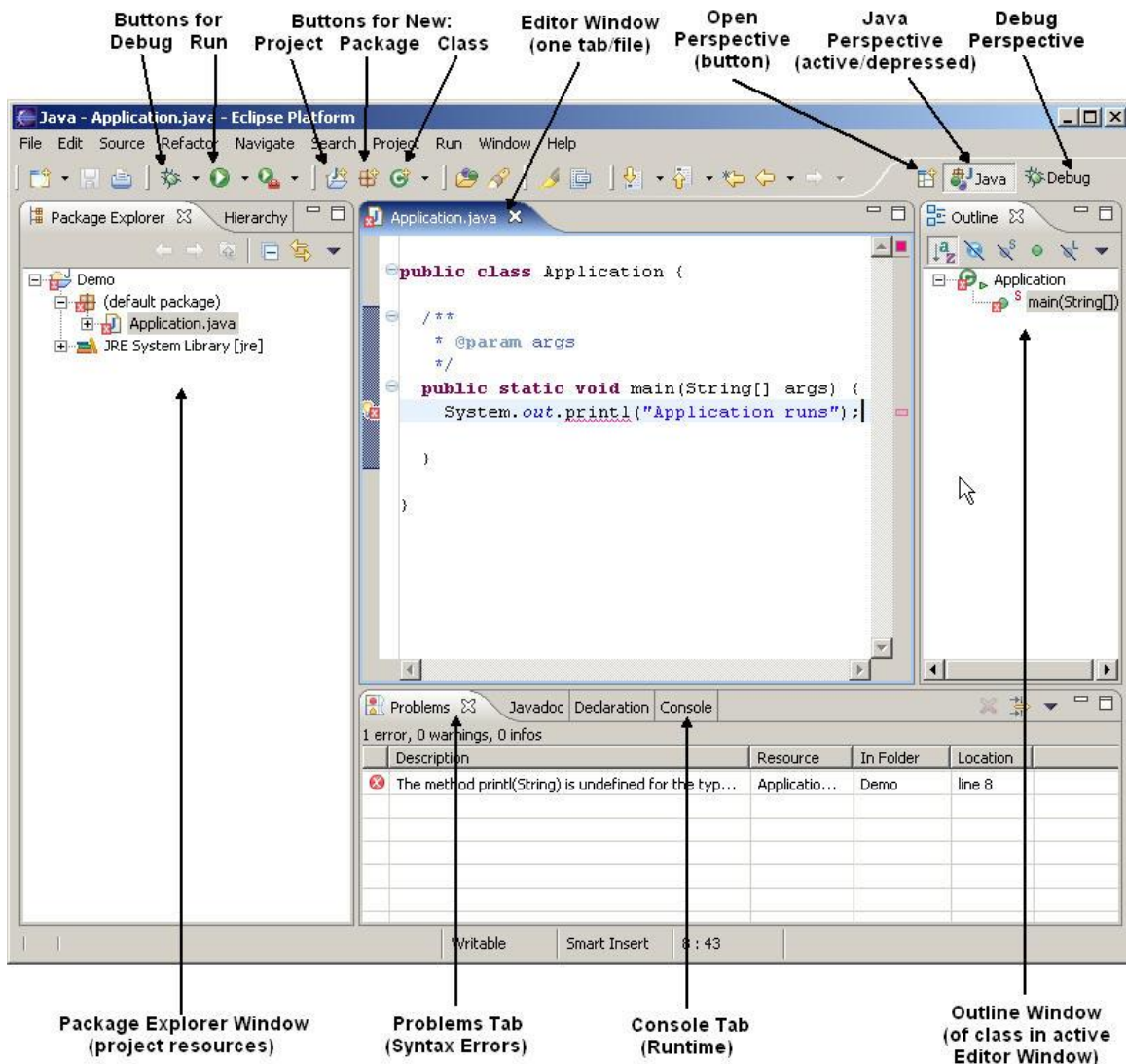


Figure 2-1 Eclipse IDE one of the most popular programming systems.

Having these basic concepts defined we can explain how errors are introduced into a system; according to Williams [84] the progression of a software failure can be explained as follows:

First a *mistake* is made and becomes a *fault* (or defect) in a software artefact such as the specification, design or code; when this happens in code, we call it a *software error* [45]. A *software error* is a *fault* that propagates as a *defect* in the executable code. When a defective piece of code is executed this leads to a *runtime fault*, in other words a machine

state that may cause a visible *failure*; when the *runtime fault* becomes visible a *failure* is perceived. However, *software errors* do not always translate into *runtime faults* and *runtime faults* not always cause *failures*, if this is the case then we say that *faults* remain latent. Figure 2-2 shows a graphic representation of this progression.

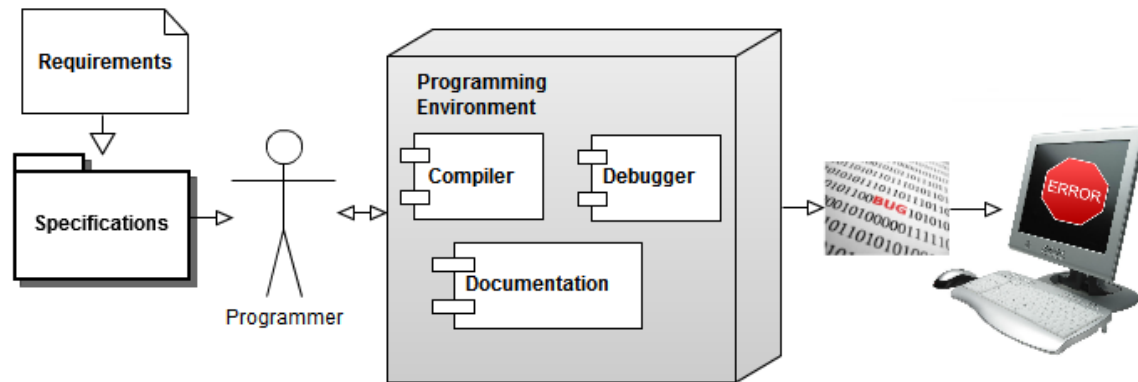


Figure 2-2 Faults are propagated into executable code and become runtime failures.

Testing is part of the software development process; it helps developers to reveal *failures*[65]. However, to solve *failures*, it is necessary that the *faults* that led to the failure are found and corrected. The process of finding the cause of a *failure* is time-consuming and unpredictable; this adds uncertainty and delays to the development process that can reflect in monetary losses. Furthermore, even when the root cause of a *failure* is detected, the cost associated to fix it may exceed the risk associated to deal with the *failure*. These faults can remain latent in the product through a follow-on release or perhaps forever.

Fixing a *fault*, once it is detected, may involve different activities such as redesign and re-code. The stage of the development life cycle in which a *failure* is detected has a direct effect on the cost of fixing them; the earlier a defect is detected the cheaper it is to fix it. To illustrate this, we can compare building software to building a house; it is easier to correct a defect in the blueprints before the house has been built, than once the construction is completed. When errors remain undetected until the software is released

the cost can be very expensive, costing companies and users billions of dollars in repairs [70], lawsuits and lost sales. The costs associated with fixing an error are not only monetary. Imagine what happens when a *failure* is detected once the software is in production. The development team might have to send someone to work on-site with the client in order to find the cause of the *failure*. It is understandable that the client will not be extremely happy if this occurs. Coding errors can also cause physical damage and, in the worst case, fatalities [73].

2.3.1 Software errors.

The process of building a software system is a complex one that involves several elements including people, processes, and technology. The construction of a system begins with a set of requirements that need to be fulfilled by the final product; those requirements become specifications that a programmer translates into code using a programming system. Each one of these phases is prone to certain types of errors and also provides certain defences. The specifications act as a high level defence mechanism against errors, but they may be incomplete, defective, or ambiguous and predispose programmers to misunderstand the system true requirements.

Programmers can use their knowledge, attention, and expertise to defend against software errors. However, programmers may have deficiencies in their defences that may turn into code errors. Another element of the process is the programming system consisting of several components such as compilers, libraries, languages, and environments. Each component has a set of defences against software errors. For example, compilers defend against syntax errors by showing warnings to programmers, but they also have latent usability issues like displaying confusing error messages, which may misguide programmers to incorrectly diagnose the cause of an error. Finally, the code may have latent errors that can eventually lead to a program's runtime *failure*.

According to Ko and Myers [45] there are four main aspects that contribute to software errors:

Surface qualities: The particular syntactic or notational anomalies that make a code fragment incorrect. Examples of this type of error are syntax oversights, trivial typos and mechanical errors simply describe unintended text in a program; erroneous assignment statements, and array references. Surface qualities of software errors are significantly influenced by the language syntax. The high frequency of these errors suggests that language syntax can be a cause of software errors on its own.

Cognitive: Programmers' lack of knowledge about language syntax, control constructs, data types, and other programming concepts may lead programmers to a situation that cause errors such as inventing language syntax, data-type inconsistencies [67], and misplaced or malformed statements. Lack of attention could result in a programmer forgetting the inclusion of a function or use the wrong variable or operand. Such problems can be attributed to distractions or a lack of vigilance. In the same category, the lack of knowledge and experience can turn into strategic issues, referring to problems like unforeseen code interactions or poorly designed algorithms.

Programming activity: Another aspect of software errors is the programming activity in which the cause of the software error occurred. For example, the code may be free of typos and syntax errors, but the algorithm implemented might be incorrect; this could be attributed to the programmer's invalid or inadequate interpretation of the requirements or problem at some stage in the specification activity.

Action type: There are different actions that can be performed during a programming activity like creation, modification, design, exploration, and understanding. Each one of these actions is prone to different kinds of errors.

Table 1 Actions in programming activity

Creating	Writing code, or creating a design and requirement specifications
Modifying	Modifying code or changing

	specifications
Designing	Considering various software architectures, data types, algorithms, etc.
Exploring	Searching for code, documentation, runtime data
Understanding	Comprehending a specification, an algorithm, a comment, runtime behaviour, etc.

All these aspects help us to understand a little more regarding what is behind a software error; however, they are limited only to a causal relationship. The interactions between the programmer and the programming environment that lead to the errors may have a higher level of complexity. For example, what appears to be an incorrect algorithm implementation may have its origin in the specifications due to lack of clarity or detail.

If we want to help developers to build better software, then we need to address the root cause of an error, and as we have seen these are very diverse and might require an entirely different strategy to address each one of them.

2.3.2 Security errors

Security errors are a peculiar type of error, they are inherently latent because their effects are not immediate and might not reflect on an evident *failure* contrary to what happens with active errors; the effects of active errors are more immediate such as when a typo prevents the program to compile or produces an incorrect output. Reason in [55] defines latent errors as issues that remain dormant for a long time and whose consequences are not evident until certain conditions are met. Security errors fall under the latter category. Many times the problem with this type of error is not always an incorrect implementation of a security feature but the total omission of security

considerations. This may be attributed to several factors, such as the lack of a formal process that integrates security tasks through the different stages of the software development life cycle, lack of security training, absence of security policies and lack of experience and awareness about possible threats.

For security errors, the first line of defence comes at the requirements engineering and design phase. Using threat modelling, security requirements are collected as functional and non-functional requirements that specify different aspects of how the system must behave. The security requirements may be formalized in a security policy. A security policy is a definition of what it means to be secure for a system, organization or other entity. For systems, the security policy addresses constraints on functions and data flow among them, constraints on access by external systems and adversaries including programs and access to data by people.

In the absence of security requirements, policies or a secure design, there are two additional lines of defense where threats can be mitigated: the programmer, and the programming environment. Ultimately security decisions relay at the programmers' discretion. This might be the case when the programmer also plays the role of architect and designer and this is common for small projects. In order for programmers, to defend an application against security errors they need to be aware of threats, attacks and countermeasures [49]. This requires awareness, education, training, and skills. Awareness by itself is not enough; even if developers are aware of the risks, they might not be able to identify instances of weaknesses or to implement the correct solutions. The industry is now aware of the importance of security, and despite some efforts of educational institutions to teach developers about security [57], cyber security is still an specialization rather than the norm. Today developers are by and large unaware of the myriad ways they can introduce security problems into their work [79].

2.4 Tools for security

Different tools have been developed to help developers reduce the number of security errors; these tools are very diverse and might be in the form of security standards, guidelines, weaknesses taxonomies, frameworks, software applications, amongst others.

Weaknesses taxonomies help to establish a common vocabulary and an understanding of the ways computer security fails. Several classification schemes have been proposed, currently the most comprehensive is the MITRE corporation's Common Weakness Enumeration (CWE) [17] that incorporates 909 elements. The main goal of the CWE initiative is to stop vulnerabilities at the source by educating software acquirers, architects, designers, and programmers on how to eliminate the most common mistakes before software is delivered. CWE serves as a resource for programmers as they design new software and write code, and supports educators in teaching security as part of the curriculum for software engineering, computer science, and management information systems; CWE ultimately helps to prevent the kinds of security vulnerabilities that have plagued the software industry and put enterprises at risk. MITRE's CWE continues to evolve as a collaborative community effort to populate a publicly available repository of software errors in code, design, architecture, and implementation for developers and security practitioners. CWE is used by tool vendors for tagging what their tool's report and claim to cover. Nevertheless, due to the high detail level of the CWE, it has inherent usability issues; developers that want to use it might get lost and confused by all the terms introduced and even if they understand them correctly they might not be able to recognize instances of the weaknesses in their work.

To help software developers and security practitioners, prioritize and allocate their security resources better, there are other classifications that focus on the most prevalent security errors like the OWASP's (Open Web Application Security Project) Top Ten project [10] for web applications, SANS institute top 25 CWE [14], and the seven kingdoms of security errors [80] for software in general.

The "seven kingdoms of security" taxonomy was designed with the primary goal of organizing sets of security rules that could be used to help software developers understand

the types of errors that have an impact on security; with the belief that one of the most effective ways to deliver this information to developers is through the use of tools. The expectation is that, by better understanding how systems fail, developers will better analyse the systems they create, more readily identify and address security problems when they see them, and generally avoid repeating the same mistakes in the future. When this set of security rules integrates with the programming environment, it becomes a powerful teaching mechanism.

Standards, guidelines, and security patterns [48], [59] help developers by collecting the knowledge and experiences of the security community in a reusable form; Some examples of these are the BSIMM [74] (Building Security In Maturity Model) “which study real-world software security initiatives”. The BSIMM does not tell what one should do; instead, it tells what everyone else is actually doing. It allows an organization to determine where it stands in terms of maturity with its software security initiative and how to evolve over time”.

To help developers in the search of countermeasures to common security problems, security patterns have been collected, classified [59] and evaluated [23],[48]. Security patterns have different levels of abstractions. There is no single correct level of detail for security patterns. Different potential consumers of security patterns work at different levels. A developer may be primarily concerned with patterns of code-level objects, an architect may build network models, and a CIO may be primarily interested in trust relationships between organizations. All are valid uses of the security patterns approach, though each target audience might find little value in patterns at a much different level of detail.

2.4.1 Software tools for security

Software tools for security can adopt very diverse forms; this section presents some instances of software security tools. This is by no means an exhaustive overview of the different tools for security available. Nevertheless, it is useful to illustrate the diversity of tools that can be found to help developers attain secure systems. Research in the area of

security tools is focused on two main areas [37]: tools that assist in the testing of software applications, and tools that help developers to create components that led to obtaining secure systems. However, software tools for security are not limited to these two branches as will be shown in this section.

Finifter and Wagner [24] carried out a comparison of how different programming languages and web frameworks influenced the security of web applications. They found that there is a relationship between the features offered by the frameworks employed were the most effective defences were those that were enabled by default or inherent in framework design or language and, that, optional protections, even when present in the frameworks were not used. The different programming languages did not show any significant advantage of one over the other.

Tools for testing fall under two categories: white box, and black box testing tools. Black box testing, also called functional testing, is testing that ignores the internal mechanism of a system or component and focuses exclusively on the outputs generated in response to selected inputs and execution conditions. White box testing, also called “structural testing” and “glass box testing”, takes into account the internal mechanism of a system or component. In recent times, the tendency is to integrate these tools in the IDE and perform the analysis on the fly at the same time developers write the code. An example of a tool that adopts this strategy is the prototype developed by Xie et al. [87], [53], [86], [88]. It offers interactive support for secure programming integrated with the Eclipse programming environment. The prototype was in the form of an Eclipse plugin that help developers to detect security errors while they are writing code. The prototype proved to be useful for novice programmers; however, their test with experienced users was not very successful, but that might be attributed in part to the experimental design they applied for the evaluation; among other usability issues of the tool. Some popular tools for source code analysis are HP’s Fortify, Coverity’s products, SSVChecker (Static Security Vulnerability Checker) and LAPSE (Lightweight Analysis for Program Security in Eclipse) [54]. Some of these tools perform static and dynamic analysis to detect vulnerabilities.

Other tools help to perform penetration testing of web applications or systems in general. The web application attack and audit framework (w3af) is an open-source web application security scanner. The project provides a vulnerability scanner and exploitation tool for Web applications. It provides information about security vulnerabilities and aids in penetration testing efforts. The Metasploit Project is a computer security project which provides information about security vulnerabilities and aids in penetration testing and signature development for Intrusion Detection Systems (IDS). Its most well-known sub-project is the open-source Metasploit Framework, a tool for developing and executing exploit code against a remote target machine. Exploit can be defined as “a piece of software, a chunk of data, or sequence of commands that takes advantage of a bug, glitch or vulnerability in order to cause unintended or unanticipated behaviour to occur on computer software, hardware, or something electronic (usually computerised). Such behavior frequently includes such things as gaining control of a computer system or allowing privilege escalation or a denial-of-service attack”.

Other tools like the one presented by Mutti et al. in “An Eclipse plug-in for specifying security policies in modern information systems” [50] take a different approach, presenting a plugin to develop security policies using ontological web language, which allows to automate part of the process of validation and verification. Or the web goat project whose goal is to: “*create a de-facto interactive teaching environment for web application security.*” [11]. This is done through the creation of a deliberately insecure web application that can be used by security practitioners to analyse and test security tools.

The Protection Analysis[4] project deserves a special mention here because its intended aim was the same as ours: to produce security tools. The main goal was to produce a tool to make protection evaluation more effective and economical by automating the detection of security flaws. A general strategy referred as “pattern-directed protection evaluation” was identified. They sketched algorithms for such tool, but the static analysis technology available at the time was not sufficient to realize them.

The SDL(Security Development Lifecycle) Threat Modeling Tool [60] according to Microsoft is the first threat modeling tool which is not designed for security experts. It makes threat modeling easier for all developers by providing guidance on creating and analyzing threat models. The tool enables any developer or software architect to:

1. Characterize systems and analyze data flow diagrams.
2. Communicate about the security design of their systems.
3. Analyze those designs for potential security issues using a proven methodology (STRIDE).
4. Suggest and manage mitigations for security issues.

The SDL Threat Modeling Tool is indeed a very good example of what represents a usable security tool. By allowing non-security experts to take advantage of Microsoft's experience regarding security with ease of use and intuitiveness.

There are other tools for analysis [7], design [18], and modelling; depending on what we are looking for there is a chance that might be a tool for that.

2.4.2 Security tools and insecurity

Often, users do not understand how a tool works and the kind of protection it provides and this prevents them from benefiting effectively from them. This can be dangerous because it creates a false sense of security; many times users believe that the mere presence of security tools automatically protects them and they are not necessarily aware of the risk they are exposed to. The correct use and understanding of the features provided by a tool falls in the realm of usability

The lack of usability is detrimental to security. In "Why Johnny Can't Encrypt" [83] a usability test of PGP 5 was performed. Originally PGP's goal was to enable users to protect their email messages' confidentiality and authenticate the source of them. In its marketing material it stated that the "significantly improved graphical user interface makes complex mathematical cryptography accessible for novice computer users." [83]. The study found that PGP 5 does not make the task manageable for average computer

users. The lack of usability had negative consequences for security; one example is that users ended up sending their private keys in plain text to the researchers while trying to send an encrypted message. Due to the lack of feedback, users could not tell if what they were doing was correct or not.

Cryptography is the foundation of cyber security, it provides the primitives that help us to attain security goals, and nevertheless it poses a usability challenge to anyone that needs it. A common problem with the use of encryption is that many times it is approached as an end instead of a medium. The mere use of encryption does not provide confidentiality protection. The security provided depends directly on the placement and management of the encryption keys. If the key used for encryption is stored along with the data it is intended to protect, the protection it provides is null. Therefore it is necessary to design security tools to be easy to use and understand.

2.4.3 Security tools and usability

According to the International Standard Organisation (ISO) [38], usability can be defined as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. Usability is a contextual property; a system deemed usable in one context may not be usable in another. This definition focuses on users' goals (effectiveness), the speed with which goals are achieved (efficiency), and users' satisfaction with the system within a specified context.

Security comes with certain costs in terms of usability. Traditionally, that is the sacrifice we make to be secure. Security many times becomes an afterthought requirement [85]. This is a common mistake, to attain security many aspects need to be considered at different stages of a project. However, people that work on security make the same mistake with usability, inventing or designing security policies and mechanisms that people cannot use. The lack of usability is one of the most recurring problems when it comes to security. When security tools are used incorrectly [83],[26] it leads to an insecurity situation.

Gutmann et al. [29] enumerate four different stands that can be adopted while balancing security and usability:

- The two should work together as equal partners.
- Security comes first, and usability should be the compromising junior partner.
- Usability comes first, and security should be the compromising junior partner.
- Security is best left as a separate product, naturally layered into the application without disturbing it and without compromising strong design principles.

The approach one chooses will influence the system architecture, the way in which systems are deployed, and the way in which security is delivered to, and experienced by users. Therefore, there is no easy answer to the trade-off question yet. To deliver security properly, we must rethink the assumption of a usability compromise.

The main challenge for current security efforts is not to find better encryption algorithms or protocols but to make the existing ones usable.

2.4.3.1 Security Usability Fundamentals

We need to understand the basic concepts of application security and usability, Gutmann defines them as follow in his work “Engineering security” [28]:

“An application, exhibits functionality if things that are supposed to happen, do happen. Similarly, an application exhibits security if things that are not supposed to happen, do not happen”

Security usability combines technical and human factors. If a highly secure system is unusable, users will move their data to less secure but more usable systems. Problems with usability are a major contributor to many high-profile security failures today.

“However, usable security is not well-aligned with traditional usability for three reasons:

1. Security is rarely the desired goal of the individual. In reality, security is often in opposition to the actual goal. Such as a locked door oppose the main purpose of the door that is allowing access through it.
2. Security information is about risk and threats. Such communication is most often unwelcome. Increasing unwelcome interaction is not a goal of usable design.

3. Since individuals must trust their machines to implement their desired tasks, risk communication itself may undermine the value provided by them.

A broader conception of both security and usability is therefore needed for usable security.” [81]

Usability, just like security, is a contextual property and has different meanings in different contexts. For some, efficiency may be a priority, for others, learnability, for still others, flexibility. In a security context, the priorities must be whatever is needed in order for the security to be used effectively.

Security software is usable if the people who are expected to use it [83]:

1. Are reliably made aware of the security tasks they need to perform;
2. Are able to figure out how to perform successfully those tasks;
3. Don't make dangerous errors; and
4. Are sufficiently comfortable with the interface to continue using it.

This is the definition we used to develop the prototype presented in later sections.

2.4.3.2 Making Security Usable

Most computer security is not easy for people to use. Ideally, they should be empowered to make and enforce their own security and privacy decisions, but the usability barrier has made this implausible so far. Whitten [82] presents a research work in which she proposes that security usability is different from usability for other kind of software. In consequence usability of computer security must be specially tailored to address the problems inherent to it.

Two techniques are presented by Whitten:

- Safe staging, which takes the basic concept of multi-level user interfaces (which are usually designed to aid learning and to support both novice and expert users), and enhances it by providing a clear theory of how to design levels and transitions that

preserve the user's security at all times.

- Metaphor tailoring, adds a new technique, risk enumeration, to existing techniques for designing visual user interface metaphors. Risk enumeration, enables us to tailor our visual representation of the most important aspects of security in a methodical and prioritized way.

However, the techniques presented may not be equally applicable to all aspects of computer security; some security may be inherently unusable, and some security may already be usable enough. The difficulty for the developer is to identify when and how to apply these concepts.

Another fundamental concept extracted from [82] is the “well-in-advance” principle; the concept of “just-in-time” help has become a popular usability design strategy. It is based on the idea that the information necessary to enable a user to perform a particular task should be triggered when the user begins to attempt that task. Whitten argues that this is a fine strategy when the task is the user's primary goal but that it is a bad strategy when the task is a secondary goal that must be attended to in order to accomplish the primary goal safely, as is very often the case in computer security.

To better understand this argument, consider Microsoft Windows Vista pop-ups requesting users' permission to perform a certain task. Figure 2-3 shows an example of how this dialog looks. Users usually just blindly press the “Continue” button and proceed. It is not a surprise that, when users are already engaged in some primary task, they will be reluctant to grant much attention to an interruption that tells them they must learn some new concepts before they can proceed safely to achieve their goals.

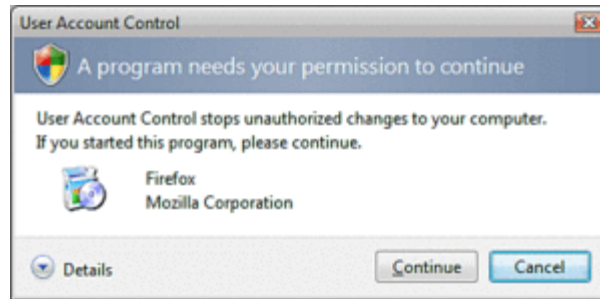


Figure 2-3 Windows Vista user account control dialog.

The “well-in-advance” principle establishes that when some primary user task requires that some security tasks be attended to in order to be safe, the user needs to have a reasonable idea of the complexity and effort required to achieve those security tasks, well in advance of deciding to tackle the primary task.

Other efforts try to integrate techniques and tools and improve them to support the design of usable and secure systems. Failys [22], developed a framework for specifying usable and secure systems. IRIS (Integrating Requirements and Information Security) considers the system design process from three different perspectives — Usability, Security, and Requirements — and guides the selection of techniques towards integrative Security, Usability, and Requirements Engineering processes. Failys’ research makes three significant contributions:

- 1) A conceptual model for usable secure Requirements Engineering is presented, upon which the IRIS framework is founded;
- 2) The CAIRIS (Computer Aided Integration of Requirements and Information Security) software tool is presented to support the elicitation and specification of usable and secure systems.
- 3) The description of how the results of applying IRIS can be used to improve the design of existing User-Centered Design techniques for secure systems design.

One has several options at the moment of designing security features to integrate into applications. However, the designer’s interpretation of these concepts is what really matters. The real challenge for many designers is how to enable users to achieve their

goals within an acceptable risk threshold in a way that is easy to understand.

2.4.3.3 How Users Make Decisions

In this section, we take a look at some of the human mental processes that are relevant to how users make decisions about security, and explore the reason why security user interfaces do not perform very well, in some cases.

The Bayesian decision-making model [1], assumes that someone making a decision will carefully take all relevant information into account in order to come up with an optimal decision. The formalization of this model is called “Subjective Expected Utility” (SEU) [63],[30] and makes the following assumptions about the decision-making process:

- 1) “The decision-maker has a utility function that allows them to rank their preferences based on future outcomes.
- 2) The decision-maker has a full and detailed overview of all possible alternative strategies.
- 3) The decision-maker can estimate the probability of occurrence of outcomes for each alternative strategy.
- 4) The decision-maker will choose between alternatives based on their subjective expected utility.”[28]

However, this is an ideal situation in which the user has enough information to make a rational decision, yet people do not always act in a rational way or have enough information, and often make their decisions based on other factors, such as emotions [3], or past experiences.

2.4.3.4 How Users Really Make Decisions

When a rational decision is not possible, humans use heuristics [47]. A heuristic is a technique designed for solving a problem more quickly when classic methods are too slow, or for finding an approximate solution when classic methods fail to find any exact solution. By trading optimality, completeness, accuracy, and/or precision for speed, a

heuristic can quickly produce a solution that is good enough for solving the problem at hand, as opposed to finding all exact solutions in a prohibitively long time.

Research from the US Department of Defense [44],[43] discovered that people under pressure do not weigh their options and choose the best one. Instead, they use what is called “recognition-primed decision making” (RPD). In which they generate options one at a time, without ever comparing any two, rejecting the ones that do not work and going with the first one that does. Humans take this approach to making a decision when they cannot hold all of the necessary information in working memory, or cannot retrieve the information needed to solve the problem from long-term memory, or cannot apply standard problem solving techniques within the given time limit.

This approach to making decisions is used under the following circumstances:

- 1) The decision-maker is under pressure.

Normally, programmers are faced with time pressures, whether from employers, assignments, or communities, i.e., social pressure.

- 2) The conditions are dynamic.

The situation may change by the time a long and detailed analysis is performed.

- 3) The goals are ill-defined.

Often security goals are not expressed due to a lack of security knowledge.

- 4) The information about the different options is incomplete or unavailable.

- 5) In the case of security, users have little knowledge on how to make a system secure and the mechanisms and actions that are required for it.

This model, along with the SEU model, represents the most general decision making process. Different factors affect them, but this generalization can give us a broad idea of how developers make decisions during software development.

2.5 Summary

Computer security can be defined as measures and controls that ensure confidentiality, integrity, and availability of information system assets including hardware, software, firmware, and information being processed, stored, and

communicated. There is no such thing as a secure system but only systems with an acceptable level of risk. Many factors can contribute to the introduction of security errors, such as lack of quality code, skills, knowledge, and concentration, which all can result in security errors. Security errors are different from common coding errors because they are latent and its effects are not immediately perceived. To help developers build secure systems, several tools have been created. Some tools make visible errors hidden in the code to assist in the code review efforts of an organization. Other tools assist in the testing and design security features. Many common security issues still remain unsupported or minimally-supported, such as input sanitation, access control, and intrusion detection. One of the bigger problems of security is not the lack of safe encryption mechanisms, but rather the usability of these encryption mechanisms for either the user or software developer, depending on the context.

3 CHAPTER 3 – CRYPTO-ASSISTANT

3.1 Introduction

This chapter describes the motivation for the development of the prototype presented. The chapter continues with the presentation of the development process, the strategy adopted, and factors that influenced the development of the prototype. These factors include an analysis of how users react to security warnings and how they make decisions. Finally a cognitive walkthrough evaluation of the *learnability* of the Crypto-Assistant is presented where no major issues were detected.

3.1.1 Motivation

With the intention of building a tool to help programmers to add security to their applications, we started the search for a security error in which we could focus our efforts. The prototype developed has the main goal to contribute to the remediation of the software weakness “CWE-311: Missing Encryption of Sensitive Data” [15] in its more specific form: “CWE-312: Clear text Storage of Sensitive Information” [16], which is a very common issue, occupying the 8th position in “SANS Top 25 Most Dangerous Software Errors” [14] and is also part of the OWASP’s Top 10 project category: “2010-A7-Insecure Cryptographic Storage” [77]. This vulnerability occurs when the application stores data in clear text in a resource that might be accessible to an attacker when information should be encrypted or otherwise protected. According to the “CWE-700: Seven Pernicious Kingdoms” taxonomy [80] this kind of weaknesses falls under the “CWE-254: Security Features” category and within the “CWE-359: Privacy Violations” class. According to Team SHATTER [72] (Security Heuristics of Application Testing Technology for Enterprise Research), unencrypted sensitive data is one of the top 10 database vulnerabilities.

