HAAGA-HELIA
University of Applied Sciences

Designing and executing a security and usability testing plan: IdeaClick Prototype

Manuel Bacso

Omar Lenin Gutiérrez Gutiérrez

Thesis

DP in Business Information Technology

28 May 2013

HAAGA-HELIA
University of Applied Sciences

Abstract

Date of presentation of thesis

Degree Programme in Business Information Technology

Authors Manuel Bacso Omar Lenin Gutiérrez Gutiérrez	**Year of entry** 2009
The title of your thesis Designing and executing a security and usability testing plan: IdeaClick Prototype	**Number of pages and appendices** 58 + 1
Supervisors Pekka Kamaja, Timo Haukola, Päivi Pöyry-Lassila	

The purpose of this thesis was to define, analyse and execute usability and security testing plans for an application prototype named IdeaClick. IdeaClick is a web application and its main purpose is to share ideas and provide a means to collaborate on them.

The usability aspects of IdeaClick were tested and analysed based on two standard web user interface methodologies. The first method followed the black-box procedure, consisting of I/O data. The second one was a heuristic evaluation done by the testing facilitator responsible and the testers. The security of this prototype was tested following the OWASP guidelines for web application security testing. This non-profit organization provides an extended guide on how to test a web application regarding various security vulnerabilities.

As a result of this thesis, two individual testing plans were created. In addition, the study includes a description of the prototype and its features.

The thesis concludes that the prototype functions well; however, the overall usability should be improved by investing more time in the further development of the user interface. Furthermore, some security flaws were found and they require attention as soon as possible.

Key words

Security, Usability, Testing, IdeaClick, Web Application

Table of contents

Table of Figures

Abbreviation

R&D	Research and development
IT	Information technology
OWASP	Open Web Application Security Project
SW	Software
VISCI	Virtual Intelligent Space for Collaborative Innovation
UI	User Interface

1 Introduction

In the 21st century software is a part of everyone's life even if we do not realize it. Software development is at the moment blooming but what is going on behind the scenes? Only a fraction of all the software that is being developed sees the end customer. A big reason for this is that the development process itself often fails. There are many root causes behind these failures: overly optimistic schedules, unrealistic expectations, excessive multi-tasking, short-changed quality assurance and many others. When realizing the extent in which this impact the field the natural question to bring up is: How can this problem be avoided? (Bootstraptoday 2012.)

The answer is not any simpler than the problem itself. According to general opinion proper planning and extensive testing are very important. The hard part is thought to know what to test, when to test and how to test. It is also vital to analyse the development at different phases and steer the project to right direction.

In this thesis a Virtual collaborative prototype is analysed, called IdeaClick, which has been researched at Enterprise Simulation Laboratory SimLab, Aalto University School of Science. This prototype was one of the outcomes of a two year long research project named VISCI Tools. One of the authors of this thesis, Manuel Bacso, was at the time working as a technical assistant at VISCI Tools and was deeply involved with the software development done during the project.

IdeaClick is a web based virtual environment, which acts as an interface to a platform called IdeaPlatform. In IdeaClick users share, manipulate and modify ideas that have been created in the environment, as well as establishing relationships between them. A relevant point concerning IdeaClick is, that it the first of its kind.

To analyse the prototype at its current development stage is very important because it provides a description of the platform as well as labels its current functionality and visualization capabilities. This thesis also increases our knowledge of both usability and

security of web applications as well as how to design and execute proper testing plans accordingly.

1.1 Objectives and scope of the study

The objective of this thesis is to generate substantial information of the IdeaClick prototype, allowing developers of IdeaClick, to have a clear understanding of the current stage. This information should consist of the usability and security of the prototype and will play a big part in the decision making process of its future utilization.

As described above, the primary objective is to analyse the IdeaClick prototype at its current development stage. This includes, providing a description of the platform as well as label its current functionality and visualization capabilities.

The secondary objective of this thesis is to increase our knowledge of both usability and security of web applications as well as how to design and execute proper testing plans accordingly. In addition to that, providing a recommendation based on the analysis results.

The Scope of the thesis is to design and execute testing plans for the web application prototype IdeaClick, as well as summarize and analyse the results of these tests.

These test cases will focus specifically on the Usability design and the Security weaknesses of this prototype. A description of the current stage of the prototype will also be defined. This Thesis is focused entirely on the IdeaClick Prototype of the IdeaPlatform.

In addition to the IdeaClick prototype, six other prototypes were developed which are all linked by a powerful core. These additional prototypes will not be analysed in the thesis, but a brief insight is provided. These prototypes were split up into three categories: The creation prototypes, allowing for users to share their ideas, relate pictures to them and easily collaborate; the organizational prototypes, which focus mainly on allowing the users to relate objects to one another as well as grouping them together; and

the visualization prototypes, which started off as demo prototypes to be used in workshops and presentations and moved on to become interfaces to easily stumble upon new ideas. Due to the platform being so large, we have decided to focus in detail on the IdeaClick prototype and disregard the rest.

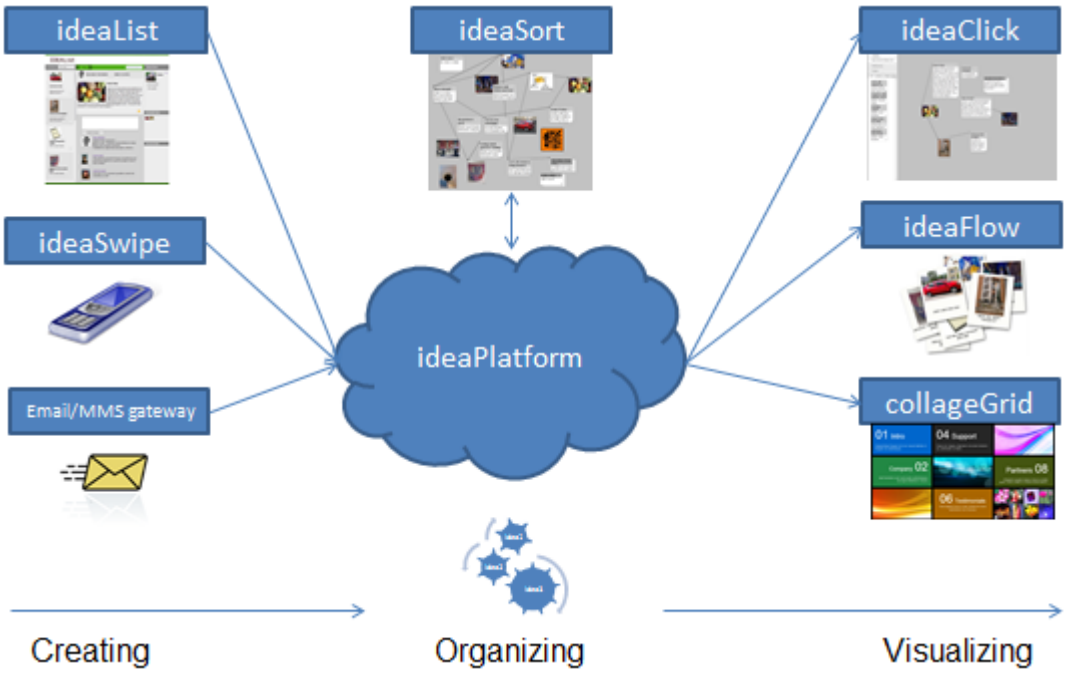


Figure 1 IdeaPlatform prototype by type

The IdeaClick prototype is still at its prototype stage and has been developed and tested only on the Windows environment using the Firefox browser. These tests were mainly to make sure the prototype functioned decently. As a result we have chosen to only test the platform on the same environment it was initially developed.

1.2 Deliverables and Environment

Each Test plan is to be executed and well documented, failed tests are to be documented as well and reasons to failure defined. Each of the authors of the thesis is responsible for their respective topics; these topics were chosen due to their prior knowledge and interest in the field. Some University courses that the authors have participated in are mentioned in header 6 Testing Plan.

- Usability testing plan, results and recommendations
    - Person responsible: Omar Gutierrez
- Security testing plan, results and recommendations

- Person responsible: Manuel Bacso

In the thesis the steps used for the development and documentation are split into three phases: 1) the data gathering phase, 2) the testing phase and the 3) documentation phase.

The first is the data gathering phase in which the team utilizes its time to gather all useful information and documentation by reading through the project material and documentation written about the prototypes. Secondly comes the testing phase, where the platform is put under usability and security testing, includes interviews with the researchers involved with the VISCI Tools Project. The tests will be performed on the Windows Operating system and will run only on the Firefox web browser, due its development and initial ad hoc testing. Finally comes the Documentation phase where the result is utilized to sum up all findings and complete the Thesis.

The Thesis project will make use of the following tools and languages, categorized by their phases:

Data Gathering

- MS-Office
- Acrobat Reader
- Dropbox (File Sharing)
- Visual Studio 2010
- Mozilla Firefox (Windows)
- WebScarab
- Firebug (Mozilla Firefox)

Testing

- Mozilla Firefox (Windows)
- Dropbox
- MS-Office
- WebScarab

- Firebug (Mozilla Firefox)

Documentation

- MS-Office
- Acrobat Reader
- Dropbox

Other Technologies Used

- UML
- XHTML
- Jquery

1.3 Summary of Study

The following two chapters contain the theoretical content of the thesis, starting with a background of the prototype, followed by the essentials of software testing. These essentials contain the basics of software testing, followed by a brief description of selected software testing types used frequently in system testing.

Next, two software testing types, Usability and Security definitions, are described in depth, providing a good understanding of the testing plans at the end of the document. Each author has chosen one of these testing types to be used for the testing of the IdeaClick prototype. The test plans can be found as attachments at the appendices section of this document. The two attachments consist of the usability testing plan and security testing plan.

The thesis is finished with a summary containing the conclusions of the testing phases, as well as opinions and other relevant information concerning to the entire process of creating this thesis.

2 Orientation to Study

This chapter is dedicated to providing a context to the usability and security theoretical analysis. The first section describes the IdeaClick prototype and its features. The following section provides a brief definition of software testing; this provides an understanding to the next chapter.

2.1 Web Applications and IdeaClick

This chapter gives a short introduction about web application´s history and the benefits that come along with its use. In addition, it tells of IdeaClick services and how to use them through the menu.

The World Wide Web started off with just very few websites. These websites were very simple, they didn't include any graphical design or fancy images, or even colors to brighten up the design. In the early days, websites consisted of only text, written from one line to the next, the biggest changed you could make to the design was writing a text in bold or changing its size.

With all these large companies providing means of communication over the internet, it is easy to understand why it has grown so widely over the past few years. A few widely used web applications include:

- Google – Search Engine
- Facebook – Social Networking
- Amazon – Online Shopping
- eBay – Auctions
- Blogger – Blogging
- Gmail – Email Service
  (Stuyyard, D. & Pinto, M. 2011, 39.)

Web Applications provide us with the ability to know what is happening around the world (News), where to get the cheapest products (e.g. Ebay) and gives us the means of sharing our thoughts and communicating with friends (Social Networking).

2.1.1 IdeaClick

The IdeaClick interface is designed for visualizing and giving the user the possibility to create relations and groupings (collages) between Ideas and Pictures. This interface is very similar to working on a table with post-it notes and pictures on it, where the user tries to arrange these objects in the best possible way. IdeaClick features the following:

- Complex interface for visualizing relationships between all Inno-Objects
- Panels for showing / hiding and filtering of Inno-Objects
- Ability to add Ideas and pictures to the platform
- An interface that allows the creation and removal of relationships and collages
- Personal workspace by constantly saving the work progress giving the ability to recall the workspace from any location or device

Below is a screenshot of the IdeaClick prototype. Visible here are different ideas, all with a light grey header. An image is also visible and a Collage, marked in dark grey, displayed as a stack of ideas. Some of these objects are referenced by arrows with each other. All of these objects can be dragged around. Additional Ideas, Pictures and Collages can be pulled into the working space from the left.

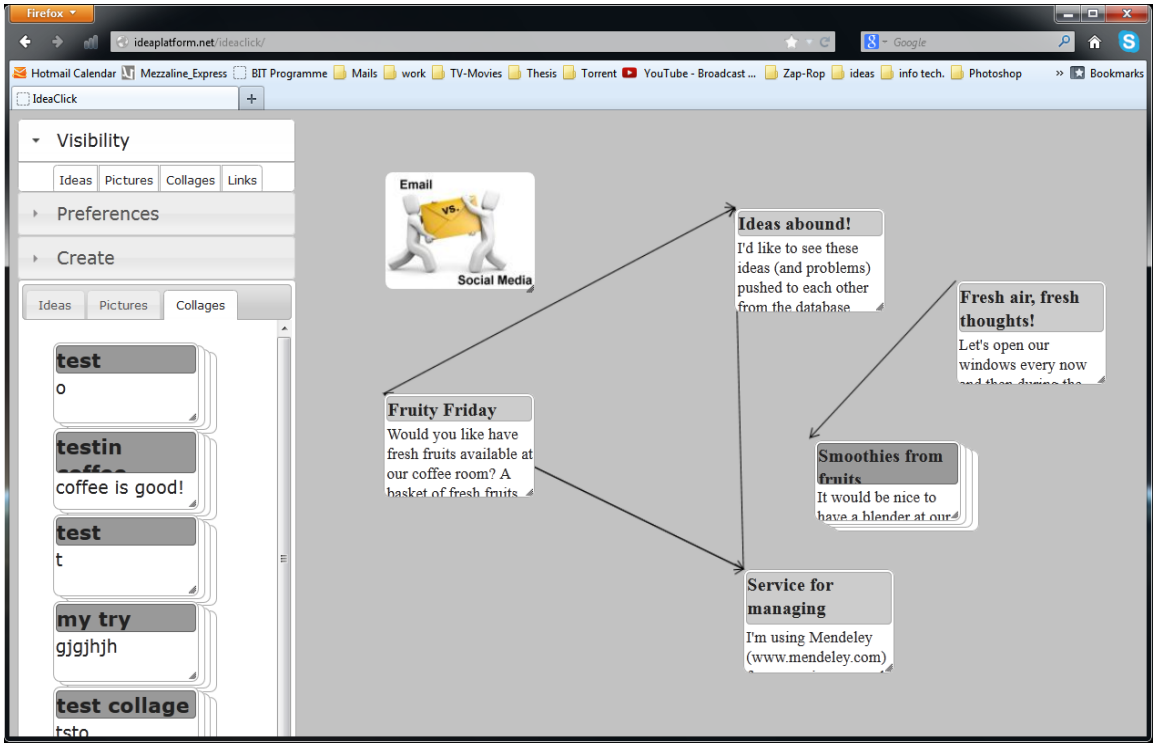


Figure 2 IdeaClick interface

IdeaClick is an online service based virtual environment. The objective of IdeaClick prototype is to promote employees participation in R&D activities for the company through IdeaClick services.

This provides a general description of the IdeaClick Menus and how they can be used. IdeaClick works as follows: every employee creates an IdeaClick account. After the account is created, the users (employees) access IdeaClick by logging in the application with the following URL: http://ideaplatform.net/ideaClick. Once the user is online, he is able to access all IdeaClick services.

The IdeaClick menus allow the following functionality:

- Create button: Creates new objects to be share, displays a sub-menu
    - Collage: to create Collages of Idea and Picture objects
    - Idea: creates a new idea object
    - Picture: creates a new picture idea object
- Visibility button: select objects to be display on the screen
    - Idea: display only idea objects

        - Picture: display only pictures
        - Collages: display only collages
        - Links: display relationships
    - Preferences button:
        - Save positions: it saves the object's position on the screen
        - Logout: to log out the application
        - The Visibility button is extremely useful when having too many objects on the working space, giving the ability to temporarily display only certain object types, such as ideas.

The Visibility button is extremely useful when having too many objects on the working space, giving the ability to temporarily display only certain object types, such as ideas.

Next figure is a first look of IdeaClick user interface, and it shows the different types of objects that the applications offer as part of its web services.