3.2 Problem definition

The networked database is crucial for the functioning of any application. The most valuable assets reside in the database. The information stored can include transaction records, financial data, and customer information. Protecting this data is very important, and failure to do so might result in financial and legal cost; but, it is also an increasingly difficult and non-trivial task.

Sensitive data stored on networked servers are at risk from attackers who only need to find a way inside the system to access this confidential information. Additionally, attackers might impersonate a user of the system and therefore one must consider internal threats including employees that can access and exploit this data. Another situation in which sensitive data can be compromised is when physical backups are stolen or lost, surprisingly, there are many examples of this kind of confidentiality breach [78],[27],[71].

To allow the reader to have a more complete picture of how the development of Crypto-Assistant evolved; some important concepts will be introduced to highlight the aspects that we had to take into consideration during the development of the prototype.

The purpose of encryption is to protect sensitive information from unauthorized readers (confidentiality) by making it unintelligible. However, the data must remain accessible for the authorized applications and users who require it for a legitimate business reason (availability). Figure 3-1 shows the interaction of the different components of encryption.

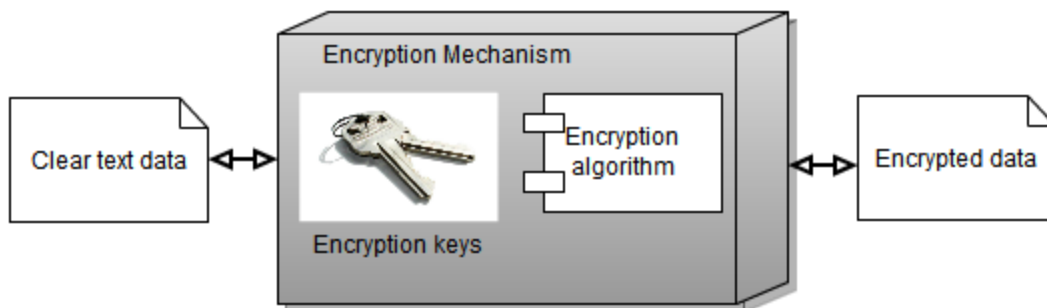


Figure 3-1 How encryption works

Encryption keys are required to encrypt and decrypt data therefore they need to be accessible in order to store or access encrypted information. When using encryption, the protection it provides is as good as the protection of the encryption keys. In the same way a company might store employee records in a locked drawer and designate a person to be responsible for the key, the same must happen with encryption keys.

Figure 3-2 and Figure 3-3 depict a broad overview of the process to implement transparent data encryption in an application with Hibernate Tools Eclipse plugin and Jasypt. Both tools will be described in greater detail in section 3.3.1.2 and 3.3.1.3.

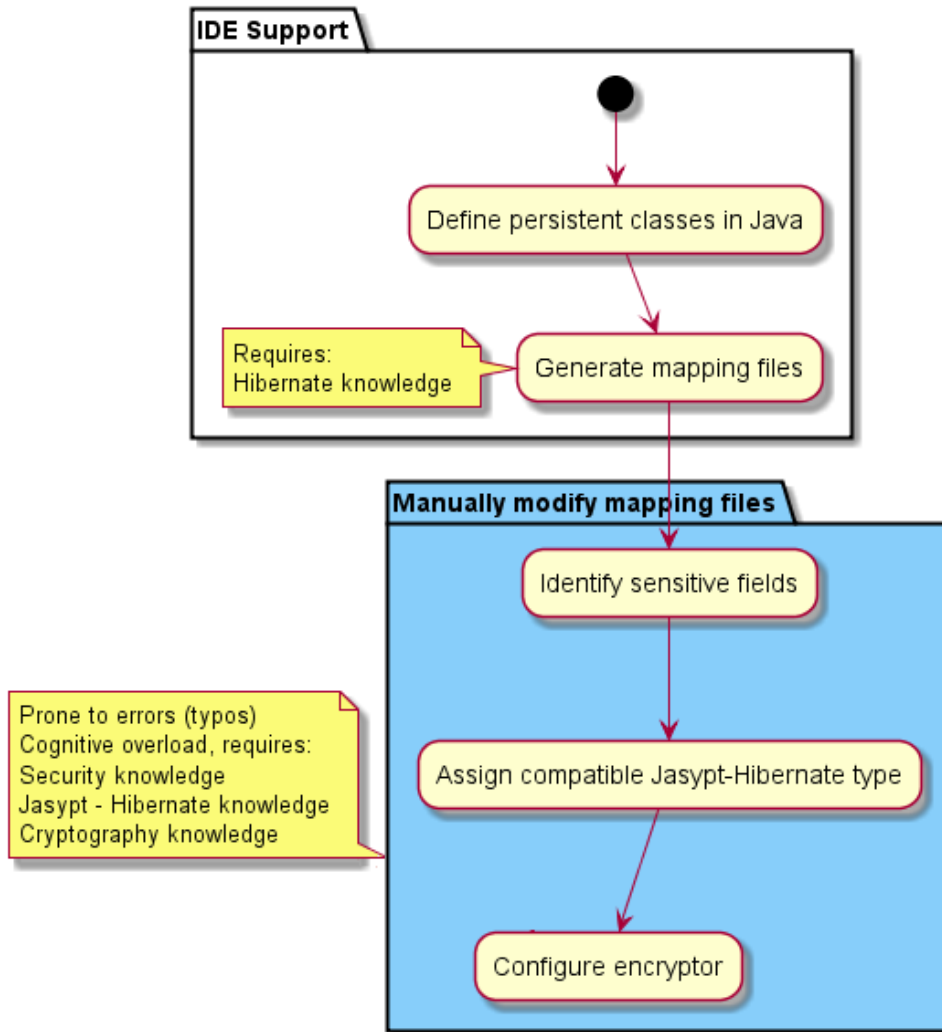


Figure 3-2 Use of Jasypt and Hibernate for encryption without Crypto-Assistant support.

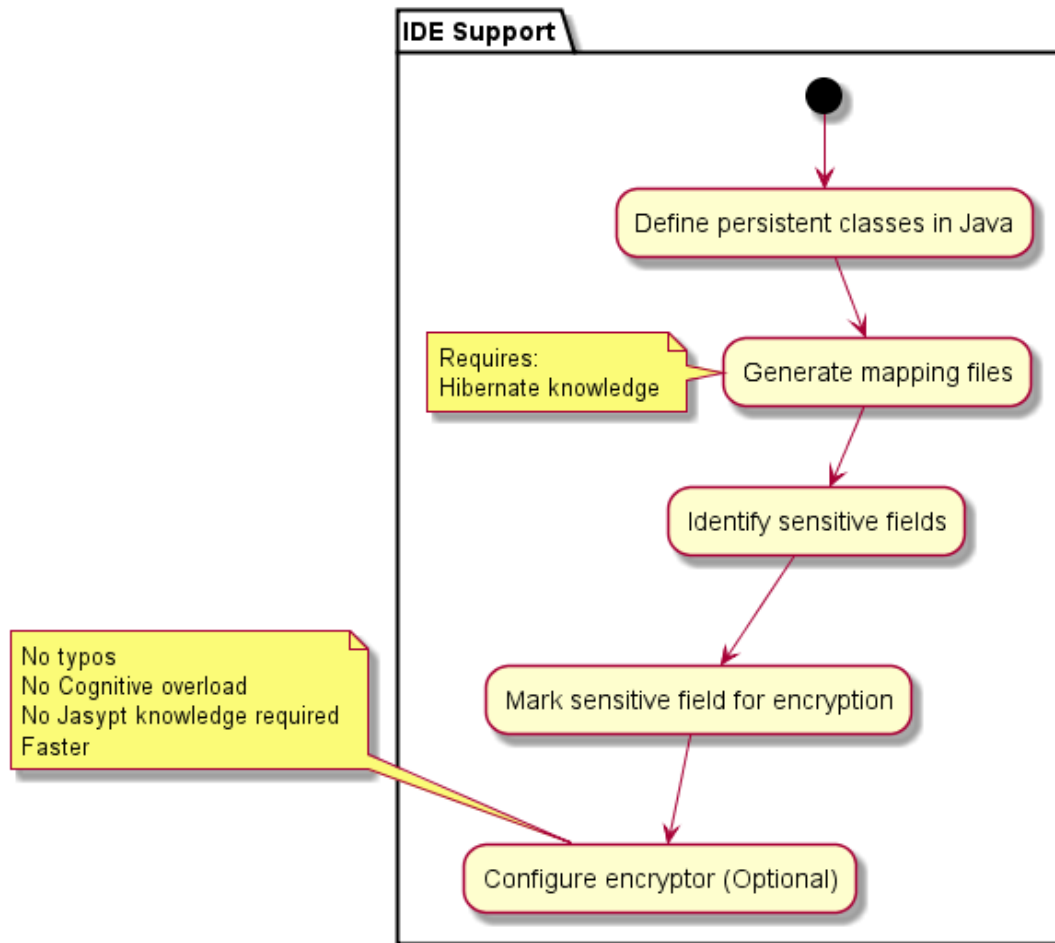


Figure 3-3 Use of Jasypt and Hibernate for encryption with Crypto-Assistant support.

3.2.1 Initial Hypothesis

Based on observations about the causes of security errors in the previous chapter and the information presented here about the use of Jasypt and Hibernate to protect sensitive data, we identified several factors that contribute to the problem of developers not implementing encryption and we can summarize them as follows:

1. Lack of awareness about the risks of storing sensitive data in plain text.

2. Lack of knowledge about available protection mechanisms and their effective use.
3. Lack of usability of the protection mechanisms.

Our initial idea was to raise awareness and let developers fix the issues. However, lack of awareness is not the only problem. Even if developers are aware of the risks, there are other problems that contribute to the problem of developers not implementing encryption. One of the most evident is the lack of training in secure programming techniques. This should come as no surprise because if developers are not even aware of the possible issues, they will not know how to fix them. Our goal was to help developers that have little or no experience about security and as a result, we could not expect that they were familiar with any security mechanism.

When a developer is aware of the risk and willing to take corrective actions the last obstacle to overcome is the effective use of a protection mechanism or a tool for security, the focus of the problem switch then to the usability of the tools.

The hypothesis we formulated is the following:

Rising awareness about the risk of storing sensitive data in plain text and putting a usable and intuitive encryption mechanism conveniently located at the reach of the developers in the IDE; will be reflected in an increase of encryption as protection mechanism.

3.2.2 Communication strategy

Developers' behaviour is influenced by their environment, this includes the programming systems and tools that they use and as a result, if security is not integrated with them, it might be perceived as an interruption in the work flow by forcing them to leave their IDE due to the lack of tool support. Therefore, we wanted to integrate security tasks within the tools developers use to build applications. The IDE was proposed as an effective teaching mechanism [80] and is the only layer that we have to communicate a problem to the developer.

Security warnings are common in computer systems to communicate a failure or deter users to engage in risky behaviours. The purpose of security warnings is to protect users and their systems. Warnings can be useful to capture user's attention and raise awareness about possible risks. But, if no remediation action is provided or if the remediation action is too complex or time consuming, some users may ignore them rendering the effectiveness to null. To increase its effectiveness, they must enable users to take a mitigation or remediation action. Based on this information we choose to use security warnings to meet the goal of raising awareness. To increase our chance of success, research was conducted on the use of computer warnings and how users make decisions.

3.2.3 Warnings and User Reactions

Computer security warnings are intended to protect users and their systems. However, users frequently ignore these warnings. In [8], the authors describe a study designed to gain insight into how users perceive and respond to computer alerts. From this study, there are some remarkable contributions that can be applied to the design of computer warnings in general.

1. There is a trade-off between the amount of information presented to a user in a warning and the chance the user will utilize that information in a useful way. It is less likely that a user will interrupt his or her work to read a long technical text.
2. Warnings must be presented only when necessary, and then with only the necessary information.
3. Only present a security warning prompt when automatically eliminating or guarding against a risk is not possible.

Warnings should only be presented with situations in which the best course of action depends on the details of the circumstances that are known to the user. Many times users are not familiar with the concepts used in security warnings and this can affect the effectiveness of the information presented. It is easy to find examples of warnings that are not effective [8]. We incorporate this information in the development of the Crypto-

Assistant. To increase the effectiveness of our warning we investigated further on how users make decisions.

The chosen strategy was to embed a warning in the workflow of the tools, deliver the warning message, and assist the user to perform a risk defusing operation. This strategy might seem too simplistic but it implies several difficulties. One of them is to find the right placement and timing for the warning message. We will refer to the placement of the warning in the workflow as the communication point. The communication point must be identified early enough in the development process when the error is cheaper to fix. The information presented must be brief, clear and easy to understand. Figure 3-4 Communication strategy shows the possible outcomes of the warning display

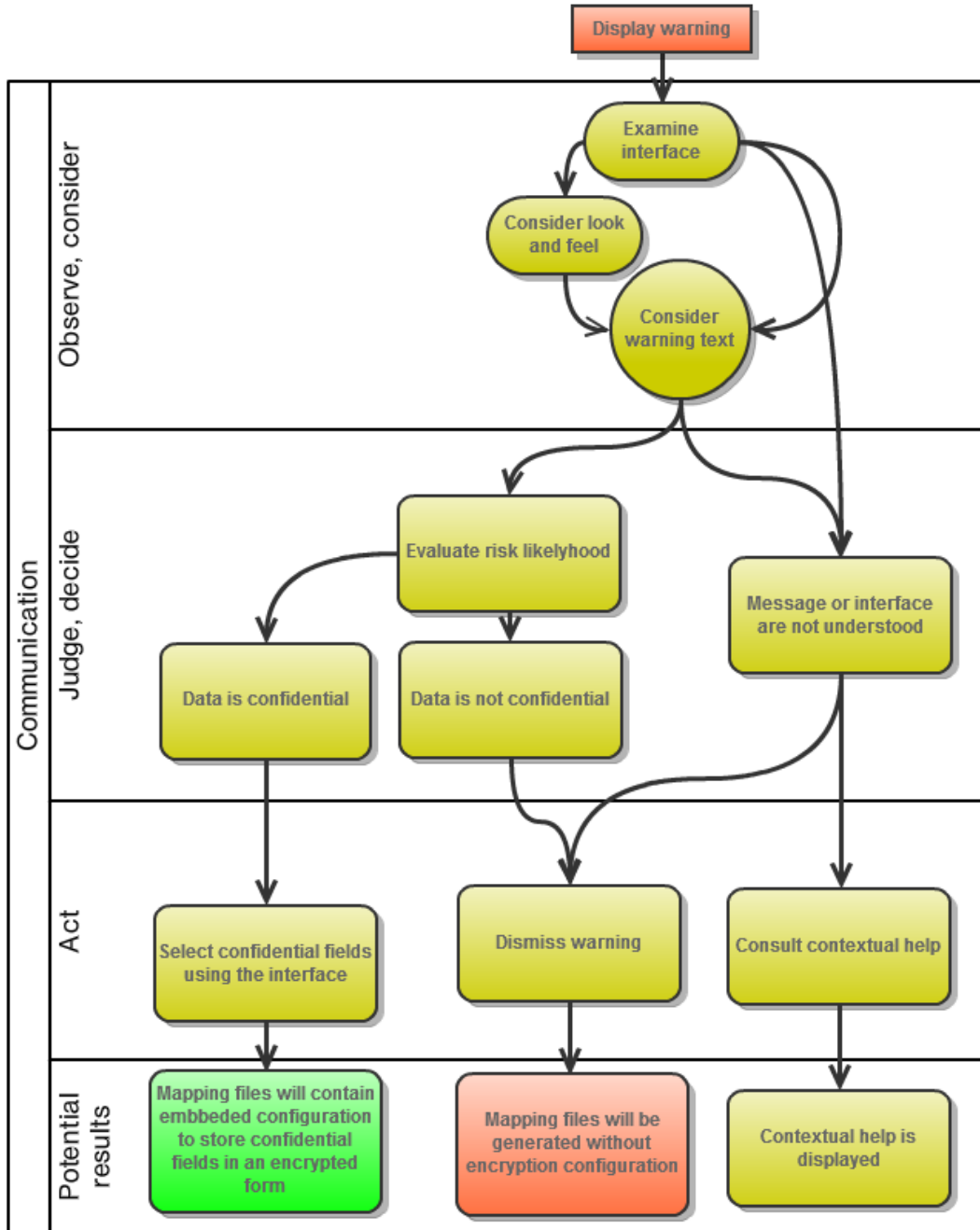


Figure 3-4 Communication strategy

3.2.4 Component selection

The design of the Crypto-Assistant prototype began with the search for a suitable tool to embed the warning message. The goal was to maximize the relevance of the message

placing it in a decisive moment of the application development integrating it in the tool workflow. We focus the search in open source components; that would allow us to modify the code to produce a prototype more quickly.

We were inspired by the scaffolding capabilities of MyEclipse [52], a closed source commercial implementation of Eclipse that only required a database schema to generate the skeleton for a CRUD (Create, Read, Update and Delete) application. The goal was to influence the development of the application in an early stage; before the cost of any necessary modification would become prohibitive.

The platform selected for the prototype was the Eclipse IDE. Its opens source nature and popularity made it an ideal candidate for our research. Hibernate was selected based on its popularity, abstraction capabilities and the availability of Hibernate Tools plugin for Eclipse. Jasypt (Java simplified encryption library) is a library for simplified symmetric encryption in Java; this library expands Hibernate user types, allowing transparent data encryption.

3.2.5 Design considerations

The first design idea was inspired in MyEclipse; we wanted to use the reverse engineering capabilities of the Hibernate Tools. We considered it a good communication point to integrate encryption before the full application was completed. The idea was to suggest the use of encryption to users trying to generate Java classes and mapping files from an existing database schema. The database schema would work as input for our tool. Then users would be able to select what field they wanted and if the database was empty use Hibernate to regenerate the database structure. However, there was a fundamental problem with this approach. More specifically, the database may contain existing records requiring encryption and changes in the schema. This last realization added a new challenge for our tool because to increase the usability of the tool, we needed to simplify the whole process or rely on the user to find a solution. However, based on the observations of the ASIDE (Assured Software Integrated Development Environment)

project [53] we cannot trust that users would know how to complete the task by themselves.

To understand better what would be required to simplify the process, we needed to consider the effects of encryption in data and what would be required to simplify the migration process.

Encryption of fields or columns with Jasypt requires changes in the structure of the database. Encryption and decryption algorithms are known as ciphers. Ciphers often use an operation mode that produce output in fixed block sizes and require the input data to match this output size or it will be padded. The effects of encryption operations might be more evident on small data items which may increase the size of the stored data. Encryption transforms character data into meaningless binary data; this has consequences not only in the size of encrypted data, but also in the data type used to store the information. Jasypt stores encrypted data encoded in character form using Base64 or hexadecimal format, which increases the data size by approximately one third than if it were stored in its original binary form. Jasypt encodes the data in Base64 by default and therefore, it is necessary to resize and update the database columns to accommodate the encrypted data.

The Jasypt default configuration uses a random salt for every value encrypted. A salt is random data that is used as an additional input to an encryption function, which makes slower the decryption of data by the use of brute-force, dictionary and rainbow tables' attacks. The use of a random salt allows that the same data encrypted always result in a unique cypher text, making impossible to perform search queries based on the encrypted field since the encryption of the same value produces different outputs due to the addition of the random salt. One must put special care before encrypting information in indexed fields. Indices are used to improve the speed of lookups, and searches may be seriously degraded by the computational overhead of decrypting the field contents each time searches are conducted. Depending on the strategy adopted, the encryption of indexed data might not be feasible (i.e. the use of a random salt makes such operations impossible to perform). Unfortunately, most often administrators index the fields that must be

encrypted. New planning considerations are needed to determine what fields must be indexed; a decision that might not be easy to take.

Referential integrity is another important factor to consider. If a field with integrity constraints have to be encrypted, that is, a field that is part of a relation (e.g., a foreign key is encrypted) then all of the tables that are part of the relation would require changes in their structure and update of its values in addition to the use of a fixed salt.

The resources needed to provide our prototype the capabilities to automate the process of modifying an existing data base schema and its data, was beyond our scope. We required that the solution proposed to the user was simple enough to be described in a warning message.

Therefore we had to reconsider our first strategy and located a better communication point; Hibernate Tools allows for the creation of mapping files from Java classes to improve the usability of Hibernate. Using this as a communication point still gave us an input structure, the Java source, but did not imply the existence of a database structure; the structure can be generated from the mapping files. This decision eliminated the need to include all the features required to simplify the change of structure in the first design.

3.2.5.1 Recommended Encryption algorithms

The encryption algorithm was an important factor to consider for our problem. Our implementation allows the developer to choose their encryption algorithm from the ones available for the virtual machine. Many databases use Data Encryption Standard (DES) to protect sensitive data. However, DES has long been considered insufficient to protect any information for a considerable amount of time. Advanced Encryption Standard or AES and triple DES (3DES) are, at the time of writing, the recommended algorithms by NIST for symmetric encryption [2]. Triple DES offers a better protection against cryptographic attacks than DES; however, the use of this algorithm comes with a trade off in performance. AES encrypts and decrypts data in 128-bit blocks, using 128, 192 or 256 bit keys. The nomenclature for AES for the different key sizes is AES-x, where x is the key size. All three key sizes are considered adequate by NIST for Federal

Government applications. Triple DES encrypts and decrypts data in 64-bit blocks, using three 56-bit keys. NIST recommends that applications should use three distinct keys.

Due to export restrictions Java puts a limit on the key size allowed for encryption, this produces a run-time exception if a key size of 128 bit is meant to be used and the Java runtime is not properly configured to enable strong encryption. To minimize the chance of using a weak encryption algorithm, we implemented a warning mechanism to alert the user if the algorithm selected for encryption is different from AES. The mechanism is limited to inform the user that AES is the recommended algorithm but it requires the strong encryption configuration.

If a user wants to use AES they would have to get the unrestricted security policy files from Oracle and install them in their Java Virtual machine:

- 1) Download the unlimited strength JCE policy files.

Go to: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

- 2) Uncompress and extract the downloaded file.

This will create a subdirectory called jce. This directory contains the following files:

README.txt	Detailed install information
COPYRIGHT.html	Copyright information
local_policy.jar	Unlimited strength local policy file
US_export_policy.jar	Unlimited strength US export policy file

- 3) Install the unlimited strength policy JAR files.

To utilize the encryption/decryption functionalities of the JCE framework without any limitation, replace the original JCE policy files (US_export_policy.jar and local_policy.jar) with the unlimited strength versions extracted in the previous step.

The standard place for JCE jurisdiction policy JAR files is:

<java-home>/lib/security [Unix]

<java-home>\lib\security [Win32]

<java-home> refers to the directory where the Java SE Runtime Environment (JRE) was installed.

3.2.5.2 Key management

Because cryptography is based on keys that encrypt and decrypt data, the database protection is only as good as the protection of the keys. Security depends on two factors: where the keys are stored, and who has access to them. Secure key management is often overlooked when planning an encryption strategy.

Some important questions to address in planning an encryption strategy include: how many encryption keys will be needed, and how they will be managed?

The answer to these questions should include careful planning of where the keys will be stored, how to protect them, and how often the keys should change.

The fewer keys you use to encrypt information, the easier the solution is to manage, but the more critical key security becomes. Crypto-Assistant uses a single key that is embedded in the mapping files of every class that have an encrypted field. This may not be the best solution from a security perspective. Part of managing keys is deciding where to store them. One easy solution is to store the keys in a restricted database table or file. But, all administrators with privileged access could also have access to these keys, decrypt any data within the system, and then cover their tracks. The recommended approach is to use a Hardware Security Module to store the keys. In this case, the keys never leave the hardware and therefore access can be controlled so neither administrators nor intruders can penetrate the machine and steal them.

We recommend the separation of the database and application servers. This architecture protects against rogue database administrators and media stealing; even if the

data can be accessed the key is needed to decrypt the data is still out of reach and vice versa.

Proper management involves restricting personal access to key storage locations, random key updates and encoded key storage servers. An effective key management system involves every aspect of key creation like distribution, revocation, network access, and personnel management. As a result, key management is outside of the scope of our study.

3.2.5.3 Database encryption strategies

There are several strategies that one can adopt when choosing a database encryption strategy. Each one of these strategies has their own advantages and risks that must be taken into consideration.

Different aspects must be taken in consideration when planning an encryption database strategy. The next sections are condensed from RSA's document: "Securing Data at Rest: Developing a Database Encryption Strategy"[61]. The information collected here has the intention to be an introduction to explain how the encryption capabilities implemented in our prototype work and other alternatives that could have been adopted.

3.2.5.3.1 Inside the DBMS.

If the DBMS (Database Management System) supports encryption, the process of encryption and decryption takes place within the database the main advantage of this process is that it is transparent to the application. The data is encrypted as soon as it is stored in the database; however, any data that enters or leaves the database, will be making it as clear text. This is one of the simplest database encryption strategies, but it presents performance trade-offs and security considerations that must be evaluated. One of the disadvantages of this strategy is the extra processing that takes place in the DBMS every time that storing or accessing data is necessary. This can have serious consequences on the performance of the entire system. This strategy implies that the DBMS has access to the encryption key, and sometimes it means that the keys are stored in the same server;

an attacker capable of gaining access to the DBMS will have access to both the data and the encryption key, gaining access to the unencrypted data. To prevent this situation a dedicated “Hardware Security Module” (HSM) can be used to store the keys, however this option is not always possible like when virtualization is used in a shared cloud infrastructure.

3.2.5.3.2 Off-loading encryption outside of the DBMS.

The recommended strategy is to consider database architectures that off-load encryption processing and secure key management to a separate, centralized “Encryption Server”. The “Encryption Server” performs the computations required by encryption and decryption. The benefits of this strategy are that it removes the computational overhead of cryptography from the DBMS or application servers, and perhaps most importantly, it allows separation of encrypted data from encryption keys. The keys in this architecture never leave the encryption server. Locking down access and monitoring the “Encryption Server” is important in this scenario as well, but easily achievable.

3.2.5.3.3 Application level encryption.

This is the architecture that allows explicit control over the information that is encrypted. The application has the chance to classify and manage who has access to the information during what times and for what purpose. This requires authorization and authentication controls, otherwise, encryption at this level provides no additional security.

In this architecture, the application server takes the responsibility to perform the cryptographic operations. Data is introduced in the application as plain text, then encrypted and sent obfuscated to the database. The keys never leave the application server and therefore the separation of the encrypted data and the encryption key is achieved in this way. Attempts to snoop or intercept writes on disk or direct access to the database would yield useless information.

However, encryption at this level puts limitations in the operations that can be performed in the database (e.g., searches or lookups that cannot be performed on the obfuscated information at the database level). Since the encryption is done on a per

application basis, if multiple applications require encryption, this will add additional complexity to the protection of data. Typically, application level encryption is software based [21] which is the case of our Crypto-Assistant prototype. Furthermore, encryption is a CPU intensive task and will compete for resources with other processes. In addition, the application server needs access to the encryption key, therefore, if an attacker breaks into the server and finds the keys, the information can be decrypted that is why it is important to separate the database and the applications server in case one of them is compromised to provide an effective protection.

3.3 Development

Development of the Crypto-Assistant had a double purpose: testing the hypothesis previously mentioned, and to help developers use encryption in their applications. Using the hypothesis as a starting point, the first high level requirements were elicited:

1. Raise awareness among non-security savvy developers about the risks and consequences of not protecting data at rest.
2. Simplify the encryption process to protect data at rest so that developers without deep cryptography knowledge or security training could benefit from its use.

3.3.1 Architecture

In this section, we discuss the architecture of the Crypto-Assistant, providing a brief explanation about the role of its various components. The Crypto-Assistant is built on top of Hibernate Tools plugin for Eclipse IDE. The prototype uses Jasypt (Java simplified encryption) library [40] to provide its encryption capabilities.

Figure 3-5 shows a broad overview of the Crypto-Assistant architecture. The selection of these components had a double purpose. First, they are tightly related to the target problem: the lack of encryption of data at rest. Therefore, they are an ideal case study for the research presented in the previous chapter. The second reason to choose them is because both of them are open source. Thanks to this, we were able to build the

prototype on top of the functionality these tools provide and speed up the development process to have an operational prototype in a relatively short time.

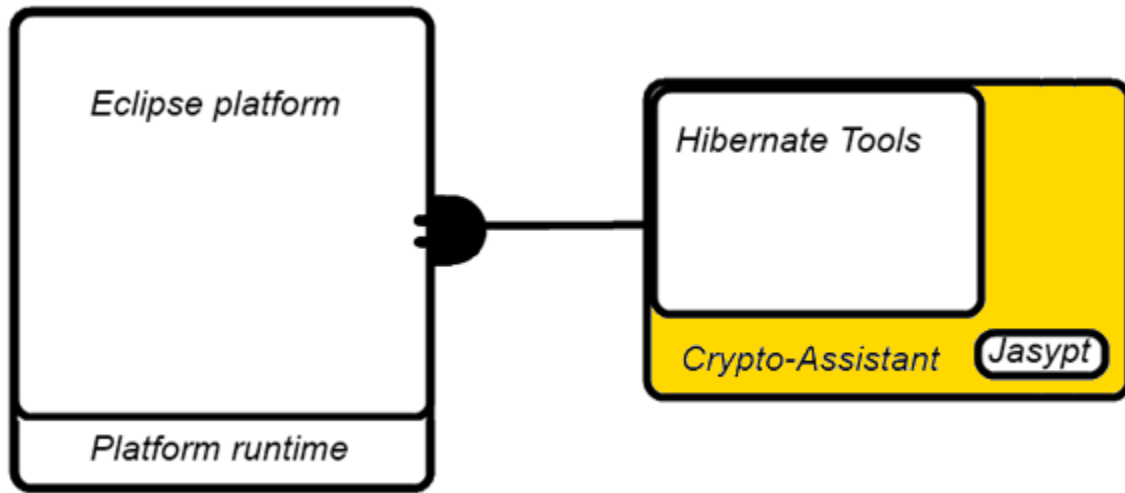


Figure 3-5 Crypto-Assistant - High level architecture

3.3.1.1 Eclipse

“The Eclipse Platform is an IDE for anything, and for nothing in particular” [20]. The Eclipse Platform is a general purpose IDE that contains the functionality required to build a specific integrated development environment (IDE). However, the Eclipse Platform is itself a composition of components; by using a subset of those components, it is possible to build arbitrary applications. One of the advantages of the Eclipse Platform is its integration capabilities. Building a tool or application on top of the Eclipse Platform enables the tool or application to integrate with other tools and applications also written using the Eclipse Platform. Thanks to its managed windowing system, it allows a rich and consistent experience for its users across multiple platforms.

The built-in functionality of the platform is very generic. It takes additional tools to extend the Platform to work with new content types, to do new things with existing content types, and to focus the generic functionality on something specific. The platform provides extension points that allow developers to integrate new functionality through executable modules called plugins.

A tool provider writes a tool as a separate plugin that operates on files in the workspace and surfaces its tool-specific UI in the workbench. When the platform is launched, the user is presented with an integrated development environment (IDE) composed of the set of available plugins. The quality of the user experience depends significantly on how well the tools integrate with the Platform and how well the various tools work with each other.

3.3.1.2 Hibernate & Hibernate Tools

Hibernate is an object relational mapping tool (ORM) [32] whose main goal is to enable developers to persist Java objects in relational databases. Hibernate abstracts the underlying database and increase developer productivity by reducing 95% of the Java code that is typically required to access databases. Hibernate provides its own data types that act as translators between the applications and the underlying database. To achieve its functionality Hibernate uses a set of XML files for configuration and data mapping; additionally data mapping can be done using code annotations embedded in Java code.