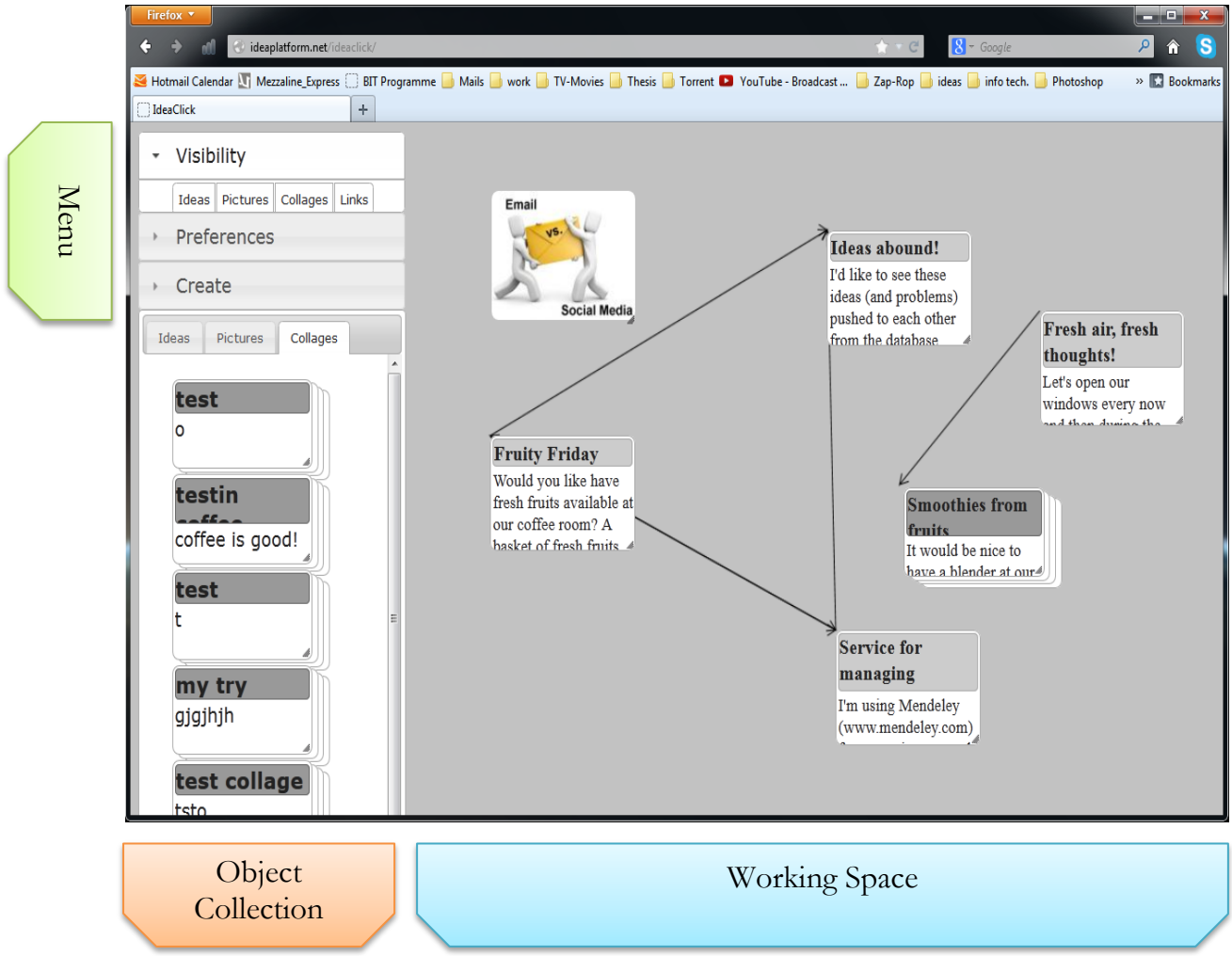


Figure 3 IdeaClick First Look

IdeaClick Visibility objects Menu allows the user to choose, what objects type will be display at the working space. By the default, all objects previously displayed and used during the last login session by the user will be shown on the Working space.

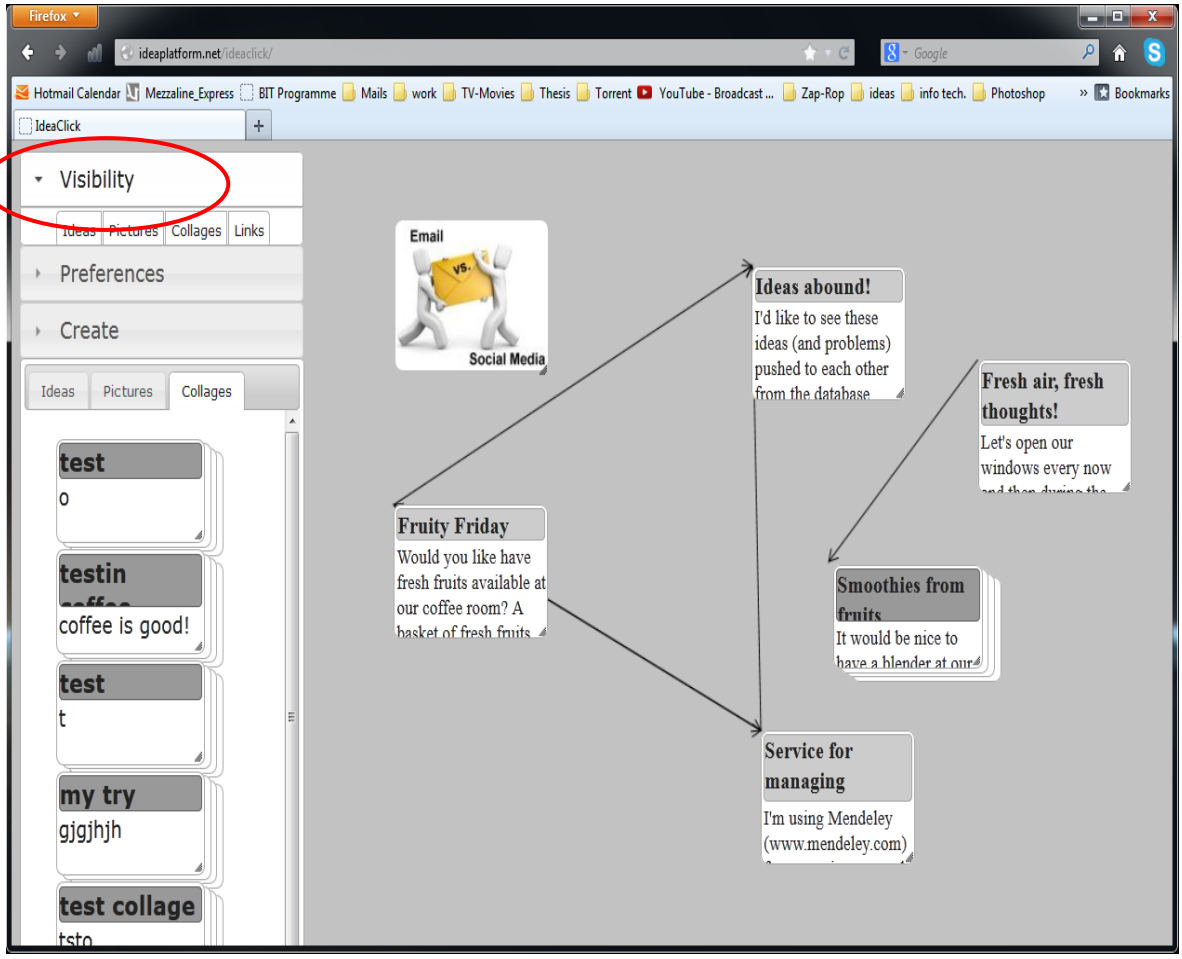


Figure 4 IdeaClick Visibility Objects

Save Position: this allows the user to save the positions of the objects on the screen manually. The positions are saved automatically every 30 seconds.

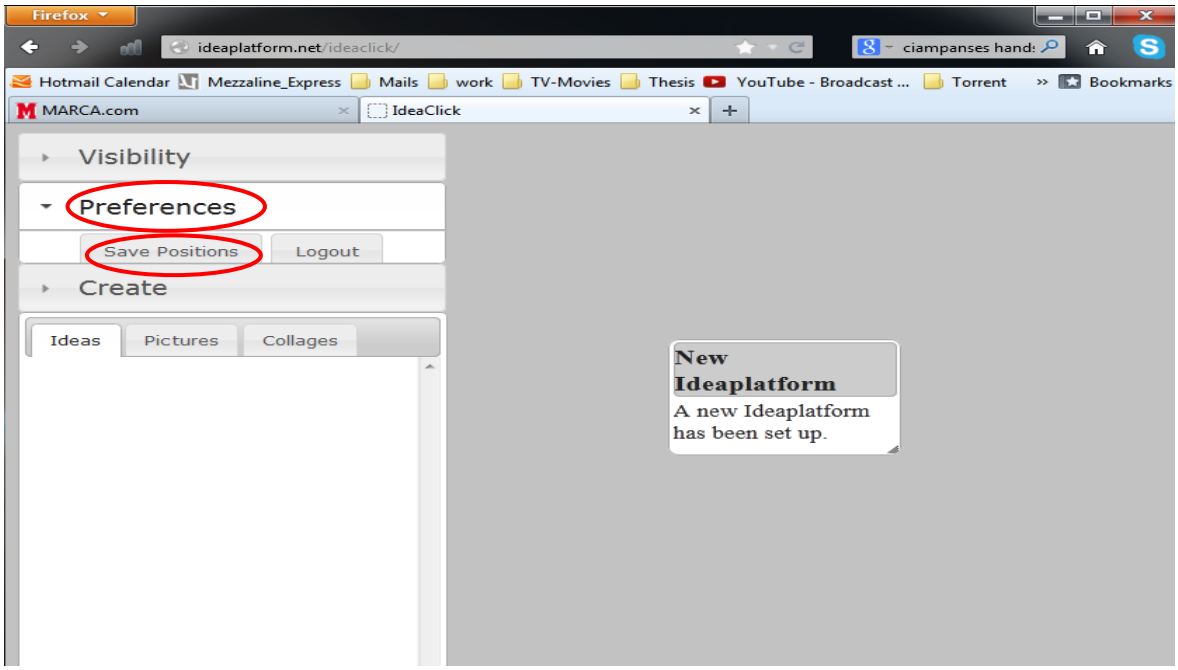


Figure 5 IdeaClick Preferences

Logout: this option allows the user to logout from the session in use

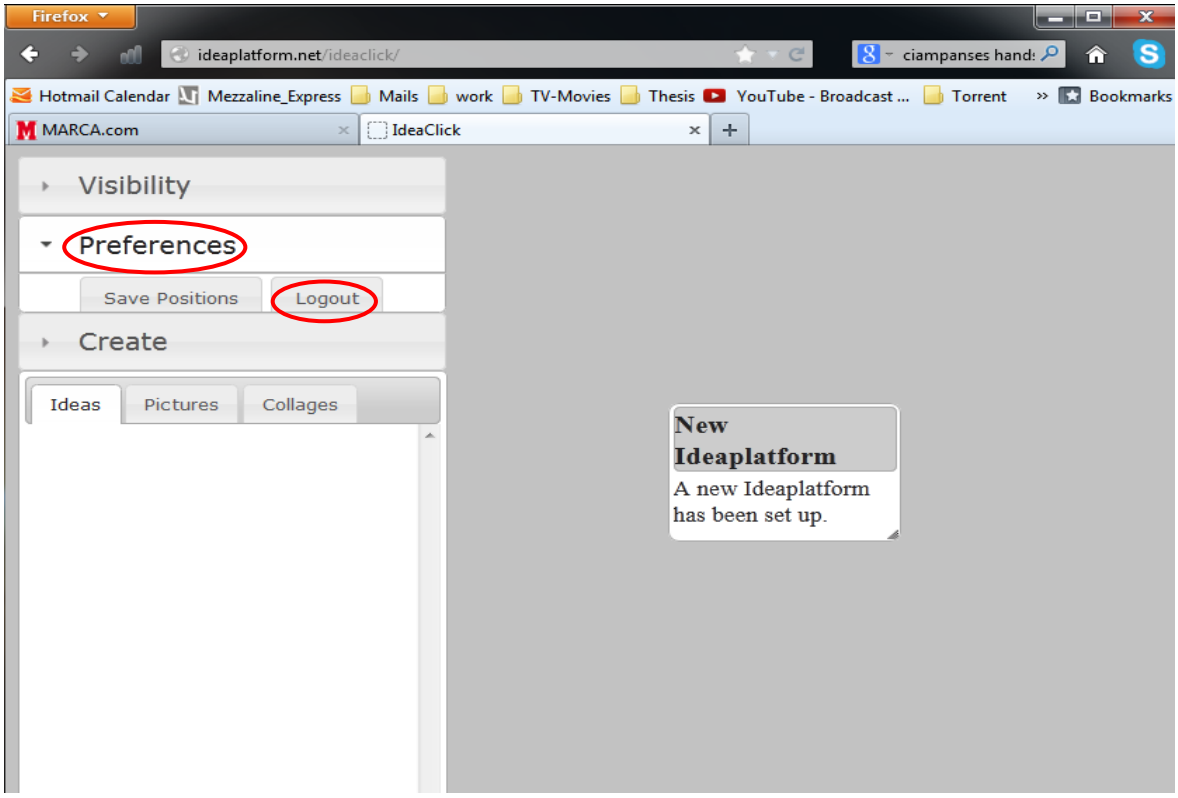


Figure 6 Preferences Logout

Idea and image object is created using the same steps described below. User creates new object by clicking on:

1. Create button
2. Click on object's type: Idea
3. Form appears on the working space
4. Define title and description
5. Click on Submit
6. Object automatically appears on the working space

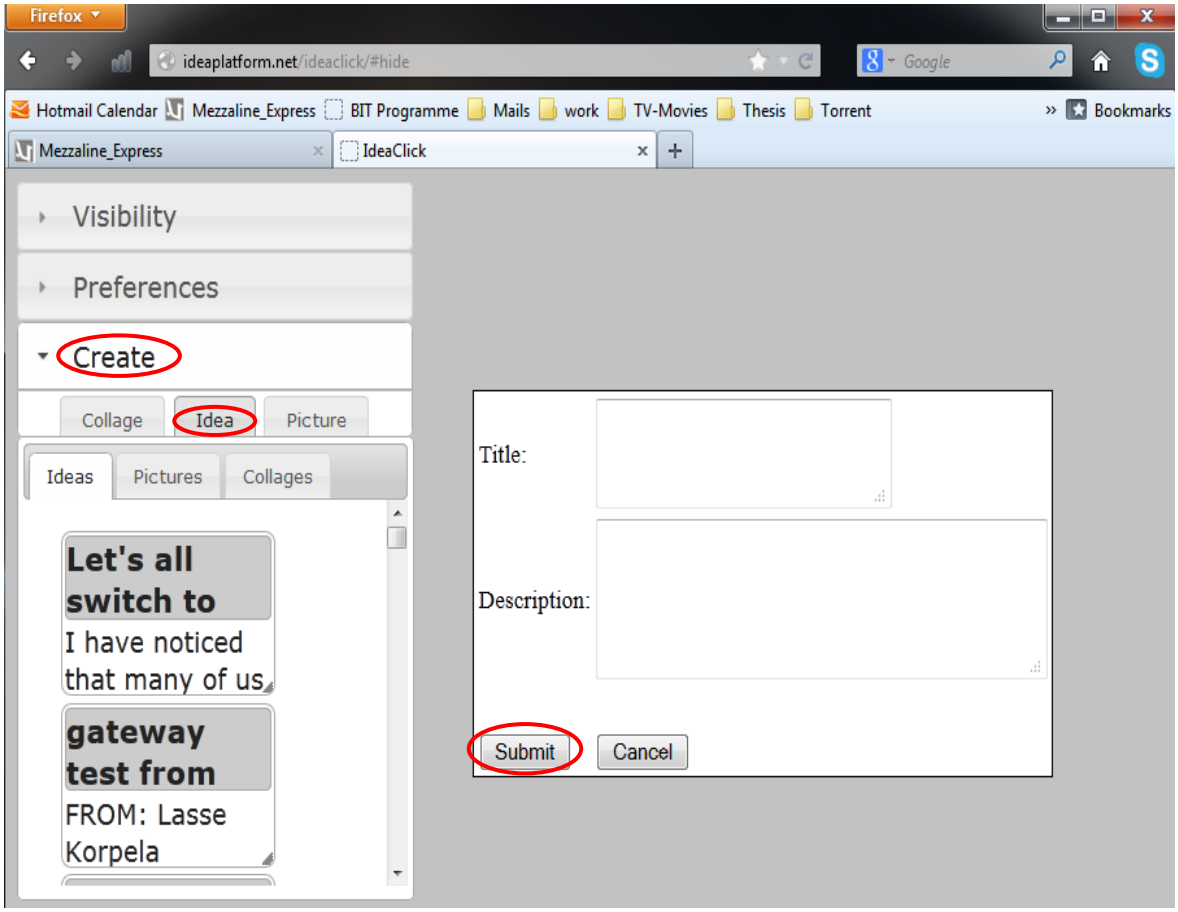


Figure 7 Create a New Object Idea

Idea and image object is created using the same steps described below. User creates new object by clicking on:

1. Create button
2. Click on object's type: Picture
3. Form appears on the working space
4. Define title and description
5. Click on Browse and choose the picture to be uploaded
6. Click on Submit
7. Object automatically appears on the working space

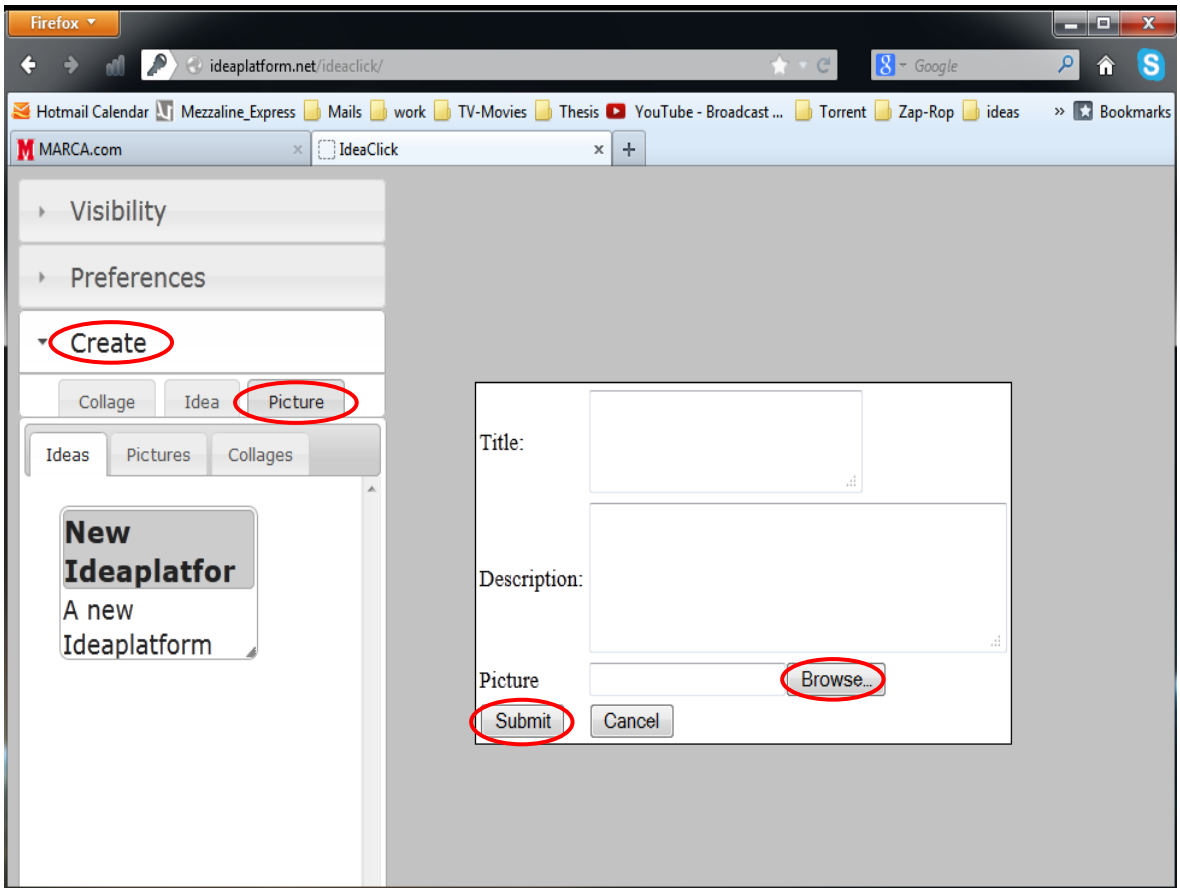


Figure 8 Create a New Object Picture

Collage is another type of object in IdeaClick. A collage is a group of objects already created, which can consist of ideas, pictures and other collages.

User clicks on:

1. Create button
2. Click on object's type: Collage
3. Draw circle around objects to be grouped
4. A menu appears on the top of the working space
5. Press Create Collage
6. Form appears on the working space
7. Define title and description
8. Object automatically appears on the working space

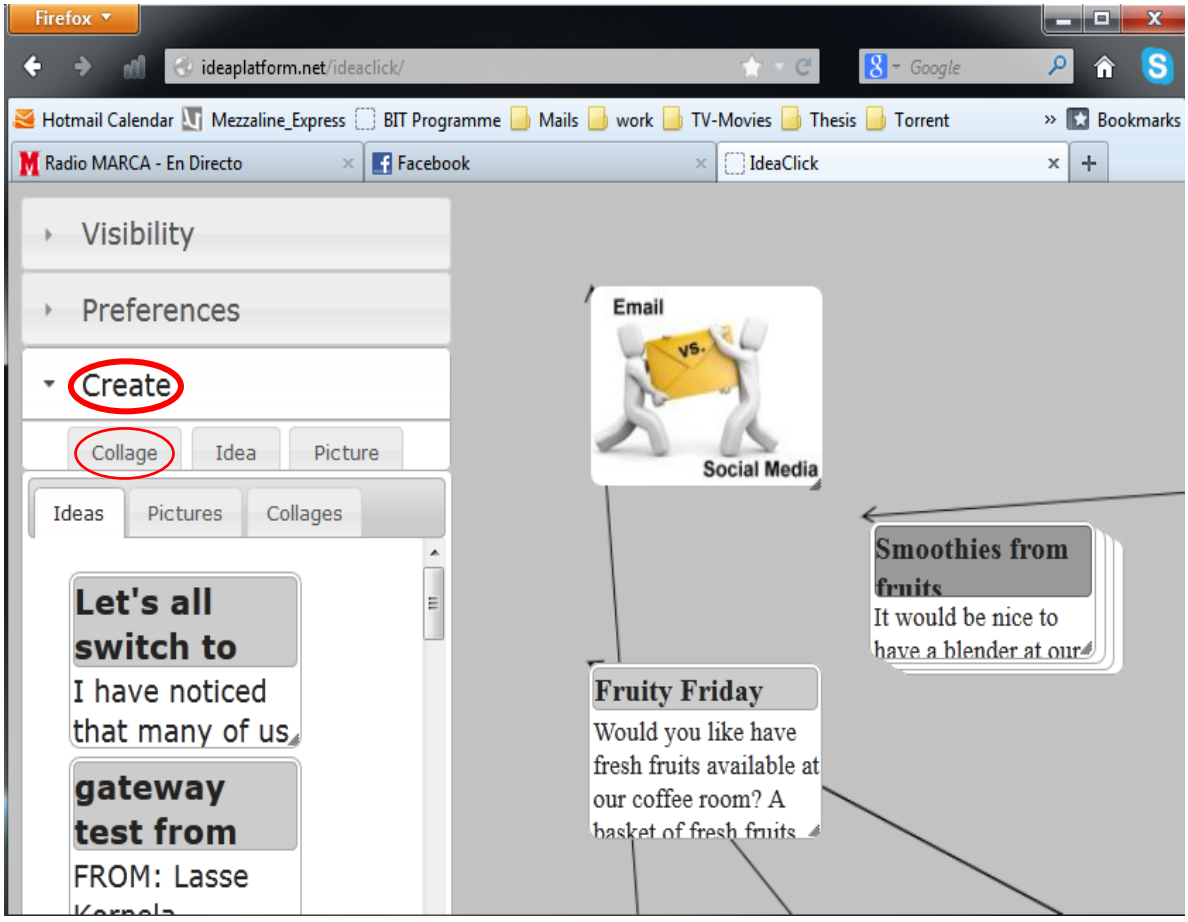


Figure 9 Create a new collage

Select object individually or by circling a bunch of them as shown in the image below.

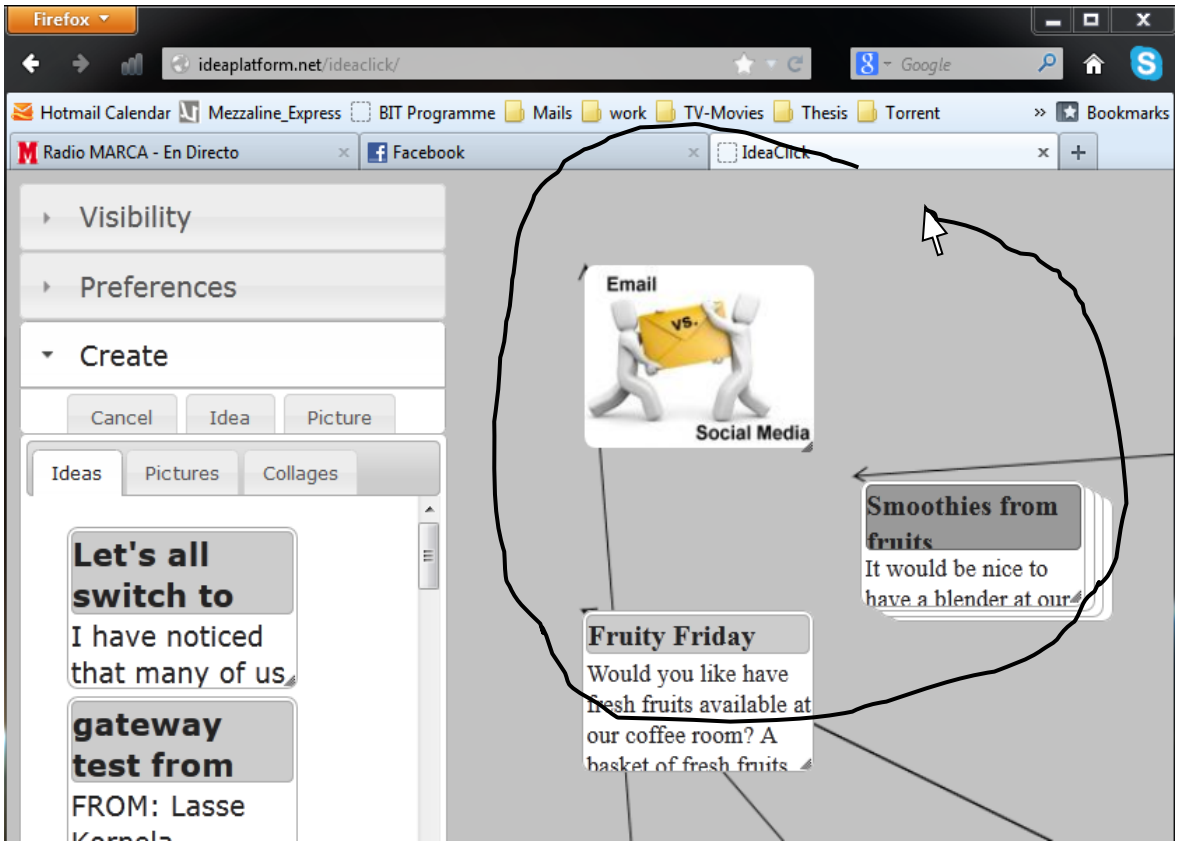


Figure 10 Encircling objects to become a collage

Select the "Create Collage" button to finalize the selection.

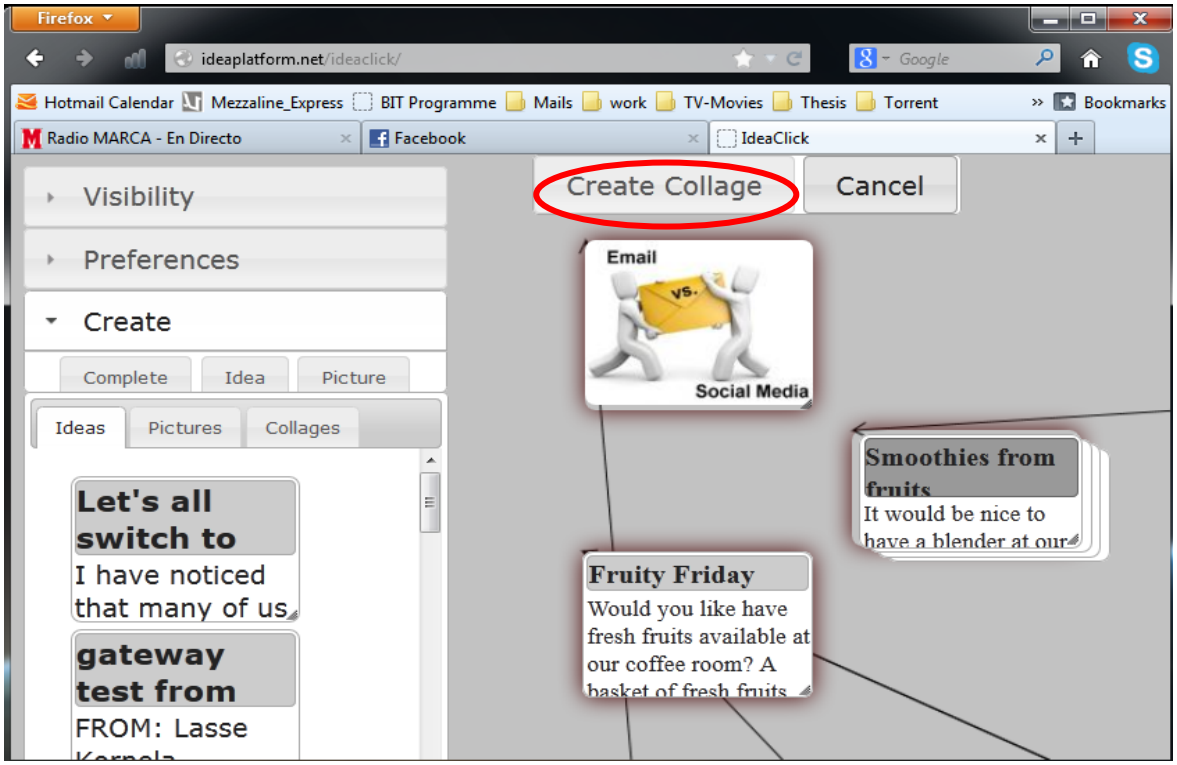


Figure 11 Create collage

Every object on the working space has a right-click menu. This menu gives the user the functionality of linking objects, removing links and hiding the object by sending it to the collection space. If the user has created this object he is also entitled to remove the object from the interface, making it invisible to all users. The menu also allows the user to "Go to" the object, sending him to the idealist prototype which is designed for collaboration with other users.