Hibernate Tools [33] makes working with Hibernate more pleasant. Hibernate Tools is a toolset for Hibernate 3 implemented as an integrated suite of Eclipse plugins, together with a unified Ant task for integration into the build cycle. An Ant task is a piece of code that extends the functionality of the Ant build system. Hibernate Tools makes the following features available within Eclipse:

Mapping Editor: An editor for Hibernate XML mapping files, supporting auto-completion and syntax highlighting. The editor even supports semantic auto-completion for class names, property/field names, table names and column names.

Console: The Hibernate Console perspective permits configuring database connections, provides visualization of classes and their relationships and allows to execute Hibernate Query Language (HQL) queries interactively against the database and browse the results.

Reverse Engineering: The most powerful feature of Hibernate Tools is a database reverse engineering tool that can generate domain model classes and Hibernate mapping files, annotated EJB3 (Enterprise Java Beans 3) entity beans, and HTML documentation.

Wizards: Several wizards are provided, including wizards to generate Hibernate configuration (cfg.xml) files that tell Hibernate how to connect to a database (which is a fundamental requirement of any application using Hibernate) and Hibernate console configurations that help Eclipse to provide auto completion and reverse engineering capabilities for Java projects.

Ant task: Apache Ant is a software tool for automating software build processes. It is similar to Make but is implemented using Java. Hibernate Tools provide a unified Ant task that allows performing schema generation, mapping generation, or Java code generation as part of the build process.

3.3.1.3 Jasypt Java Simplified Encryption Library

Jasypt [40] is a Java library which allows developers to add symmetric encryption capabilities to their projects with minimum effort, and without the need of having deep knowledge on how cryptography works. Normally, the use of encryption in Java requires the programmer to have a broad understanding of Java and cryptography recommended modes of use. Jasypt simplifies the use of encryption providing a more clear and concise application programming interface (API) that is easy to understand and use. With Jasypt, encrypting and checking a password can be as simple as...

```
BasicPasswordEncryptor passwordEncryptor = new BasicPasswordEncryptor();
String encryptedPassword = passwordEncryptor.encryptPassword(userPassword);
...
if (passwordEncryptor.checkPassword(inputPassword, encryptedPassword)) {
    // correct!
} else {
    // bad login!
}
```

And encryption and decryption of text:

```
BasicTextEncryptor textEncryptor = new BasicTextEncryptor();
textEncryptor.setPassword(myEncryptionPassword);
String myEncryptedText = textEncryptor.encrypt(myText);
...
String plainText = textEncryptor.decrypt(myEncryptedText);
```

And the encryption of sensitive data directly from Hibernate

```
<class name="Employee" table="EMPLOYEE">
  ...
  <property name="address" column="ADDRESS" type="encryptedString" />
  <property name="salary" column="SALARY" type="encryptedDecimal" />
  ...
</class>
```

These are steps in the right direction, but further steps can be taken to simplify the process even more.

This is why the Crypto-Assistant was developed. Its development was based on the assumption that its simplicity would encourage developers to encrypt their application's sensitive data and that they do this encryption correctly. By combining the power of these tools together, the Crypto-Assistant simplifies the process of incorporating a database encryption strategy into an application under development.

The database encryption strategy implemented by our prototype takes encryption and decryption out of the DBMS, the workload takes place at the application server where Hibernate is running and it integrates transparently with the application; this means that the application does not require any changes in its code. Some code changes may be required but only to support cryptography best practices (e.g. key rotation[2] that involves decryption of the data with the old key and re-encryption of it using a new key).

3.4 Usage

The development of Crypto-Assistant simplifies the process of using encryption with Hibernate.

In order to protect sensitive data within an application with the help of Hibernate Tools Eclipse plugin and Jasypt, assuming that the developer already has a Hibernate configuration file, a developer must perform the following actions:

1. First, mapping files must be generated for the persistent classes using the “New Hibernate Mapping File (*.hbm)” wizard. This involves:
 - a. Selecting the classes for which we want to generate mapping files.
 - b. Generating the mapping files.
2. Manually modify each one of the mapped files generated for classes that contain sensitive data. This involves:
 - a. Opening the mapping files of the classes that contain sensitive information.
 - b. Choose the properties that contain sensitive information.
 - c. For each one of the properties chosen, a developer has to:
 - i. Modify the data type assigned by Hibernate tools during the mapping file generation and assign instead a Jasypt Hibernate type compatible with the original data type that was assigned to the property.
 - ii. Select a password, encryption algorithm, and key derivation cycles.
 - iii. While doing this, the user must be careful not to select properties that will be used as primary or foreign keys because this would break the relationships among database tables.
3. Finally update the configuration files to recognize the mapping files that were created and modified previously

The use of Crypto-Assistant changes the original procedure in the following form:

1. First, mapping files must be generated for the persistent classes using the “New Hibernate Mapping File (*.hbm)” wizard.
 - a. This involves selecting the classes for which we want to generate mapping files.
 - b. Selecting the fields that contain sensitive data.
 - c. Generating the mapping files

Most of the Crypto-Assistant functionality is not visible to the user. The only visible modification consists on the addition of a new page in the “New Hibernate Mapping File (*.hbm)” wizard. This wizard generates Hibernate XML mapping files taking as input a set of Java classes.

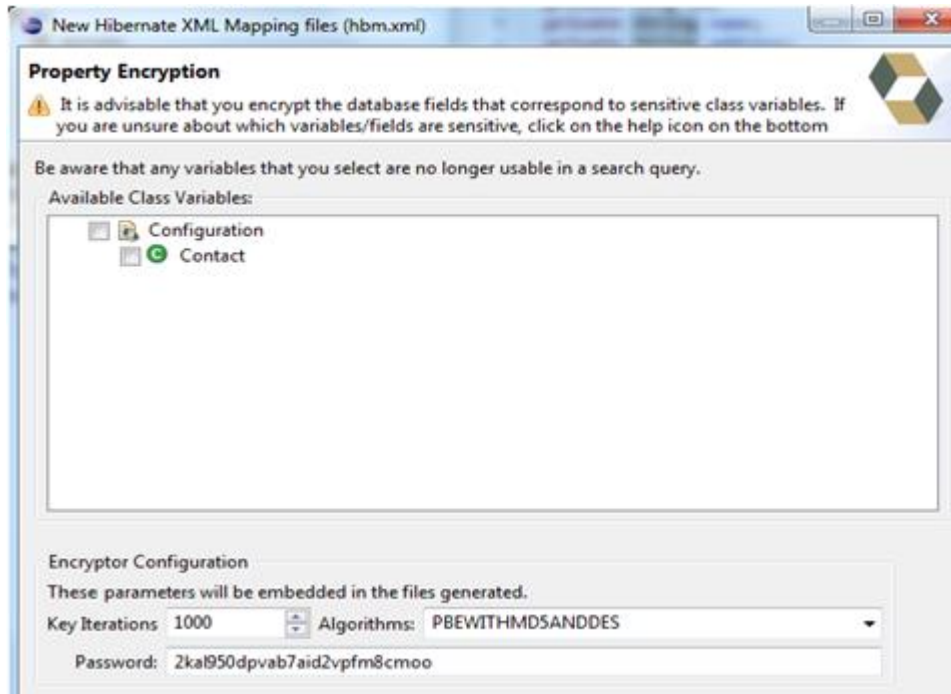


Figure 3-6 Property Encryption page added by the Crypto-Assistant.

The new page added by the Crypto-Assistant, Figure 3-6, presents a security warning whose intention is to raise awareness about the risk of storing sensitive data without encryption, and it offers a course of action to mitigate that risk, allowing developers to select the properties or fields of a class containing sensitive information. Crypto-Assistant uses password based encryption, where the encryption key is generated by applying a hash function to the password provided and at least 1000 times. On this screen, it is possible to configure the password, encryption algorithm, and key iterations used to generate the encryption key for the fields selected, the selection of these values was determined by the configuration parameters required by the Jasypt encryptor.

The prototype helps to reduce the chance of human error in several ways. More specifically, it hides properties such as the ones used as primary or foreign keys whose encryption would break the entity relations. This might be confusing if users are looking for these specific fields but it prevents them from breaking the relations in the database by mistake. For the encryption algorithm, AES and 3DES are the recommended algorithms by NIST; however, the default security policy of the JVM put limits on the cryptographic strength available by default. The process to enable stronger cryptography requires the manual installation of unrestricted policy files. Development of a tool to assist developers in the installation and the detection of this file requires a considerable effort; in consequence it was outside the scope of the prototype we present. The algorithm selected by default in the prototype is DES, this was done to provide an “out of the box” experience for the users, and avoid confusion about why the application would throw a run time exception related to security if a strong encryption algorithm is selected and the strong encryption policy files are not installed for the Java virtual machine. Other algorithms can be selected if available but a warning message will be shown in the wizard page if the algorithm selected is not AES which is the recommended one.

To avoid using a default password, a random one is generated every time the wizard is used. The passwords generated are stored in the mapping files. The developer is responsible for keeping track of the password in case the mapping files are regenerated using the wizard. . Optionally, the users can choose their own password. If the wizard is used to make modifications to the configuration files, a new password will be generated by default, users would have to enter it manually each time they make changes and want to keep the same encryption key.

Once the user decides to proceed to the preview page, heuristics are applied to assign a suitable encrypted type to each one of the properties selected. At the end the mapping files generated contain embedded configuration settings to allow Hibernate to use Jasypt's custom data types to perform the encryption and decryption of the selected properties. **Error! Reference source not found.** shows the preview screen of the mapping wizard with an encrypted type being used.

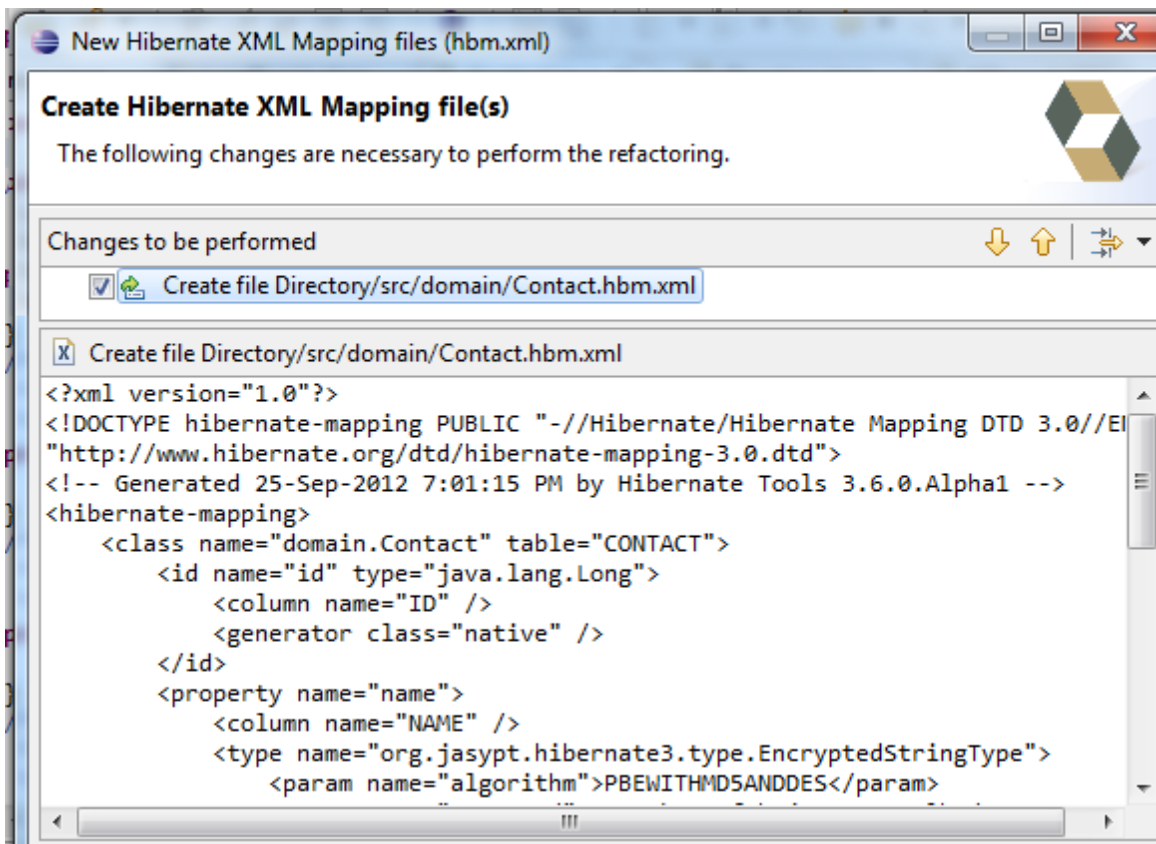


Figure 3-7 Mapping files wizard, preview screen showing the use of Jasypt Hibernate types for encryption.

The encryption passwords are stored in the mapping files. This is not the most secure approach; however, using the recommended strategy of separating the database server from the application server to protect the keys will provide protection for the data at rest in the event that an attacker gains access to it. Jasypt supports other options but the

complexity associated with them would not allow the simplification that we were trying to achieve. Key management is out of the scope of this prototype version and therefore it relies on the developer's effort to protect the encryption keys.

3.5 Installation

The installation of Crypto-Assistant is no different from any other Eclipse plugin. The best way to install it is to use the update manager. Once Crypto-Assistant is compiled it is packaged in a zip file that can be used as an update site.

- 1) Select Help > Install New Software. The install dialog will appear Figure 3-8

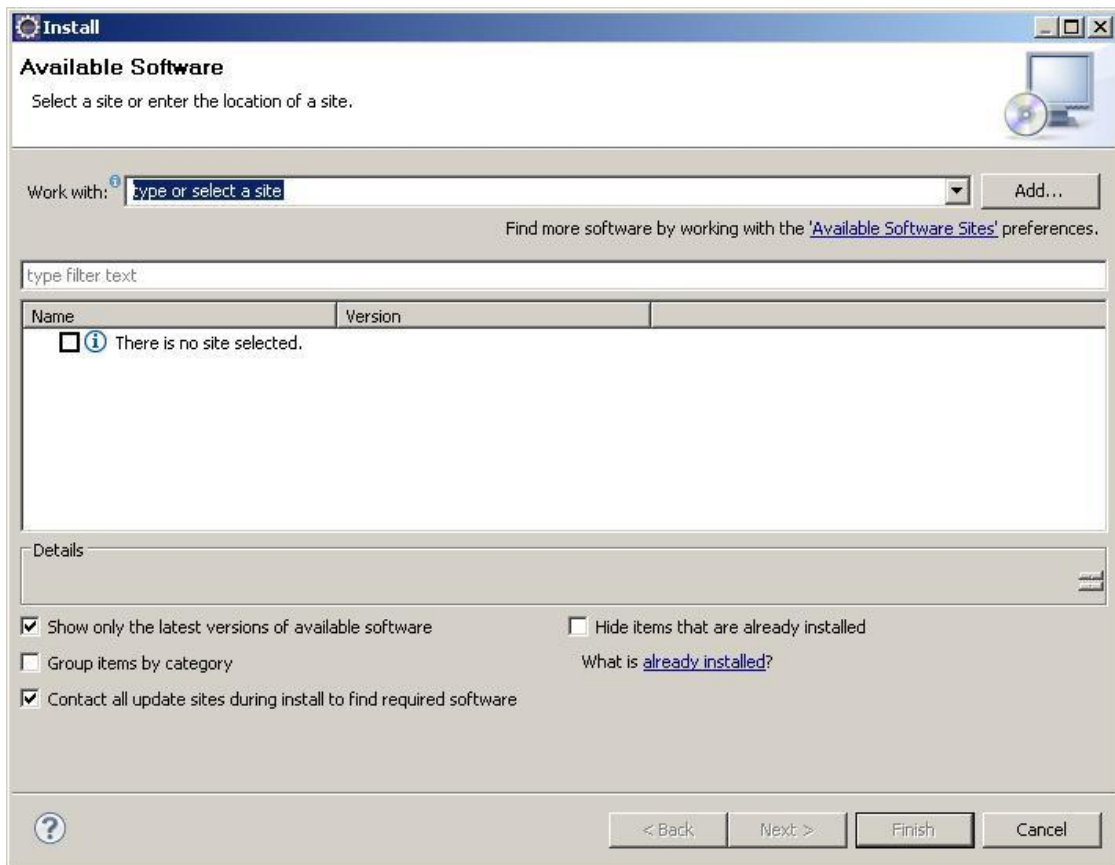


Figure 3-8 Install new software dialog.

- 2) Click **Add...** and type in the name and locate the zip file containing the update site for the Crypto-Assistant plugin, as in Figure 3-9.

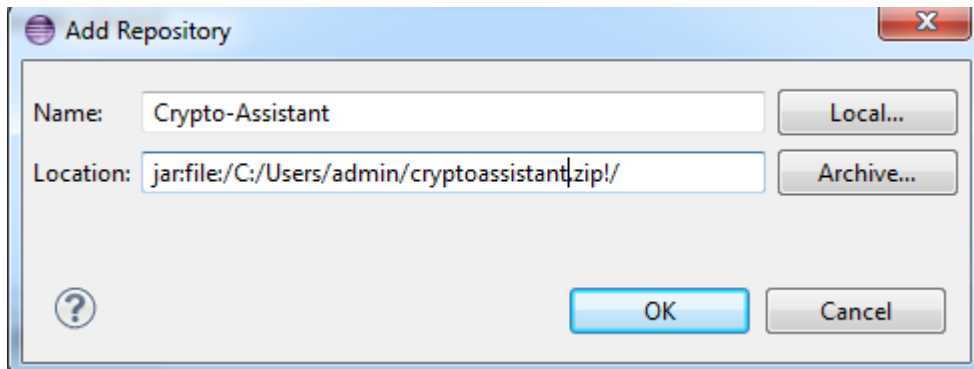


Figure 3-9 Add repository dialog.

- 3) Click ok and select the components to install from the window that appears Figure 3-10.

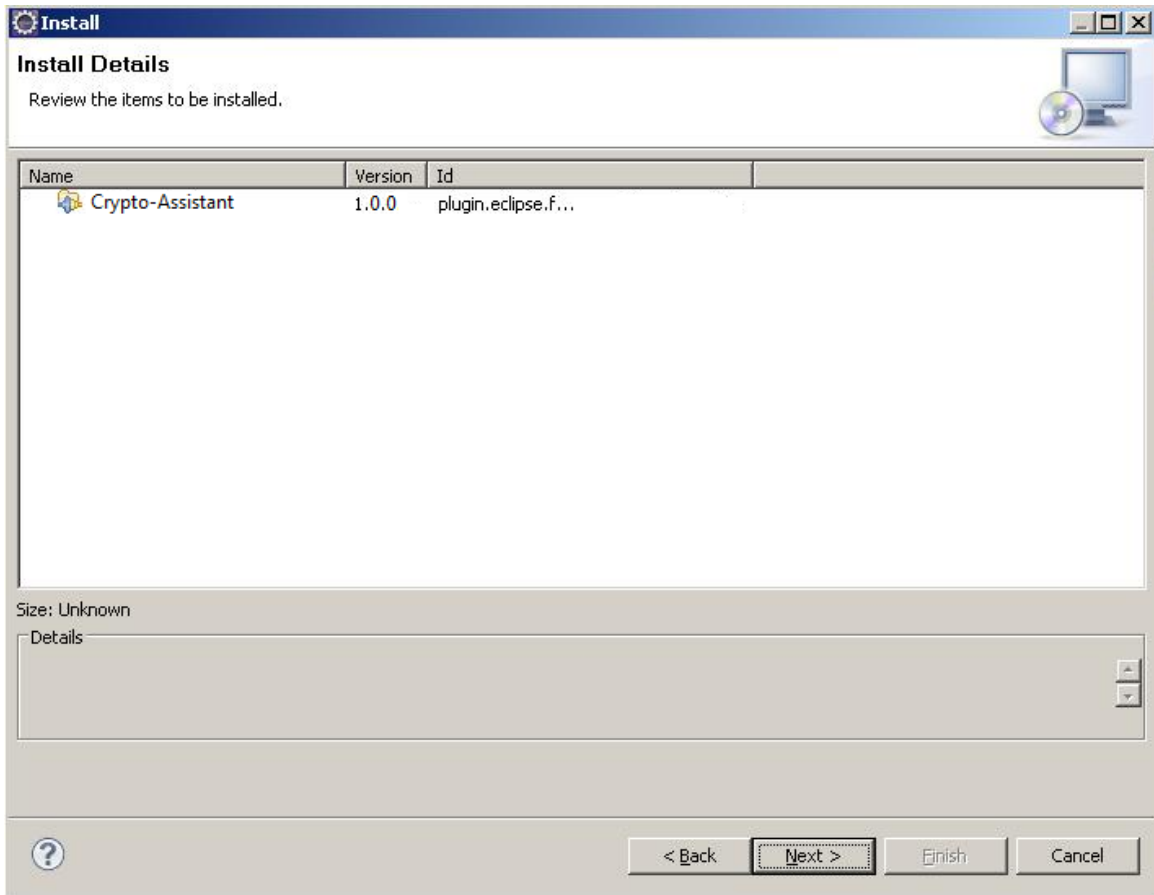


Figure 3-10 Installing Crypto-Assistant

- 4) Click the checkbox next to the update site you just added in this case is Crypto-Assistant. Click Next.
- 5) The dialog box in Figure 3-11 will appear. Click Yes and you will be ready to use Crypto- Assistant

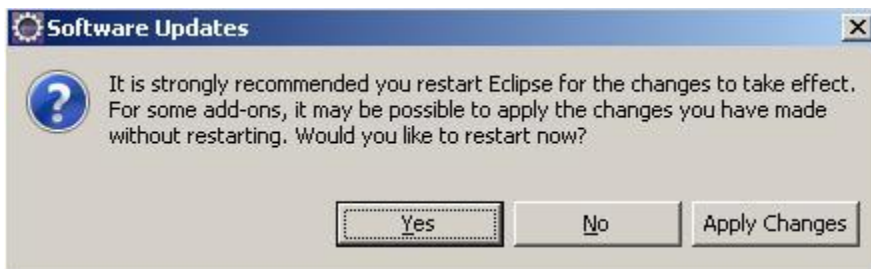


Figure 3-11 Restart eclipse dialog.

3.6 Usability evaluation

Usability is one of the main goals of the Crypto-Assistant. To evaluate the usability of our prototype we used several methods. *Learnability* was one of the main aspects of usability that we tried to address.

One of the most important aspects of the Crypto-Assistant is learnability. Learnability refers to the skills or knowledge that a new user requires in order to use the system effectively. Since our target audience were developers with little or no security training this aspect was important and required evaluation.

3.6.1 Cognitive Walk-through

The cognitive walk-through method described in [66], [56], allows to perform an evaluation of the *learnability* of our prototype without user intervention. To carry out this activity there are some prerequisites:

1. A general description of who the users will be and what relevant knowledge they possess.
2. A specific description of one or more representative tasks to be performed with the system.
3. A list of the correct actions required to complete each of these tasks with the interface being evaluated.

The targeted users are developers in general, who have little knowledge regarding security. The actions to be performed by the user are the selection of the properties that require encryption. The correct action is the expansion of entities and clicking on the properties that require protections marking them as checked.

The cognitive walk-through consists of an evaluator answering the following questions in a believable way.

1. Will the user try to achieve the right effect?

The display of a warning message has the purpose to influence the developer to incorporate the classification and selection of sensitive data for encryption as part of her current goals.

2. Will the user notice that the correct action is available?

The page does not explicitly indicate how to carry out the correct action. However, the selection area stands out from the other components by taking most of the space available suggesting some interaction must take place.

3. Will the user associate the correct action with the effect that the user is trying to achieve?

All of the controls and messages on the screen are associated with the protection of sensitive data throughout encryption. Therefore, one can assume that the user will associate the checking of the items in the tree with the protection of data. However, this might be an opportunity area to implement a visual metaphor, by placing a lock icon that would be open or closed to reflect the status of the check box.

4. If the correct action is performed, will the user see that progress is being made toward the solution of the task?

A check mark will be displayed next to the item checked.

We explored alternative behaviours that might be not completely satisfactory in section 3.2.2. For example users might not understand the messages or the behavior that is expected from them or they might simply skip the warning page and continue without reading the warnings. Despite the possible issues the answers to the questions posed by the cognitive walk-through appeared to be satisfactory, therefore, we assumed that there were not outstanding problems with respect to the learnability (skills/knowledge) required to use Crypto-Assistant.

3.7 Summary

In this chapter a brief introduction to the problem that motivated the creation of the Crypto-Assistant was given. The principal motivations to build this prototype are:

1. Test the hypotheses introduced in this chapter.
2. Make developers aware of the dangers to which sensitive data at rest is exposed.
3. Increase the usability of encryption to a level that any developer can use it effectively without having to be an expert on security or cryptography.

A brief overview of the main aspects to have in consideration at the moment of choosing a design strategy was presented. Then our prototype and the different components that form part of it were introduced. Finally, a small evaluation of its usability focusing in *learnability* was performed and no major problems were found.

4 CHAPTER 4 – PILOT USER STUDY

4.1 Introduction

In this research, several philosophical stands were adopted. First a positivist stand is adopted, this means that we believe the ideas we are testing can be reduced and analysed in an experimental setting and the results obtained from them can be applied to real situations. Software security deals with psychological and sociological aspects and therefore our model cannot take into account all of the different variables that influence this activity. Our theory and hypothesis may be incomplete and therefore we adopt a pragmatic approach. This stance tries to qualitatively [31] assess the feelings of the subjects that were exposed to our prototype, and to identify the most relevant factors for our purposes. The Crypto-Assistant was designed based on the hypothesis presented in the previous chapter, that is, to test this hypothesis; we designed an experiment with a scenario that resembles a typical situation that developers have to deal with and assigned them a task that involved the use of our prototype. Our product is still a prototype and it has many areas of opportunity to improve. The evaluation we performed is formative [25] in the hope to advance our knowledge and the final product with the results obtained. The plan is to make the source code available for review and use with an open source license.

4.2 Purpose/Background Information

The purpose of this experiment is to test the effectiveness of our prototype against the hypothesis formulated and presented in Chapter 3.

If our participants use the prototype to encrypt sensitive data we would have achieved the goal of influencing developers to produce more secure applications. The focus of the experiment is to learn about the usability of the Crypto-Assistant and more specifically, its effectiveness towards the aforementioned hypothesis. The following questions were formulated to help us in its evaluation and improvement.

1. Is it effective at encouraging the use of encryption as a protection mechanism?

2. Is it effective at raising awareness about the risk to which data at rest is exposed?
3. Are the features provided easy to understand and use?
4. In case the users do not use the encryption capabilities suggested and provided by the Crypto-Assistant, what is the reason?

Question three was answered in part by the learnability evaluation presented in the previous chapter. However this pilot user study was complementary to that evaluation.

4.3 EXPERIMENTAL DESIGN

In this section we will describe how we performed our evaluation of the Crypto-Assistant. Several difficulties presented that were not part of the original scope of this research. The first problem was the difficulty to find suitable candidates for our experiment. Even though Hibernate is a popular ORM tool used extensively in industry, it is not very popular among students or hobbyists that were the source of our research subjects. Therefore, we had to design a workshop that would prepare the research subjects to perform the task we had prepared for them. This was especially difficult due to the many features and complexity associated to Hibernate and the time constraints of the participants. Didactic material had to be created especially for the occasion which was not contemplated when we first envisioned the experiment.

A programming task had to be designed to resemble a realistic situation and in some way lead the participants to interact with our prototype. The task we envisioned was a maintenance task that involved the modification of a web application. Therefore we had to develop a small web application that would be easy to modify and needed to have the basic functionality of any popular application such as, for example, the registration of users, and the ability to log in/out and manipulation of users' data.

Once the design of the programming task was done we had to refine our experimental design to move out of the picture elements that were irrelevant to our purpose and could prevent the participants from completing the task. One of those elements was the difficulty to learn Hibernate in a short period of time. For this purpose we provided the participants with stub classes, and example source code, configuration

files, and a cheat sheet for quick reference. Another element we wanted to mitigate was the effect that the experimental setting could have on the responses of the participant. More specifically, we wanted to ensure that the participants did not simply please us with their answers and therefore we had to conceal the real purpose of the experiment.

4.3.1 Recruitment process

Participants were recruited from UOIT campus through the use of posters and an email directed to all the students at the university using the university distribution list for official announcements. More information about the participants is available in Appendix A.

4.3.1.1 Initial contact

Once a volunteer contacted the research personnel, they were sent an email along with the pre-screening consent form requesting them to complete an electronic screening questionnaire. The time required to complete the questionnaire was about 15 minutes and its purpose was to assess the eligibility of the volunteers. Upon completion of the questionnaire, the participants were informed about their eligibility via email by the research personnel.

4.3.1.2 Eligibility

Participants were selected based on the number of correct answers in the screening questionnaire. The total number of questions was 31, and the first eight questions helped us develop a profile of the participants, while the rest of the questions had the intention to gauge the participants' knowledge about SQL (Structured Query Language) & Java technology. From those questions, 11 were about SQL and with 12 about general Java knowledge, Servlets and JSP; this questionnaire along with the answers provided by the participants is included in Appendix A.

The criterion to select the participants was to have a minimum of 12 correct answers which amounted to more than 50% of the total number of questions about SQL and Java.

4.3.2 The study

The study consisted of two phases: (1) a workshop to provide the participants with the necessary information about Hibernate and Hibernate Tools plugin, and (2) the experiment, where participants had to perform a short programming task.

4.3.2.1 Workshop

The workshop took place at the UOIT North Campus the first week of October 2012. Participants were welcomed by a member of the research team and instructed to begin with the setup of their equipment. The session was started with a quick overview of the workshop, and the tools to be used. After this brief introduction, participants were instructed to set up the development environment needed for the workshop. Storage devices were handed out to participants loaded with the files required to participate in the session. The files included documents and source code to follow the workshop and perform the experiment.

The development environment was composed by a virtual machine loaded with Ubuntu 12.04, Eclipse IDE and MySQL database server. The eclipse IDE was preconfigured with our prototype.

During the workshop participants were not informed explicitly about the changes introduced by the Crypto-Assistant. This was covered during the workshop as if it was another feature of Hibernate and part of the process to generate mapping files with the help of Hibernate Tools. This was done to avoid contamination in the behaviour of the participants. At the end of the workshop, participants could choose to leave without compensation aside from the free lunch and knowledge gained, or continue and participate in the experiment.

4.3.2.2 Experiment

For the experiment, participants were required to modify a small web application to use Hibernate instead of JDBC (Java Database Connectivity), and perform any improvements they deemed necessary to improve the quality of the application. This was deliberate to

simulate a typical situation developers face when they have to meet functional requirements and deal with vague requirements, unfamiliarity with code and technology used, and time constraints.

There was an incentive for the top three applications of one gift cards with a value of \$150 for the best quality. A link to Wikipedia's article about software quality was provided as a reference. This had the goal to encourage them to look for possible defects including security ones.

The rules of the experiment were:

- They could use any resources from the internet.
- They could not communicate to any other person.
- They could not speak to each other.

The application to modify was small enough to be considered a toy program but was complex enough so they would struggle to understand the whole code at first sight. To help them to overcome this difficulty an overview of the application architecture was provided before they started the task and a list of the specific steps needed to complete the migration from JDBC to Hibernate was given to all of them, the application contained stub classes and an example method implementation using Hibernate was included to help them to understand what they have to do.

At the end of the study participants were compensated with \$30 each. A link to the exit questionnaire was emailed to them so they could answer it at their convenience and employ the time required to provide quality answers.

4.3.3 Data collection and evaluation

Data was collected using logs produced by the software, source code and questionnaires produced by the participants. Logs and questionnaires can be found in Appendix A.

The logs collected the interaction of users with the wizard page added by the Crypto-Assistant to the "New mapping file wizard". Any selection or manipulation of the interface within this screen produced an entry in the log file. Including in particular any

fields they selected for encryption, we planned to use this information to analyze what users did with the prototype.

A screening questionnaire was used to assess the suitability of the participants for the study. Another questionnaire was used at the end of the experimental phase to gauge the participants acceptance of the prototype. The software artefacts produced by the participants were analysed too.

4.3.4 Ethics

All of the experiments abide by the University of Ontario Institute of Technology Ethics Review process for experiments involving human participants. None of the participants were put at risk at any moment and they were informed of their right to withdraw from the beginning and through the course of the experiment.

4.4 Results

In our small pilot study we started with four participants that qualified through the process described before.

For several reasons including a fire drill and a building evacuation, the commencement of the workshop was delayed approximately 30 minutes. One of the participants did not have the equipment necessary but was provided with a laptop by the research personnel. Another participant had problems setting up the software necessary, and the research personnel tried to assist the participant in the set up but the cause of the error was unknown. After several delays and malfunctions the participant decided to withdraw from the experimental session. The workshop continued without any additional delays.

The duration of the workshop, including lunch, was estimated to be 2.5 hours. However, due to the multiple delays during the workshop this was extended to about 3.5 hours from the 4 that were originally allocated. All three participants decided to stay and continue with the experimental session. However, the time pressure became a great issue

because the task was complex enough to at least take them an hour. The experimental session required an introduction to explain the rules of the experiment and the architecture of the software to modify which took about 15 minutes of the half hour that was left. Because of this situation, participants were allowed to work at their discretion on the task, and all of them dedicated approximately one hour to complete the task.

We expected users would use the cheat sheet (Appendix B 8.9) as reference to carry out the task assigned. We assumed that the participants would select some of the sensitive fields during the interaction with the wizard to generate the mappings files. Analysing the logs collected, we discovered that the subjects did not interact with our prototype as we expected. The only interaction that appears in the logs is the examination of the combo box containing the list of encryption algorithms.

Through the answers extracted from the exit questionnaire we were able to extract some qualitative data from the participants:

- Two of them identified the difficulty of the task as average and one as easy; the source code collected from them corroborated this with its completeness level.
- Participants declared that none of them had received any formal training about security even when two of them had professional experience developing software and one was enrolled in a program related to security. However, the study took place at the beginning of the school year and the participant was a new student.
- One participant (P3) that correctly identified the application to being vulnerable to network attacks. The other two could not tell if it was vulnerable to any attacks.
- All of the participants were aware of the presence of sensitive data in the application. They identified password, credit card and social insurance numbers as the most sensitive information and two of them identified the entire table as sensitive for containing personal information.
- When asked what would be their suggestion to protect this data and only two provided an answer: encryption was suggested by both, but, one of them explicitly indicated Hibernate's encryption capabilities as a protective mechanism.

- Two of them qualified the difficulty of implementing encryption in their programs as average and the other one as easy.
- The lack of time and the focus on functionality was identified as the main reason for not using the features added by the prototype. One of the participants declared that she had the intention to go through it later.
- All participants had a good opinion about the usefulness of the encryption capabilities; their answers were measured using a likert scale with values that went from “not useful” (1) to “essential” (5), which is the maximum level. Their answers were for participant “useful” (3) to “very useful” (4) and “essential” (5).
- Only one subject (P3) used the contextual help button and found the information presented relevant and the difficulty to understand it as average.
- The easy encryption capabilities was one of the features that were well received by the users, one of the participants wrote: “I like how the tools had a simple way of implementing after the initial setup, as well as an easy way of adding encryption to sensitive user information.”
- When answering the question about what they did not like about the tools, we received only one answer referring to Hibernate basic functionality: “They were very clunky to use for a small program; there was a lot of setup for a small amount of payoff. But this is necessary for larger applications to make proper use of them.”

4.5 Analysis

With the result at hand, we prepared to answer the questions posed at the beginning. It is important to highlight that these answers are based in the observations extracted from this small pilot study and they are not definitive or intended to be generalized. This results are only applicable the situation described here and further study is needed to draw more general conclusions.

- Is Crypto-Assistant effective at encouraging the use of encryption as a protection mechanism?

Under the laboratory conditions described and with an external factor of extreme time pressure, the Crypto-Assistant will not be effective to encourage the use of encryption.

- Is it effective at raising awareness about the risk to which data at rest is exposed?

All the participants were aware about the threat of lack of encryption of data at rest and its potential disclosure to unauthorized parties.

- Are the features provided easy to understand and use?

Even when the participants did not make use of the encryption features, one stated that one of the features she liked most was how easy it was to add encryption. These results along with the learnability evaluation performed suggest that user acceptance and effortlessness of use was attained.

- In case the users do not use the encryption capabilities suggested and provided by the Crypto-Assistant, what is the reason?

Time constraints were mentioned by all the participants, this element plays an important external factor that was not in our consideration through the development of the prototype.

4.5.1 Lessons learned and experimental limitations

There are some limitations with the approach of our experimental design. In this section we try to acknowledge the most relevant and explain how they might have influenced the results we observed and what was learnt from this experience.

The main limitations are: the limited number of participants in the study and the design of the experiment itself. Documentation about design and test of security tools is still scarce in consequence we had to develop our own methodology. Our ad-hoc approach was more focused on the testing of the hypothesis presented than in the improvement of the tool we were developing. It would have been better to first focus only in the development and evaluation of the prototype and then with the prototype ready, focus on the hypothesis test.

The prototype design is another factor to consider. While designing the prototype, usability was the top most priority. We did not take into consideration the effect of external factors in users' goals. Even when we were aware of them and tried to use one in the form of an incentive offered with the purpose of including security indirectly as one of the participant's goals. A redesign of the prototype would include making explicit the use of encryption by adding a new menu item to Eclipse user interface indicating clearly that encryption will be available. By indicating explicitly the use of encryption before even starting the process; we align with Witten's [82] well in advance principle mentioned in section 2.4.3. The selection of this explicit menu item would imply the intention of the user to protect the data with encryption aligning the purpose of the prototype with the user intentions.

The developers' goal was to perform the migration from JDBC to Hibernate and it was stated that this goal was the main task of the experiment. It was also mandatory to complete it in order to be eligible for one of the gift cards; therefore, the warning presented might have been perceived as an interruption that was on their way to finish the task requested. The use of encryption to protect the data was not an explicit goal. A new study comparing the performance of participants with the explicit goal of encrypting data with and without the support of the Crypto-Assistant prototype would shed new light on the efficiency of the prototype.

The theoretical "information disclosure" risk might have been perceived as non-existent due to the experimental nature of the task. Even when we tried to recreate a realistic setting the participants knew that it was just an experiment and the release of the data in the prototype would not affect them directly.

The limitation in time was important too as the results obtained may differ if time pressure was not a factor. By adding a time constraint, the subjects had to optimize the resources they had. In this case the alternative presented to mitigate the risk involved the allocation of time to perform the risk mitigation task. Participants might have decided that the cost associated to perform the risk defusing task was not worth the potential benefit since there was no real threat and this was not an explicit requirement. Participants had

explicit functional goals to meet and missing those goals represented a greater risk in the context of the experiment. The time constraint was a determinant factor to prioritize functionality over any other feature.

The unfamiliarity of participants with the prototype is another factor to consider. Participants' lack of experience with the technologies and functions added by the prototype might be significant for the results of the experiment. This might have been an issue but, the demographics we were targeting justify this condition.

There is also the threat of over encryption. Users might find that all the fields in a table are sensitive risking to over encrypting data which might render it unsearchable. This problem was not addressed and it is still present in the final version of the prototype.

4.5.2 Implications of the results obtained from the pilot user study

The results show that user goals are hard to change and external factors such as time constraints are an issue and suggest that a redesign must take place to improve the effectiveness of the Crypto-Assistant.

In an effort to better understanding of what parts required more work, we found that the security threat model presented in [41] was ideal to evaluate our prototype. This model presents several factors that affect the security of a system and helps to evaluate if the system contributes to insecure behaviours by evaluating the security and usability in a user-centric way. By using that model it was possible to determine that there were some threats to security that our prototype was not addressing. Three of the security factors that are part of the model apply directly to the design of our prototype:

Vigilance—secure systems tend to expect users to be alert and proactive in assessing the security state of a system. Even experts (people who understand the working of a secure system) are not always alert. Tasks that pose this security risk tend to be those that require users to divert attention from a primary task in order to attend to a security task. Such tasks should be analysed and integrated into users' workflow or eliminated if possible.

The prototype tried to incorporate encryption into the workflow of developers, in a non-intrusive way. This approach requires a user to be vigilant and proactive to defuse the information disclosure risk.

Motivation—users have different levels of motivation to perform security tasks in different circumstances. Participants would be more motivated to perform a risk defusing operation if they perceive that a risk affects them more directly than in a case where the risk is perceived to be low or directed at someone else.

As mentioned before the lab setting and the experimental nature of the activity might be determined in the perception of participants about the risk mentioned in the warning. Time constraints also affected how the participants responded to the stimulus presented.

Conditioning—repetitive security tasks for which users can predict an outcome can become a threat to the security of a system. A security-usability analysis of a system should assess whether security tasks have the potential for condition users.

By using a warning we preconditioned the behaviour of the participants. People are used to dismiss warnings, a behaviour that could be explained by great exposure to many ineffective security warnings on computer systems.

In the light of these results the need of a change in the design of our prototype is required. A solution to the defects detected in the version evaluated might be mitigated by adopting a different stand.

Mitigation of the threats identified can be achieved through separation of concerns making awareness and functionality separate goals. The incorporation of a new menu item that explicitly enables the functionality added by the prototype addresses the three issues detected. First, it would make the prototype compatible with the principle of “well in advance” [82] information introduced by Whitten. The explicitness implies motivation from part of the user. It also defuses the need of vigilance and by moving away from the warning design conditioning is also addressed. This change in the design and approach taken comes with a change in the profile of the target users. The users must be willingly

and proactively looking for the incorporation and use of encryption in their applications which implies that they already performed an assessment and decided to use Jasypt in their database encryption strategy. The increase of awareness can be achieved through the same warning strategy adopted in this version and described in section 3.2.2. Embedding warnings in the workflow of other tools whose activities might be related to certain design risks and display security warnings at relevant points having in consideration that the earlier an error is detected the cheaper is to fix.

4.6 Summary

Despite the limitations in the realization of the user study, it provided valuable information about the developers' mental model. The data collected shows that in general the prototype had good acceptance. Nevertheless, encryption was not used and despite the usability improvements the results suggest it was perceived as a time consuming task that was not aligned with the participants' functional goals. An evaluation of the results obtained suggested that a change in the design might be beneficial.

5 CHAPTER 5 – DISCUSSION AND CONCLUSION

5.1 Introduction

In this chapter we will discuss the results of the user study and the contributions of our work in addition to providing some suggestions for future directions for researchers in the field. The user study we performed showed that even when the participants received the functionality provided by the prototype with enthusiasm and were made aware of the risk of storing sensitive information in clear text by the warning presented, they did not use it to encrypt any data due to a lack of time.

5.2 Discussion

The results of the prototype evaluation suggest that the prototype was not effective in improving the use of encryption under time constraints. Functional requirements have a higher priority for developers over security concerns. The warning presented by the prototype was not able to persuade the participants to mitigate the risk they were being informed of, with the suggested strategy, even with the learnability and efficiency added by the tool.

The experiments carried by Xie et al. [87] with professional developers influenced the development of our evaluation methodology and therefore have several similarities. Instead of a virtual machine, a laptop was set up and loaded with their prototype. Their ASIDE (Assured Software Integrated Development Environment) prototype performed static analysis on participants' code and presented several warnings for problems detected. The fact that the purpose of the experiment was to test the prototype was hidden from participants. Their participants were assigned the task to build an online stock trading system; a project with basic functionality was included to help developers getting started.

The results from [87] are similar to the ones we collected. They show that warnings had some success raising awareness about possible errors. In total, 22% to 27%

of the warnings presented were clicked on. Despite that, warnings were not successful in protecting user or their systems from the risk they were intended to prevent or mitigate. In both experiments participants showed their willingness to address those concerns if they have had more time. These results suggest that environmental factors such as time constraints have a significant and detrimental effect on software security and should be considered in the design of a security tool.

From a psychological standpoint, there are several experiments that examine human behaviour on risky decisions [34],[19],[46]. Some of these studies focus on the effects of time pressure on the search for risk defusing operators [35]. A risk defusing operator (RDO) is an action intended by the decision maker to be performed in addition to an otherwise attractive alternative and expected to decrease the risk. However, their approach is different from ours, since those situations are purely hypothetical and do not require to perform any action which reduces their realism. In our experiment we had two risks:

1. Disclosure of sensitive data, which was presented through the warning.
2. Failure to deliver the required functionality that was given by the task context.

The results of our experiment suggest that people perceived as a greater risk to their immediate goals, not having completely implemented the functionality we requested them. This is understandable because there is no benefit on fixing the security of a product that does not fulfil its functional requirements. These results could be explained by the findings of Kocher et al. [46] which found that when there are mixed losses and gains involved at the same time, subjects become more loss averse and more gain seeking under time pressure, depending on the framing of the prospects. Their results suggest the importance of goals or aspiration levels, as they refer to them, under time pressure.

These findings reveal that the use of warnings to encourage the use of the functionality provided by our prototype might have not been the best approach. Warnings impose limitations about the amount and type of information that can be presented at once to the user. The adoption of a security solution must be carefully evaluated to assess if it provides the right protection and the trade-offs in usability are acceptable for the users of

the system. This type of assessment is difficult to perform when participants are in a rush to achieve a functional goal and the information presented to them is limited.

5.3 Future research

There are some avenues for future work in the functionality of the Crypto-Assistant, In future research the lessons learned and documented in section 4.5.1 would need to be applied to improve the Crypto-Assistant such as making explicit in the user interface that encryption is available.

The evaluation methodology can be improved conducting a performance comparison, e.g. assigning the task to encrypt entity data to a set of participants both with and without Crypto-Assistant support. This approach would more accurately evaluate the benefits provided by our prototype. Another change of the evaluation would be to focus exclusively on usability problems and the improvement of the tool, using other techniques such as Cranor’s “the human in the loop” security framework [13] to analyze and improve the design of the prototype.

The prototype can be improved by adding support for key management and key rotation. There is already some progress done to move the definition of the encrypted types to a separate file that would be managed by the Crypto-Assistant to minimize the exposure of the encryption keys.

5.4 Conclusion

The technical community has underestimated the security problem. We feel this work was not the exception. The development and testing of the Crypto-Assistant was more demanding and complex than we initially anticipated. However, the results and experience gained through the whole process was worth overcoming all the existing difficulties.

The main goals of this research were producing a tool for security and help in the application of the “Built Security In” concept. We achieved those goals with the Crypto-Assistant. Another goal was to help the future developers of security. We cannot say we

achieved this goal yet but hope that with the theoretical framework introduced in Chapter 2; the design process in Chapter 3; and the user study in Chapter 4; we have condensed enough knowledge to serve as reference point for other developers.

The material condensed in Chapter 2, the documentation of the development process, the design and the source code of Crypto-Assistant along with the data collected from the pilot study presented here are the contributions of this thesis. It is important that to remark that the results presented here are not conclusive, they are based in a small sample and are not intended to be generalized but rather a point towards ways that the design of Crypto-Assistant and the experiment can be improved, as previously mentioned in section 5.4. In the future the Crypto-Assistant code will be released for the benefit of the community. Future work includes more realistic evaluations with actual users, and work in areas that were left out of the scope of the work presented here, .i.e., key management, integration of visual metaphors and evaluation of the changes suggested by this initial evaluation.

6 Bibliography

- [1] M. Baddeley, “Herding, social influence and economic decision-making: socio-psychological and neuroscientific analyses,” *Philos Trans R Soc Lond B Biol Sci*, vol. 365, no. 1538, pp. 281–290, Jan. 2010.
- [2] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, “NIST SP800-57: Recommendation for Key Management – Part 1: General(Revised),” Mar. 2007 [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf. [Accessed: 20-Nov-2012]
- [3] A. Bechara, “The role of emotion in decision-making: Evidence from neurological patients with orbitofrontal damage,” *Brain and Cognition*, vol. 55, no. 1, pp. 30–40, Jun. 2004.
- [4] R. BisbeyII. and D. Hollingworth, “Protection Analysis: Final Report,” May 1978.
- [5] M. Bishop, *Computer Security: Art and Science*, 1st ed. Addison-Wesley Professional, 2002.
- [6] M. Bishop, “What is computer security?,” *IEEE Security Privacy*, vol. 1, no. 1, pp. 67 – 69, Feb. 2003.
- [7] D. Bolchini and P. Paolini, “Capturing Web Application Requirements through Goal-Oriented Analysis,” *PROCEEDINGS OF THE WORKSHOP ON REQUIREMENTS ENGINEERING (WER 02)*, pp. 16–28, 2002.
- [8] C. Bravo-Lillo, L. Cranor, J. Downs, and S. Komanduri, “Bridging the gap in computer security warnings: a mental model approach,” *Security & Privacy, IEEE*, no. 99, pp. 1–1, 2011.
- [9] “Build Security In Home,” Available: <https://buildsecurityin.us-cert.gov/bsi/home.html>. [Accessed: 18-Oct-2012]
- [10] “Category:OWASP Top Ten Project - OWASP,” Available: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project. [Accessed: 04-Nov-2012]
- [11] “Category:OWASP WebGoat Project - OWASP,” Available: https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project. [Accessed: 10-Dec-2012]
- [12] “CIA Triad «CIPP Guide,” Available: <https://www.cippguide.org/2010/08/03/cia-triad/>. [Accessed: 22-Oct-2012]
- [13] L.F. Cranor, “A framework for reasoning about the human in the loop,” in *Proceedings of the 1st Conference on Usability, Psychology, and Security*, Berkeley, CA, USA, 2008, pp. 1:1–1:15 [Online]. Available: <http://dl.acm.org/citation.cfm?id=1387649.1387650>. [Accessed: 20-Aug-2012]

- [14] “CWE - 2011 CWE/SANS Top 25 Most Dangerous Software Errors,” Available: <http://cwe.mitre.org/top25/index.html#CWE-311>. [Accessed: 06-Nov-2012]
- [15] “CWE - CWE-311: Missing Encryption of Sensitive Data (2.3),” Available: <http://cwe.mitre.org/data/definitions/311.html>. [Accessed: 06-Nov-2012]
- [16] “CWE - CWE-312: Cleartext Storage of Sensitive Information (2.3),” Available: <http://cwe.mitre.org/data/definitions/312.html>. [Accessed: 06-Nov-2012]
- [17] “CWE - VIEW SLICE: CWE-2000: Comprehensive CWE Dictionary (2.3),” Available: <http://cwe.mitre.org/data/slices/2000.html>. [Accessed: 04-Nov-2012]
- [18] A.K. Dalai and S.K. Jena, “Evaluation of web application security risks and secure design patterns,” in *Proceedings of the 2011 International Conference on Communication, Computing & Security - ICCCS '11*, Rourkela, Odisha, India, 2011, p. 565 [Online]. Available: <http://dl.acm.org.uproxy.library.dcuoit.ca/citation.cfm?id=1948057>. [Accessed: 31-Aug-2011]
- [19] I.E. Dror, J.R. Busemeyer, and B. Basola, “Decision making under time pressure: an independent test of sequential sampling models,” *Memory & Cognition*, 1999 [Online]. Available: <http://eprints.soton.ac.uk/18349/>. [Accessed: 19-Nov-2012]
- [20] “Eclipse Platform Technical Overview,” Available: <http://www.eclipse.org/resources/resource.php?id=131>. [Accessed: 29-Nov-2012]
- [21] EMC, “Approaches for Encryption of Data-at-Rest in the Enterprise: A Detailed Review| Whitepapers | TechRepublic.” [Online]. Available: <http://www.techrepublic.com/whitepapers/approaches-for-encryption-of-data-at-rest-in-the-enterprise-a-detailed-review/1007473>. [Accessed: 29-Nov-2012]
- [22] S. Faily, “A framework for usable and secure system design,” PhD, University of Oxford, 2011 [Online]. Available: http://oxford.academia.edu/ShamalFaily/Papers/728979/A_framework_for_usable_and_secure_system_design. [Accessed: 23-Aug-2012]
- [23] E.B. Fernandez, N. Yoshioka, H. Washizaki, and M. VanHilst, “Measuring the Level of Security Introduced by Security Patterns,” in *ARES '10 International Conference on Availability, Reliability, and Security, 2010*, 2010, pp. 565–568.
- [24] M. Finifter and D. Wagner, “Exploring the Relationship Between Web Application Development Tools and Security,” in *Proceedings of the 2nd USENIX Conference on Web Application Development. USENIX (June 2011)*, 2011 [Online]. Available: http://www.usenix.org/event/webapps11/tech/final_files/webapps11_proceedings.pdf#page=107. [Accessed: 16-Sep-2012]
- [25] “Formative and Summative Evaluations in the Instructional Design Process,” Available: http://www.nwlink.com/~donclark/hrd/isd/types_of_evaluations.html. [Accessed: 07-Nov-2012]
- [26] S. Furnell, “Why users cannot use security,” *Computers & Security*, vol. 24, no. 4, pp. 274–279, Jun. 2005.

- [27] “GE Money Alerting Clients About A Data Security Breach. These Guys Act Like Pros, No Matter What - AlertBoot Endpoint Security,” Available: http://www.alertboot.com/blog/blogs/endpoint_security/archive/2008/01/08/ge-money-alerting-clients-about-a-data-security-breach-these-guys-act-like-pros-no-matter-what.aspx. [Accessed: 07-Nov-2012]
- [28] P. Gutmann, “Engineering Security,” New Zealand, May-2012 [Online]. Available: <http://www.cs.auckland.ac.nz/~pgut001/pubs/book.pdf>. [Accessed: 26-Oct-2012]
- [29] P. Gutmann and I. Grigg, “Security Usability,” *IEEE Security Privacy*, vol. 3, no. 4, pp. 56 – 58, Aug. 2005.
- [30] P.J. Hammond, “6 SUBJECTIVE EXPECTED UTILITY,” *Handbook of Utility Theory: Volume 1: Principles*, p. 213, 1999.
- [31] O. Hazzan, “Qualitative Research in Software Engineering” [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.137.7613&rep=rep1&type=pdf>. [Accessed: 16-Sep-2012]
- [32] “Hibernate - JBoss Community,” Available: <http://www.hibernate.org/>. [Accessed: 12-Mar-2013]
- [33] “Hibernate Tools - JBoss Community,” Available: <http://www.hibernate.org/subprojects/tools.html>. [Accessed: 03-Dec-2012]
- [34] O. Huber, “Behavior in risky decisions: Focus on risk defusing,” *Uncertainty and risk*, pp. 291–306, 2007.
- [35] O. Huber and U. Kunz, “Time pressure in risky decision-making: effect on risk defusing,” *Psychology Science*, vol. 49, no. 4, p. 415, 2007.
- [36] “IEEE Standard Glossary of Software Engineering Terminology,” *IEEE Std 610.12-1990*, p. 1, 1990.
- [37] Ion Ivan and Luckacs Breda, “Informatics Security Metrics Comparative Analysis,” *Informatica Economica*, vol. XI, no. 4, pp. 107–110, 2007.
- [38] *ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 9: Requirements for non-keyboard input devices*, 2000.
- [39] “ISO/IEC 27002 code of practice,” Available: <http://www.iso27001security.com/html/27002.html>. [Accessed: 25-Oct-2012]
- [40] “Jasypt: Java simplified encryption - Main,” Available: <http://www.jasypt.org/>. [Accessed: 16-Sep-2012]
- [41] R. Kainda, I. Flechais, and A.W. Roscoe, “Security and Usability: Analysis and Evaluation,” in *ARES '10 International Conference on Availability, Reliability, and Security, 2010*, 2010, pp. 275 –282.
- [42] R. Kissel, P.D. Gallagher, and D. Introduction, *Revision 1 Glossary of Key Information Security Terms*. 2011.

- [43] G.A. Klein, "A recognition-primed decision (RPD) model of rapid decision making," in *Decision making in action: Models and methods*, G.A. Klein, J. Orasanu, R. Calderwood, and C.E. Zsombok, Eds. Westport, CT, US: Ablex Publishing, 1993, pp. 138–147.
- [44] G.A. Klein, "Recognition-Primed Decisions.," KLEIN ASSOCIATES INC YELLOW SPRINGS OH, 1998 [Online]. Available: <http://en.scientificcommons.org/18500209>. [Accessed: 26-Oct-2012]
- [45] A.J. Ko and B.A. Myers, "A framework and methodology for studying the causes of software errors in programming systems," *Journal of Visual Languages & Computing*, vol. 16, no. 1–2, pp. 41–84, Feb. 2005.
- [46] M.G. Kocher, J. Pahlke, and S.T. Trautmann, "Tempus Fugit: Time Pressure in Risky Decisions," University of Munich, Department of Economics, Discussion Papers in Economics 12221, 2011 [Online]. Available: <http://ideas.repec.org/p/lmu/muenec/12221.html>. [Accessed: 19-Nov-2012]
- [47] E. Kurz-Milcke and G. Gigerenzer, "Heuristic decision making," *Marketing JRM*, no. 1, pp. 48–60, 2007.
- [48] Markus Schumacher, "Security Patterns," 12-Jul-2005. [Online]. Available: <http://www.securitypatterns.org/patterns.html>. [Accessed: 19-Aug-2011]
- [49] Microsoft Corporation, *Improving Web Application Security: Threats and Countermeasures*, 1st ed. Microsoft Press, 2003.
- [50] S. Mutti, M.A. Neri, and S. Paraboschi, "An Eclipse plug-in for specifying security policies in modern information systems" [Online]. Available: <http://digiway.novasemantics.it/attach/MuArPa11/eclipseIt11.pdf>. [Accessed: 05-Nov-2012]
- [51] S. Myagmar, A.J. Lee, and W. Yurcik, "Threat modeling as a basis for security requirements," in *Symposium on Requirements Engineering for Information Security (SREIS)*, 2005 [Online]. Available: <http://craigchamberlain.com/library/security/Threat%20Modeling%20as%20a%20Basis%20for%20Security%20Requirements.pdf>. [Accessed: 26-Nov-2012]
- [52] "MyEclipse for Spring: Spring MVC Scaffolding," Available: <http://www.myeclipseide.com/documentation/quickstarts/scaffoldingtutorial/scaffolding.html>. [Accessed: 16-Jan-2013]
- [53] "OWASP ASIDE Project - OWASP," Available: https://www.owasp.org/index.php/OWASP_ASIDE_Project. [Accessed: 06-Nov-2012]
- [54] "OWASP LAPSE Project - OWASP," Available: https://www.owasp.org/index.php/OWASP_LAPSE_Project. [Accessed: 08-Nov-2012]
- [55] J. Reason, *Human Error*, 1st ed. Cambridge University Press, 1990.

- [56] J. Rieman, M. Franzke, and D. Redmiles, "Usability evaluation with the cognitive walkthrough," in *Conference companion on Human factors in computing systems*, 1995, pp. 387–388 [Online]. Available: <http://dl.acm.org.uproxy.library.dc-uoit.ca/citation.cfm?id=223735>. [Accessed: 05-Dec-2012]
- [57] G.W. Romney, C. Higby, B.R. Stevenson, and N. Blackham, "A teaching prototype for educating IT security engineers in emerging environments," in *Information Technology Based Higher Education and Training, 2004. ITHET 2004. Proceedings of the Fifth International Conference on*, 2004, pp. 662–667.
- [58] B. Schneier, *Secrets and lies: digital security in a networked world*. John Wiley, 2000.
- [59] M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann, and P. Sommerlad, *Security Patterns: Integrating Security and Systems Engineering*. John Wiley & Sons, 2006.
- [60] "SDL Threat Modeling Tool," Available: <http://www.microsoft.com/security/sdl/adopt/threatmodeling.aspx>. [Accessed: 11-Dec-2012]
- [61] "Securing Data at Rest: Developing a Database Encryption Strategy," RSA Security Inc, 2002 [Online]. Available: http://www.rsa.com/products/bsafe/whitepapers/DDES_WP_0702.pdf. [Accessed: 19-Sep-2012]
- [62] "security," *Department of Defense Dictionary of Military and Associated Terms*. [Online]. Available: http://www.dtic.mil/doctrine/dod_dictionary/data/s/6926.html. [Accessed: 20-Oct-2012]
- [63] H. Simon, *Reason in Human Affairs*. Stanford University Press, 1990.
- [64] B. Snow, "We need assurance![assurance of computing quality, reliability, and safety]," in *Computer Security Applications Conference, 21st Annual*, 2005, p. 7–pp [Online]. Available: http://ieeexplore.ieee.org.uproxy.library.dc-uoit.ca/xpls/abs_all.jsp?arnumber=1565230. [Accessed: 29-Nov-2012]
- [65] "Software testing (chapt.5)," in *Guide to the Software Engineering Body of Knowledge SWEBOK*, .
- [66] R. Spencer, "The streamlined cognitive walkthrough method, working around social constraints encountered in a software development company," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2000, pp. 353–359 [Online]. Available: <http://dl.acm.org.uproxy.library.dc-uoit.ca/citation.cfm?id=332456>. [Accessed: 03-Dec-2012]
- [67] J.G. Spohrer and E. Soloway, "Analyzing the high frequency bugs in novice programs," in *Papers presented at the first workshop on empirical studies of programmers on Empirical studies of programmers*, Norwood, NJ, USA, 1986, pp. 230–251 [Online]. Available: <http://dl.acm.org/citation.cfm?id=21842.28897>. [Accessed: 30-Oct-2012]

- [68] G. Stoneburner, "SP 800-33. Underlying Technical Models for Information Technology Security," National Institute of Standards & Technology, Gaithersburg, MD, United States, Dec. 2001 [Online]. Available: <http://csrc.nist.gov/publications/nistpubs/800-33/sp800-33.pdf>. [Accessed: 22-Oct-2012]
- [69] G. Stoneburner, A. Goguen, and A. Feringa, "Risk Management Guide for Information Technology Systems," 800-30, Jul. 2002 [Online]. Available: <http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf>. [Accessed: 26-Nov-2012]
- [70] G. Tasse, "The economic impacts of inadequate infrastructure for software testing," 2002.
- [71] "TD Bank says it 'worked diligently to find' lost tapes | The Morning Sentinel, Waterville, ME," Available: http://www.onlinesentinel.com/news/td-bank-says-it-worked-diligently-to-find-lost-tapes_2012-10-10.html. [Accessed: 07-Nov-2012]
- [72] "Team Shatter," Available: <http://www.teamshatter.com/>. [Accessed: 16-Sep-2012]
- [73] "The Big Cost of Software Bugs: When Coding Goes Awry," *Bloomberg*. [Online]. Available: <http://www.bloomberg.com/slideshow/2012-08-03/the-big-cost-of-software-bugs.html>. [Accessed: 30-Oct-2012]
- [74] "The Building Security In Maturity Model (BSIMM)," Available: <http://bsimm.com/>. [Accessed: 14-Nov-2012]
- [75] "The CIA principle," Available: <http://www.doc.ic.ac.uk/~ajs300/security/CIA.htm>. [Accessed: 22-Oct-2012]
- [76] "The CIA Triad | TechRepublic," Available: <http://www.techrepublic.com/blog/security/the-cia-triad/488>. [Accessed: 22-Oct-2012]
- [77] "Top 10 2010-A7-Insecure Cryptographic Storage - OWASP," Available: https://www.owasp.org/index.php/Top_10_2010-A7. [Accessed: 07-Nov-2012]
- [78] "TRICARE discloses SAIC breach: stolen backup tapes held data on 4.9 million (updated): Office of Inadequate Security," Available: <http://www.databreaches.net/?p=20816>. [Accessed: 07-Nov-2012]
- [79] K. Tsipenyuk, "Seven pernicious kingdoms: A taxonomy of software security errors," in *NIST Workshop on Software Security Assurance Tools, Techniques, and Metrics, November, 2005*, 2005, pp. 36–43.
- [80] K. Tsipenyuk, B. Chess, and G. McGraw, "Seven pernicious kingdoms: a taxonomy of software security errors," *IEEE Security Privacy*, vol. 3, no. 6, pp. 81 – 84, Dec. 2005.
- [81] "USEC '12," Available: <http://infosecnet/usec12/index.php>. [Accessed: 05-Nov-2012]

- [82] A. Whitten, “Making Security Usable,” PhD, Carnegie Mellon, 5000 Forbes Avenue Pittsburgh, PA 15213-3890, 2004 [Online]. Available: <http://www.gaudior.net/alma/MakingSecurityUsable.pdf>. [Accessed: 22-Aug-2012]
- [83] A. Whitten and J.D. Tygar, “Why Johnny can’t encrypt: A usability evaluation of PGP 5.0,” in *Proceedings of the 8th USENIX Security Symposium*, 1999, vol. 99 [Online]. Available: http://www.usenix.org/events/sec99/full_papers/whitten/whitten.ps. [Accessed: 23-Aug-2012]
- [84] D.L. Williams, “A (Partial) Introduction to Software Engineering Practices and Methods,” *NCSU CSC326 Course Pack*, vol. 2009, 2008 [Online]. Available: <https://online.ist.psu.edu/sites/ist412/files/williamstext.pdf>. [Accessed: 27-Oct-2012]
- [85] T. Wilson, “Security Still An Afterthought, Study Says - Dark Reading,” *Dark Reading*, 04-Nov-2011. [Online]. Available: <http://www.darkreading.com/security-monitoring/167901086/security/application-security/231902431/security-still-an-afterthought-study-says.html>. [Accessed: 05-Nov-2012]
- [86] J. Xie, B. Chu, and H. Richter Lipford, “Idea: interactive support for secure software development,” *Engineering Secure Software and Systems*, pp. 248–255, 2011.
- [87] J. Xie, H. Lipford, and B.-T. Chu, “Evaluating interactive support for secure programming,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2012, pp. 2707–2716 [Online]. Available: <http://doi.acm.org/10.1145/2207676.2208665>. [Accessed: 10-Nov-2012]
- [88] J. Xie, H.R. Lipford, and B. Chu, “Why do programmers make security errors?,” in *2011 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2011, pp. 161–164.