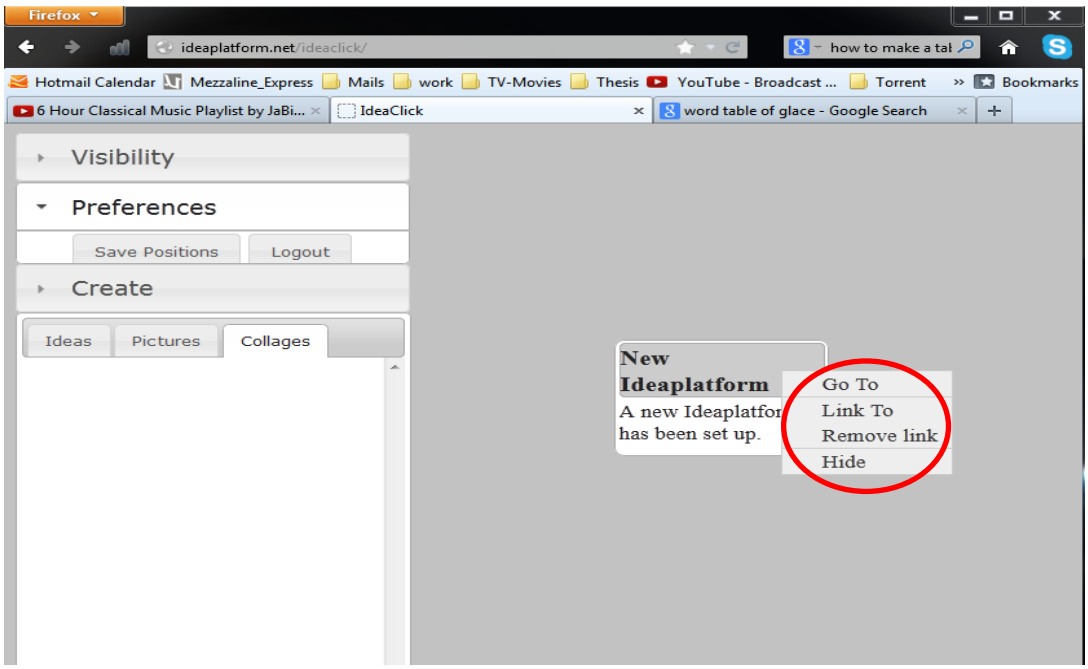


Figure 12 Object right-click options

Right-click an object on the working space to see additional options, such as linking and removing:

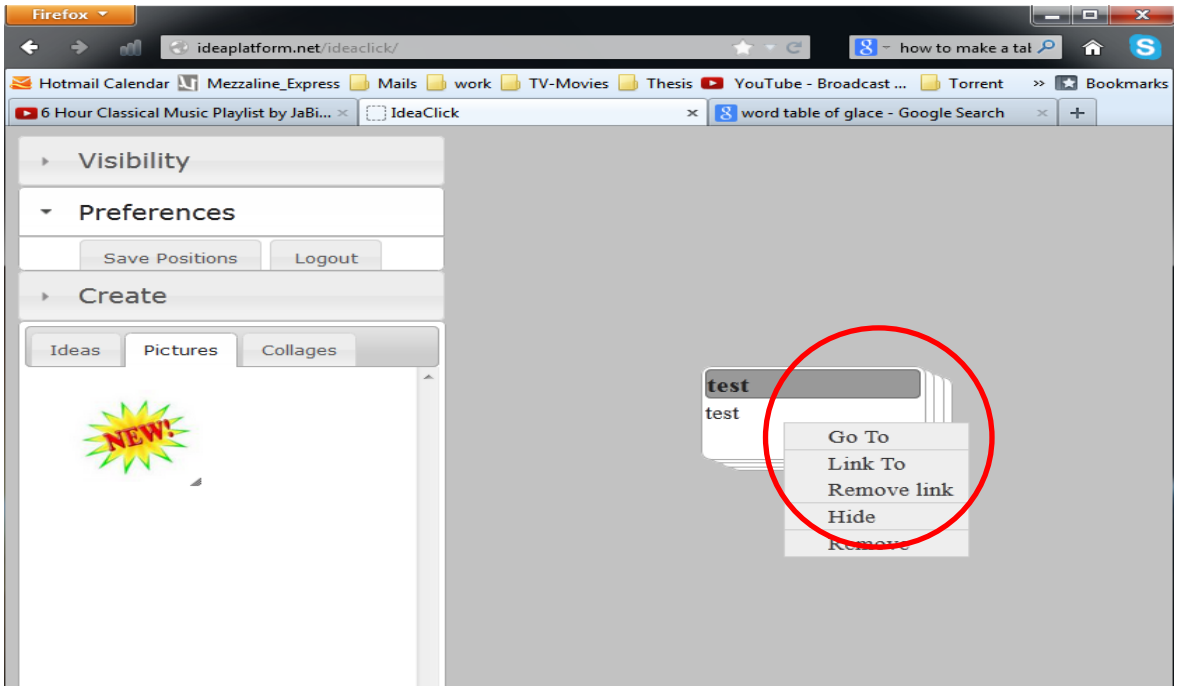


Figure 13 Object right-click options when owning the object

Objects that are created are shared to all users. The user that has created the object will immediately have it in his working space. All other users can find this object in their collection space.

2.1.2 IdeaPlatform

The IdeaClick prototype is only a small part of the entire platform. This prototype may have evolved to be the most interesting and have caused most discussions between researchers, but it just a single interface behind a powerful core, extending to many other prototypes.

The idea behind the platform was to develop a highly modular core, which allowed for easy extensions to be added to the platform without having to modify the core structure too much, if at all. To achieve this, the Database and Class architecture were designed to dynamically expand through the creation of a single InnoObject parent, derived from Innovation Object. This InnoObject Parent could store any data, from user friendly pictures and text, to raw data which could be read and translated by a prototype.

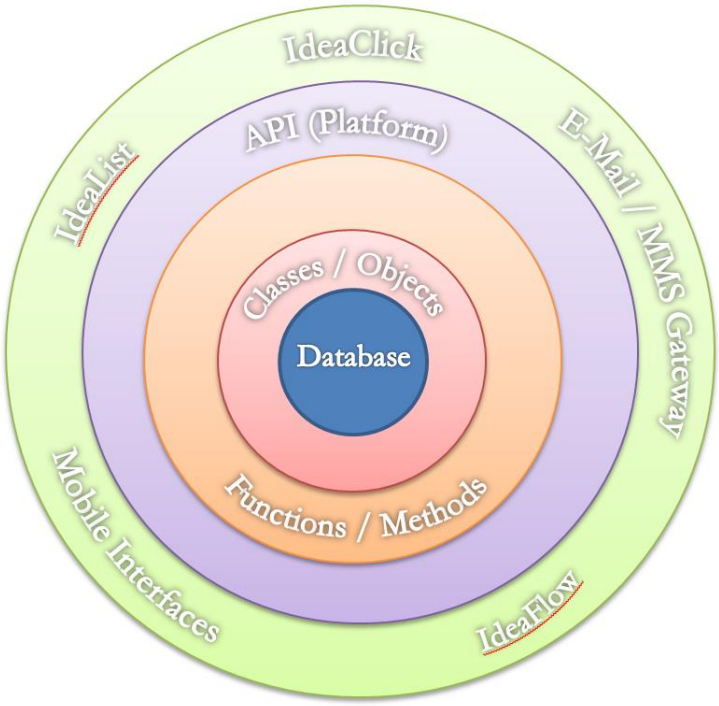


Figure 14 IdeaPlatform layer architecture

Represents the platform in a layer architecture from its central storage to its prototypes.

2.2 Software Testing

This chapter gives the essential theory and definitions related to software testing, software testing rules and a brief history of software testing. In addition, there is a brief explanation of the most uses types of software testing.

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. (Wikipedia. 2013.)

According to definition given by The Institute of Electrical and Electronics Engineers, USA (IEEE) - Software testing is the process of analyzing a software item to detect the differences between existing and required conditions (that is, bugs) and to evaluate the features of the software item.

According to the definition given by Dave Gelperin and William C. Hetzel - Software testing can be stated as the process of validating and verifying that a software program/application/product:

- meets the requirements that guided its design and development
- Works as expected
- Can be implemented with the same characteristics
- Satisfies the needs of stakeholders (Sofwareqatestings Sqat. 2013.)

Debugging is a methodical process of finding and reducing the number of bugs, or defects, in a computer program or a piece of electronic hardware, thus making it behave as expected. Debugging tends to be harder when various subsystems are tightly coupled, as changes in one may cause bugs to emerge in another. (Wikipedia. 2013.)

2.2.1 Brief history of Software testing

The origins of software testing can actually be traced back to the fifties, when the primary method of testing anything was debugging. In the late seventies, the approach evolved to one of destruction; basically, the testers would break down the code to find holes or gaps in it. This method was effective but it was not until the advent of prevention oriented methodologies that we began to enjoy the benefits of more robust software applications. In 1979, Glenford J. Myers correctly hypothesized that there must be a distinction between debugging, which means identifying and eliminating bugs in the software code, and actually testing the software in real world settings. It was during this time that there was a distinct shift toward software testing as we know it today. (SlideShare. 2011.)

The sewparation of debugging from testing was initially introduced by Glen ford J. Myers in 1979. Although his attention was on breakage testing ("a successful test is one that finds a bug" it illustrated the desire of the software engineering community to separate fundamental development activities, such as debugging, from that of verification. Dr. Dave Gelperin and Dr. William C. Hetzel classified in 1988 the phases and goals in software testing in the following stages:
Until 1956 – Debugging oriented

1957-1978 – Demonstration oriented

1979-1982 – Destruction oriented

1983-1987 – Evaluation oriented

1988-2000 – Prevention oriented (Prabhakaran, R. 2013.)

2.2.2 Software Testing Purpose and Rules

A primary purpose of testing is to detect software failures so that defects may be discovered and corrected. Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions. The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to and do what it needs to. In the current culture of software development, a testing organization may be separate from the development team. There are various roles for testing team members. Information derived from software testing may be used to correct the process by which software is developed. (Wikipedia. 2013.)

In software testing development is recommends to follow a series basic testing rules, which allows testing responsible to have strong bases that will follow more efficiencies testing plans. QA and Testing Tutorial (2011) point outs the next rules:

- Perform the software Test early and test the software often.
- Integrate the application development and testing life cycles during software testing.
- Formalize a Software testing methodology; this will help test everything the same way and help with uniform results.
- Develop a comprehensive Software Test plan. It forms the basis for the Software Testing methodology.
- Use both static and dynamic testing during the software testing phase.
- Define the expected results early during software testing.
- Understand the business reason behind the application or software on which you are testing. You'll write a better test cases or scripts.

- Use multiple levels and types of testing (regression, systems, integration, stress and load) during the entire software testing cycle.
- Review and inspect the work.
- Don't let your developers check their own work during software testing. They will miss their own defects.

2.2.3 Types of testing

In the IT field there are more than one hundred different types of services for testing products in their release stage. Main testing types (SeaNergyPro) are:

- Functional Testing
- Non-Functional Testing
- Automated Testing - for both functional and non-functional testing

Functional testing ensures that the business requirements are met and validate that the system functions as intended. These types of tests use the functional specifications provided by the client to ensure its requirements are met. Functional testing has only two outcomes, whether the results are met or not. Here are some of the functional tests:

- Unit Testing
- Smoke testing / Sanity testing
- Integration Testing (Top Down, Bottom up Testing)
- Interface & Usability Testing (including Independent Focus Groups)
- System Testing
- Regression Testing
- Pre User Acceptance Testing (Alpha & Beta)
- User Acceptance Testing
- White Box Testing, Black Box Testing
- Globalization and Localization Testing

- Web site functional testing & Testing for completeness and consistency of web pages. (SeaNergyPro.)
- Ad-hoc Testing
- Negative Testing

Non-Functional testing on the other hand focuses on the quality of the product rather than its functionality. These tests have a large impact on the satisfaction of users, since they focus further on the reliability and user satisfaction. Here are some examples (SeaNergyPro) of non-functional tests:

- Load and Performance Testing
- Ergonomics Testing
- Stress & Volume Testing
- Compatibility & Migration Testing
- Data Conversion Testing
- Security / Penetration Testing
- Operational Readiness Testing
- Installation Testing
- Security Testing
- Usability Testing
- Exploratory Testing

Automated testing is as the title describes a way to automate the manual tests performed by testers. Automating tests reduces the execution time when having to retest several times by only having to define the tests once. The following extract by SeaNergyPro provides additional information on automated testing:

> Automated Testing is automating the manual testing process currently in use. This requires that a formalized "manual testing process" exist in the company or organization. Minimally, such a process includes: Detailed test cases, including predictable "expected results", which have been developed from Software Requirements Specifications and Design documentation. A standalone Test Environment, including a Test Database that is restor-

able to a known constant, such that the test cases are able to be repeated each time there are modifications made to the application. (SeaNergyPro.)

Automated testing can be used for any testing type, such as functional and non-functional testing, irrelevant of its purpose. This only provides the means to automate the testing process, allowing easy retesting of the same test.

Below are the definitions of selected functional and non-functional testing types:

Functional Testing	
Functional Testing	Functionality testing is performed to verify whether the product/application meets the intended specifications and the functional requirements mentioned in the documentation. Functional tests are written from a user's perspective. These tests confirm that the system does what the users are expecting it to do. Both positive and negative test cases are performed to verify the product/application responds correctly. Functional Testing is critically important for the products success since it is the customer's first opportunity to be disappointed.
Unit Testing	Testing of individual software components or groups of related components Testing conducted to evaluate whether systems or components pass data and control correctly to one another
Integration Testing	Integration testing is the activity of software testing in which individual software modules are combined and tested as a group. Testing in which software components or hardware components or both are combined and tested to evaluate the interaction between them.
System Testing	System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's

	compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. System testing is performed on the entire system with reference of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS).
Regression Testing	When a defect is found in verification and it is fixed we need to verify that 1) the fix was done correctly 2) to verify that the fix doesn't break anything else. This is called regression testing. Regression testing needs to be performed to ensure that the reported errors are indeed fixed. Testing also needs to be performed to ensure that the fixes made to the application do not cause new errors to occur. Selective testing of a system or component to verify that modifications have not caused unintended effects.
Alpha Testing	Testing performed by actual customers at the developer's site.
Beta Testing	Testing performed by actual customers at their site (customer's site).
Acceptance Testing	Formal testing conducted to enable a user, customer or other authorized entity to determine whether to accept a system or component.
White-Box Testing	White-box testing focuses on the internal structure of a system, this usually requires some programming skills to evaluate the code behind. The most common pieces of code tested here are the loops and if-statements.
Black-Box Testing	Black-box testing focuses on the externals of a system, this doesn't require basically any programming skills or knowledge of the internal structure of the system. Black-box testing is also called behavioral testing, focusing on the behavior of the system from a user's point of view.

Ad-hoc Testing	Adhoc testing is a commonly used term for software testing performed without planning and documentation. The tests are intended to be run only once, unless a defect is discovered.
Negative Testing	Negative Testing is testing the application beyond and below of its limits. For ex: If the requirement is to check for a name (Characters), 1) We can try to check with numbers. 2) We can enter some ascii characters. 3) First we can enter some numbers and then some characters. 4) If the name should have some minimum length, we can check beyond that length.

Figure 15 Functional testing table

Non-Functional Testing	
Performance Testing	Performance test is testing the product/application with respect to various time critical functionalities. It is related to benchmarking of these functionalities with respect to time. This is performed under a considerable production sized set-up. Performance Tests are tests that determine end to end timing (benchmarking) of various time critical business processes and transactions, while the system is under low load, but with a production sized database. This sets 'best possible' performance expectation under a given configuration of infrastructure. Some examples of the Performance parameters (in a Patient monitoring system - Healthcare product) are, 1. Real-time parameter numeric values match the physiological

	inputs 2. Physiological input changes cause parameter numeric and/or waveform modifications on the display within xx seconds. 3. The system shall transmit the numeric values frequently enough to attain an update rate of x seconds or shorter at a viewing device.
Stress Testing	1. Stress Tests determine the load under which a system fails, and how it fails. 2. Testing conducted to evaluate a system or component at or beyond the limits of its specified requirements to determine the load under which it fails and how. 3. A graceful degradation under load leading to non-catastrophic failure is the desired result. Often Stress Testing is performed using the same process as Performance Testing but employing a very high level of simulated load. Some examples of the Stress parameters (in a Patient monitoring system - Healthcare product) are, 1. Patient admitted for 72 Hours, and all 72 hours of data available for all the parameters (Trends). 2. Repeated Admit / Discharge (Patient Connection and Disconnection) 3. Continuous printing 4. Continuous Alarming condition
Compatibility Testing	Compatibility testing is done to check that the system/application is compatible with the working environment. For example if it is a web based application then the browser compatibility is tested. If it is an installable application/product then the Operating system compatibility is tested.

	Compatibility testing verifies that your product functions correctly on a wide variety of hardware, software, and network configurations. Tests are run on a matrix of platform hardware configurations including High End, Core Market, and Low End.
Security Testing	Security testing focuses on the safety of the system, ensuring that the functionality remains as intended and prohibiting the bypass of authentication procedures.
Usability Testing	Usability testing ensures the users satisfaction, providing an easy to use and intuitive application by making the use of the product as easy as possible.
Exploratory Testing	Exploratory testing is a method of manual testing that is described as simultaneous learning, design and execution.