7 Appendix A - Data Collection

7.1 Screening questionnaire

Screening questionnaire - Suitability of code for industrial settings

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada

Participant background

First tell us a little more about you.

1*
What level of study are you currently engaged in?

- Some High School
- High School Diploma
- College
- Undergraduate
- Masters Degree
- PhD
- Not in school

2*
What is your primary field of study?

3*
How would you rate your experience programming web applications and servlets?

- None
- Novice
- Some Knowledge
- Highly Knowledgeable
- Expert

4
If your answer to the previous question is different from none, please provide more details.

Hibernate

5*
How would you rate your experience with Hibernate?

Screening questionnaire - Suitability of code for industrial settings

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada

- None
- Novice
- Some Knowledge
- Highly Knowledgeable
- Expert

6

If your answer to the previous question is different from none, please provide more details.

Professional experience

7*

Do you have any professional experience developing software?

- yes
- no

8*

Please specify what your professional experience is

SQL

This section is to assess your knowledge about SQL

Answer the following questions assuming we have the following "Persons" table

Persons				
P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

9

With SQL, how do you select the column named "FirstName" from table "Persons"?

Screening questionnaire - Suitability of code for industrial settings

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada

- SELECT FirstName FROM Persons
- EXTRACT FirstName FROM Persons
- SELECT Persons.FirstName

10

With SQL, how do you select all the columns from table "Persons"?

- SELECT Persons
- SELECT *.Persons
- SELECT [all] FROM Persons
- SELECT * FROM Persons

11

With SQL, how do you select all the records from table "Persons" where the value of the column "FirstName" is "Peter"?

- SELECT * FROM Persons WHERE FirstName <> 'Peter'
- SELECT * FROM Persons WHERE FirstName='Peter'
- SELECT [all] FROM Persons WHERE FirstName='Peter'
- SELECT [all] FROM Persons WHERE FirstName LIKE 'Peter'

12

With SQL, how do you select all the records from table "Persons" where the value of the column "FirstName" starts with an "a"?

- SELECT * FROM Persons WHERE FirstName LIKE '%a'
- SELECT * FROM Persons WHERE FirstName='a'
- SELECT * FROM Persons WHERE FirstName='%a%'
- SELECT * FROM Persons WHERE FirstName LIKE 'a%'

13

The OR operator displays a record if ANY conditions listed are true. The AND operator displays a record if ALL of the conditions listed are true

- False
- True

Screening questionnaire - Suitability of code for industrial settings

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada

14

With SQL, how do you select all the records from table "Persons" where the "FirstName" is "Peter" and the "LastName" is "Jackson"?

- SELECT * FROM Persons WHERE FirstName<>'Peter' AND LastName<>'Jackson'
- SELECT FirstName='Peter', LastName='Jackson' FROM Persons
- SELECT * FROM Persons WHERE FirstName='Peter' AND LastName='Jackson'

15

With SQL, how can you insert a new record into the "Persons" table?

- INSERT VALUES ('Jimmy', 'Jackson') INTO Persons
- INSERT ('Jimmy', 'Jackson') INTO Persons
- INSERT INTO Persons VALUES ('Jimmy', 'Jackson')

16

With SQL, how can you insert "Olsen" as the "LastName" in the "Persons" table?

- INSERT INTO Persons (LastName) VALUES ('Olsen')
- INSERT ('Olsen') INTO Persons (LastName)
- INSERT INTO Persons ('Olsen') INTO LastName

17

How can you change "Hansen" into "Nilsen" in the "LastName" column in the Persons table?

- UPDATE Persons SET LastName='Hansen' INTO LastName='Nilsen'
- UPDATE Persons SET LastName='Nilsen' WHERE LastName='Hansen'
- MODIFY Persons SET LastName='Hansen' INTO LastName='Nilsen'
- MODIFY Persons SET LastName='Nilsen' WHERE LastName='Hansen'

18

With SQL, how can you delete the records where the "FirstName" is "Peter" in the Persons Table?

- DELETE ROW FirstName='Peter' FROM Persons
- DELETE FirstName='Peter' FROM Persons
- DELETE FROM Persons WHERE FirstName = 'Peter'

Screening questionnaire - Suitability of code for industrial settings

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada

19

With SQL, how can you return the number of records in the "Persons" table?

- SELECT COUNT() FROM Persons
- SELECT COLUMNS(*) FROM Persons
- SELECT COLUMNS() FROM Persons
- SELECT COUNT(*) FROM Persons

Java

Some questions to assess your Java knowledge

20

What is the correct way to declare an executable class in java?

a)

```
class ExampleProgram {  
    public static void main(String[] args, int size){  
        System.out.println("I'm a Simple Program");  
    }  
}
```

b)

```
import java.lang.System.out;  
class ExampleProgram {  
    public static void main(String[] args, int size){  
        System.out.println("I'm a Simple Program");  
    }  
}
```

c)

```
class ExampleProgram {  
    public static void main(String[] args){  
        System.out.println("I'm a Simple Program");  
    }  
}
```

- a
- b
- c

21

What reserved word makes a variable read-only?

- const
- final
- readOnly

Screening questionnaire - Suitability of code for industrial settings

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada

22

What is a servlet?

- A class that runs on the server
- A web page with embedded Java code
- All of the above

23

What's the difference between servlets and applets?

1. Servlets execute on Servers, whereas Applets execute on Browsers
2. Servlets have no GUI, whereas an Applet has a GUI
3. Servlets create static web pages, whereas Applets create dynamic web pages
4. Servlets can handle only a single request, whereas Applets can handle multiple requests

- 1,2,3 are correct
- 1,2 are correct
- 1,3 are correct
- 1,2,3,4 are correct

24

Which of the following are the session tracking techniques?

- URL rewriting, using a session object, using a response object, using hidden fields
- URL rewriting, using a session object, using cookies, using hidden fields
- URL rewriting, using a servlet object, using a response object, using cookies
- URL rewriting, using a request object, using a response object, using a session object

25

What will be the result of running the following line in a jsp file taking into account that the Web server has just been started and this is the first page loaded by the server?

```
<%=request.getSession(false).getId() %>
```

- It won't compile.
- It will print the session id.
- It will produce a NullPointerException.
- It will produce an empty page.

Screening questionnaire - Suitability of code for industrial settings

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada

26

A deployment descriptor describes

- Web component response settings
- Web component settings
- Web component request objects
- All of the above

27

Name the class that includes the getSession method that is used to get the HttpSession object.

- HttpServletRequest
- HttpServletResponse
- SessionContext
- SessionConfig

Java

Consider the following class:

```
public class IdentifyMyParts {  
    public static int x = 7;  
    public int y = 3;  
}
```

28*

a. What are the class variables?



29*

b. What are the instance variables?



30*

Screening questionnaire - Suitability of code for industrial settings

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada

c. What is the output from the following code:

```
IdentifyMyParts a = new IdentifyMyParts();
IdentifyMyParts b = new IdentifyMyParts();
a.y = 5;
b.y = 6;
a.x = 1;
b.x = 2;
System.out.println("a.y = " + a.y);
System.out.println("b.y = " + b.y);
System.out.println("a.x = " + a.x);
System.out.println("b.x = " + b.x);
System.out.println("IdentifyMyParts.x = " + IdentifyMyParts.x);
```



31*

What gets printed when the following code is compiled and run? Select the one correct answer.

```
public class test {
    public static void main(String args[]) {
        int i = 1;
        do {
            i--;
        } while (i > 2);
        System.out.println(i);
    }
}
```

- 0
- 1
- 2
- 1

Screening questionnaire - Suitability of code for industrial settings

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada

Results

*Answers for questions Q4 and Q6 were not captured due to a bug with the capture system.

Date taken	24/09/2012 03:15	24/09/2012 23:59	24/09/2012 03:07
Tracking code	P1	P2	P3
Q1	Undergraduate	Undergraduate	Masters Degree
Q2	Game development (programming)	Game Development and Entrepreneurship	IT Security
Q3	Novice	Some Knowledge	Some Knowledge
Q5	Novice	None	None
Q7	yes	No	yes
Q8	Freelance work for various companies, ranging from web-based Java applications to game building tools in Unity.	None	I worked for 2.5 years in an IT company. my project mainly dealt with JAVA coding
Q9	SELECT FirstName FROM Persons	SELECT FirstName FROM Persons	SELECT FirstName FROM Persons
Q10	SELECT * FROM Persons	SELECT * FROM Persons	SELECT * FROM Persons
Q11	SELECT * FROM Persons WHERE FirstName='Peter'	SELECT * FROM Persons WHERE FirstName='Peter'	SELECT * FROM Persons WHERE FirstName='Peter'
Q12	SELECT * FROM Persons WHERE FirstName LIKE 'a%'	SELECT * FROM Persons WHERE FirstName LIKE 'a%'	SELECT * FROM Persons WHERE FirstName LIKE 'a%'
Q13	TRUE	TRUE	TRUE
Q14	SELECT * FROM Persons WHERE FirstName='Peter' AND LastName='Jackson'	SELECT * FROM Persons WHERE FirstName='Peter' AND LastName='Jackson'	SELECT * FROM Persons WHERE FirstName='Peter' AND LastName='Jackson'
Q15	INSERT INTO Persons VALUES ('Jimmy', 'Jackson')	INSERT INTO Persons VALUES ('Jimmy', 'Jackson')	INSERT INTO Persons VALUES ('Jimmy', 'Jackson')
Q16	INSERT INTO Persons (LastName) VALUES ('Olsen')	INSERT ('Olsen') INTO Persons (LastName)	INSERT INTO Persons (LastName) VALUES ('Olsen')
Q17	UPDATE Persons SET LastName='Nilsen' WHERE LastName='Hansen'	MODIFY Persons SET LastName='Nilsen' WHERE LastName='Hansen'	UPDATE Persons SET LastName='Nilsen' WHERE LastName='Hansen'
Q18	DELETE FROM Persons WHERE FirstName = 'Peter'	DELETE FROM Persons WHERE FirstName = 'Peter'	DELETE FROM Persons WHERE FirstName = 'Peter'
Q19	SELECT COUNT(*) FROM Persons	SELECT COLUMNS(*) FROM Persons	SELECT COUNT(*) FROM Persons
Q20	c	C	c
Q21	final	Const	const
Q22	A class that runs on the server	A class that runs on the server	All of the above
Q23	1,2 are correct	1,3 are correct	1,2,3 are correct

Screening questionnaire - Suitability of code for industrial settings

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada

Q24	URL rewriting, using a servlet object, using a response object, using cookies	URL rewriting, using a servlet object, using a response object, using cookies	URL rewriting, using a servlet object, using a response object, using cookies
Q25	It will produce an empty page.	It will print the session id.	It will print the session id.
Q26	All of the above	All of the above	All of the above
Q27	HttpServletRequest	HttpServletRequest	HttpServletResponse
Q28	x	X	variables that can be accessed within a class
Q29	y	Y	variables that can be defined in a class with each object of a class has a copy of it
Q30	a.y = 5 b.y = 6 a.x = 2 b.x = 2 IdentifyMyParts.x = 2	a.y = 5 b.y = 6 a.x = 1 a.y = 2 IdentifyMyParts.x = 7	a.y = 5 b.y = 6 a.x = 1 b.x = 2 IdentifyMyParts.x =
Q31	0	0	1

7.2 Exit questionnaire

Exit questionnaire - Suitability of code for industrial settings

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada

Exit questionnaire

For the purposes of our research the answers you provide to us in this questionnaire are very valuable. Please try to answer in the most honest, accurate and complete way you can.

1*

Do you think that your application has some defects? If so please list them here.

2*

What improvements would you do the program code if you had more time to finish this task?

3*

If you were working for a company, what recommendations would you make to improve the application? Please explain.

4*

How would you rate your computer security knowledge?

Novice

Some
Knowledge

Average

Highly
Knowledgeable

Expert

5*

If you had received any kind of security training, please tell us what kind of training it was and what it was about.

6

In case your answer is different from "Novice", please list all the threats or attacks that you think this application is vulnerable to

7*

Do you think there is sensitive information in the data base? If so, which tables/fields are those? And why you consider them as sensitive?

Exit questionnaire - Suitability of code for industrial settings

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada

8

What would you suggest to protect the sensitive data in the application?

9*

If you were asked to implement encryption on your programs how difficult it would be?

Very easy	Easy	Average	Difficult	Very difficult
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

10*

How difficult do you consider the task assigned to you in this experiment?

Very easy	Easy	Average	Difficult	Very difficult
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

11*

The tools you used to develop your program presented some information related to encryption. Do you recall what it was? Please tell us what you remember and understood.

12*

Did you use the tool to encrypt any field in the database? (Yes/No) please specify why.

13

In case your answer is no, did you try to use it? What stopped you from completing the process?

14

How relevant do you find the information presented in the property encryption page?

Not relevant	Slightly relevant	Relevant	Very relevant	Critical
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Exit questionnaire - Suitability of code for industrial settings

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada

15
How easy to use are the tools you were asked to use for the experiment?

Very easy	Easy	Average	Difficult	Very difficult
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

16*
How useful do you consider the encryption capabilities of the tool?

Not useful	Slightly useful	Useful	Very useful	Essential
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

17
If you used the help button in the encryption section how relevant do you find the information presented?

Not relevant	Slightly relevant	Relevant	Very relevant	Critical
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

18
How difficult it was to understand its use?

Very easy	Easy	Average	Difficult	Very difficult
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

19*
What do you like about the tools?

20
What you do not like about the tools and what improvements do you suggest for the tools?

Results

Date taken	09/10/2012 16:35	06/10/2012 17:39	06/10/2012 01:06
Tracking code	P1	P2	P3
Q1	It didn't run because I ran out of time and didn't finish making it.	Due to time constraints, I could not fully finish my application. So there are several defects that were not resolved.	No

Exit questionnaire - Suitability of code for industrial settings

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada

Q2	I would make it run, first of all.	I would add checks to make sure that the user could only enter valid data, such as only allowing valid credit card numbers.	I would complete the requirements listed and run the application
Q3		Add checks so that only one user can have a certain value, as an example only one user can have a social insurance number.	More time and more explanation on the technology
Q4-1	Novice	Novice	Average
Q4-1-other			
Q5	None	I have received no formal security training.	No training received
Q6			network attacks
Q7	Passwords should always be considered sensitive data, since people tend to use them on multiple sites. SIN numbers are also very important to keep private.	Yes, the entire table holds personal information, but especially the credit card and social insurance number fields are especially sensitive information.	name etc
Q8		Encrypt the table using hibernate's built in encryption.	encryption
Q9-1	Average	Easy	Average
Q9-1-other			
Q10-1	Easy	Average	Average
Q10-1-other			
Q11	Hibernate offered a range of encryption methods when setting up the config file, and the ability to select which fields would be encrypted with that method.	Hibernate had a built in encryption feature when setting up, you just had to check a box and chose the encryption type from a drop down menu.	while creating the config file we can encrypt the columns if needed
Q12	No, didn't get around to it. I was intending to though.	No, I did not include encryption because at the time I was more focused on trying to get the rest of the application working.	No.. time constrain
Q13	Lack of time		Time constrain
Q14-1	Relevant		Relevant

Exit questionnaire - Suitability of code for industrial settings

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada

Q14-1- other			
Q15-1	Average	Average	Average
Q15-1- other			
Q16-1	Useful	Essential	Very useful
Q16-1- other			
Q17-1			Relevant
Q17-1- other			
Q18-1			Average
Q18-1- other			
Q19	Based on my brief exposure, it seems like they have decent scalability, and it seemed like a fairly well designed database abstraction tool.	I like how the tools had a simple way of implementing after the initial setup, as well as a easy way of adding encryption to sensitive user information.	user friendly
Q20	They were very clunky to use for a small program; there was a lot of setup for a small amount of payoff. But this is necessary for larger applications to make proper use of them.		NA

7.3 Pilot study logs

Legend

Yellow fill with dark yellow text	Interaction with the mapping wizard
Green text	Making visible the property encryption page
Light purple	Interaction with the encryption page
Red text	Hiding the property encryption page
Black on white	Plugin component message

Participant 01		
Date	Component	Message
2012-10-05 11:51:55,57 1	bernate.eclipse.jdt.ui.wizards.NewHibernateMappingFileWizard	Changing page name: title: Create Hibernate XML Mapping file(s)
2012-10-05 11:51:58,91 1	bernate.eclipse.jdt.ui.wizards.NewHibernateMappingFileWizard	Changing page name: ColumnEncryption title: Property Encryption
2012-10-05 11:51:59,18 9	te.eclipse.jdt.ui.wizards.NewHibernatePropertyEncryptionPage	Showing page
2012-10-05 11:52:07,49 4	bernate.eclipse.jdt.ui.wizards.NewHibernateMappingFileWizard	Changing page name: PreviewPage title: Create Hibernate XML Mapping file(s)
2012-10-05 11:52:07,49 4	te.eclipse.jdt.ui.wizards.NewHibernatePropertyEncryptionPage	Processing selected items
2012-10-05 11:52:08,17 6	bernate.eclipse.jdt.ui.wizards.NewHibernateMappingPreviewPage	Create textFileChange/resourceChange for new hbm.xml /Directory/src/domain/Contact.hbm.xml
2012-10-05 11:52:08,24 6	te.eclipse.jdt.ui.wizards.NewHibernatePropertyEncryptionPage	Hiding page
2012-10-05 12:07:28,84 1	bernate.eclipse.jdt.ui.wizards.NewHibernateMappingFileWizard	Changing page name: title: Create Hibernate XML Mapping file(s)
2012-10-05 12:07:56,52 0	bernate.eclipse.jdt.ui.wizards.NewHibernateMappingFileWizard	Changing page name: ColumnEncryption title: Property Encryption
2012-10-05	te.eclipse.jdt.ui.wizards.NewHi	Showing page

12:07:56,544	bernamePropertyEncryptionPage	
2012-10-05 12:26:53,903	bername.eclipse.jdt.ui.wizards.NewHibernateMappingFileWizard	Changing page name: PreviewPage title: Create Hibernate XML Mapping file(s)
2012-10-05 12:26:53,904	te.eclipse.jdt.ui.wizards.NewHibernatePropertyEncryptionPage	Processing selected items
2012-10-05 12:26:53,938	bername.eclipse.jdt.ui.wizards.NewHibernateMappingPreviewPage	Create textFileChange/resourceChange for new hbm.xml /Directory/src/domain/Contact.hbm.xml
2012-10-05 12:26:53,953	te.eclipse.jdt.ui.wizards.NewHibernatePropertyEncryptionPage	Hiding page
2012-10-05 12:27:04,911	bername.eclipse.jdt.ui.wizards.NewHibernateMappingPreviewPage	perform textFileChanges changes

Participant 02		
Date	Component	Message
2012-10-05 12:26:08,556	bername.eclipse.jdt.ui.wizards.NewHibernateMappingFileWizard	Changing page name: title: Create Hibernate XML Mapping file(s)
2012-10-05 12:26:16,219	bername.eclipse.jdt.ui.wizards.NewHibernateMappingFileWizard	Changing page name: ColumnEncryption title: Property Encryption
2012-10-05 12:26:16,292	te.eclipse.jdt.ui.wizards.NewHibernatePropertyEncryptionPage	Showing page
2012-10-05 12:26:26,804	te.eclipse.jdt.ui.wizards.NewHibernatePropertyEncryptionPage	Encryption algorithm selected PBEWITHMD5ANDES
2012-10-05 12:26:51,739	bername.eclipse.jdt.ui.wizards.NewHibernateMappingFileWizard	Changing page name: PreviewPage title: Create Hibernate XML Mapping file(s)
2012-10-05 12:26:51,739	te.eclipse.jdt.ui.wizards.NewHibernatePropertyEncryptionPage	Processing selected items

2012-10-05 12:26:52, 111	ernate.eclipse.jdt.ui.wizards.NewHibernateMappingPreviewPage	Create textFileChange/resourceChange for new hbm.xml /Directory/src/domain/Contact.hbm.xml
2012-10-05 12:26:52, 144	te.eclipse.jdt.ui.wizards.NewHibernatePropertyEncryptionPage	Hiding page
2012-10-05 12:27:20, 939	ernate.eclipse.jdt.ui.wizards.NewHibernateMappingPreviewPage	perform textFileChanges changes

Participant 03

Date	Component	Message
2012-10-05 12:26:18,7 63	ernate.eclipse.jdt.ui.wizards.NewHibernateMappingFileWizard	Changing page name: title: Create Hibernate XML Mapping file(s)
2012-10-05 12:26:34,5 61	ernate.eclipse.jdt.ui.wizards.NewHibernateMappingFileWizard	Changing page name: ColumnEncryption title: Property Encryption
2012-10-05 12:26:34,6 86	te.eclipse.jdt.ui.wizards.NewHibernatePropertyEncryptionPage	Showing page
2012-10-05 12:27:11,6 58	ernate.eclipse.jdt.ui.wizards.NewHibernateMappingFileWizard	Changing page name: PreviewPage title: Create Hibernate XML Mapping file(s)
2012-10-05 12:27:11,6 98	te.eclipse.jdt.ui.wizards.NewHibernatePropertyEncryptionPage	Processing selected items
2012-10-05 12:27:12,7 51	te.eclipse.jdt.ui.wizards.NewHibernatePropertyEncryptionPage	Hiding page
2012-10-05 12:27:34,8 45	ernate.eclipse.jdt.ui.wizards.NewHibernateMappingPreviewPage	perform textFileChanges changes
2012-10-05 13:34:28,4 42	ernate.eclipse.jdt.ui.wizards.NewHibernateMappingFileWizard	Changing page name: title: Create Hibernate XML Mapping file(s)
2012-10-05 13:34:30,8 15	ernate.eclipse.jdt.ui.wizards.NewHibernateMappingFileWizard	Changing page name: ColumnEncryption title: Property Encryption
2012-10-05	te.eclipse.jdt.ui.wizards.New	Showing page

13:34:30,8 77	HibernatePropertyEncryption Page	
2012-10-05 13:34:36,5 50	ernate.eclipse.jdt.ui.wizards. NewHibernateMappingPreviewPage	Create textFileChange/resourceChange for new hbm.xml /Experiment/src/ca/uoit/dao/HibernateUtil.hbm.xml
2012-10-05 13:39:12,1 70	ernate.eclipse.jdt.ui.wizards .NewHibernateMappingFileWizard	Changing page name: title: Create Hibernate XML Mapping file(s)
2012-10-05 13:39:14,3 62	ernate.eclipse.jdt.ui.wizards .NewHibernateMappingFileWizard	Changing page name: ColumnEncryption title: Property Encryption
2012-10-05 13:39:14,4 10	te.eclipse.jdt.ui.wizards.New HibernatePropertyEncryption Page	Showing page
2012-10-05 13:39:16,8 83	ernate.eclipse.jdt.ui.wizards. NewHibernateMappingPreviewPage	Create textFileChange/resourceChange for new hbm.xml /Experiment/src/ca/uoit/dao/DAOUtil.hbm.xml
2012-10-05 13:39:16,8 93	ernate.eclipse.jdt.ui.wizards. NewHibernateMappingPreviewPage	perform textFileChanges changes
2012-10-05 13:50:28,4 15	ernate.eclipse.jdt.ui.wizards .NewHibernateMappingFileWizard	Changing page name: title: Create Hibernate XML Mapping file(s)
2012-10-05 13:50:30,1 04	ernate.eclipse.jdt.ui.wizards .NewHibernateMappingFileWizard	Changing page name: ColumnEncryption title: Property Encryption
2012-10-05 13:50:30,1 35	te.eclipse.jdt.ui.wizards.New HibernatePropertyEncryption Page	Showing page
2012-10-05 13:50:32,0 76	ernate.eclipse.jdt.ui.wizards .NewHibernateMappingFileWizard	Changing page name: PreviewPage title: Create Hibernate XML Mapping file(s)
2012-10-05 13:50:32,0 77	te.eclipse.jdt.ui.wizards.New HibernatePropertyEncryption Page	Processing selected items
2012-10-05 13:50:32,1 09	ernate.eclipse.jdt.ui.wizards. NewHibernateMappingPreviewPage	Resource already exist on project src/ca/uoit/dao/HibernateUtil.hbm.xml
2012-10-05 13:50:32,1 10	ernate.eclipse.jdt.ui.wizards. NewHibernateMappingPreviewPage	Create textEdit change to replace the content of HibernateUtil.hbm.xml
2012-10-05 13:50:32,5 86	te.eclipse.jdt.ui.wizards.New HibernatePropertyEncryption Page	Hiding page
2012-10-05	ernate.eclipse.jdt.ui.wizards.	perform textFileChanges changes

13:50:35,6 NewHibernateMappingPrevie
97 wPage

8 Appendix B – Experiment Material

8.1 Data opt-out & removal form

Data Opt Out & Removal Request Form– Suitability of code for industrial standards
University of Ontario Institute of Technology
Faculty of Business and Information Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada
Ethics Approval #11-096

In order for us to assist you in the removal of your identifying information from our records please complete this form it is vital that you submit this form fully filled out as instructed below. I hereby request to remove data about me from your records. I understand my request will be processed within two weeks after I respond to the confirmation email I will receive after my data Opt-Out request has been received. All information you provide through the data opt out request process is ONLY used for the purposes of removing data from our records.

My signature below indicates that I choose not to participate in this study any further and remove any data about my participation on it.

_____	_____	_____
Print Name	Signature	Date

Email address		

8.2 Consent form

Consent Form – Suitability of code for industrial standards

University of Ontario Institute of Technology
Faculty of Business and Information Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada
Ethics Approval #11-096

Research Personnel:

Ricardo Rodriguez Garcia
Graduate Student, MSc
Computer Science

Dr. Julie Thorpe
Assistant Professor

Dr. Miguel Vargas Martin
Associate Professor

Purpose: The purpose of this study is to determine the suitability of the code produced to meet industry-standards by using a tool called Hibernate. Hibernate is a tool that helps create mappings between objects and relational databases.

Task Requirements: This study is comprised of three tasks. **Task 1:** Complete a training workshop on the use of Hibernate. **Task 2:** Complete a small application with the support of Eclipse and Hibernate tools, and finally **Task 3:** Fill out an exit questionnaire. You will need to bring your own computer to the experiment, and it should run Windows and have at least 7 GB of free space. All required software will be provided to you as well as assistance to set it up.

Duration and Location: The session will take place at the North Campus of the University of Ontario Institute of Technology. The total amount of time needed to complete the study is about 5-6 hours. The training workshop (Task 1) will last for about 2.5 hours (including setup and free lunch break) and you will have three hours to complete the assigned application (Task 2). We expect that Task 2 will only take about two hours, but you may stay for up to three hours if needed. The exit questionnaire (Task 3) should take 15-30 minutes. Participants will be compensated with \$30 for participating in the experiment portion of the session (Tasks 2 and 3). The top 3 applications (created during Task 2) will get a \$150 future shop gift card (applications will be evaluated according to the guidelines set at http://en.wikipedia.org/wiki/Software_quality).

Potential Risk/Discomfort: There are no psychological or physical risks associated with this experiment.

Anonymity/Confidentiality: All data collected will be held completely confidential. The data will only be made available to the Research Personnel mentioned above. Data will be coded for identification purposes, which means that the data will be associated with an arbitrary identifier (such as “participant 20”) which could not be linked back to a specific person. All data containing personal or identifiable information including this form will be kept in a locked drawer. Once the study is finished, all electronic files will be erased and all paper records will be shredded.

Voluntary Participation: Any participation in a research study is completely voluntary. You are free to decline to participate for any reason. You may also stop participating at any time or refuse to answer any individual questions.

Right to Withdraw: You may choose not participate in any part of this study or not to answer certain questions or to withdraw from this study at any time for any reason. If you withdraw before beginning “Task 2” you will not be eligible to receive any compensation; Withdrawing during any of the remaining tasks or refusing to answer the exit questionnaire does not affect your \$30 compensation, the only exception is that you may not be eligible for the \$150 gift card if you withdraw because you will not have a complete product to be evaluated.. You may contact any member of the research team either verbally or electronically to tell them you wish to withdraw from the study; you may provide a reason for withdrawing, but it will not be necessary to do so. Additionally, the research personnel may ask you to fill out a short survey asking why you decided to opt out the study, but this will be completely optional.