Figure 16 Non-Functional testing table

The graphics described bellowed is a representation of the Software testing services divided by areas, and services' options in each of those areas

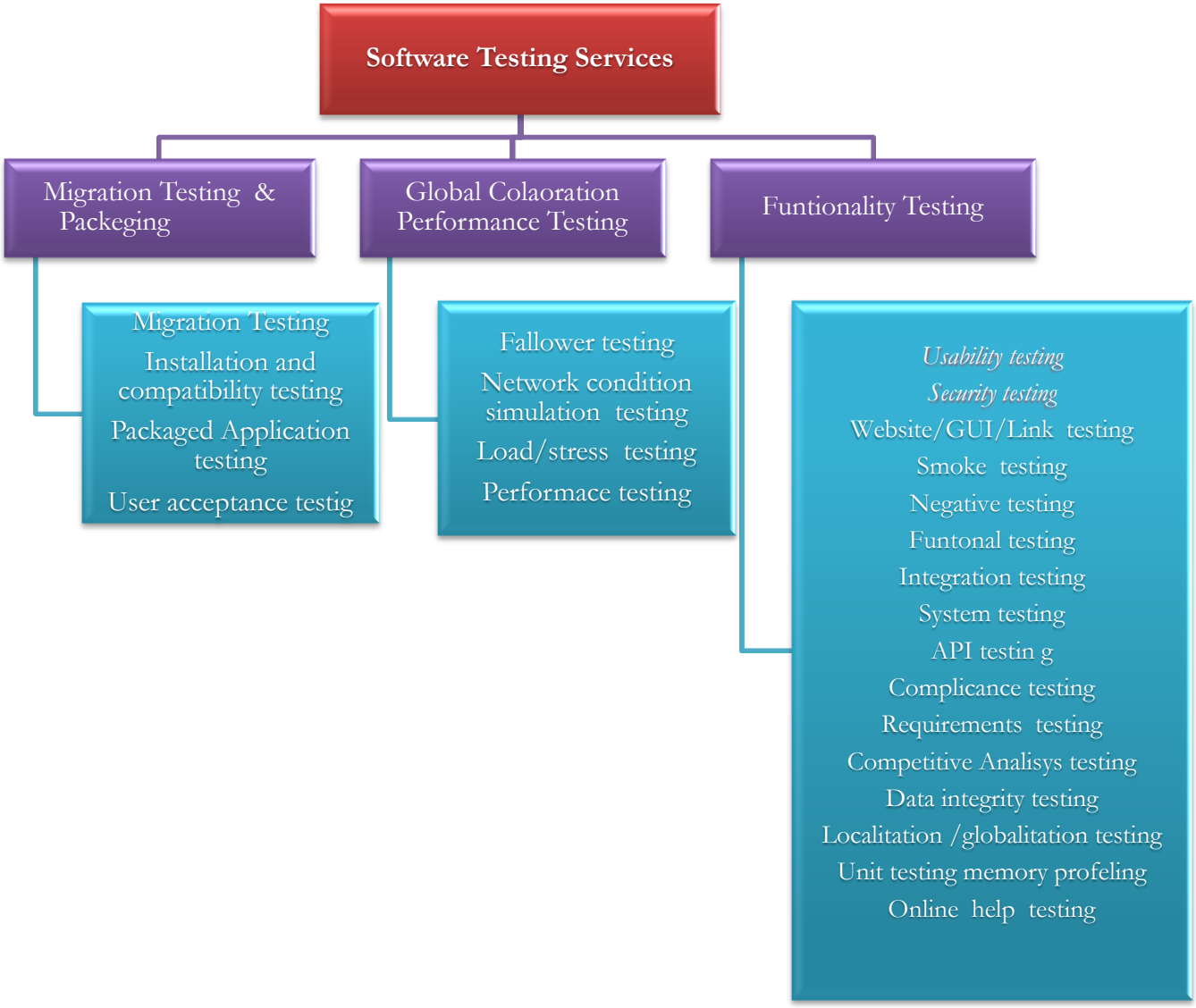


Figure 17 Hierarchical testing types services and its areas

3 The Two Dimensions of Testing

3.1 Usability

This chapter gives a brief introduction of what usability testing is about, essential definitions related to usability testing, and a brief history of usability testing. In addition, it is mentioned the importance of good UI, usability evaluations and its relevance. Finally, there is a short summary of important elements to keep in mind related to usability.

"Usability testing is a technique used to evaluate a product by testing it with representative users. In the test, these users will try to complete typical tasks while observers watch, listen and takes notes." (Usability.gov2. 2013.)

"Your goal is to identify any usability problems, collect quantitative data on participants' performance (e.g., time on task, error rates), and determine participant's satisfaction with the product." (Usability.gov2. 2013.)

"Ergonomics is an essential term related to usability. Ergonomics basically looks for designing everyday things that are used by people, so they are functional and easy to use. Therefore, Ergonomics' primary target is to achieve usability" (Patton, R. 2006, 169.)

Usability testing is often used rather indiscriminately to refer to any technique used to evaluate a product or system. In addition, o refer to a process that employs people as testing participants who are representative of the target audience to evaluate the degree to which a product meets specific usability criteria. Criteria generally tend to change, not drastically at all, based on the type of service to be fulfilled. (Rubin, J. & Chisnell, D. 2008, 21)

According to Carol M. Barnum in her book Usability Test Essential ready, set, test, she explains: "when I refer to usability testing, I mean the activity that focuses on the ob-

serving users with a product, performing tasks that are real and meaningful to them." (Barnum, C. 2011, 11.)

In order to understand what makes a product Usable, it is imperative to know what Usability means. Jeffefrey Rubin and Dana Chisnell in their book Hand Book of Usability testing describes it as: when a product of service is truly usable, the user can do what He or She wants to do the way He or She expects to be able to do it, without hindrance, hesitation, or questions. Rubin, J. & Chisnell, D. (2008, 4) in addition, it is good to remember that the concept of Usability include a series of attribute. A product or a service in order to usable should be as well efficient, effective, satisfying learnable, accessible and useful.

Carol M. Barnum in her book Usability Test Essential ready, set, test; uses the best known definition of usability, the one that is defined by the ISO, The international Organization for standardization (9241-11): "The extent to which a product can be used by specified user to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use." (Barnum, C. 2011, 11.)

"You should test early and test often. Usability testing lets the design and development teams identify problems before they get coded (e.g. set in concrete). The earlier those problems are found and fixed, the less expensive the fixes are." (Usability.gov2. 2013.)

You DO NOT need a formal usability lab to do testing. You can do effective usability testing in any of these settings:

- a fixed laboratory having two or three connected rooms outfitted with audio-visual equipment
- a conference room, or the user's home or work space, with portable recording equipment
- a conference room, or the user's home or work space, with no recording equipment, as long as someone is observing the user and taking notes
- remotely, with the user in a different location (Usability.gov2. 2013)

3.1.1 Brief history of usability testing

"Those who don't know history are destined to repeat it" Edmund Burke. It is relevant to know where Usability practice came from and how it is practice today.

Traditional usability testing relies on the practice of experimental design. Usability testing was integrated as a formal process during the 1990's as a formal method of experimental design. People in change of conducting the tests, had typically knowledge in cognitive scientist, experimental psychologist or human factors engineers. The general question always was if it was affordable, so not much usability testing was done.

Nielsen and Landauer determined that the maximum cost-benefit ratio, derived by weighing the cost of testing and the benefits gained, is achieved when you test with three to five participants. (Barnum, C. 2011, 15)

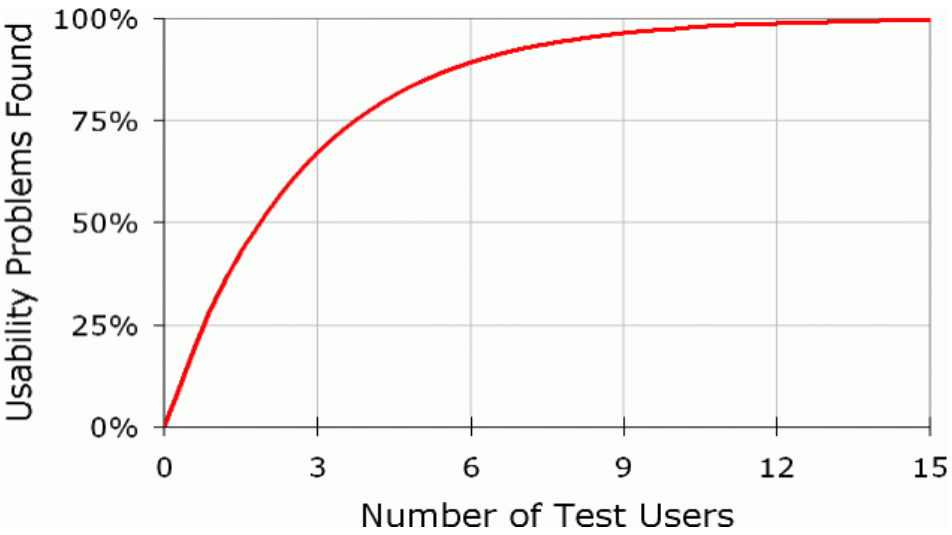


Figure 18 Nielsen testing curve. (Nielsen Norman Group, 2009)

Nielsen says, the most striking truth of the curve is that zero users give zero insight. As soon as you collect data from a single test user, your insights shoot up and you have already learned almost a third of all there is to know about the usability of the design. According to Nielsen, you should stop after the fifth user because you are seeing the same things repeated, and you will have reached the optimal return 85% of the findings to be uncovered (Barnum, C. 2011, 15.)

3.1.2 What makes a good User interface UI

Many companies put a lot of effort and money in order to make the best out of their User interface design. The companies make use of special Labs to run up tests under very controlled environments. Video cameras, that record how users proceed with the test, and how they use the application. Everything that users do during the testing phase can be analysed, such as pressing a button and moving the mouse. It is also important to analyse mistakes and especially those that confuse them, so that corrections and improvements can be done to the UI. (Barnum, C. 2011, 170-171.)
Patton, R. (2006, 173-178) in his book Software Testing second edition mention 7 important traits that commonly can be describe a good designed UI.

- Follow Standards and guidelines: if the application is running on existing platform (Windows, Ubuntu, Mac) there are standards already set.
- Intuitive: Functions required and responds should be obvious and be there when expect them. It is excessive functionality? Is what to do next obvious?
- Flexible: User's possibility to choose, if there is the possibility to do so, this is concerned of course to usable matters. e.g windows calculator allows the user to choose between to views, standard and scientific view.
- Consistent: User after using an application tends to follow patrons and expect that other programs will work in the same way. A button search should locate at the same place in two different applications running over the same platform. Button search in notepad is located in different place compare to the one in WordPad. Consistence, among others, includes terminologies and naming, shortcuts keys and menu selections, placement of buttons such as OK and Cancel.
- Comfortable: The software should be comfortable to use. It means, that the user should feel comfortable, this is more related to his inner feelings, feeling confident going through the different program's menus.
- Correct: tests the UI what it's supposed to do
- Usable: User interfaces whether it is useful, not that the software itself is useful.

This observes that if any particular future in the software is usable.

3.1.3 Website Usability Importance

The main reason that usability is so important is because there are so many similar websites, and people will go to the next website if the first one they visit is not usable. You can have the most beautiful website in the world, but people will leave immediately if they are unable to figure out how to navigate your site quickly.

> As stated in the article Why Web Site Usability is Important for a Company: a company's web site is the only point of contact that a company has with anyone who is interested in it. Thus, companies entirely rely on their web presence in order to achieve their online goals. Similarly, a user of a company's web site will formulate a judgment about that company that is strongly correlated with the way they perceive its web site. Furthermore, usable websites increase user satisfaction whereas web sites which violate usability conventions confuse users and result in a loss of revenue for the companies behind them. This is because improving usability is a great way to encourage users to visit your site instead of the sites that belong to your competitors and is often an approach that keeps customers coming back to your site again and again. Indeed, high-quality websites that are easy to use bring in customers and give a particular site a competitive edge over the competition.(Usabilitygeeg. 2013.)

3.1.4 Types of Usability Testing Methods

The following is a brief description of the main usability testing methods that are used. The descriptions below were written by Thomas Churm in his online article: An introduction to website Usability Testing (2012).

- **Hallway Testing:** Using random people to test the website rather than people who are trained and experienced in testing websites. This method is particularly effective for testing a new website for the first time during development.
- **Remote Usability Testing:** Testing the usability of a website using people who are located in several countries and time zones. Sometimes remote testing is performed using video conferencing, while other times the user works separately from the evaluator. Nowadays, there are various software available at a relatively low cost that allow remote usability test-

ing to be carried out even by observers who are not usability experts. Typically, the click locations and streams of the users are automatically recorded and any critical incidents that occurred while they were using the site are also recorded, along with any feedback the user has submitted. Remote usability testing allows for the length of time it took each tester to complete various tasks to be recorded. It is a good method of testing because the tests are carried out in the normal environment of the user instead of a controlled lab.

- **Expert Review:** An expert in the field is asked to evaluate the usability of the website. Sometimes the expert is brought to a testing facility to test the site, while other times the tests are conducted remotely and automated results are sent back for review. Automated expert tests are typically not as detailed as other types of usability tests, but their advantage is that they can be completed quickly.
- **Paper Prototype Testing:** Quite simply, this usability testing method involves involves creating rough, even hand-sketched, drawings of an interface to use as prototypes, or models, of a design. Observing a user undertaking a task using such prototypes enables the testing of design ideas at an extremely low cost and before any coding has been done. For additional details about paper prototype testing, please read the article
- **Questionnaires and Interviews:** Due to their one-on-one nature, interviews enable the observer to ask direct questions to the users (apart from double checking what they are really doing). Similarly, the observer can also ask questions by means of questionnaires. The advantage of questionnaires is that they allow more structured data collection. However, they are rigid in nature as opposed to interviews.
- **Do-it-Yourself Walkthrough:** Just as the name implies, in this technique, the observer sets up a usability test situation by creating realistic scenarios. He or she then walks through the work themselves just like a user would. A variation of this technique is the group walkthrough where the observer has multiple attendees performing the walkthrough.
- **Controlled Experiments:** An approach that is similar to scientific exper-

iments typically involving a comparison of two products, with careful statistical balancing in a laboratory. This may be the hardest method to do "in the real world" but due to its scientific nature, it yields very accurate results that can eventually be published

- **Automated Usability Evaluation:** Probably the Holy Grail of usability testing. Various academic papers and prototypes have been developed in order to try and automate website usability testing, all with various degrees of success. One interesting approach has been discussed in this blog is Justin Mifsud's USEFul Framework.

Usability evaluation techniques require a considerable amount of judgment on the part of the evaluators and usually do not include representative users. Evaluation techniques include: surveys/questionnaires, observational evaluations, guideline based reviews, cognitive walkthroughs, expert reviews, and heuristic evaluations. (Usability.gov. 2013.)

You can conduct a usability evaluation as soon as you have a prototype. Many usability professionals first do a usability evaluation and then follow it up with a usability test. They use the results of the evaluation to develop hypotheses about what could be serious problems and then develop the usability test around those hypotheses. (Usability.gov. 2013.)

Probably the most popular evaluation method is referred to as a heuristic evaluation. In general, this is a method for finding usability issues in a user interface by having a small number of evaluators (usually one to five) examine the interface and judge its compliance with usability principles (heuristics). The resulting observations represent the evaluator's opinion about what needs to be improved in a user interface. (Usability.gov. 2013.)

To assess the usability of any product, including Web sites, you can use any or all of several methods. We divide these methods into two major types:

- Usability tests, which focus on users working with the product

- Usability evaluations, which typically do not include users working with the product. (Usability.gov. 2013.)

Usability tests always include test participants; usability evaluations usually do not. Usability testing is the only way to know if the Web site actually has problems that keep people from having a successful and satisfying experience (Usability.gov. 2013.)

Generally, we are not interested in what testers think will be a problem; we want it demonstrated by having one or more users actually struggle with some aspect of the site. A usability test provides an opportunity for the site to allow users to succeed, succeed with difficulty, or totally fail. (Usability.gov. 2013.)