8.3 Opt out survey

Opt-out Survey – Suitability of code for industrial settings
University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada
Ethics Approval #[11-096](#)

This form is optional

Please answer the following with YES or NO

I choose to withdraw or not to participate any further in this study.	
I understand that by submitting this form, I am withdrawing from this study.	
I give permission for my data collected so far to be used (tools used, steps completed). I understand that this information will be used anonymously.	

Please check all that apply	
<input type="checkbox"/>	I could not understand the concepts explained in the workshop.
<input type="checkbox"/>	I think the tasks assigned are too difficult or complex.
<input type="checkbox"/>	I think the tasks assigned are too time-consuming.
<input type="checkbox"/>	I do not have time to complete this study
<input type="checkbox"/>	I do not want to participate in this research study anymore.
<input type="checkbox"/>	Other: (please specify)

My signature below indicates that I choose not to participate in this study any further.

Print Name

Signature

Date

This form is optional

8.4 Pre-screening consent form

Pre- screening Consent Form – Suitability of code for industrial standards

University of Ontario Institute of Technology
Faculty of Business and Information Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada
Ethics Approval #11-096

Research Personnel:

Ricardo Rodriguez Garcia
Graduate Student, MSc
Computer Science

Dr. Julie Thorpe
Assistant Professor

Dr. Miguel Vargas Martin
Associate Professor

Purpose: The purpose of this study is to determine the suitability of the code produced to meet industry-standards by using a tool called Hibernate. Hibernate is a tool that helps create mappings between objects and relational databases.

Task Requirements: In order to assess your eligibility to participate in this experiment you need to complete a pre-screening questionnaire to assess your knowledge about SQL and Java. In case you are eligible to participate in the study you will be contacted via email with further details about the next phase of the experiment; in case you are not eligible to participate in further phases of the experiment you will be entitled to get a chocolate bar.

Duration and Location: You can complete the pre-screening questionnaire by accessing the link provided in the email you receive along this consent form, from any computer with access to internet. The estimated time to complete the questionnaire is about 15 minutes.

Potential Risk/Discomfort: There are no psychological or physical risks associated with this experiment.

Anonymity/Confidentiality: All data collected will be held completely confidential. The data will only be made available to the Research Personnel mentioned above. Data will be coded for identification purposes, which means that the data will be associated with an arbitrary identifier (such as "participant 20") which could not be linked back to a specific person. All data containing personal or identifiable information including this form will be kept in a locked drawer. Once the study is finished, all electronic files will be erased and all paper records will be shredded.

Voluntary Participation: Any participation in a research study is completely voluntary. You are free to decline to participate for any reason. You may also stop participating at any time or refuse to answer any individual questions.

Right to Withdraw: You may choose not participate in any part of this study or not to answer certain questions or to withdraw from this study at any time for any reason. If you choose to exercise this right before completing the pre-screening questionnaire, you will not be eligible to receive any compensation for your participation. You may contact any member of the research team via email to let them know you wish to withdraw from the study; you may provide a reason for withdrawing, but it will not be necessary to do so. Additionally, the research personnel may ask you to fill out a short survey asking why you decided to opt out the study, but this will be completely optional.

Data Opt-Out & Removal: If a withdrawing participant wishes, they can contact the research team to receive the Data Opt-Out & removal Form; if the participant completes this form or tells a member of the research team that they wish their data be removed, the research personnel will ensure removal of any/all data related to the user's participation in the study, as long as the request to withdraw are received before linking personal identifiers are destroyed (i.e., by November 30, 2012).

Signing this form does not waive any of your legal rights or alter your ability to stop participating at a later time.

Consent Form – Suitability of code for industrial settings

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada

Ethics Approval #11-096

Reporting concerns: If you have concerns about the ethics of this research, please contact the University of Ontario Institute of Technology’s Ethics and Compliance Officer. For other questions about the research, please contact Dr. Julie Thorpe or Dr. Miguel Vargas Martin:

Ethics and Compliance Officer,
Office of Research Services,
University of Ontario Institute
of Technology
Tel: 1 905-721-8668 ext 3693
Email: compliance@uoit.ca

Dr. Julie Thorpe
Assistant Professor
Faculty of Business and
Information Technology
University of Ontario Institute of
Technology
Tel: 1 905-721-8668 ext. 6585
Email: julie.thorpe@uoit.ca

Dr. Miguel Vargas Martin
Associate Professor
Faculty of Business and Information
Technology
University of Ontario Institute of
Technology
Tel: 905-721-8668 ext. 2834
Email: miguel.vargasmartin@uoit.ca

I have read and understand the above terms of testing and I understand the conditions of my participation.
My signature below indicates that I agree to participate in this experiment.

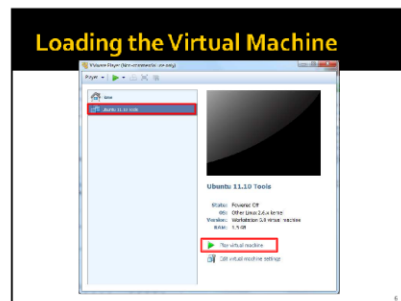
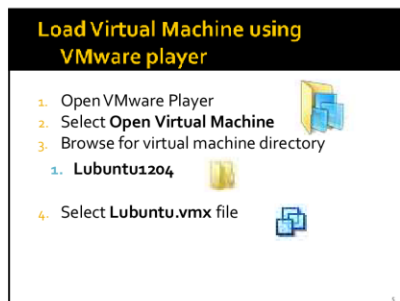
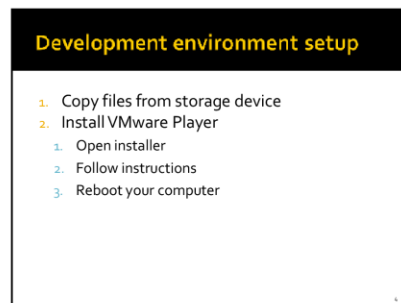
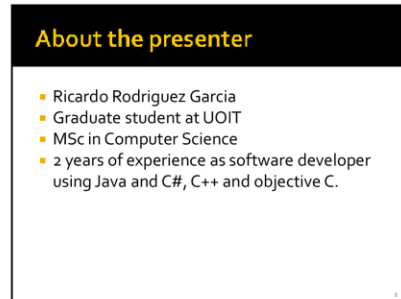
Print Name

Signature

Date

8.5 Workshop slides

Nov-12



1

Workshop Objectives

- Learn about Hibernate
- Build demo application
 - Eclipse.
- Provide you references to get more advanced information.

Workshop Outline

1. Hibernate Introduction
 1. Why Hibernate was created?
 2. What is Hibernate?
 3. What do you need to use Hibernate?
2. Hibernate Example
 1. Minimalistic Hibernate app example
 2. Hibernate architecture and API
3. Build your own Hibernate application

1.1 Why learn Hibernate?

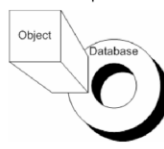
- Reason 1 – Developers' Productivity
 - Less code = Less bugs
 - Focus on the main problem

1.1 Why learn Hibernate?

- Reason 2 – Performance & Scalability
 - Fine Tuning=Performs better than handcrafted JDBC
 - Big data
- Reason 3 - Effective Cross-Database Portability
 - Abstracts SQL differences and incompatibilities

Why Hibernate was created?

- Object Relational Impedance Mismatch



O/R Mapping is hard

- Many factors to consider:
 - Identity
 - Granularity
 - Associations
 - Sub types
 - Type mismatches

Granularity

- Table structures
 - "De-normalized"
 - Rows map to multiple objects
- Objects
 - Fine grained

Associations

Java objects

- Does address know about the customer?
- Can a customer have more than one address?
- Can an address be owned by more than one customer?

Data tables

Inheritance

- Table per concrete class.
- Table per class hierarchy.
- Table per subclass.

Data type mismatch

- Data types
 - There is no one-to-one mapping between SQL and Java types

JDBC Type	Java Object Type
CHAR	String
VARCHAR	String
CLOB	String
NUMERIC	java.math.BigDecimal
DECIMAL	java.math.BigDecimal
BIT	Boolean
TINYINT	Integer
SMALLINT	Integer
MEDIUMINT	Integer
BIGINT	Integer
REAL	Float
DOUBLE	Double
NUMERIC	BigDecimal
DECIMAL	BigDecimal
DATE	java.sql.Date
TIME	java.sql.Time
TIMESTAMP	java.sql.Timestamp
STRUCT	Object (map of attribute types)
SQLXML	XML
ARRAY	Array
STRUCT	Struct or SQLData
REF	Referencing java class
SQL_OBJECT	Referencing java class

Hibernate to the rescue

- It adapts to your needs, so you don't have to adapt to it!
- Database can exist already
- Or be created at the same time the application

Workshop Outline

- Hibernate Introduction
 - Why Hibernate was created?
- Hibernate architecture and API
 - Minimalistic Hibernate app example
- Build your own Hibernate application
 - What do you need to use Hibernate?

Minimalistic Hibernate App

1. Open Eclipse
2. Open **BasicHibernate** project

Running the app

- Right click on **HibernateExample.java**
- Run as -> Java Application

```

Integer id1 =saveStudent("Alice","UOIT");
Integer id2 =saveStudent("Bob","UOIT");
Integer id3 =saveStudent("Carl","UOIT");
listStudents();
updateStudent(id3, "Claire");
listStudents();
deleteStudent(id2);
listStudents();
    
```

Hibernate Architecture and API

- Application Code
 - basic.HibernateExample.java
- Plain Old Java Objects (POJOs)
 - basic.model
 - Mapping files (*.hbm.xml)
- Hibernate runtime.
 - Hibernate Jars
- Hibernate Configuration
 - hibernate.cfg.xml
- Database.

The Configuration Object

HibernateUtil.java

The Session Factory

basic.HibernateExample.java

- persistStudent()
- updateStudent()
- deleteStudent()
- listStudents()

Manipulating Objects

basic.HibernateExample.java

```

Session session = factory.openSession();
Transaction transaction = null;
try {
    transaction = session.beginTransaction();
    //Do something
    ...
    transaction.commit();
} catch (HibernateException e) {
    transaction.rollback();
    e.printStackTrace();
} finally {
    session.close();
}

```

26

Sessions and Transactions

- Session objects
 - Main interface to work with the database
 - **Methods to create, read, update, delete**
 - Light weight and inexpensive to create
 - Shouldn't be kept open for long
 - Not thread safe (don't pass between threads)
 - Lifecycle is bounded to transactions.

27

Transactions

- **Transaction**
 - Keep data integrity
 - Delimits a unit of work with the database
 - Transaction tx = session.beginTransaction();
 - ... (database manipulation)
 - tx.commit();
 - Created from session objects when a database modification is required.
 - Should be kept open the shortest time possible.
 - If an exception occurs it must be rolled back

27

Instance states

- **transient:**
 - never persistent, no Id, not associated with any Session
- **persistent:**
 - Id value, associated with an unique Session
- **detached:**
 - Database representation, not associated with any Session, container

28

Objects Creation

- **saveStudent()**

```
Student student= new Student(); transient
```
- **persistStudent()**

```
session.saveOrUpdate(student); persistent
```

29

Read & Delete instances

- **get(Class clazz, Serializable id)**
 - (Student) session.get(Student.class, studentId);
 - Persistent until the session is closed
- **delete(Object object)**
 - session.delete(student);
 - Removes a persistent instance from the datastore.

30

Retrieving persistent instances

- Hibernate query language
 - Similar to SQL
 - `createQuery(String queryString)`
- Criteria queries
 - Object oriented
 - `createCriteria(Class persistentClass)`
- Native SQL
 - Traditional approach
 - `createSQLQuery(String queryString)`

20

Query Examples: listStudents()

- Hibernate Query Language
 - `session.createQuery("from Student").list();`
- Criteria Query
 - `session.createCriteria(Student.class).list();`
- Native SQL
 - `session.createSQLQuery("Select * from STUDENT").addEntity(Student.class).list();`
- `uniqueResult()`

21

Criteria queries by example

- `org.hibernate.criterion.Example`

```
Cat cat = new Cat();
cat.setSex('F');
cat.setColor(Color.BLACK);
List results = session.createCriteria(Cat.class)
    .add(Example.create(cat) )
    .list();
```

22

Workshop Outline

1. Hibernate Introduction
2. Hibernate architecture and API
3. Build your own Hibernate application
 1. What do you need to use Hibernate?
 2. Directory application

23

Build you own app!

- What do you need?
 1. A database
 1. MySQL 5
 2. An existing schema
 1. A schema represents the database structure
 3. DB user with appropriate permissions
 2. JDBC drivers
 3. Hibernate 3.6
 - www.hibernate.org



24

Directory application

- Contacts class
 - Name
 - Address
 - Phone Number
 1. Create
 2. Update
 3. Search
 4. Delete

25

What do we need to do?

1. Database and Class path set up
2. Create domain classes and their corresponding XML mapping files
3. Create Hibernate configuration file (Hibernate.cfg.xml)
4. Add application code

Database and Class path set up

- Directory project
 - Hibernate and dependencies
 - App structure and stub classes
 - Database
 - tutorial scheme
 - User: root
 - Password:



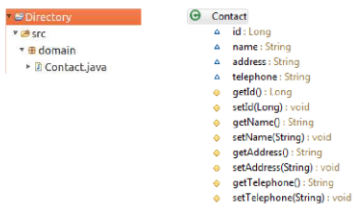
What do we need to do?

1. Database and Class path set up
2. Create domain classes and XML mapping files
3. Create Hibernate configuration file (Hibernate.cfg.xml)
4. Add application code

Persistent Classes

1. Implement a no-argument constructor
2. Provide an identifier property
3. Prefer non-final classes (semi-optional)
4. Declare accessors and mutators (getters/setters) for persistent fields (optional)

Contact class



Mapping files *.hbm.xml

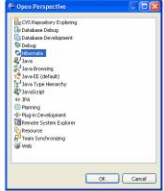
- Associate Java classes with database tables and columns when performing queries.
- Hibernates hardest part is getting your mapping files right
 - Properties.
 - Associations.
 - Inheritance tree.

Hibernate tools plugin

- Wizards
 - Configuration files
 - Mapping files
- Editors
 - Auto completion
- Reverse engineering
 - Java Code generation
 - Mapping files

Enable Hibernate perspective

- Go to Window -> Open Perspective -> Other,
- Select **Hibernate**,
- Click the **Ok** button.

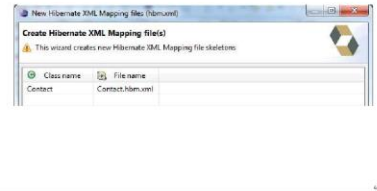


Using Hibernate tools plugin to generate mapping files

New -> Hibernate XML mapping file.



Hibernate Xml Mapping file wizard



Property encryption

It is advisable that you encrypt the database fields that correspond to sensitive class variables. If you are unsure about which variables/fields are sensitive, click on the help icon on the bottom.

Available Class Variables:

- Configuration
- Contact

Encrypt Configuration

Key location: Algorithm: Password:

Mapping files preview

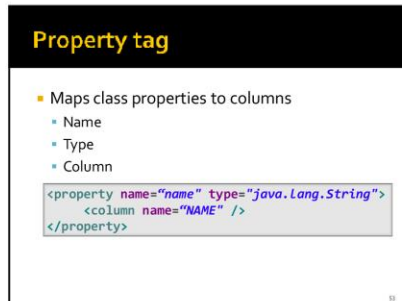
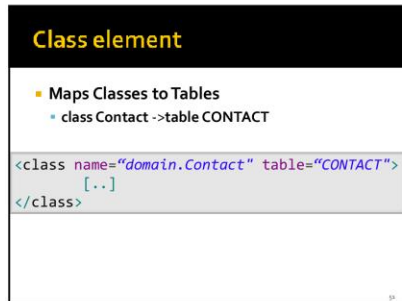
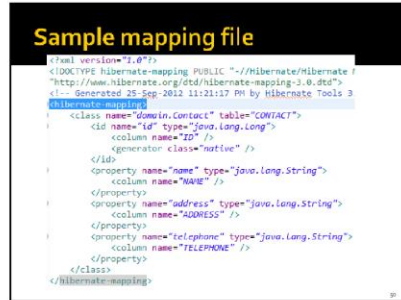
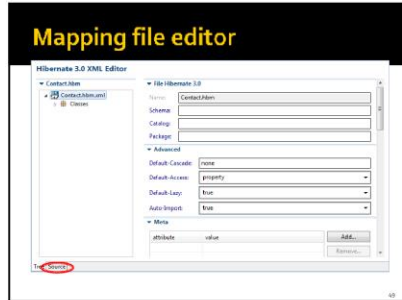
Changes to be performed

- Create file Directory:\src\domain\Contact.hbm.xml

```

<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate.org/Hibernate Mapping 3.0.dtd"
["http://www.hibernate.org/3/DTD/hbmata-mapping-3.0.dtd"]
[<!-- generated 20-SEP-2012 7:00:13 PM by Hibernate Tools 3.6.0.Alpha1 -->
<hibernate-mapping>
<class name="domain.Contact" table="Contact">
<id name="id" type="java.lang.Long">
<column name="ID" />
<generator class="native" />
</id>
<property name="name">
<column name="name" />
</property>
</class>
</hibernate-mapping>

```

Mapping types

Mapping type	Java type	ANSI SQL Type
integer	int or java.lang.Integer	INTEGER
long	long or java.lang.Long	BIGINT
short	short or java.lang.Short	SMALLINT
float	float or java.lang.Float	FLOAT
double	double or java.lang.Double	DOUBLE
big_decimal	java.math.BigDecimal	NUMERIC
character	java.lang.String	CHAR(n)
string	java.lang.String	VARCHAR
byte	byte or java.lang.Byte	TINYINT
boolean	boolean or java.lang.Boolean	BIT
yes/no	boolean or java.lang.Boolean	CHAR(1) ('Y' or 'N')
true/false	boolean or java.lang.Boolean	CHAR(1) ('T' or 'F')

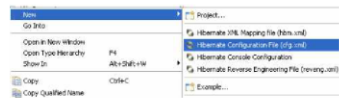
What we need to do?

1. Database and Class path set up
2. Create domain classes and their corresponding XML mapping files
3. Create configuration file (Hibernate.cfg.xml)
4. Write application code

Use Hibernate Tools for Eclipse to create the configuration file

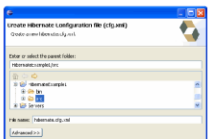
To create the Hibernate Configuration File,

1. Right click the project folder node,
2. Select New -> Hibernate Configuration File (cfg.xml)

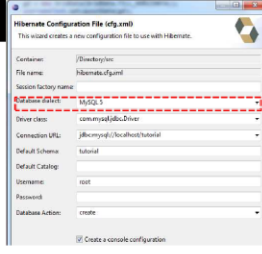


hibernate.cfg.xml

- By default the file name will be hibernate.cfg.xml,
- select the src directory and click Next.

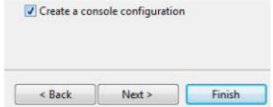


hibernate.cfg.xml



HIBERNATE.CFG.XML

- Finally check the create console configuration option.
- Click Finish



Sample Hibernate.cfg.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
"/Hibernate/Hibernate Configuration DTD 3.0/EN"
"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
    <property name="hibernate.connection.password">root</property>
    <property name="hibernate.connection.username">root</property>
    <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
    <property name="hibernate.connection.url">jdbc:mysql://localhost/tutorial</property>
    <property name="hibernate.default_schema">tutorial</property>
    <!-- List of XML mapping files -->
    <mapping resource="Contact.hbm.xml"/>
  </session-factory>
</hibernate-configuration>
    
```

Mapping resources

```
<session-factory>
...
<mapping
resource="domain/contact.hbm.xml"/>
</session-factory>
```

What we need to do?

1. Database and Class path set up
2. Create domain classes and their corresponding XML mapping files
3. Create configuration file (Hibernate.cfg.xml)
4. Write application code

Write application code

- Create configuration object
- Create session factory
- Add methods for:
 - Add a contact
 - Update contact details
 - Search for contacts
 - Delete a contact

Start up and helpers

- HibernateUtil helper
 - Startup.
 - Convenient access to SessionFactory.
- Obtain sessions

```
package util;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class HibernateUtil {
    private static final SessionFactory sessionFactory;
    static {
        try {
            sessionFactory = new Configuration().configure()
                .buildSessionFactory();
        } catch (Throwable ex) {
            System.err.println("Initial SessionFactory
                creation failed." + ex);
            throw new ExceptionInInitializerError(ex);
        }
    }
    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
}
```

The main Directory class

- Main method
 - Create a new contact
 - Search contact
 - Update contact name
 - Delete contact from directory
- saveOrUpdateContact
- deleteContact
- searchByName

Generic method idiom

```

Session session = HibernateUtil.getSessionFactory().openSession();
Transaction transaction = null;
try {
    transaction = session.beginTransaction();
    //Do something
    ...
    transaction.commit();
} catch (HibernateException e) {
    transaction.rollback();
    e.printStackTrace();
} finally{
    session.close();
}

```

60

saveOrUpdateContact

```

public long saveOrUpdateContact (Contact contact) {
    long result=-1;
    try{
        ...
        session.saveOrUpdate(contact);
        transaction.commit();
        result=contact.getId();
        ...
    }
    return result;
}

```

61

deleteContact

```

public boolean deleteContact (Contact contact) {
    boolean result=false;
    try{
        ...
        session.delete(contact);
        transaction.commit();
        result= true;
        ...
    }
    return result;
}

```

62

searchByName(String query)

```

public List<Contact> searchByName(String query) {
    List<Contact> result;
    try{
        ...
        result=session.createQuery("Select * from
CONTACT where name like '"+query+"%' "
).addEntity(Contact.class).list();
        ...
    }
    return result;
}

```

63

Test the application

- Run the Directory class
 1. Right click the main.Directory.java file
 2. Select run as->Java application
- Verify everything works

64

The end

- Congratulations you just created your first Hibernate web application.
- Hibernate reference documentation:
 - <https://docs.jboss.org/hibernate/core/3.6/reference/en-US/html/index.html>

65

Appendix A Hibernate 3.6 jars

- Hibernate 3.6 libraries and required dependencies
 - antlr-2.7.6.jar
 - commons-collections-3.1.jar
 - dom4j-1.6.1.jar
 - javassist-3.12.0.GA.jar
 - jta-1.1.jar
 - slf4j-api-1.6.1.jar
 - hibernate-jpa-2.0-api-1.0.1.Final.jar
 - hibernate3.jar
 - jasypt-1.9.0.jar or jasypt-1.9.0-lite.jar
 - jasypt-hibernate3-1.9.0.jar

79

Retrieving persistent instances

- Hibernate query language
 - <https://docs.jboss.org/hibernate/orm/3.5/reference/en/html/queryhql.html>
- Criteria queries
 - <https://docs.jboss.org/hibernate/orm/3.5/reference/en/html/querycriteria.html>

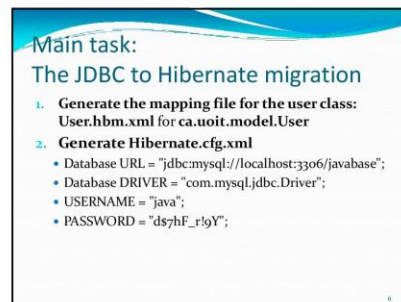
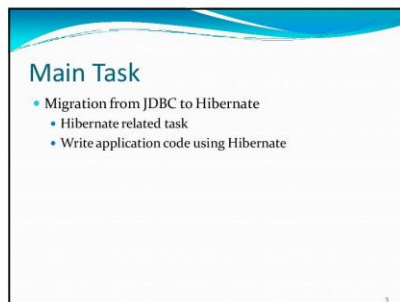
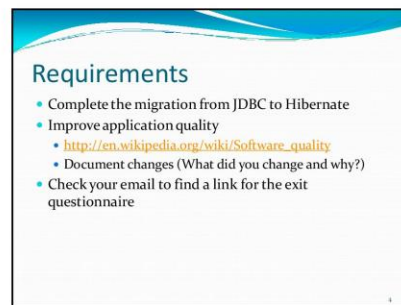
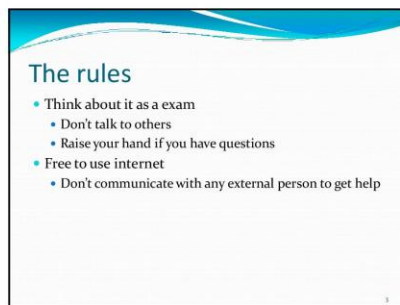
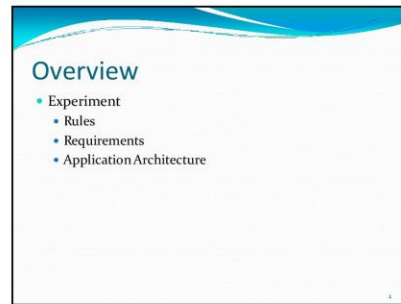
79

Virtual Machine users and passwords

- Ubuntu 11.10
 - standard user:
 - username: user
 - password: password
 - administrative user:
 - username: root
 - password: password
- MySQL server
 - username: root
 - password: root

79

8.6 Experiment slides



Main task:
The JDBC to Hibernate migration

4. Add a reference to `User.hbm.xml` to `Hibernate.cfg.xml`
5. **Implement User DAO interface**
 - `UserDAOHibernate`
6. **Modify DAOFactory to return instances of `UserDAOHibernate` instead of `UserDAOJDBC`**

Web Application

Experiment project

Functionality

- Register new users
- Modify existing records
- Login/logout

Architecture

- Java code
- Web Content

```

• JAX-WS Web Services
• Deployment Descriptor: Task
• Java Resources
  • src
    • ca.uoit.control
    • ca.uoit.dao
    • ca.uoit.model
  • Libraries
  • JavaScript Resources
  • build
  • WebContent
    • META-INF
    • WEB-INF
      • index.jsp
      • login.jsp
      • userEdit.jsp
      • userRegister.jsp
      • task database.sql
  
```

Packages

- control
- model
- DAO

control

- Edit
- Login
- Logout
- Register

```

• ca.uoit.control
  • EditController.java
  • FormValidator.java
  • FormValidatorUtil.java
  • LoginController.java
  • LogoutController.java
  • RegisterController.java
  • ValidatorException.java
  
```

model

```

class User {
    id: Long
    username: String
    password: String
    firstName: String
    lastName: String
    email: String
    sinNumber: String
    creditCardNumber: String
    address1: String
    address2: String
    User()
}

```

- Domain class
- Validation code

```

package ca.uoit.model {
    class User.java
    class UserFormValidator.java
}

```

DAO (Data Access Object)

UserDAO

UserDAO

- DAOFactory
 - Produces instances of UserDAO
- Implemented by:
 - UserDAOJDBC
 - UserDAOHibernate

User DAO interface

- User findById(Long id)
- User findByName(String username)
- List<User> list()
- void create(User user)
- void update(User user)
- void save(User user)
- void delete(User user)
- boolean existUsername(String username)
- boolean existEmail(String email)

Web content

- User interface
 - Index
 - Login
 - UserEdit
 - UserRegister