3.1.5 Usability Summary

Barnum, C. 2011 in her book Usability Testing Essentials ready, set … Test!
makes a precise summary of important elements that should be keep in mind , while developing a Usability work based.

A focus on users, not products—it's all about the user's experience, not the product's performance.
Usability, which encompasses:

- The product's effectiveness and efficiency for users, as they work with the product
- The elusive quality of user satisfaction, which is based on users' perceptions entirely

Usability testing, which focuses on observing real users performing real tasks that are meaningful to them, and which can be classified into two types:

- Formative testing, done during product development to diagnose and fix problems

Summative testing, done at the end of product development to confirm that the product meets requirements

The key elements for conducting effective small studies, which include:

- Identifying a specific user profile for the study
- Creating scenarios that are task based and goal directed
- Encouraging users to think out loud as they work
- Testing again to confirm that the changes work for users

The need for bigger studies

- when the test is a summative evaluation and metrics are the goal, or
- when more users are needed to see different user groups, or
- when risk or personal safety is an issue, or
- when management needs bigger numbers to be convinced that
- the results are representative of users

The factors affecting the type of study you conduct based on balancing your goals, management support, your budget, and your time.

3.2 Security

Security is as we all know an essential part of many products. We rely on these products and expect them to work as intended. The following citation by BusinessDictionary (2013) describes the definition of security very well.
"Security is a state in which something is secure or safe. In computing, security is the extent to which a computer system is protected from data corruption, destruction, interception, loss, or unauthorized access."

3.2.1 History and definition of web application Security

At the beginning of the World Wide Web, all websites were public, without any private content to hide, so there was no real threat of stealing confidential information, other than attackers attempting to take a website offline. With no information to hide, there was nothing to break into.

As the World Wide Web started getting more sophisticated with user authentication for private websites, hackers had a reason to start attempting to retrieve this hidden information. The hacking of websites began and developers had to be more and more careful to make sure they had no security vulnerabilities. Unfortunately that was not the case and nowadays up to 94% of web applications have vulnerabilities. (Stuyyard, D. & Pinto, M. 2011, 8.)

Most companies rely on websites to distribute their content to their customers. This content may include advertising of the company, sharing sensitive information to clients around the world or selling products through the World Wide Web. Advertising a company's information may not sound like it may require much security, but when is comes to the latter alternatives, companies need to implement huge amounts of work force to ensure their websites are safe and reliable.

Most risks associated with websites include taking a website down or stealing sensitive information, such as credit card numbers stored in the Databases of online stores.

Many developers spend a great deal of time on implementing features and functionality to their work, but spend only little time on making these secure. This is the main reason why web applications have so many security vulnerabilities.

There are many different ways on how an attacker can gain access to a web application. If a web application has not been thoroughly tested for vulnerabilities, it is not very hard for any rookie to find his way around and steal sensitive information. (AppliCure. 2013.)

The most common ways of breaking into an application are:

- SQL Injection
- XSS (Cross Site Scripting)
- Remote Command Execution

3.2.2 Security Testing Definition

"Security Testing is the process to determine that an information system protects data and maintains functionality as intended" (AllInterview. 2013). This process helps identify flaws in the product, which is essential to providing a secure product.

Security testing is, as described in (SoftwareTestingMentor. 2013), based on 6 basic security concepts: confidentiality, integrity, authentication, authorization, availability and non-repudiation.

- Confidentiality is a security measure which protects against the disclosure of information to parties other than the intended recipient that is by no means the only way of ensuring the security.
- Integrity is a measure intended to allow the receiver to determine that the information which it is providing is correct. Integrity schemes often use some of the same underlying technologies as confidentiality schemes, but they usually involve adding additional information to a communication to form the basis of an algorithmic check rather than the encoding all of the communication.

- Authentication is the process of establishing the identity of the user. Authentication can take many forms including but not limited to: passwords, biometrics, radio frequency identification, etc.
- Authorization is the process of determining that a requester is allowed to receive a service or perform an operation. Access control is an example of authorization.
- Availability assures information and communications services will be ready for use when expected. Information must be kept available to authorized persons when they need it.
- Non-repudiation is a measure intended to prevent the later denial that an action happened, or a communication that took place etc. In communication terms this often involves the interchange of authentication information combined with some form of provable time stamp

The goal of security testing is to identify the flaws of a product and measure its potential functional vulnerabilities.

3.2.3 When to test Web application Security

Every type of testing has its special purpose during the development process of a product. Some of these types of tests are even used several times during the production process. In many cases a product is tested for usability on two occasions, first a mockup of the product is developed, this doesn't consume much time, but allows for an initial insight on how user friendly it is, without having to complete the development and identifying that the product does not meet the expectations that were defined in the beginning. The second time the product is tested is at its prototype stage, where a completed product can give much more understanding on how usable the product really is. Should the product not meet the expectations, the developers usually go back to the first step and draw up another mockup and the process begins from the start.

In the case of security testing it is quite problematic drawing up a mockup in several cases to get an initial insight. In this case the testing is done at the end of its develop-

ment stage. Each type of testing has its unique purpose throughout the development process. To read more about types of testing refer to point 5 of this document.

3.2.4 What makes a Web Application Secure

There are many methods of breaking into a Web Application, both through software and hardware, but most importantly an application is considered secure when it has been thoroughly tested and confirmed that there are no vulnerabilities that may allow an attacker to either retrieve any sensitive information, or place malicious code which attack visitors of the site or render the application inoperable.

3.2.5 Future of Security in Web Applications

The original plan of the WWW was a network through which public content could be shared throughout the entire world. Private companies which would need to secure their information were usually located behind a firewall. During that time a firewall blocked all incoming traffic, securing the premises from any outside attacks. Throughout the years, companies started having their own private Web servers inside the premises, therefore forcing them to open certain areas of their firewall and increasing the chances of attackers gaining access to their sensitive information.

Even though developers and system administrators have slowly become more aware resulting in older attack methods to be less common, new vulnerabilities are discovered all the time. With new technologies such as cloud computing, new vulnerabilities present themselves allowing attackers to continuously finding new ways of breaking into systems. (Stuyyard, D. & Pinto, M. 2011, 50.)

3.3 Results - Testing Plan

Creating and executing testing plans for the IdeaClick prototype and analysing the results are the core of this thesis. In addition, and more important, is to make a clear summary of the results and hand it out to IdeaClick's project management, so they can have more information in order to better assess the future of the prototype.

IdeaClick usability testing plan follows basic structure and procedures studied at the university courses, Usability testing by Seija Wolfer and Software QA and testing by Raine Kauppinen. IdeaClick Security testing plan follows the structure studies at the university courses, Software QA and Testing by Raine Kauppinen and Corporate and IT Security by Markku Somerkivi, as well as the OWASP Foundation principles. IdeaClick's testing plan is described in the IdeaClick's testing plan document. The document covers all essential procedures to describe the most appropriate security and usable testing plan for IdeaClick.

For the Usability testing plan methodology, I have choose to follow a baseline methodology, where 10 users make use of a black-box methodology through predefine use cases. Users do not have any source code interaction; they see the prototype as it is, as a black box. The rest of the prototype's architect is behind and therefore not perceived in any way from the tester.

The Security Testing Plan follows the OWASP Security Testing methodologies, based on the black box approach. In this testing plan, 3 people have been testing due to the extended time required for Security testing. See IdeaClick Security Testing plan appendix for objectives, testing process and results. IdeaClick testing plan describes objectives, use cases, results and most relevant information concerned to the testing process. See IdeaClick Usability and Security Testing plans appendix.

4 Conclusions and Criticism

The main objective of the thesis was to describe and execute a security and usability testing plan, in order to evaluate the prototype IdeaClick. IdeaClick is the result of the VISCI Tools research project, which took place at SimLab, Aalto University. The purpose of it is to develop a virtual environment where users can share and collaborate on ideas.

VISCI Tools research project has been working on IdeaClick prototype. The project has run a few ad-hoc tests to ensure basic functionality. Due to those results and the prototype present stage, a usability test plan was required in order to fulfill a proper project development procedure. With that a more accurate decision can be done over IdeaClick further development.

This chapter will present the conclusion from the two areas covered by the thesis. Security related by Manuel Bacso and Usability related by Omar Lenin Gutierrez Gutierrez.

4.1 Usability Conclusions

Based on the testing plan results, IdeaClick has the potential to become an excellent web application. All IdeaClick testers agreed on the good benefits that its services can bring, since it could be well adapted to different working areas. Unfortunately, the test results also present a variety of inconsistences and errors that need to be fixed first.

Most of the errors found are related to the subjective satisfaction factor, which is associated to how much a user wants and likes to use a system. The comments related to this subject are link to IdeaClick's menu design and locations, UI colors applied and button naming. These inconsistencies and errors could cause a negative effect over the user by reducing their desire to use the application.

Finally, the study indicates that IdeaClick must go through a user interface re-design, since clearly most of the negative remarks were related to it. Therefore, if the responsible people of IdeaClick projects are willing to proceed and give further continuity to

IdeaClick development, the final suggestion is to invest more time and effort to improve the actual user interface of it.

4.1.1 Author's reflections and thoughts

I, Omar Lenin Gutiérrez Gutiérrez, am pleased to participate in this project. It is an excellent way to culminate my studies at HAAGA-HELIA by making use of the knowledge gathered during the last years, and evaluating myself for what is to come in life.

IdeaClick was an excellent thesis subject to work on, very interesting topic and definitely a new subject to learn from the technological point of view. In addition, IdeaClick was very interest for me, because I got to work on usability and user experience subject, which is has been my main interest during my studies.

Another important element during my thesis work was to face and deal with problems constantly appearing. It forced me to come across with many of my weaknesses and more important, was the process to learn how to deal with them.
The entire thesis work was a constant learning process, from the scientific point of view but also from my personal experience. These are important elements I'm looking forward to improve in order to become a better work partner and a better person.

To conclude, here is a list of pros and cons that were present during the thesis work process and which I had to face constantly, in order to give a final end to this thesis work.

Pros:

- Interesting topic, Usability testing
- Plan testing structure
- Source information access

Cons:

- Motivation

- Commitment to thesis work
- Commitment to what is what is was planned
- Registering work load
- Backing up documents

4.1.2 Fears and improvement

After the first meeting at Aalto University with the project representatives, I started to have my first concern about the project. I was wondering if I was able to present a final document that could enclose substantial information, in quantity and quality, so it would be very useful for IdeaClick project managers. Over more, I was slightly concerned of the work load share. What could be the proper amount of work share between us? Since this is a shared thesis. And finally, What if the document was not sufficiently good to fill the thesis objectives? These were the first questions popping in head at the beginning.

Another concern was related to my regular everyday activates, how possible they could affect to the development of my thesis. I have a 4 year daughter that I have to take care, a part time job that I have to attend, and by the time a few courses that I had still pending at the university. I knew that all these things could cause a negative impact over my thesis work. And as I feared, all of them were present at certain point during the development of the thesis. Therefore, the thesis was not finished on time, as we predicted.

From this experience and for my future projects, I would definitely put much more effort on the work process phase, in terms of commitment to project plan, and as well to the agreements made in the group. In order to have a constant and progressive work, with no long pauses that could interrupt a continue work development.

4.1.3 Recommendations

Based on the entire thesis work process, that includes the creation of project plan documentation, testing plan design and execution, and finally our thesis document, I would like to make a few recommendations for students and people interested on developing a similar thesis final work in the future. These are general recommendation over the entire thesis work.

Next recommendations are dedicated to whole documentation work process:

- Keep a detailed track of the daily work process.
- Commit project definition
- Take plenty of time to defining all documents, such as goals, scope, possible outcomes, Etc.
- Do not take long breaks while executing any phase of the work thesis, it might cause a drop back on the whole process
- keep a constant work development
- keep constant contact with adviser and all the people involved in the project
- keep document backups
- Thesis references update them while creating the document; do not leave it for the last.
- take a break from work or any other circumstance that could make you lose concentration on keeping a constant work rhythm
- Set dead line for every phase and commit to it, very important.

Recommendations for the usability work process that includes testing plan:

- Set, carefully, scope and goal for the project
- Take enough time to find the most appropriated testing method
- Take enough time to find the most appropriated evaluation method
- Read note and documents from previous courses

- Get enough time to know the application
- Test yourself
- Do not hurry defining testing cases
- Pay attention to document consistency
- Pay attention to case testing consistency
- Follow strictly testing plan definition

Therehere are specific recommendations for IdeaClick managers, explaining how IdeaClick UI can be improve. These recommendations can be found at the section: Reporting result, in the Usability testing plan attachment.

4.2 Security Conclusions

As seen by the test results, the IdeaClick prototype is potentially a very powerful interface, harnessing many features for easy establishment of relational diagrams. Unfortunately, it is still lacking quite a few aspects which need attention.

A few security vulnerabilities have been found on the IdeaClick prototype. One very troubling issue is the ability for a hacker to hijack another user's session quite easily allowing him to do practically everything the actual user himself could do. One additional issue is the ability for anyone to retrieve any objects stored on the database, even though this does not allow for the hacker to retrieve users' information, it does allow for all InnoObjects to be retrieved, including archived ones. All vulnerabilities found are quite easily fixed and won't require much time by the developer assigned. Nevertheless these are issues which should be attended if the prototype and the platform are to be developed further and be used in future scenarios.

The prototype is as stated still in its prototype stage and is to provide a proof of concept, and so by fixing these security issues, the prototype would be sufficiently secure to allow further use of it.

4.2.1 Author's Reflections and Thoughts

We have been very interested in this prototype from the beginning and hope that our findings will provide as much information as possible to allow the future development of the prototype. We are satisfied that we have been able to provide everything we had set out for in our project plan.

I, Manuel Bacso, am happy about providing the information necessary to evaluate the prototype at its current stage and hopefully contributing in the future development of IdeaClick. However, it should be clear that this is a prototype and not a tool to be deployed in everyday use, but much rather a milestone for what may become a tool.

4.2.2 Recommendations

The IdeaClick prototype was my contribution to my work placement at the SimLab department. I loved to see how it grew and became more and more as an extended prototype that could be used both on a desktop computer as well as on a touch screen device, such as a tablet. The opportunity to continue my work on this prototype was a great chance to improve it even further.

This thesis was a team effort, and that is at times very taxing, but it does have its benefits, even though it's difficult to collaborate and write a single paper with more people, it does provide the chance to easily combine two minds and therefore have a better result. This also gave me a good change to once again learn how to work as a group and focus on team effort, rather than just my own.

I recommend for people to attempt to develop their own applications, since this is the only way we will come up with great new tools. It's not that hard and the satisfaction in the end is great. I have learned many things about my own prototype after I had seized its development for the time. If you have the possibility to write your thesis about it and see it as a worthy topic, then that is a great opportunity to learn much more about your own tools.

4.3 Time management

This chapter explains the time used over the entire thesis process development. It described a usability and security time management separately. In addition, a time distribution table during the whole project development.

4.3.1 Usability time management

Unfortunate, I believe that the time management of IdeaClick project, design and execution, was not efficient. Therefore, the documentation process took more time than expected. A series of difficulties encountered during the whole process, leaded me to rush up during the last two weeks, so I could meet the thesis dead line presentation. The thesis work requires an estimated time of 410 hrs. I estimated 420hrs of personal work on the thesis.

The graphic shows the amount of hours used per day and how regularly I work on the thesis. unfortunatly the graphics is to long to show every day of work during the last 8 month.