```

WebContent
├── META-INF
├── WEB-INF
│   ├── index.jsp
│   ├── login.jsp
│   ├── userEdit.jsp
│   └── userRegister.jsp

```

index

Welcome,

Please select the action you wish to perform

[Register New User](#)

[Edit User Data](#)

Login

- Verify user credentials

Welcome! Please enter your Name and Password to log in.

Name:

Password:

UserEdit

UPDATE

Here you can register yourself.

Email address

First Name

Last Name

Address1

Address2

SIN Number

Credit Card

[home](#) [logOut](#)

UserRegister

REGISTER

Here you can register yourself.

Username *

Password *

Confirm password *

Email address

First Name

Last Name

Address1

Address2

SIN Number

Credit Card

[home](#)

Finishing

- Answer the exit questionnaire
- Once you have finished rise your hand
 - collect artefacts (code produced, logs, etc)
 - further instructions

8.7 Email correspondence

Email correspondence– Suitability of code for industrial standards
University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada
Ethics Approval #11-096

Contents

- MyCampus recruitment..... 2
- Screening questionnaire invitation 3
- Subject not suitable..... 4
- Subject suitable..... 5
- Final date and location 6
- Contest results 7
- Feedback Letter..... 8

Email correspondence– Suitability of code for industrial standards

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada
Ethics Approval #11-096

MyCampus recruitment

Subject: Java/Hibernate workshop and research study - \$30, free lunch, plus the chance to win a \$150 Future Shop gift card!

Body:

We are looking for participants to take part in a study that aims to determine the suitability of code to meet industry quality standards. Participants will be compensated with \$30 dollars for participation on the study. The top 3 programs based on the software quality criteria stated in this link https://en.wikipedia.org/wiki/Software_quality will receive a 150\$ Future Shop gift card.

Participants will be required to modify a small web application with the help of Eclipse and adapt it to use a tool called Hibernate instead of JDBC. Persons willing to participate must be familiar with Java, IDEs (e.g., Eclipse, NetBeans, Visual Studio) and have a basic understanding of SQL and HTML. There will be a screening process to assess the suitability of participants.

In order to participate in the study, participants must first attend a free introductory training workshop on the use of Hibernate. Free pizza and soft drinks will be served during the workshop, which will take approximately 2.5 hours.

The workshop and study will take place at UOIT (North Campus); a poll must be completed to establish a date suitable to most participants in late September and will have total duration of approximately 5.5-6 hours.

If you are interested or would like further information, please send your contact information before September 20th to:

UOITSoftwareStudy@gmail.com
Faculty of Business & Information Technology
Graduate Research Study
Ethics #: 11-096

Email correspondence– Suitability of code for industrial standards

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada
Ethics Approval #11-096

Screening questionnaire invitation

Subject: Software study: screening questionnaire invitation

Body:[attached consent form]

Dear [participant]

Thank you for your interest in participating in our research!

We are looking for participants to take part in a study that aims to determine the suitability of the code to meet industry quality standards. Participants will be compensated with \$30 dollars for participation in the study. The top 3 programs based on the software quality criteria stated in this link https://en.wikipedia.org/wiki/Software_quality will receive a \$150 Future Shop gift card.

Participants will be required to modify a small web application with the help of Eclipse and adapt it to use Hibernate instead of JDBC. Persons willing to participate must be familiar with Java, Servlets, IDEs (e.g., Eclipse, NetBeans, Visual Studio) and have a basic understanding of SQL and HTML. There will be a screening process to assess the suitability of participants.

The workshop and study will take place at UOIT (North Campus); a poll must be completed to establish a date suitable to most participants this session will have a total duration of approximately 5-6 hours.

In order to participate in the study, you will need to complete a screening questionnaire. Please take some minutes of your time to complete it. Your answers will be kept **confidential** and will not be distributed.

Take some time to review the attached consent form. If you agree with the terms in the consent form proceed to answer the questionnaire, to begin click the link below or copy it into your browser.
[Survey link]

Once we receive your questionnaire results, we will get in contact with you to let you know further details.

Sincerely,
Ricardo Rodríguez García,
MSc Student (Computer Science)
Faculty of Business & Information Technology
Graduate Research Study
Ethics #:

Email correspondence– Suitability of code for industrial standards

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada
Ethics Approval #11-096

Subject not suitable

Subject: Software study, screening questionnaire result

Body:

Dear [participant],

Thank you for taking the time out of your busy schedule to complete our screening questionnaire. Unfortunately, the responses provided in the screening questionnaire show that you are not part of the population we are aiming to study in our research; however, we wish to thank you for your participation and time.

No information about your participation in this study will be kept. You are entitled to receive a chocolate bar for the time you spent answering our screening questionnaire. You can pick up your chocolate bar at [location and time].

Thank you again for your time and consideration.

Best regards,
Ricardo Rodríguez García,
MSc Student (Computer Science)
Faculty of Business & Information Technology
Graduate Research Study
Ethics #:

Email correspondence– Suitability of code for industrial standards

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada
Ethics Approval #11-096

Subject suitable

Subject: Software study, screening questionnaire result

Body:

[Attached consent form]

Dear [participant],

Thank you for taking the time out of your busy schedule to complete our screening questionnaire. The responses provided in the screening questionnaire show that you are eligible to take part in our research study and we will like to provide you with more details about the study;

The workshop and study sessions will be held at the UOIT North Campus on a Saturday between 20/04/2012 and 30/05/2012.

Please take some time to review the consent form attached and answer a small poll that will help us to decide a suitable date to perform our study. To begin the poll, click the link below or copy it into your browser.

[Survey link]

We will get in contact with you to communicate the final date and location as soon as possible.

Best regards,
Ricardo Rodríguez García,
MSc Student (Computer Science)
Faculty of Business & Information Technology
Graduate Research Study
Ethics #:

Email correspondence– Suitability of code for industrial standards

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada
Ethics Approval #11-096

Final date and location

Subject: Software study, final location and date

Body:

[Attached consent form]

Dear [participant],

The responses provided in the screening questionnaire showed that you are eligible to take part in our research study and we will like to provide you with more details about the study.

The workshop and study sessions will be held at UOIT North Campus on Saturday [date] at [location] from 10 am to 4 pm.

Please confirm your participation by replying to this email.

Best regards,
Ricardo Rodríguez García,
MSc Student (Computer Science)
Faculty of Business & Information Technology
Graduate Research Study
Ethics #:

Email correspondence– Suitability of code for industrial standards

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada
Ethics Approval #11-096

Contest results

Subject: Software study, contest result

Body:

Dear [participant],

The results from the contest are as follows:

[Contest result identified by participant code]

The participants whose ID corresponds to the 3 first positions will be contacted by the research team with instructions to claim your prize.

Thank you very much to all for your participation in this study! We will communicate with you all once the results of the study from which this experiment are part are ready to be published.

Thanks again to all participants.

Best regards,
Ricardo Rodríguez García,
MSc Student (Computer Science)
Faculty of Business & Information Technology
Graduate Research Study
Ethics #:

Email correspondence– Suitability of code for industrial standards

University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada
Ethics Approval #11-096

Feedback Letter

Subject: Software study, results

Body:

Dear [participant],

We finally have the results of the study in which you contributed with your participation.

[Information to access the results]

Thank you all for your contribution to this project.

We appreciate your participation and wish you the best in all your endeavours.

Best regards,
Ricardo Rodríguez García,
MSc Student (Computer Science)
Faculty of Business & Information Technology
Graduate Research Study
Ethics #:

8.8 Recruitment poster

Poster – Suitability of code for industrial settings
University of Ontario Institute of Technology
2000 Simcoe St. North, Oshawa, ON, L1H 7K4, Canada
Ethics Approval # 11-096

DO YOU KNOW JAVA, SQL AND HTML?

FREE  **HIBERNATE Workshop**

FREE PIZZA

FREE DRINKS

\$30 CASH



**THE CHANCE TO WIN 1 OF 3 FUTURES SHOP
150\$ GIFT CARDS**

ALL THIS FOR PARTICIPATING IN A RESEARCH STUDY*

Improve your skills & productivity, contribute to science, and show off your programming skills! The study will be conducted at UOIT North Campus in late September; time limit to register September 20th.

Would you like further information? Please send your contact information to:

Email: UOITSoftwareStudy@gmail.com

Faculty of Business & Information Technology
Graduate Research Study

* To receive all this you will need to qualify for this study through a pre-screening questionnaire. Participants must be familiar with Java, IDEs (e.g., Eclipse, Netbeans, Visual Studio) and have a basic understanding of servlets, SQL, HTML. Participants need to bring their own laptops to the study. Participant will modify a small application to use Hibernate instead of JDBC, the top 3 applications will get a \$150 Future Shop gift card; applications will be evaluated according to the guidelines set at http://en.wikipedia.org/wiki/Software_quality.

UOITSoftwareStudy@gmail.com
UOITSoftwareStudy@gmail.com
UOITSoftwareStudy@gmail.com
UOITSoftwareStudy@gmail.com
UOITSoftwareStudy@gmail.com
UOITSoftwareStudy@gmail.com
UOITSoftwareStudy@gmail.com
UOITSoftwareStudy@gmail.com
UOITSoftwareStudy@gmail.com
UOITSoftwareStudy@gmail.com
UOITSoftwareStudy@gmail.com
UOITSoftwareStudy@gmail.com
UOITSoftwareStudy@gmail.com
UOITSoftwareStudy@gmail.com
UOITSoftwareStudy@gmail.com

8.9 Reference cheat sheet

How to build a Hibernate application

Step	Description	Slide
1	Set up class path and database	37
	Add Hibernate jars and your DB driver in your class path. Make sure to have a DB schema and a user with enough privileges.	
2	Create domain classes	38
	To take advantage of all the features of Hibernate it is necessary that our instance classes follow these rules: <ol style="list-style-type: none">1. They must have a property to work as an index or identity field2. Have an empty constructor3. Provide accessor methods (getters/setters) for its persistent attributes http://docs.jboss.org/hibernate/orm/3.6/reference/en-US/html/persistent-classes.html	
3	Generate mapping files for each one of your persistent classes	
	To generate the mapping files: Click on [File -> New -> Other -> Hibernate -> Hibernate XML mapping file] and select your persistent classes to create the corresponding mapping files for each one of them.	
4	Generate Hibernate.cfg.xml configuration file	53
	Click on [File -> New -> Other -> Hibernate -> Hibernate Configuration File] and create a cfg file. The following properties should be specified: jdbc url, username, password, DB schema, driver class and dialect.	
5	Add mapping file references to the hibernate.cfg.xml configuration file	59
	Before closing the session-factory element add a mapping tag indicating in the resource parameter the full path to your mapping files	
6	Loading and storing objects	64
7	Create HibernateUtil Class	62
	You can take a look at the examples presented and reuse the code found in there. http://docs.jboss.org/hibernate/orm/3.6/reference/en-US/html/tutorial.html#tutorial-firstapp-helpers	
8	Use session objects to change the persistence of records in the database.	24
	http://docs.jboss.org/hibernate/orm/3.6/javadocs/org/hibernate/Session.html	
9	Retrieve records using one of the three strategies available	30
	HQL: The Hibernate Query Language: http://docs.jboss.org/hibernate/orm/3.6/reference/en-US/html/queryhql.html Criteria objects: http://docs.jboss.org/hibernate/orm/3.6/reference/en-US/html/querycriteria.html Native SQL: http://docs.jboss.org/hibernate/orm/3.6/reference/en-US/html/querysql.html	

9 Appendix C - Research Ethics Board Documentation

9.1 Application for ethical review of research involving human participants



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

In order for us to direct your application to the appropriate Research Ethics Board, please answer the following questions.

I am a(n):

Faculty Researcher

Graduate Student Researcher

Is this research for academic credit? Yes No

Undergraduate Student Researcher

Is the research for academic credit? Yes No

External (to UOIT) Researcher

In the faculty of:

Social Science & Humanities

Business and Information Technology

Science

Engineering and Applied Science

Health Sciences

Energy Systems and Nuclear Science

Education

Undergraduate Student Research

UOIT has established a number of Faculty Research Ethics Boards (FREBs) for the review of course, individual and group thesis projects by undergraduate students in addition to the main UOIT Research Ethics Board (UOIT REB).

Office of Research Services
Fax: 905-721-3210

1



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

The FREBs shall review research projects conducted by undergraduate students when (1) it is conducted as part of an undergraduate course offered and (2) it is not part of a faculty member’s research programme already subject to review by the UOIT REB.

The FREBs may refer an application to the UOIT REB. Examples of situations in which referral would be appropriate are research that the FREB thinks may be of more than minimal risk, research involving ethical or legal issues for which it does not have adequate expertise, and in cases for which conflicts of interest reduce its size to less than two members.

Faculty Research

All faculty and external research proposals involving human participants will be reviewed by the UOIT Research Ethics Board.

University of Ontario Institute of Technology Research Ethics Board (REB) Application for Ethical Review of Research Involving Human Participants	File #	
---	---------------	--

	Name & UOIT Banner ID (If Applicable)	Rank (e.g., faculty, student, visiting professor, other affiliation... etc.)	Faculty/ Dept./Address	Phone No.	E-Mail



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

Principal Investigator	Ricardo Rodriguez Garcia 100436549	Graduate Student	FBIT		ricardo.rodriguezgarca@uoit.c
Co-Investigator(s)					
All co-investigators must be listed. Attach additional investigator information/signature pages to this application.					
Faculty Supervisor(s) <small>(for student PIs)</small>	Miguel Vargas Martin	Faculty / Professor	FBIT	905 721 8668 ext 2834	miguel.vargasmartin@uoit.ca
Faculty Supervisor(s) <small>(for student PIs)</small>	Dr. Julie Thorpe	Faculty / Professor	FBIT	1 905 721 8668 ext 6585	julie.thorpe@uoit.ca

Submit the application, all consent materials and instruments to the Office of Research Services (see guide below)

Hard copy: Original + 2 additional copies of the following documents, and

Electronic: Electronic file of all documents to compliance@uoit.ca

<p>Recruitment Materials</p> <ol style="list-style-type: none"> 1. Letter of invitation 2. Verbal script 3. Telephone script 4. Advertisements (newspapers, posters) 5. Email correspondence 6. Other 	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	<p>Letter of Approval/Permission (if applicable)</p> <p><i>(Not letters of support)</i></p> <ol style="list-style-type: none"> 1. cooperating organizations 2. school board(s) 3. hospitals 4. community agencies or other institutions (university/college) 5. other 6. other 	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
--	--	---	--

Office of Research Services
Fax: 905-721-3210



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

			<input type="checkbox"/>
Consent Materials 1. Consent form 2. Assent form for minors 3. Parental/3rd party permission forms 4. Transcriber confidentiality agreement 5. Other	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Plan for Dissemination/Communication of Results to Participants 1. Thank you letter 2. Feedback letter 3. Workshop 4. Verbal thank you 5. Debriefing letter 6. Other	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
Data Gathering Instruments 1. Questionnaires 2. Interview guides 3. Tests 4. Other	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Any previously approved protocol to which you refer Information/ Signatures of all investigators/co-investigators (attach additional signature pages)	<input type="checkbox"/> <input checked="" type="checkbox"/>

<p>Considering the risks involved in the proposed research, you are applying for Expedited Review <input checked="" type="checkbox"/> Full Review <input type="checkbox"/></p> <p>Please note Expedited Review involves review by 1 REB member and the REB Chair. Expedited review does not mean a rapid review of the application. You will be notified if your application has been sent for Full review. (See UOIT and TCPS policy for more information on risks)</p>

Section A: General Information, Rational and Purpose of Research

A1. Title of the Research Project: The influence of development tools on aspects of software security

A2. Proposed Date (dd/mm/yyyy)

Office of Research Services
 Fax: 905-721-3210



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

(a) of commencement: 13/094/2012

(b) of completion: 30/105/2012

(Note, please allow sufficient time for REB to review request.)

A3. Indicate the location(s) where the research will be conducted.

- University of Ontario Institute of Technology
- Community Site(s) Specify
- School Board(s) Specify
- Hospital(s) Specify
- Other Specify

A4. Other Ethics Approval/Permission:

(a) Is this a multi-centered study? (when several university/hospital REBs consider the same proposal from the perspectives of their respective institutions)	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No
(b) Has any other Canadian University Research Ethics Board approved this research?	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No
If NO, will any other Research Ethics Board be asked for approval? <i>Specify university/hospital to be approached or explain why approval will not be sought</i>	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No
If YES, please complete Section A of the application only; unless you are accessing the UOIT student base, then you must complete the entire application. Title of the project approved elsewhere: Name of the Other Institution: Name of the Other Board: Date of the Decision:	



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

<p>A contact name and phone number for the other Board:</p> <p>Provide a copy of the application from the other institution together with all accompanying materials and a copy of the clearance certificate / approval. Ensure all investigator information and signature pages are attached with your submission.</p>	
<p>(c) Has any other person(s) or institution(s) granted permission to conduct this research? <i>Specify (e.g., school boards, community organizations, proprietors)</i></p>	<p><input type="checkbox"/> Yes <input checked="" type="checkbox"/> No</p>
<p>If NO, will permission/approval be sought? <i>Specify Agency/College, Government Agency, NGO etc.</i></p>	<p><input type="checkbox"/> Yes <input checked="" type="checkbox"/> No</p>
<p>If YES, Name of the Other Institution: Date of the Decision: A contact name and phone number for the other Board: Provide a copy of the clearance certificate / approval.</p>	
<p>(d) Are you signing an external agreement with an institution governing the use of data? <input type="checkbox"/> Yes <input checked="" type="checkbox"/> No If yes, please submit with your application.</p>	
<p>(e) Has this research application received a peer/scientific review?</p>	<p><input type="checkbox"/> Yes <input checked="" type="checkbox"/> No</p>

A5. Level of the Research (determined by status of Principal Investigator):

- Undergraduate Research
- Graduate Research
- Quality Assurance/Program Evaluation
- Faculty Course Based Research
- Faculty Research



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

A6. Professional Expertise/Qualifications:

1. Does this research require professional/specialized expertise qualifications other than your own?
2. Yes No

If YES, specify:

Does the researcher (or your supervisor if the Principal Investigator is a student researcher), or any members of your research team have the professional expertise/recognized qualifications required to carry out this research?

3. Yes No

If YES, specify: Dr. Julie Thorpe has done several studies into graphical passwords schemes development experiments. Dr Miguel Vargas Martin has experience in several experiments involving humans. Ricardo Rodriguez has worked as a programmer. This experience provides the research team with the skills necessary to carry out this research.

A7. If you are a UOIT researcher, please complete the following section.

<p>If you have recently received an Assessment of Research Compliance (ARC) form from the Grants Officers, please provide the 5 digit award file number here:</p> <ol style="list-style-type: none">1. Internal: <input type="checkbox"/> Yes <input checked="" type="checkbox"/> No2. Research Account Number # (if known)3. Name Source:<ul style="list-style-type: none"><input type="checkbox"/> TIFF<input type="checkbox"/> Start Up<input type="checkbox"/> Professional Development<input type="checkbox"/> Other (Specify):
<p>External: <input checked="" type="checkbox"/> Yes <input type="checkbox"/> No</p> <p>a) Please indicate period of funding Start: 08/04/01 End: 13/03/31</p>



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

If funding is being sought or has been received, please indicate Agency or Sponsor Applied for:

CIHR NSERC SSHRC Other (Specify):

Please indicate Name of PI who holds the grant (Name and Institution if external to UOIT): Miguel Vargas Martin / Julie Thorpe

Please provide Project Title on Grant/Funding Application : "Network Security with Automatic Mitigation of Disruptive Traffic, Attack Containment, and Intrusion Detection" / "Improving Computer Security Through the Analysis of Human Factors"

Multiple funding sources: Yes No

If Yes, please specify: NSERC and UOIT Graduate scholarship

Please attach additional funding documentation including period of funding, Agency or Sponsor, Name of PI and Title on Funding Application

A8. Conflict of Interest:

Will the researcher(s), members of the research team, and/or their partners or immediate family members receive any personal benefits related to this study - for example: a financial remuneration, patent and ownership, employment, consultancies, board membership, share ownership, stock options (Do not include details regarding Release Time Stipend, conference and travel expense coverage, possible academic promotion, or other benefits which are integral to the conduct of research generally).

Yes No

1. If Yes, please describe the benefits below.

Describe any restrictions regarding access to or disclosure of information (during or at the end of the study) that the sponsor has placed on the investigator(s).



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

[Empty rectangular box]

A9. Rationale:

Describe the purpose and background rationale for the proposed project, as well as the hypothesis(es)/research question(s) to be examined.

Purpose/Background Information:

The purpose of this experiment is to test the hypothesis that the design of development tools can influence the adoption of security best practices in a positive way.

Problem Definition

The networked database is the heart of any application. It is where the most valuable assets reside – the information that is the foundation of business, transaction records, financial data, and customer information. Protecting this data is very important, however it is also an increasingly difficult and non-trivial task.

Sensitive data stored on networked servers are at risk from attackers who only need to find one way inside the network to access this confidential information. Additionally, perimeter defenses like firewalls cannot protect stored sensitive data from the internal threat – employees with the means to access and exploit this data.

According to Team SHATTER (Security Heuristics of Application Testing Technology for Enterprise Research), unencrypted sensitive data is one of the top 10 database vulnerabilities.

The purpose of encryption is to protect sensitive information from being accessed by unauthorized persons; however data must still be accessible for the authorized applications and users who need it.

So far, development tool support for this kind of protection is minimal or non-existent on the eclipse Integrated Development Environments (IDEs) used by many programmers today.

In order to improve or alleviate this situation, we have designed the “Crypto Assistant”, a



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

tool prototype that brings together three tools that are commonly used by Java developers to add encryption support to their application: Eclipse, Hibernate, and Jasypt.

The goal of Hibernate is to increase developer productivity by reducing 95% of the Java code that is normally needed to access databases; Jasypt is an encryption library whose aim is to simplify the use of encryption methods. Normally, the use of encryption in Java requires the programmer to have a broad understanding of Java and cryptography, Jasypt simplifies the use of encryption providing a more clear and concise application programming interface (API) that is easy to understand and use. These are steps in the right direction, but we believe further steps can be taken to simplify the process and ensure that developers encrypt their application's sensitive data (and that they do this encryption correctly).

By combining the power of these tools together, Crypto Assistant simplifies the process of incorporating encryption into an application under development.

Normally, the developer must take the following actions to add encryption:

- Select the fields that contain sensitive information;
- Modify the database as encrypted data changes the format of data stored and requires more space to be stored.
- Change the type of data mapped by Hibernate during code generation. This is done through modifications to mapping files or source code. Jasypt integrates with Hibernate to provide encryption capabilities in a transparent way for the application code.
- Add initialization code to the application for the encryptor that will be used by Jasypt.

The prototype

We have developed the Crypto Assistant to help simplify the required actions of the developer.

Crypto Assistant is a set of modifications made to the Eclipse plug-in called Hibernate tools, which adds a wizard that guides developers in the mapping of fields containing sensitive information into its encrypted equivalent. There are two main goals that we want to achieve with the tool:

- First, to raise awareness among non-security savvy developers about the risks and consequences of not protecting data at rest.
- Second, we want to make it easier for developers to use encryption to protect data at rest, which we hypothesize will result in more developers implementing encryption successfully.



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

The information presented must be directly related to the developer's immediate perceived needs and delivered at a significant 'decision moment'. Our assumption is that a good moment to influence developers' actions is when they are selecting the database tables to use as part of the process of using Hibernate tools.

Purpose.

The purpose of this study is to investigate to what degree the above-stated goals are achieved by the Crypto Assistant and what changes would make the tool more effective.

Central Hypothesis.

The central hypothesis is that work patterns are influenced by the tools that developers use; as such, if the tools used don't put an emphasis on security, security will always be relegated to a secondary and almost demisable feature of the final product. A change in the focus that these tools put on security would be beneficial to the community in general by raising awareness about possible risks and solutions that are often overlooked, such as encryption of sensitive data in the present tool we are studying.

If the tools prime the developers with the idea that encryption of sensitive information is an important security feature, assist them in the implementation in order to reduce human error, and speeding up the process, then we expect to see an improvement in the security of applications by incorporating this technology. People that are not familiar with encryption will at least get primed with the idea and even if they opt to not use the Crypto Assistant they will hopefully consider it for future projects.

Particular hypotheses:

- 1) The prototype being evaluated will enhance the security (in terms of whether encryption was properly and appropriately implemented) of final software products.
- 2) Regardless of whether the developers use the prototype features, the information presented will raise developers' awareness about certain security issues (in particular, encrypting sensitive data).

Research questions:

- 1) Can the use of tools that emphasize security, in particular the Crypto Assistant, influence the quality of the resulting code produced?
- 2) Is the Crypto Assistant effective at encouraging the use of encryption and raising



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

awareness about its use as a protection mechanism?

3) Is the Crypto Assistant easy to understand and use?

4) In case the users don't use the encryption capabilities suggested and provided by the Crypto Assistant, what is the reason for not doing it?

A10. Participants: The REB is mandated by the guiding principles of the TCPS. In this section we are primarily concerned with the following two principals: Respect for Vulnerable Populations and Respect for Justice and Inclusiveness. (See TCPS for more information)

<http://www.pre.ethics.gc.ca/english/policystatement/context.cfm#C>

- 1. Is this a vulnerable population? Yes No
- 2. Are issues of inclusiveness being respected? Yes No
- 3. Describe the number of participants and any required demographics characteristics (e.g., age, gender).

We hope to recruit a sample of 8 to 16 participants; male and/or female (gender is not an issue here). They all must be 18 or older and must correctly answer at least 50% of the questions in each section of the screening questionnaire. The participants must have good understanding of Java, basic knowledge of SQL and HTML and it would be desired (but not required) that they have some experience using Eclipse, but knowledge of other IDEs like Visual Studio or NetBeans will be sufficient.

Section B: Methodology, Data Exchange, Risk Management

B1. Methods: Are any of the following procedures or methods involved in this study? Check all that apply.



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

- | | |
|---|---|
| <input type="checkbox"/> Questionnaire (mail) | <input type="checkbox"/> Participant Journals |
| <input checked="" type="checkbox"/> Questionnaire (email/web) | <input type="checkbox"/> Audio/video taping |
| <input type="checkbox"/> Questionnaire (in person) | <input checked="" type="checkbox"/> Unobtrusive observations |
| <input type="checkbox"/> Interview(s) (telephone) | <input type="checkbox"/> Invasive physiological measurements (e.g., venipuncture, muscle biopsies) |
| <input type="checkbox"/> Interview(s) (face to face) | <input type="checkbox"/> Non-invasive physical measurement (e.g., exercise, heart rate, blood pressure) |
| <input type="checkbox"/> Secondary Data | <input type="checkbox"/> Analysis of human tissue, body fluids, etc. |
| <input checked="" type="checkbox"/> Computer-administered tasks | <input checked="" type="checkbox"/> Other: (specify) |
| <input type="checkbox"/> Focus Groups | |

B2. Data Exchange Procedures: Describe sequentially, and in detail, all procedures in which the research participants will be involved (e.g., paper and pencil tasks, interviews, questionnaires, physical assessments, physiological tests, time requirements, etc.) **Remember to attach a copy of all questionnaire(s), interview guides, or other test instruments. Remember also to describe the procedures for all stages of the research (e.g., pre-tests, etc.) where applicable.**

Recruitment process

1) Initial contact

Once a volunteer gets in touch with the research personnel, they will be sent an email along with the pre-screening consent form requesting they complete the screening questionnaire using the ID assigned to them (which will also be provided in the email). The time required to complete the questionnaire is about 15 minutes. Upon completion of the questionnaire, the participants will be informed about their eligibility via email by the research personnel.

2) Eligibility

A) If the candidate participant is not eligible, the research personnel will send an email informing them of this and thanking the volunteer for participating. If a volunteer is not eligible to participate after completing the screening, they will be entitled to receive a chocolate bar.

B) The volunteers who are eligible will receive an email with further instructions about the



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

experiment and the specific place and time the workshop/study will be carried out. They will also be reminded that they need to bring their own university laptop to the workshop.

3) The study

The study will consist of two phases: (1) a workshop to provide the participants with necessary information about Hibernate, and (2) the experiment, where participants are asked to perform a short programming task.

3.1 - Workshop

The workshop will take place at the UOIT North Campus. We may organize several sessions depending on the availability of participants.

- 1) Participants will be welcomed by a member of the research team.
- 2) We will start the experiment with a quick overview about the experiment, the workshop, and the tools they will use. After this brief introduction, they will be instructed to set up the development environment needed for the workshop; USB keys or hard drives will be handed to participants, which will be loaded with the files needed to participate in the workshop and experiment. The files include a virtual machine loaded with Ubuntu 11.10, documents and source code to follow the workshop and perform the experiment, the eclipse IDE and MySQL database server populated with 2 database schemas: one called 'tutorial' used for the workshop, and another one called 'javabase'. The eclipse IDE will be with or without the prototype, depending on the group the participants are in; the transfer of the files should take between 15 to 30 minutes in total for each one of the participants, several USB keys will be used in order to accelerate the process.
- 3) To complete this task, the participants need to unpack the zip file they copied or downloaded before and place the content on the desktop for simplicity, install VMware player, and use VMWare player to load the virtual machine files.

The workshop will begin when the participants have everything ready -- the research personnel will assist them during this time.

- 4) A free pizza lunch will be provided during a break, likely after the setup in step (3) above has been completed.

Workshop Outline

The workshop will begin with a Hibernate tutorial covering the following material outlined



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

here:

- Introduction to Hibernate
- The Development Environment
- The Database
- Creating a Web Project
- Create a Sample Page
- Deploy the Page
- Object/Relational Mapping with Hibernate
- Class Path set up
- The Hibernate Configuration File
- The Event Class
- The Mapping File
- Start up and Helper class
- Manipulating Objects
- Storing and Loading Objects
- Delete
- Update

The duration of the workshop (including lunch) is estimated to be 2.5 hours. The information will be extracted from the official Hibernate documentation chapter one; this information will be taken from the following URL: http://docs.jboss.org/hibernate/core/3.6/reference/en-US/html_single/

At the end of the workshop, participants can choose to leave without compensation (aside from the free lunch and knowledge gained), or continue and participate with the study. If a participant leaves at this moment, they won't receive any cash compensation. If they decide to participate in the experiment, a task will be assigned to them as described below in Section 3.2.

3.2 - The experiment

Depending on the number of participants, we will perform a first study with the participants, using our prototype in the development environment. If we have more than 10 participants, we plan to run a second study condition using the original development environment without modifications (for comparison).

1) We we will briefly explain the purpose of the study based on the information in the consent form. They will also be informed that participation in this study is not mandatory, that all information provided will be anonymized, and that they can withdraw by leaving the workshop or experiment phase and this won't affect the compensation they will receive.

2) We will provide them with a consent form, as well as an opt out form, on which they can indicate their reason for withdrawing from the study if they so choose. Participants will be



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

encouraged to ask any questions to the research personnel. Participants will be given a new ID at this time; they won't need to provide their names in any way except by the consent form, and will include the new ID assigned to them on all documentation they hand in, including their code or application. The relationship between the ID and real name will be kept separate in a document owned by the research team for the duration of the study, after which it will be destroyed.

3) For the experiment, participants will need to modify a small web application that uses the Java specification for databases (JDBC), such that it uses Hibernate instead, and perform any improvements they think are necessary.

The document included with this REB application, named "The experiment", contains the description of the application architecture, and the rules to observe during the task, and the instructions to deliver the product once finished.

4) An overall overview of the tools and how to use them will be presented to participants before they start. Documentation with this same information will be included as reference with the files provided.

The main function of the web application the participants will modify, is to manage personal information, protected by a password. We will explain the function of the base web application and will suggest they use the eclipse plugin to generate mapping files and source code from the existing database schema. This schema contains a table which will include sensitive data such as a SIN number, credit card number; access password, telephone number, email address, and home address.

5) There will be a time limit of 3 hours to complete the task, but it will be simple enough that it should be completed in less than 2 hours, this time limit is set to not rush the participant and avoid producing a low quality product due to time limitations. By using the source code of the web application as a base, this task should be relatively simple to complete for the participants (who have been pre-screened to have the pre-requisite knowledge).

The participants will be asked to develop the solution in a professional way, and will be informed that it is what will ultimately be deployed for a client in a business environment. They will also be informed that their final application's readiness for a business environment is what will be tested, focusing on the 10 Software Quality concepts indicated in the following link: https://en.wikipedia.org/wiki/Software_quality -- note that security is one of these 10 concepts.

We also will encourage them to produce the best quality software by reminding them that the top 3 applications (rated according to the definition of "Software Quality" from the link



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

supplied above) will receive a \$150 Future Shop gift card.

Instructions on what data they need to include, and an email address to send their finished product to, will be provided to each participant in a printed document.

3.3 - At the end of the study

The exit questionnaire will be provided to them at the end of the study, in order to assess their knowledge about security and inquire if they used the security features provided by the tool, what they think about their own work, what changes they would have made if they had more time and why, how the lab setting influenced the product they developed, and if they would do it in a different way in a real-life scenario.

When they leave, participants will receive their compensation.

Debriefing letter

All participants will be contacted via email with the debriefing letter to inform them about the real purpose of the study and give them chance to drop out of the study if they wish to. The debriefing letter will also inform them about the hypothesis and rationale of the experiment, and possible date for the release of the results and how they may access them if they are interested to do so.

- B3. Recruitment:** Describe how and from what sources the participants will be recruited, including any relationship between the investigator(s), sponsor(s) and participant(s) (e.g., family member, instructor-student; manager-employee). Attach a copy of any poster(s), advertisement(s) or letter(s) to be used for recruitment. Remember also to describe the procedures for all stages of the research (e.g., pre-tests, etc.) where applicable.

Participants will be recruited from the UOIT campus, [technology user groups on the internet, IT companies and IT consultants](#). Posters and [a-myCampus emails](#) will be used to attract participants indirectly and the primary investigator and supervisor will directly recruit from classes running at the school. Recruitment will not be from any classes taught by the primary investigator or the research supervisor. Classes targeted will be selected through contacts of the research supervisors. We will give a short verbal presentation to each selected class.



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

Pre-screening of Potential Participants

Participants will need to fill out a web based questionnaire to assess their suitability for the experiment. Depending on the outcome, they will be informed if they may- to participate or not. After this initial phase, they need to complete the mandatory workshop to participate in the study.

- B4. Compensation: The REB is concerned with potential feelings of coercion on the part of the participant. Please provide details addressing coercion issues. (For more information see TCPS discussion regarding compensation under "Minimal Risk" Section C, Article 1 <http://www.pre.ethics.gc.ca/english/policystatement/section1.cfm#IC1> and "Voluntariness" Section B, Article 2.2 <http://www.pre.ethics.gc.ca/english/policystatement/section2.cfm#2B>)

(a) Will participants receive compensation for participation? Yes No

(b) If yes, please provide details.

Participants will be compensated with \$30 dollars for participation in the study but not for participating in the workshop. The workshop is free and will provide the participants with valuable information that they can use in future software development projects. Free pizza and soft drinks will be served during the workshop and the study. If participants finish their application and fill-out the questionnaires, they will have a chance to compete for one of 3 \$150 Future Shop gift cards. We will be rewarding the top 3 applications with a gift card. The top 3 applications will be determined based on the overall "Software Quality" as described to the participants and explained in the following link:
https://en.wikipedia.org/wiki/Software_quality .

- B5. Possible Risks:

a) Physical risks (including any bodily contact, physical stress, or Yes No



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

administration of any substance)?

b) Psychological risks (including feeling demeaned, embarrassed
worried or upset, emotional stress)? Yes No

c) Social risks (including possible loss of status, privacy, and / or
reputation)? Yes No

d) Are any possible risks to participants greater than those that the
participants might encounter in their everyday life? Yes No

e) Is there any deception involved? Yes No

f) Is there potential for participants to feel coerced into contributing to
this research (e.g., because of regular contact between them and the researcher)? Yes No

B6. Description of Risks: If you answered 'yes' to any of the above, please explain the risk.

This study involves deception. Potential participants will be informed that the purpose of the study is to measure the level of readiness that the solutions they provide have for use in a real industrial or commercial environment, and how the tools influence the overall quality of the produced software. This is true, but the main purpose of the study is to assess if the prototype being tested achieved its goals of encouraging the use of encryption to protect sensitive information, thus improving security. Not knowing the main purpose will not disadvantage the participants at all; to be eligible for a Future Shop gift card, their code will be evaluated against the list of software quality metrics provided to them (of which security is one), so they should not feel that the deception reduced their chances of obtaining a prize. They may feel deceived, but this is required in order to avoid influencing the participants' awareness of security by telling them that the study is specifically about security. This concealed purpose is needed as it avoids the participant



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

focusing (unrealistically) on their applications security in order to please the researchers, as in reality, security is often not an explicit requirement in the majority of business projects. If the participants are focused on security more than they would be in a normal situation, it could bias our results for the "Crypto Assistant".

- B7. Management of Risks:** Describe how the risks will be managed (include the availability of appropriate medical or clinical expertise, qualified persons). Give an explanation as to why less risky alternative approaches could not be used. Remember also to describe the procedures for all stages of the research (e.g., pre-tests, etc.) where applicable.

All participants will receive a debriefing letter after the study session, explaining them the real purpose of the study and reminding them about the right to withdraw if they wish to do so.

- B8. Possible Benefits:** Discuss any potential direct benefits to the participants from their involvement in the project. Comment on the (potential) benefits to the scientific community/society that would justify involvement of participants in this study.

The participants will directly receive valuable knowledge about the tools and technologies used and they will also be compensated for their time invested in the study. The benefits to the scientific community are to help determine if the goals of the Crypto Assistant prototype were achieved, which may help reduce security risks in the future, by shaping the way in which developer tools are designed.

- B9. The Consent Process:** Describe the process that the investigator(s) will be using to obtain informed consent. Include a description of who will be obtaining the informed consent. If there will be no written consent form, explain why not. See samples) If applicable, attach a copy of the Letter of Invitation, the Consent Form, the content of any telephone script and any other material that will be utilized in the informed consent process. Remember also to describe the procedures for all stages of the research (e.g., pre-tests, etc.) where applicable.

When the subject gets in contact for recruitment they will be given an overview about what the study consists of, along with a pre-screening consent form. Participants will be provided a link to the pre-screening questionnaire; we will not report any data from



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

volunteers that did not provide consent in the screening process. In the workshop session, they will be given this information again and be presented with the experiment consent form and opt out form before beginning the workshop, thereby allowing them to decide if they want to participate or not. To begin the experiment portion of the study, the participants must sign the consent form. The participants can withdraw from the study at any point; if this occurs their data will be destroyed.

B10. Consent by an authorized party: If the participants are minors or for other reasons are not competent to consent, describe the proposed alternative source of assent (agreement to participate in research from minors), including any permission form to be provided to the person(s) providing the alternative consent.

B11. Alternatives to prior individual consent (e.g. Naturalistic Observation): If obtaining individual participant consent prior to commencement of the research project is not appropriate for this research, please explain and provide details for a proposed alternative consent process.

B12. Acknowledgement/Feedback to Participants: Explain what feedback/ information will be provided to the participants after participation in the project. Include, for example, appreciation for participation, a more complete description of the research purpose, any results that may be available, and participant access to a final results summary. Also, describe the method and timing for delivering the feedback.

There will be a feedback letter through which the participants will be informed of the outcomes of the study via email, with further information about the research and information



Research Ethics Board (REB)
Application for Ethical Review of Research Involving Human Participants

about how to get access to the full documentation.

B13. Participant withdrawal:

- Describe how the participants will be informed of their right to withdraw from the project. Outline the procedures that will be followed to allow the participants to exercise this right. **Remember also to describe the procedures for all stages of the research (e.g., pre-tests, etc.) where applicable.**

Participants will be informed of their right to withdraw on written form and verbally if possible by the facilitator of the study at the following junctures:

When they get in contact with the research personnel for the first time:

Before beginning the pre-screening questionnaire they will receive an email and a consent form indicating that participation is voluntary, and they can withdraw by sending an email to the recruiter's email indicating that they don't wish to participate anymore.

Participants that begin the pre-screening questionnaire will be entitled to get a chocolate bar if they are not eligible or if they decide to withdraw the study.

At the beginning of the workshop:

They will be provided with the experiments consent form and informed verbally by the study facilitator that participation is voluntary and they can withdraw by talking to or emailing the research personnel, and will be invited to fill out the opt out form, if they would like (as the form is optional), to inform the researchers of their reason to leave.

Participants that withdraw from the study during workshop and before the beginning of the programming task won't be entitled to any compensation; participants that withdraw during the programming task will receive \$30 for their time but won't be eligible to participate for the chance to win the \$150 gift card.

- Indicate what will be done with the participant's data and any consequences that withdrawal might have on the participant, including any effect that withdrawal may have on participant compensation.



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

If they decide that they don't want their data to be used as part of the study their data will be removed (all electronic data will be deleted and hard copies will be shredded, including all pre-screening and other questionnaire data and forms).

Section C) Safeguards in place to protect participant and data

Confidentiality: information revealed by participants that holds the expectation of privacy (this means that all data collected will not be shared with anyone except the researchers listed on this application).

Anonymity: information revealed by participants will not have any distinctive character or recognition factor, such that information can be matched to individual participants (any information collected using audio-taping or video recording cannot be considered anonymous).

Remember also to describe the procedures for all stages of the research (e.g., pre-tests, etc.) where applicable.

C1. Given the definitions above:

Confidentiality	
1.	Will the data be treated as confidential? <input checked="" type="checkbox"/> Yes <input type="checkbox"/> No
2.	Describe the procedures to be used to ensure the confidentiality of data both during the conduct of the research and in the release of its findings. All the data will be identified by anonymized codes which will be assigned to each participant at the time of the experiment. All the files containing personal or identifiable information will be stored in a secure location at UOIT, and only the researchers will have access to this data. This information will be keep separate from the data collected during the experiment.
3.	If participant confidentiality is not appropriate to this research project, explain, providing details, how all participants will be advised that data will not be confidential.



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

Anonymity	
4. Are the data anonymous? Participants will use anonymized codes to communicate and deliver any product for this research, the relation key linking participants identity with this code will be keep in an encrypted USB drive, in possession of the researchers [Ricardo Rodriguez Garcia, Miguel Vargas Martin and Julie Thorpe]. This data will be destroyed along with any other identifiable information no later than August <u>October</u> 30 th 2012.	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No
5. Describe the procedures to be used to ensure anonymity of participants in the release of its findings. The data will be condensed and anonymized and no information will be provided about the identity of any of the participants, except for their group demographic characteristics.	
6. If participant anonymity is not appropriate to this research project, explain, providing details, how all participants will be advised that data will not be anonymous.	

C2. State who will have access to the data.

Only the researchers of this project listed here will have access to any data collected:

Ricardo Rodriguez Garcia, Miguel Vargas Martin and Julie Thorpe.

C3. Explain how written records, video/audio tapes, and questionnaires will be secured, and provide details of their final disposal or storage (including for how long they will be secured and the disposal method to be used). Remember also to describe the procedures for all stages of the research (e.g., pre-tests, etc.) where applicable.

- I plan to keep raw data and aggregate data indefinitely, without identifiers.
- I plan to keep raw data and aggregate data indefinitely, with identifiers. Describe the storage method.



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

I plan to keep raw data and aggregate data, with identifiers for the time period of (please specify date, where you are storing the data and why you are keeping the identifying data):

I plan to destroy and/or dispose of the data. Please describe how and when it will be destroyed (remember this should be communicated to participants during the consent process).

All data collected will be stored in an encrypted USB drive in possession of the research personnel during the duration of the study; hard copies will be stored in a locked drawer in the office of Dr. Julie Thorpe at UOIT. Once the research from which this study is part of, is completed, and no later than ~~October~~ August 30 2012, all data linking personal identifiers to anonymized data will be deleted or destroyed. Raw data without identifiers will be kept indefinitely. Thus, the consent forms provide a dead line to withdraw from the study that will be the same date listed here for the destruction of the linking personal identifiers.

C4. SECONDARY USE OF DATA

- 1. I understand that if I use the data for purposes other than described in this application that consent must be sought from participants.
 I agree to this statement.

(b) If there are no plans to use the data with identifiers for secondary purposes and yet, you wish to keep the data indefinitely, please briefly explain why.

[Empty rectangular box for explanation]

C5. Study Completion and Annual Report/Continuing Review Form: For the purposes of monitoring ongoing research, the REB requires the completion of the "Study Completion Report/Form" form at the completion of the research and an "Annual Report/Continuing Review Form" at least annually.

- 1. Identify approximate dates when the REB should expect to receive reports on the progress or final report on the research.
2. Indicate whether any additional monitoring or review would be appropriate for this project. (Consider risks of research)



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

Final reports about the progress of the research will be available no later than ~~October~~^{August} 30th 2012.

Due to the harmless nature of this experiment we consider that no additional monitoring is necessary for this research.

Section D: Signature pages

SIGNATURES : All investigators (Principal and Co-investigators) are required to sign the ethics application. (Fax signatures are acceptable for our records.) Student researchers are required to obtain approval and signature from his/her faculty supervisor.

Please indicate that you have read and fully understand all ethics obligations by checking the box beside each statement. Failure to submit the signatures may result in a delay in processing your request for ethics approval.

Name of Principal Investigator:	(e.g., faculty, student, visiting professor, other affiliation)	Dept./Address	Phone No.	E-Mail
Ricardo Rodriguez Garcia	Graduate Student	FBIT		ricardo.rodriuezgarcia@uoit.ca

<input checked="" type="checkbox"/>	I have read the University of Ontario Institute of Technology Research Ethics Policy and Procedures and agree to comply with the policies and procedures outlined therein.
<input checked="" type="checkbox"/>	I will report any Adverse/Unanticipated Events (unanticipated negative consequences or results affecting participants) to REB Administration and the REB Chair, as soon as possible and in any event, no more than 3 days subsequent to their occurrence to the Research Ethics Board (REB).
<input checked="" type="checkbox"/>	Any additions or changes in research procedures after approval has been granted will be submitted to the REB.
<input checked="" type="checkbox"/>	I agree to complete an Annual Report/Continuing Review and a Change Request form for any project continuing beyond the expected date of completion or for more than one year.
<input checked="" type="checkbox"/>	I will submit a final report to the Office of Research Services once the research has been completed.
<input checked="" type="checkbox"/>	I take full responsibility in ensuring that all other investigators involved in this research follow the protocol as outlined in the application.



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

X

PLEASE NOTE: If you are unable to insert your digital signature please print, sign, and fax or scan/email it to us. Thanks.

All co-investigators must be listed and signatures obtained.

Attach additional investigator information/signature pages to this application (found at [Forms | Research](#))

Name of Co-Investigator(s)	(e.g., faculty, student, visiting professor, other affiliation)	Dept./Address	Phone No.	E-Mail

- I have read the [University of Ontario Institute of Technology Research Ethics Policy and Procedures](#) and agree to comply with the policies and procedures outlined therein.
- I will report any Adverse/Unanticipated Events (unanticipated negative consequences or results affecting participants) to REB Administration and the REB Chair, as soon as possible and in any event, no more than 3 days subsequent to their occurrence to the Research Ethics Board (REB).
- Any additions or changes in research procedures after approval has been granted will be submitted to the REB.
- I agree to complete an Annual Report/Continuing Review form or a Change Request form for any project continuing beyond the expected date of completion or for more than one year.
- I will submit a final report to the Office of Research Services once the research has been completed.
- I take full responsibility in ensuring that all other investigators involved in this research follow the protocol as outlined in the application.

X

PLEASE NOTE: If you are unable to insert your digital signature please print, sign, and fax or scan/email it to us. Thanks.

Name of Faculty Supervisor(s) <i>(for student PIs only)</i>	(e.g., faculty, student, visiting professor, other affiliation)	Dept./Address	Phone No.	E-Mail



Research Ethics Board (REB)

Application for Ethical Review of Research Involving Human Participants

Miguel Vargas Martin	Faculty / Professor	FBIT	905 721 8668 ext 2834	miguel.vargasmartin@uoit.ca
----------------------	---------------------	------	-----------------------	-----------------------------

- I agree to provide the proper surveillance of this study to ensure that the rights and welfare of all human participants are protected.
- I will ensure an Annual Report/Continuing Review form or a Change request form for any project continuing beyond the expected date of completion or for more than one year is completed.
- I have read and approved the application and proposal.

X

PLEASE NOTE: If you are unable to insert your digital signature please print, sign, and fax or scan/email it to us. Thanks.

Name of Faculty Supervisor(s) <i>(for student PIs only)</i>	(e.g., faculty, student, visiting professor, other affiliation)	Dept./Address	Phone No.	E-Mail
Dr. Julie Thorpe	Faculty / Professor	FBIT	1 905 721 8668 ext 6585	julie.thorpe@uoit.ca

- I agree to provide the proper surveillance of this study to ensure that the rights and welfare of all human participants are protected.
- I will ensure an Annual Report/Continuing Review form or a Change request form for any project continuing beyond the expected date of completion or for more than one year is completed.
- I have read and approved the application and proposal.

X

9.2 Change renewal request

1



RESEARCH ETHICS BOARD
OFFICE OF RESEARCH SERVICES

Change Request and/or Study Renewal Form

For Office Use Only:	
Date Received:	REB # _____

1.0 Purpose

This form must be filled out for all Research Projects that are:

1. Requesting Changes to a previously approved Protocol, or
2. Renewing your Protocol AND Requesting Changes, or
3. Wishing to Renew your Protocol without any Changes.

Please note that if there are significant deviations from the original approved protocol, the REB may request a new REB Application or additional information.

2.0 Instructions

For all users of this form, fill out Sections 1, and then follow directions from each Option. Submit **ONE Signed Softcopy** of this form along with all attachments to compliance@uoit.ca. Hand written forms will NOT be accepted.

Section 1a: Principal Investigator Information	
REB File #:	11-096
Project Title:	The influence of development tools on aspects of software security
First Name:	Ricardo
Last Name:	Rodriguez Garcia
Email:	ricardo.rodriguezgarcia@uoit.ca

Section 1b: Status of Protocol	
<input type="checkbox"/> Option 1 (Changes ONLY) Proceed to Section 2 and complete all relevant sections	There have been Changes to the Protocol since receiving original REB Approval. I am requesting the changes found in this form approved.
<input checked="" type="checkbox"/> Option 2 (Changes AND Renewal) Proceed to Section 2 and complete all relevant sections	There have been Changes to the Protocol since receiving original REB Approval. I am requesting to have the changes found in this form approved. This study is continuing and requires renewal until Research Project Completion Date.
<input type="checkbox"/> Option 3 (Renewal ONLY) Proceed to Section 6	There have been NO Changes to the Protocol since receiving original REB Approval and I am requesting a Study Renewal.
If you have selected Options 1 or 2 , continue and complete all sections of this form.	

Office of Research Services ~ 2000 Simcoe St. N. Oshawa ON ~ FAX (905) 721-3210
Change Request and/or Study Renewal Form

Section 2 ~ Leave BLANK if there are NO Changes Requested	
2a: Co-Investigator (list ONE, if applicable)	
First Name:	
Last Name:	
Position/Affiliation:	
Email:	
2b: Faculty Supervisor (for Student Projects only)	
First Name:	Dr. Miguel Vargas Martin and Julie Thorpe
Last Name:	
Position/Affiliation:	Associate Professor and Assistant Professor (respectively), FBIT
Email:	miguel.vargasmartin@uoit.ca, julie.thorpe@uoit.ca

Section 3 ~ Leave BLANK if there are NO Changes Requested	
3a: General Project Information	
Title of Project:	
Faculty Investigators:	
Student Investigators:	
Co-Investigators:	
Research Start & End Dates:	13/09/2012, 30/10/2012
Locations:	
Other REB Approvals:	
Risk/Level of Project:	
Funding of Project:	
Conflict of Interest:	
3b: General Project Information	
Purpose/Rationale for Research:	
Methodology/Procedures:	
Previous Experience/Expertise:	
Participants Involved in Study:	
Recruitment Process/Materials:	We will be recruiting additional participants from outside UOIT, sending emails to user groups on the internet, including IT companies or IT consultants.
Compensation for Participants:	
3c: Benefits and Risk	
Possible Benefits:	
Possible Risks:	
3d: Invitation/Consent Process	
Informed Consent/Absence of Consent:	
Use of Deception:	
Process of Parental/Guardian Consent:	
3e: Confidentiality	
Procedures to ensure confidentiality:	
Who will have access to the data? List ALL individuals:	
3f: Secondary Use of Data	
Plans for Using Data for Other Purposes:	

Section 4 ~ Leave BLANK if there are NO Changes Requested
If the revision(s) is/are to a Questionnaire, Interview Script, Verbal Script, Information Letter, Consent Form, Thank you Letter, or any other material with previous ethics clearance, please attach

Office of Research Services ~ 2000 Simcoe St. N. Oshawa ON ~ FAX (905) 721-3210
Change Request and/or Study Renewal Form

the entire document and highlight the sections that are modified. Describe the changes below.
Addition of questions 4-8 to the screening questionnaire, addition of questions 18,20, 21 and changes in the wording used in the exit questionnaire.
Section 5: Other ~ Leave BLANK if there are NO Changes Requested
Any other changes (please specify and describe changes below)
We were not able to obtain a significant number of participants from UOIT. Therefore, we are requesting a change consisting of being able to also send out email invitations (using same email script as before) to people outside UOIT, including IT companies or IT consultants.

Section 6: Adverse or Unexpected Events	
Have there been any Adverse/Unanticipated Events that occurred? (if Yes, please submit an Adverse Event form immediately)	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No

Section 7: Signatures
Principal Investigator: I/We respectfully request Ethics Approval of the modifications/revisions described above for the review and approval. All relevant documentation has been included for review in this submission (if applicable).

 Invalid signature

X Ricardo Rodriguez Garcia

Signed by: Ricardo Rodriguez Garcia

Signature of Principal Investigator

Date: 21/08/2012

 Invalid signature

X 

Signed by: Miguel V. Martin

Signature of Faculty Supervisor (if applicable)

Date: 21/08/2012

REB Chair:

I Approve the Modifications/Revisions described above and/or included with submission.

X

Signature of REB Chair

Date:

Office of Research Services ~ 2000 Simcoe St. N. Oshawa ON ~ FAX (905) 721-3210
Change Request and/or Study Renewal Form

9.3 Change request approval



RESEARCH ETHICS BOARD
OFFICE OF RESEARCH SERVICES

Date: September 5th, 2012

To: Ricardo Rodriguez Garcia (Graduate Student), Dr. Miguel Vargas Martin (Supervisor) and Dr. Julie Thorpe (Supervisor)

From: Amy Leach, REB Chair

REB File #: 11-096

Project Title: The influence of development tools on aspects of software security

DECISION: CHANGE REQUEST APPROVED

CURRENT EXPIRY: April 25th, 2013

The University Of Ontario Institute Of Technology Research Ethics Board has reviewed and approved the change request. The application in support of the above research project has been reviewed by the Research Ethics Board to ensure compliance with the Tri-Council Policy Statement: Ethical Conduct for Research Involving Humans (TCPS2) and the UOIT Research Ethics Policy and Procedures.

Please note that the Research Ethics Board (REB) requires that you adhere to the protocol as last reviewed and approved by the REB.

Always quote your REB file number on all future correspondence.

Please familiarize yourself with the following forms as they may become of use to you.

- **Change Request Form:** any changes or modifications (i.e. adding a Co-PI or a change in methodology) must be approved by the REB through the completion of a change request form before implemented.
- **Adverse or unexpected Events Form:** events must be reported to the REB within 72 hours after the event occurred with an indication of how these events affect (in the view of the Principal Investigator) the safety of the participants and the continuation of the protocol. (I.e. un-anticipated or un-mitigated physical, social or psychological harm to a participant).
- **Research Project Completion Form:** must be completed when the research study has completed.
- **Renewal Request Form:** any project that exceeds the original approval period must receive approval by the REB through the completion of a Renewal Request Form before the expiry date has passed.

All Forms can be found at http://research.uoit.ca/EN/main/231307/Research_Forms.html.

REB Chair Dr. Amy Leach, SSH Amy.leach@uoit.ca	Ethics and Compliance Officer compliance@uoit.ca
--	---

University of Ontario, Institute of Technology
2000 Simcoe Street North, Oshawa ON, L1H 7K4
PHONE: (905) 721-8668, ext. 3693

10 Appendix D Code documentation

Crypt-o-assistant code documentation

What has changed?

Reverse engineering

Hibernate tools allow to generate Java classes and mapping files from database tables; through the configuration file reveng.xml the default mapping types can be overridden.

Valid type values in the reveng.xml are:

- Hibernate type names
- Java types with a direct map to Hibernate types
- User defined types by class name

Taking advantage of how the code generation works we expanded Hibernate tools to recognize aliases or named types defined in mapping files as valid type names in the reveng.xml.

The crypto assistant helps developers to generate the TypeDef.hbm.xml file containing all the required named types for the encrypted field and the reveng.xml file to override the default types and use the ones generated instead.

Finally the developer must manually update its database structure and existing data to fit the encrypted data.