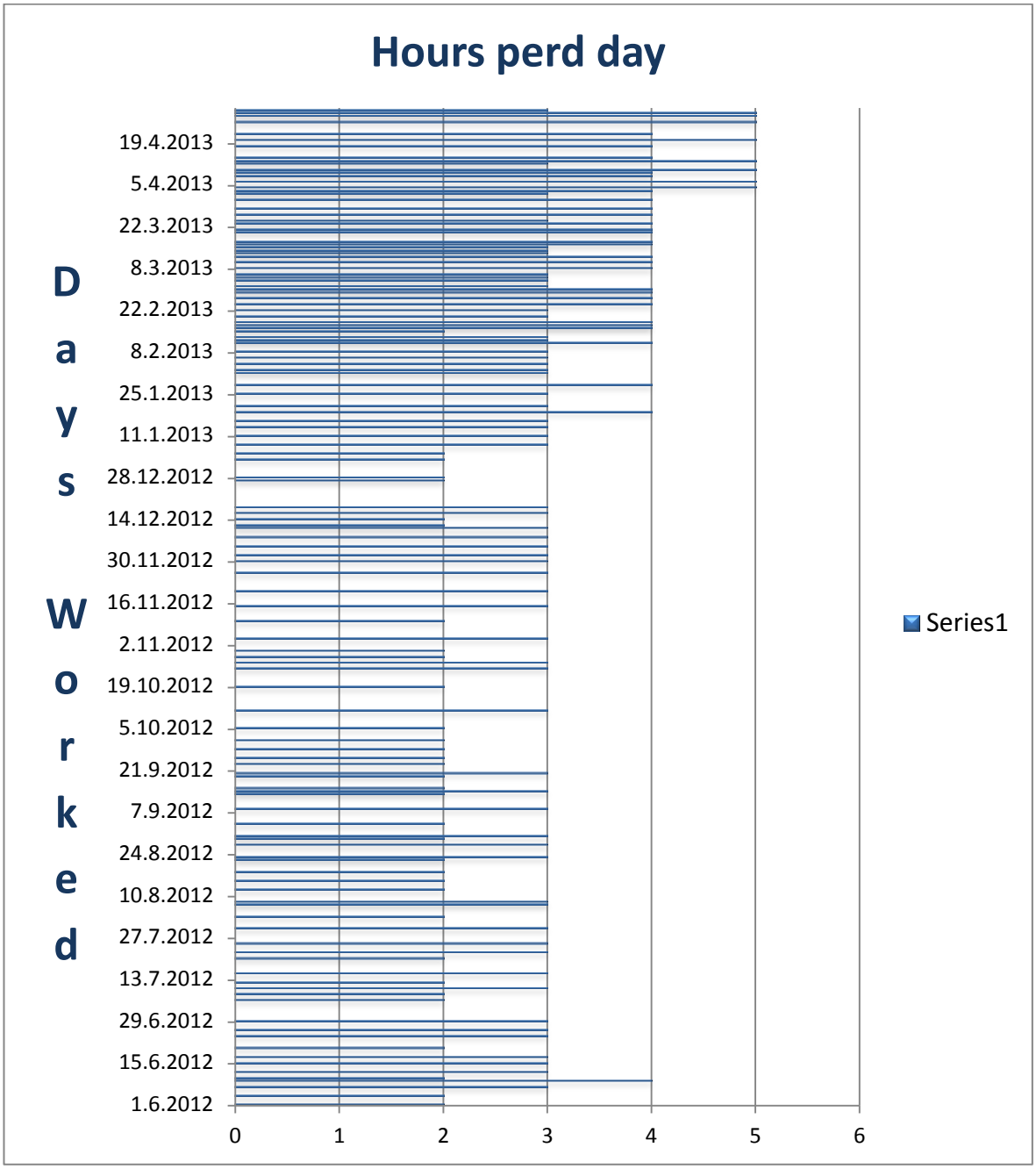


Figure 19 Time management daily hours work

4.3.2 Security time management

Initially the thesis was intended to be completed much faster. Due to some complications this was unfortunately not the case. Everything started as planned, all meetings were occurring as scheduled and the progress was on track, even with a minor set-back from HAAGA-HELIA, which had difficulties in arranging a supervisor for this thesis, despite an external teacher offering her assistance in the matter. Once the time for the Testing plan definitions came along, the amount of work to be done was underestimated and the process took much longer than expected. Slight changes were made to the project plan definition to redefine and narrow the scope of the thesis, giving a better view of what needed to be done. Nevertheless the necessary time was taken and the thesis completion was postponed. From then on the things slowly progressed with some minor loss of motivation, which was acquired back in the end. A total of around 450 hours were spent by Manuel Bacso on the completion of the necessary documents.

4.3.3 Thesis time consumption chart

The following section presents the time management over the whole thesis process. More specifically, the time expected and used during every thesis phases.

Tasks	Expected	Estimated Gutierrez	Estimated Bacso	Note
SimLab representatives meetings	20 hrs.	10hrs.	30 hrs.	
Project plan definition	40 hrs.	60 hrs.	70 hrs.	Document analysis and structuration
Test Planning definition	60 hrs.	130 hrs.	150 hrs.	Gutierrez: Fist document was lost. Computer crashed down, 2/3 of the document were ready when this hap-

				pened
Usability test plan execution	4 hrs.	15 hrs.	12 hrs.	Tests had to be split up into different days.
Usability test plan analysis and report	20 hrs.	55 hrs.	50 hrs.	Analysis of the test results and documentation.
Thesis final document	80 hrs.	150 hrs.	140hrs	This does not include the time used in the document for its correction.
Total hours	**224hrs.**	**420hrs.**	**452hrs.**	

Figure 20 Time management table

References

AllInterview 2013. What is definition of functional and security testing.
URL: http://www.allinterview.com/showanswers/27963.html.
Accessed: 27 March 2013.

AppliCure 2013, Web Application Security.
URL: http://www.applicure.com/solutions/web-application-security.
Accessed: 23 February 2013.

Barnum, C. 2011. Usability Testing Essentials ready, set … Test! Morgan Kaufmann. USA

Bootstraptoday 2012. Classic Mistakes of Software Development and How to Avoid Them
URL :http://blog.bootstraptoday.com/2012/03/21/classic-mistakes-of-software-development-and-how-to-avoid-them/.
Accessed: 22 April

BusinessDictionary 2013, Security.
URL: http://www.businessdictionary.com/definition/security.html
Accessed: 23 March 2013.

Churm T. 2012. An introduction to Website Usability Testing.
URL: http://usabilitygeek.com/an-introduction-to-website-usability-testing/.
Accessed: 28 November 2012

Irrmann, O. 2012. EURAM Virtual collaboration tools and the front end of an innovation process.

Kronqvist, J. & Salmi,H. 2011. Start with a small ball of snow – Presenting Multiple Meanings as a Challenge and Basis for Participatory Innovation

Kronqvist, J., Salmi, A. & Pöyry-Lassila, P. 2009. SEBPSS- Supporting Empathy in Business Process Simulation with Scenarios

Kronqvist, J., Salmi, A. & Pöyry-Lassila, P. CNFTIC - Collaboratively Narrating the Future Though Idea Cards.

Nielsen Norman Group 2009. Why You Only Need to Test with 5 Users
URL: http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/.
Accessed: 8 march 2013.

Patton, R. 2006. Software Testing, San Publish, The united State of America.

Prabhakaran, R. 2013. Blog. Software Testing.
URL: http://rajeevprabhakaran.wordpress.com/software-testing/.
Accessed: 27 December 2012.

QA and Testing Tutorial. 2011. Welcome to QA and Testing Tutorial.
URL: http://www.qatutorial.com/.
Accessed: 17 December 2012.

QA and Testing Tutorial. 2011. QA and Testing Tutorial, Types of Testing.
URL: http://www.qatutorial.com/?q=Types_of_Testing
Accessed: 18 December 2012.

Rubin, J. & Chisnell, D. 2008. Hand book Usability Testing. Wiley Publishing, Inc. Indianapolis, Indiana.

SeaNergyPro 2013. Testing life cycle.
URL: http://seanergypro.com/testinglifecycle.html.
Accessed: 15 Marzo 2013.

SeaNergyPro, Testing services and types
URL: http://seanergypro.com/testingservicestypes.html.
Accessed: 27 February 2013.

SlideShare. 2011, Basic history of software testing.
URL: http://www.slideshare.net/UniversalExams/basic-history-of-software-testing.
Accessed: 3 April 2013.

Sofwareqatestings Sqat 2013. Answers. What is the definition of software testing?
URL: http://wiki.answers.com/Q/What_is_the_definition_of_software_testing.
Accessed: 27 March 2013.

SoftwareTestingMentor 2013. Security Testing.
URL: http://www.softwaretestingmentor.com/types-of-testing/security-testing/.
Accessed: 27 March 2013.

Stuyyard, D. & Pinto, M. 2011. The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, Wile Publishing, Inc. Indianapolis, Indiana.

UsabilityGeeg 2013, Introduction to website Usability Testing
URL: http://usabilitygeek.com/an-introduction-to-website-usability-testing/.
Accessed: 1 January 2013.

UsabilityGeeg 2013, Why Web Site Usability is Important for a Company
URL: http://usabilitygeek.com/why-web-site-usability-is-important-for-a-compan/.
Accessed: 1 January 2013.

Usability.gov. 2013. Your guide for developing usable & useful web sites

URL: http://www.usability.gov/methods/test_refine/learneval.html
Accessed: 1 march 2013.

Usability.gov. 2013. Your guide for developing usable & useful web sites
URL:
http://www.usability.gov/methods/test_refine/learnusa/index.html#.UTdddjeFAik
Accessed: 1 march 2013.

Wikipedia. 2013, Debugging
URL: http://en.wikipedia.org/wiki/Debugging.
Accessed: 8 march 2013.

Wikipedia. 2013. Software testing.
URL: http://en.wikipedia.org/wiki/Software_testing.
Accessed: 2 march

Appendices

Usability Testing Plan

The Usability Testing Plan can be found as an attachment under the name "Attachment 1 - Usability Testing Plan".

Security Testing Plan

The Security Testing Plan can be found as an attachment under the name "Attachment 2 - Security Testing Plan".

HAAGA-HELIA
University of Applied Sciences

IdeaClick Prototype

Web Application Usability Testing Plan

Omar Lenin Gutiérrez Gutiérrez

Usability Testing Plan

DP in Business Information Technology

24 April 2013

Table of Contents

1 Introduction

This Usability Testing Plan (UTP) has been creating for the VISCI Tools project, which is developing at Aalto University School of Science and Technology, SimLab department, Espoo-Finland

IdeaPlatform and its IdeaClick prototype are a result from the research project development. IdeaClick is an online service for the company's employees sponsoring the project; IdeaClick is now at its prototype stage. IdeaClick is a visual environment where users can share their ideas; suggest possible modification for already existing one.

The user's ideas are share by creating objects ideas; these are already predefined objects in the prototype. The objects are created, modified and deleted, by drag-and-drop method; this is done in a so call, objects working space or drag-and-drop area.

A Usability testing plan is required in order to provide more data to project's managers, so they can make a further analysis and additional consideration for the next step of the project.

> The research team's uses a frond-end of innovation (FEI), is the phase of the new product development process during which ideas are born and further developed, ending with the decision to start a new prototype development project (Virtual Collaboration tools and the front end of an innovation process, Olivier Irramann 2011, 1).

2 Scope

The scope of this document is to describe a Usability Testing Plan (UTP) in order to conduct a usability test, over IdeaClick prototype at the SimLab department in Aalto University.

The document's approach is to set down a scheme for the testing activities related to IdeaPlatform and the IdeaClick prototype. Also a heuristically analysis will be described and developed.

2.1 Test objectives

The Usability testing main objective is to identify the present status of IdeaPlatform, especially the IdeaClick prototype, in term of its usability interface performance.

- Provide test metrics / testing summary reports
- Identify defects/errors
- Identify design inconsistency
- Verify that software requirement are accurate
- Verify that software requirements are complete
- Verify user friendliness
- Verify the appearance
- Verify usefulness
- Verify performance

3 Methodology

This testing plan will consist of two different parts. The first one is concerned to the usability side of the IdeaClick, which includes Use case execution by testers and a second one, a heuristically evaluation also performed by the tester.

A Usability testing is required due to outcome data generated by it, which is essential for further possible adjustment in the prototypes.

Outcome data expected are:

- Time used by the tester to complete a use case.
- Time taken by the tester to understand the prototype
- Determinate the amount of mistakes during the use case execution
- Determinate time by the user familiarizing with the prototype
- User feelings when uses the prototype
- Interface intuitiveness for the user

3.1 Methods

These two methods, Use case execution and a heuristically evaluation, have been chose due to present stage development of IdeaClick, which is prototype stage. Therefore, there is no intention of using other methodologies such as:

- Heuristic estimation
- Cognitive walkthrough
- Pluralistic walkthrough
- Feature inspection
- Formal usability inspection

Methods that involves more specific definition and procedures in terms of design and analysis

3.1.1 Use cases

A use case is a description of how users will perform tasks on the Web site and includes two main parts:

The steps a user will take to accomplish a particular task on your site

The way the Web site should respond to a user's actions

A use case begins with a user's goal and ends when that goal is fulfilled. (Creating uses cases, Kenworthy, E. 1997)

3.1.2 Usability Heuristic Evaluation: Advantages

This method can provide some quick and relatively inexpensive feedback to designers. Feedback can be obtained early in the design process. Assigning the correct heuristic can help suggest the best corrective measures to designers. (Heuristic Evaluations, Molich, R. and Nielsen, J. 1990)

4 Executive summary

SimLab department, in order to give continuity to the front-end project development, located at Aalto University, Espoo – Finland, has agreed with Gutierrez Gutierrez Omar, (Haaga-Helia students) to crate and develop a UTP for IdeaPlatform and IdeaClick prototype.

The UTP and execution will follow up methodologies and approaches gain by the participants in charge of the UTP during their studies at Haaga-Helia. The design and execution will also be under observation by the representatives of Aalto University and Haaga-Helia University of applied sciences.

The UTP will focus on the practical part of testing the platform on real life scenarios. Currently the platform is in its prototype stage, which is a beta stage. Testing result will be analyzed and documented

- Test current performance and reliability from IdeaPlatform
- Test current performance, reliability and usability of IdeaClick
- Assess result
- Document testing execution

5 Participants

The participants for this QCUTP will be integrated by 6-8 participants, 6 testers and 2 observers. The participants don't have any prior knowledge from the IdeaPlatform and IdeaClick prototype, in terms of its usability. This is in order to measure the application's intuitiveness design.

The participants are related to the IT field. Therefore, students from Haaga-Helia are being selected to perform the test use cases during the actual testing event. This is because, the student can have a better understanding regarding to the actual test and its meaning.

The participant will execute a series of use case. Each use case describes as a set of steps, clearly defined, in order to accomplish the tasks and goal required by each of them.

The participant will receive general information, related to testing goals, procedure, and short description of the platform and prototype. As well, some information related to the testing ethics and expected acquitted during the resting execution.

During testing process, two of the participant will play different roles, as observer and facilitator.

6 Trainings

This usability testing does not require any prior training related to the applications under test. The participant will have a short introduction about the IdeaPlatform and IdeaClick uses and purpose.

7 Procedure

The participants will make carry on with this testing plan at Aalto's University facilities, Espoo.

A single laptop HP EliteBook, intel core i7 will be gave to every tester in order to access IdeaClick prototype web site. In addition, the observers and facilitators are presented to the tester. Their responsibilities during the testing development are also described to testers. The testers' tasks development will be followed up by the observers situated in the same room. There are not devices that make perform any kind of observation, the only participants during the testing are the observers and facilitator.

The testers are committed to go through the entire testing process, until they finish all the use cases designed for the testing. There is no room for the tester to leave the testing precedent in the middle of it, testers are informed of this prior the activity.

The facilitator will give a short description of the web site, IdeaClick, the uses and purpose of it. There are not specific descriptions regarding the IdeaClick button events, design, web page location, or information that possible could make the tester navigation easier. The facilitator is not entitled to give further information that might enhance or change the use case itself.

Facilitator wills handout a printed copy of the testing case to testers, the facilitator asks to tester to read out loud the testing case to ensure that tester is going through the entire use case and make sure that he knows the amount of time that he has in order to complete the use case. When the tester feels ready can begin to execute the series of steps described in the use case. Observers ask to tester to speak out his thoughts related to the test use case while he executes it. Facilitator asks to tester if there is any question or doubts related to the testing proceeds.

The observer will write down the process of the tester entering the use case data, any comment that the tester could make, as well facial expression and body language.

After each use is completed, the observer will hand out a blank page to the tester so he can write any comment or thoughts from the testing case and it execution. After all the

use cases are completed, the facilitator will ask to tester to answer an after-test questioner.

The Facilitator will dismiss the tester.

8 Roles

An individual may play multiple roles and testing process requires all roles described to be present during the actual event.