```
<typedef name="encryptedString"
class="org.jasypt.hibernate3.type.EncryptedStringType">
  <param name="algorithm">PBEWithMD5AndDES</param>
  <param name="password">jasypt</param>
  <param name="keyObtentionIterations">1000</param>
</typedef>
```

An example of a named type definition in a typeDef.hbm.xml file.

```
<hibernate-reverse-engineering>
  <table-filter match-catalog="tutorial" match-name="events" />
  <table catalog="tutorial" name="events">
    <column name="title" type="encryptedString"></column>
  </table>
</hibernate-reverse-engineering>
```

The reveng.xml file using the named type "encryptedString" this is possible thanks to the changes introduced by the prototype.

Mapping file(s) generation from Java classes

Hibernate tools allows to generate mapping files from Java classes.

The developers select the files to use for the mapping generation, this files are then analyzed by Hibernate and the mapping files are generated finally a preview screen shows the results before finishing putting the files in the user workspace.

The Crypt-o-Assistant allows the user to select the fields that must be encrypted in the database, when the mapping files are generated they contain the embedded types definition to store encrypted data.

```
<property name="name">
  <column name="NAME" />
  <type name="org.jasypt.hibernate3.type.EncryptedStringType">
    <param name="algorithm">PBEWITHMD5ANDES</param>
    <param name="password">hgblapv1fv0lo69pg1uf02qg81</param>
    <param name="keyObtentionIterations">1000</param>
  </type>
</property>
```

An embedded parametrized type definition in the mapping file; in this case the column "NAME" will store encrypted data.

Code changes

The changes performed to the code are divided in 2:

HibernateTools jar library:

- <https://github.com/fulano2040/hibernate-tools/commits/master>
- hibernate-tools.zip

HibernetTools eclipse plugin:

- <https://dl.dropbox.com/u/82168038/hibernatetools.zip>
- JBTOOLS.zip
- A diff file is included for this changes

The prototype provides new functionality and improves or solves some of the deficiencies found in the previous version, this will be classified as:

New functionality:

Type of change	New functionality
Description	Generation of mapping files and annotated Java code from a database schema to enable the use of Jasypt encryption

Files affected	<p>A brief explanation of the changes is provided followed by bullets indicating the files affected</p> <p>The "Reverse engineering file" wizard was modified</p> <ul style="list-style-type: none"> • plugins/org.hibernate.eclipse.console/src/org/hibernate/eclipse/console/wizards/NewReverseEngineeringFileWizard.java <p>Warning screen was added to the reverse engineering wizard to encourage users to use encryption</p> <ul style="list-style-type: none"> • plugins/org.hibernate.eclipse.console/src/org/hibernate/eclipse/console/wizards/encryption/EncryptionWarningWizardPage.java <p>A new wizard page to select the fields that need encryption was created</p> <ul style="list-style-type: none"> • plugins/org.hibernate.eclipse.console/src/org/hibernate/eclipse/console/wizards/ColumnEncryptionPage.java <p>Preview screen was created to review the content of the files generated</p> <ul style="list-style-type: none"> • plugins/org.hibernate.eclipse.console/src/org/hibernate/eclipse/console/wizards/encryption/NewRevEngPreviewPage.java <p>A class to provide automatic mapping between SQL and Jasypt types was created, it also provides automating mapping from Hibernate types to Jasypt types</p> <ul style="list-style-type: none"> • plugins/org.hibernate.eclipse.console/src/org/hibernate/eclipse/codegen/JasyptTypeHelper.java <p>Column and table object models were implemented to temporarily store information and customize the different parameters used during the reverse engineering file definition generation:</p> <ul style="list-style-type: none"> • plugins/org.hibernate.eclipse.console/src/org/hibernate/eclipse/console/model/impl/RevEngColumnImpl.java • plugins/org.hibernate.eclipse.console/src/org/hibernate/eclipse/console/model/impl/RevEngTableImpl.java • plugins/org.hibernate.eclipse.console/src/org/hibernate/eclipse/console/model/impl/ReverseEngineeringDefinitionImpl.java
----------------	--

	<p>Class to pass properties between wizard pages</p> <ul style="list-style-type: none"> • <code>plugins/org.hibernate.eclipse.console/src/org/hibernate/eclipse/console/wizards/RevEngWizardProperties.java</code> <p>Table filter page was modified to prevent continue with the wizard without a hibernate configuration file</p> <ul style="list-style-type: none"> • <code>plugins/org.hibernate.eclipse.console/src/org/hibernate/eclipse/console/wizards/TableFilterWizardPage.java</code>
--	--

Bug fixes

A custom checked tree selection was created to fix a bug in the eclipse implementation: selected elements didn't correspond with the underlying structure, if two nodes had the same content it just returned the first element it was finding and the selected element didn't correspond to what was shown in the screen

- `plugins/org.hibernate.eclipse.console/src/org/hibernate/eclipse/console/wizards/encryption/CustomContainerCheckedTreeView.java`

Migration to Hibernate 3.6

The migration to Hibernate 3.6 was required to use its dynamic type binding capabilities to enable the use of named types in the reverse engineering file definition. This change affected the templates used for code generation and some libraries that required an update.

- `plugins/org.hibernate.eclipse.console/src/org/hibernate/eclipse/console/views/QueryParametersPage.java`

The migration to version 3.6 required a bunch of libraries to be updated this was reflected in this file

- `plugins/org.hibernate.eclipse.libs/.classpath`

Migration from Apache commons logging to slf4j

Templates update to use new validator for xml

Mapping files generation from Java classes

Type of change	New feature
----------------	-------------

Description	Mappings file generation
Files affected	<p>The mapping file wizard was modified to integrate Hibernate 3.6 functionality and to add the property encryption page</p> <ul style="list-style-type: none"> • plugins/org.hibernate.eclipse.jdt.ui/src/org/hibernate/eclipse/jdt/ui/wizards/NewHibernateMappingFileWizard.java <p>A new page for the selection of properties to encrypt was added</p> <ul style="list-style-type: none"> • plugins/org.hibernate.eclipse.jdt.ui/src/org/hibernate/eclipse/jdt/ui/wizards/NewHibernatePropertyEncryptionPage.java <p>logger facilities were added to the preview page</p> <ul style="list-style-type: none"> • plugins/org.hibernate.eclipse.jdt.ui/src/org/hibernate/eclipse/jdt/ui/wizards/NewHibernateMappingPreviewPage.java

Contextual help for property encryption page

Type of change	New feature
Description	Contextual help to aid in the identification of sensitive information was added to the plugin, the following files were modified.
Files affected	<ul style="list-style-type: none"> • plugins/org.hibernate.eclipse.jdt.ui/build.properties • plugins/org.hibernate.eclipse.jdt.ui/HelpContext.xml • plugins/org.hibernate.eclipse.jdt.ui/html/SensitiveDataExamples.html • plugins/org.hibernate.eclipse.jdt.ui/html/WhatIsSensitiveData.html • plugins/org.hibernate.eclipse.jdt.ui/plugin.xml

ConfigurationActor class

Type of change	New feature
Description	This class was reworked to provide the encryption capabilities needed; it encapsulates the main functionality for the generation of mapping files and in consequence it suffered several changes.

	<p>Many changes were done to include the use of Hibernate version 3.6 as well as a bunch of bugs fixes and improvement to existing functionality. A brief explanation of what changed is provided next:</p>
Files affected	<ul style="list-style-type: none"> • src/org/hibernate/eclipse/jdt/ui/wizards/ConfigurationActor.java
Methods affected	<p>A brief explanation of the modifications to each method is given in this file due to the large amount of changes:</p> <p>Exclude static and final fields from being selected as entity id, if no id field is present in the Java class</p> <ul style="list-style-type: none"> • org.hibernate.eclipse.jdt.ui.wizards.ProcessEntityInfo.visit(TypeDeclaration) <p>Add an id column to the entity if no id field was present and none of its properties is suitable to be used as an Id</p> <ul style="list-style-type: none"> • org.hibernate.eclipse.jdt.ui.wizards.ProcessEntityInfo.endVisit(TypeDeclaration) <p>Proper generation strategy of id values in mapping files</p> <ul style="list-style-type: none"> • org.hibernate.eclipse.jdt.ui.wizards.ProcessEntityInfo.visit(FieldDeclaration) <p>New table generation for primitive arrays collections that represent a one to many relationship (3.6)</p> <ul style="list-style-type: none"> • org.hibernate.eclipse.jdt.ui.wizards.TypeVisitor.visit(ArrayType) <p>Increase accuracy of mapping file generation for parametrized type collections (generics) for different cardinalities(1-to-1,1-to-many,many-to-many)</p> <p>This improves the mappings generated for relations defined by maps and the table names used for many-to-many relationships taking in account the entity that own the relation.</p> <ul style="list-style-type: none"> • org.hibernate.eclipse.jdt.ui.wizards.TypeVisitor.visit(ParameterizedType) <p>3.6 migration and separate table for collections of simpleValues (primitives)</p> <ul style="list-style-type: none"> • org.hibernate.eclipse.jdt.ui.wizards.TypeVisitor.visit(SimpleType) <p>3.6 migration</p> <ul style="list-style-type: none"> • org.hibernate.eclipse.jdt.ui.wizards.TypeVisitor.buildSimpleValue(String) <p>3.6 migration, instead of assigning a String type by default it tries to get the correct key type</p> <ul style="list-style-type: none"> • org.hibernate.eclipse.jdt.ui.wizards.TypeVisitor.buildCollectionValue(ITypeBinding[])

Misc improvements:

This file was modified to allow use of the UI design editor

- `plugins/org.hibernate.eclipse.console/src/org/hibernate/eclipse/console/wizards/TreeToTableComposite.java`

Hibernate configuration wizard

The driver class definition for mysql 5 was added to make it available from the wizard page

- `plugins/org.hibernate.eclipse.console/src/org/hibernate/eclipse/console/utils/DriverClassHelpers.java`

The capability to select the value of the hbm2ddl on the wizard page was added

- `plugins/org.hibernate.eclipse.console/src/org/hibernate/eclipse/console/wizards/NewConfigurationWizard.java`
- `plugins/org.hibernate.eclipse.console/src/org/hibernate/eclipse/console/wizards/NewConfigurationWizardPage.java`