Facilitator

- Presents an initial idea of the product to be test
- Explains the purposes of the testing to the participants
- Assist the testers during the test process
- Gives the assistance to tester during the case taste process development
- Handout use cases to tester
- Handout after-test questioner
- know how uses of IdeaClick

Test Observers

- Entitle to only make note from the testing process
- Silent to testers
- know how uses of IdeaClick
- can act as facilitator

Tester

- Execute test cases
- Responds after-test questioner
- Write down thoughts regarding IdeaClick interface.
- Does not have knowledge from IdeaClick, more than the information given during the introduction of this testing.

9 Ethics

All the personal I involved in the testing event will have to agree on following described ethical rules:

- The tester's names will not be registered at any time, as well during the execution of the assigned tasks
- there will be not be an tester's evaluation from tasks execution

10 Usability tasks

From the book *Paper Prototyping* by Carolyn Snyder, published by Morgan Kaufmann Publishers. Page 1

Instructions for Use

- **Task # and name.** Give each task a brief descriptive name and a number. The name helps you remember what its purpose is, and the numbers are useful in usability testing because you can ask the observers things such as, "Shall we skip 3 this time and go right to 4?" without discussing the content of the tasks in front of the users.

- **Goal/outputs.** What will users have accomplished when they're done with the task? Is there a tangible output? How will they know the task is complete? What might the users do to be sure?

- **Inputs.** List all the information or resources—tangible and intangible—that a user would need to complete this task. Examples include, a valid log-in, business policies, physical objects such as a textbook or a credit card, file names, and so on. Real users may have some of this information in their heads—in your usability task you might have to provide this information. For example, a network administrator probably knows the network configuration by heart, but for your task you'd need to create a network schematic with relevant details, such as server names and IP addresses.

- **Assumptions.** Assumptions are the conditions and prerequisites that are in place at the start of the task. The assumptions depend on what you want to learn from the task. For example, if a task explores how users recover from an error caused by a duplicate record, your assumptions include the condition(s) that cause the error to occur, such as, "An employee with the same name already exists in the database."

- **Steps.** Write down the steps you expect the user will go through in completing the task. This helps you identify the prototype pieces that you'll need to create.

Writing down the expected steps can also be helpful if there will be observers who aren't as familiar with the interface as you are. Keep the steps mostly at a screen level—no need to list every field on the order form, just say "order form." Some tasks have multiple paths that lead to success, so jot down any variations, such as "Search OR navigate to lawn & garden page." Put optional steps in parentheses, such as (Review privacy policy).

- **Time estimate for expert.** Estimate how long it would take an expert (someone on the core team) to complete the task. Ignore any time needed for the system to do its processing and focus on the time spent entering data and clicking buttons. Some tasks, such as composing an email, require time for thinking or creative effort, so allow time for that. In deciding how many tasks you'll need to fill your test time, multiply this estimate by an factor appropriate for your interface (typically, a number between 3 and 10).

- **Instructions for users.** Don't write the instructions for the users when you're filling in the rest of the template. Although task design works well as a group activity, writing the instructions can be done by one person after you've drafted your set of tasks.

- **Notes.** The notes section might have several types of information, including the reasons why you created the task, how you'll conduct it, specific things to watch for, and questions to ask users after the task is complete. Information to include in the notes varies depending on what's being tested. Write down whatever information you think will be useful to have on hand during the usability tests, and give copies of the completed task templates to usability test observers.

- **NA.** Stands for, No Arguments.

Task1- <Register a new user>

Goal/Output:	To test if IdeaClick interface is capable of registering a new user. Code validation
Inputs:	First name Surname E-mail address Password Confirm password
Assumptions	The user has not created an account prior to this test case.
Steps:	1. go to the URL:http//ideaplatform.net 2. click on the button PROTOTYPE #2 IdeaClick 3. type in First name 4. type in Surname 5. type in E-mail address 6. type in Password 7. type in Confirm password

Time for expert:	0:50 minute(s)
Instructions for user:	N/A
Notes:	Tester can use his/her own personal E-mail account, which will be removed after the test is completed.

Task2- <Log in>

Goal/Output:	To test if IdeaClick interface is able to perform the actions: • Log in Code validation.
Inputs:	E-mail address Password
Assumptions	The user has created an account already
Steps:	1. Go to the URL:http//ideaplatform.net 2. click on the button PROTOTYPE #2 IdeaClick 3. type in Username 4. type in Password 5. click on Log in button
Time for expert:	0:40 minute(s)
Instructions for user:	N/A
Notes:	N/A

Task3- <Log out >

Goal/Output:	To test if IdeaClick interface is able to perform the actions: • Log out. Code validation
Inputs:	E-mail address Password
Assumptions	The user has created an account prior to this test.
Steps:	1. click on Preferences tag 2. click on log out
Time for expert:	1 minute(s)
Instructions for user:	Please log In first in the IdeaClick, in order to perform this use case.
Notes:	N/A

Task4 - <Create a new idea object to be shared with the users >

Goal/Output:	To test if IdeaClick interface is able to perform the action: • Create and share new idea. A new idea is created and shared with the users' of IdeaClick Code validation
Inputs:	Idea name Idea description in the description text box
Assumptions	The user has created an account prior to this test. The user has and is currently logged in
Steps:	1. Go to the URL:http//ideaplatform.net 2. click on the button PROTOTYPE #2 IdeaClick 3. click on Create tag 4. click on Idea 5. type in the tittle box for the idea: idea test 1 6. type in the description box for the idea : description test 1 7. click on the button submit
Time for expert:	0:30 minute(s)
Instructions for user:	NA
Notes:	NA

Task5- <Create a new picture object to be shared with the users >

Goal/Output:	To test if IdeaClick interface is able to perform the action, • Create Picture. A new picture is created and shared with the users' environment successfully. Code validation
Inputs:	Picture name Picture description in the description text box Picture file
Assumptions	The user has created an account prior to this test.
Steps:	1. go to the URL:http//ideaplatform.net 2. click on the button PROTOTYPE #2 IdeaClick 3. click on Create tag 4. click on Picture 5. type in the tittle for the Picture: picture test 2 6. type in the description for the idea: description test 2 7. click on Browse button 8. search for the picture 9. open 10. click on the button Submit

Time for expert:	1:45 minute(s)
Instructions for user:	N/A
Notes:	N/A

Task6- <Create a new collage object to be share with the users >

Goal/Output:	To test if IdeaClick interface is able to perform the action, create collage. A new picture is created and share with the users' environment successfully Code validation
Inputs:	Collage tittle Collage description
Assumptions	The user has created an account prior to this test.
Steps:	1. go to the URL:http//ideaplatform.net 2. click on the button PROTOTYPE #2 IdeaClick 3. drag and drop an idea object to the working space 4. drag and drop an picture object to the working space 5. encircle the two objects with a line (see instructions for user) 6. click on Create button 7. type a tittle for the collage in the tittle box: collage test 3 8. type a description for the collage at the description box: collage test 3 9. click on button Submit
Time for expert:	1:45 minute(s)
Instructions for user:	Click left mouse button, hold it and make a make a circle figure around the objects
Notes:	N/A

Task7- <Create a link between two objects>

Goal/Output:	To test if IdeaClick interface is able to perform the action • Create a link between two objects. A new picture is created and shared with the users' environment successfully Code validation
Inputs:	Collage title Collage description
Assumptions	The user has created an account prior to this test. There are at least to two objects are set at the object dropping space.
Steps:	1. go to the URL:http//ideaplatform.net 2. click on the button PROTOTYPE #2 IdeaClick

	3. right click on the first object 4. click on Link To 5. click on the second object
Time for expert:	0:50 minute(s)
Instructions for user:	N/A
Notes:	The other two objects required in the object dropping space can be any type of objects.

Task8- < disable object idea from Visibility tag>

Goal/Output:	To test if IdeaClick interface is capable to disable the visibility of the objects of type Idea from the working area view. All idea objects type are no longer visualized at the object dropping space. Code validation
Inputs:	N/A
Assumptions	The user has created an account prior to this test. There are at least three idea objects at the object dropping space.
Steps:	1. go to the URL:http//ideaplatform.net 2. click on the button PROTOTYPE #2 IdeaClick 3. click on visibility tag 4. click on Idea button 5. objects are hided
Time for expert:	1 minute(s)
Instructions for user:	N/A
Notes:	N/A

Task9- < disable picture´s view from Visibility tag >

Goal/Output:	To test if IdeaClick interface is capable to disable the visibility of the objects, type Picture, from the working area view. All Pictures objects type are no longer visualized at the object dropping space. Code validation
Inputs:	N/A
Assumptions	The user has created an account prior to this test. There are at least three picture objects at the object dropping space.
Steps:	1. go to the URL:http//ideaplatform.net 2. click on the button PROTOTYPE #2 IdeaClick 3. click on visibility tag 4. click on Picture button 5. Objects Picture(s) are hide from the object dropping

	space.
Time for expert:	1 minute(s)
Instructions for user:	N/A
Notes:	N/A

Task10 - < disable Collage´s view from Visibility tag >

Goal/Output:	To test if IdeaClick interface is capable to disable the visibility of the objects of type Collage from the working area view. All the Pictures objects are no longer visualized at the working space. Code validation
Inputs:	N/A
Assumptions	The user has created an account prior to this test. There are at least three Collages objects at the object dropping space.
Steps:	1. go to the URL:http//ideaplatform.net 1. click on the button PROTOTYPE #2 IdeaClick 2. click on visibility tag 3. click on Collage button 4. objects Collage(s) hidden
Time for expert:	1 minute(s)
Instructions for user:	N/A
Notes:	N/A

Task11- < disable Link's view from Visibility tag >

Goal/Output:	To test if IdeaClick interface is capable of hiding all visibility references between the objects in the object dropping space. Code validation
Inputs:	N/A
Assumptions	The user have ready created an account There are at least three Collages objects at the object dropping space.
Steps:	1. go to the URL:http//ideaplatform.net 2. click on the button PROTOTYPE #2 IdeaClick 3. click on visibility tag 4. click on Link 5. All references are hidden
Time for expert:	1 minute(s)
Instructions for user:	

Notes:	N/A

Task12- < disable Link's view from Visibility tag >

Goal/Output:	To test if IdeaClick interface is capable of hiding all visibility references between the objects in the object dropping space. Code validation
Inputs:	N/A
Assumptions	The user have ready created an account There are at least three Collages objects at the object dropping space.
Steps:	1. go to the URL:http//ideaplatform.net 2. click on the button PROTOTYPE #2 IdeaClick 3. click on visibility tag 4. click on Link 5. All references are hidden
Time for expert:	1 minute(s)
Instructions for user:	
Notes:	N/A

Task13- <Remove link from two ideaobjects>

Goal/Output:	To test if IdeaClick interface is capable of removing an arrow link between two different objects in the object dropping space. Code validation
Inputs:	N/A
Assumptions	The user have ready created an account There are at least three Collages objects at the object dropping space. The are two object already linked in the objects dropping space
Steps:	1. go to the URL:http//ideaplatform.net 2. click on the button PROTOTYPE #2 IdeaClick 3. Right click on the idea object: idea test 1 4. Click on remove Link 5. Click left on the object : picture
Time for expert:	1 minute(s)
Instructions for user:	
Notes:	N/A

Task14- <Hide an idea object>

Goal/Output:	To test if IdeaClick interface is capable of hiding an Idea Object from the all visibility in the object dropping space. Code validation
Inputs:	N/A
Assumptions	The user have ready created an account An idea object already placed in the object dropping space
Steps:	1. go to the URL:http//ideaplatform.net 2. click on the button PROTOTYPE #2 IdeaClick 3. Right click on the picture idea object 4. Click on: Hide
Time for expert:	1 minute(s)
Instructions for user:	
Notes:	N/A

Task15- <Hide a collage object>

Goal/Output:	To test if IdeaClick interface is capable of hiding a collage from the all visibility in the object dropping space. Code validation
Inputs:	N/A
Assumptions	The user have ready created an account A collage object already placed in the object dropping space
Steps:	1. go to the URL:http//ideaplatform.net 2. click on the button PROTOTYPE #2 IdeaClick 3. Right click on the collage 4. Click on: Hide
Time for expert:	1 minute(s)
Instructions for user:	
Notes:	N/A

11 Usability metrics

Since there is not method capable to measure software, application or website usable design, this testing plan will use already standardized usability metrics, in order to assess IdeaClick in terms of usability performance and design.

For IdeaClick there has been selected the next 4 essential usability metrics

11.1 Task Time

Total task duration is the de facto measure of efficiency and productivity. Record how long it takes a user to complete a task in seconds and or minutes. Start task times when users finish reading task scenarios and end the time when users have finished all actions

11.2 Task Level Satisfaction

After users attempt a task, have they answered a few or just a single question about how difficult the task was. Task level satisfaction metrics will immediately flag a difficult task, especially when compared to a database of other tasks. After Scenario Questionnaire ASQ, a three-item after-scenario questionnaire used for measuring user satisfaction with existent system or prototypes of futures systems (Apex 1)

11.3 Test Level Satisfaction

At the conclusion of the usability test, have participants answered a few questions about their impression of the overall ease of use. For websites is use the SUPR-Q. (Apex 2)

11.4 Page Views/Clicks

For websites and web-applications, these fundamental tracking metrics might be the only thing you have access to without conducting your own studies. Clicks have been shown to correlate highly with time-on-task, which is probably a better measure of efficiency. The first click can be highly indicative of a task success or failure. (Measuring usability 2011.)

12 Usability goals

IdeaClick usability testing plan pursues essentials goal pre-established as basic elements for every web site. Foraker Labs (2012) defines four main goals to par strict attention.

Memorability of a Website

Once a user has taken the time to learn how to navigate a website and find what they are looking for; they need to be able to remember how to do it when they come back. A website needs to have high memorability. Memorability is a measure of how easy a website is to remember after a substantial time-lapse between visits.

Efficiency of Website Designs

Efficiency is a measure of how well a website does what it should do. Assuming that the utility and effectiveness goal are fulfill, efficiency is the next usability goal to take into consideration. Efficiency of the tools introduced into the website is just as important as the presence of the tools themselves. As Susan Dray says, "If the user can't use it, it doesn't work."

Effectiveness in Website Design

Website effectiveness is measure by its ability to do what it should do.

Effectiveness is actively measure by task-completion rates & other test metrics. Information architecture and semiotics play a large role in the effectiveness of a website. Other important areas that affect effectiveness are; page layout, image selection, and content.

This is where we test our website designs to make sure that users are getting information that they expect when they click on any link on a website we design.

Learnability of Websites Designed

Learnability is a measure of how easy a website is to learn, or how fast first time visitors can complete tasks on a website. On the internet-, learnability could be the most important of all usability goals. The reason for this is that if users are not able to find what they are looking for, or get a hint of how they can get to the information they want, they are only a back click away from finding another source. Learnability can be closely tied to the effectiveness of a website.

13 Problem severity

The identified severity for each problem implies a general reward for resolving it, and a general risk for not addressing it, in the current release.Usability Test Plan Template(2013) present the next possibilities.

Severity 1 - High impact problems that often prevent a user from correctly completing a task. They occur in varying frequency and are characteristic of calls to the Help Desk. Reward for resolution is typically exhibited in fewer Help Desk calls and reduced redevelopment costs.

Severity 2 - Moderate to high frequency problems with moderate to low impact are typical of erroneous actions that the participant recognizes needs to be undone. Reward for resolution is typically exhibited in reduced time on task and decreased training costs.

Severity 3 - Either moderate problems with low frequency or low problems with moderate frequency; these are minor annoyance problems faced by a number of participants. Reward for resolution is typically exhibited in reduced time on task and increased data integrity.

Severity 4 - Low impact problems faced by few participants; there is low risk to not resolving these problems. Reward for resolution is typically exhibited in increased user satisfaction.

14 Reporting results

This UTP will deliver a result after finalized the usability test. The report will hand out the final analysis from data and information collect from the test cases and the subjective evaluation given by the participants (testers, observers) of the test.
The report will make recommendations to improve any possible usable deficiency presented in the interface. There will not be an analysis regarding the usable metrics

against the pre-approved goal from the IdeaClick prototype, and this is due to the present stage of it, a prototype. The report is anticipated to be hand out VISCI Tools department and Haaga-Helia University.

14.1 Results

The UTP's testing execution brought satisfactory results. It has exposed the prototype's weaknesses in a very detailed way. Questions by the testers during the test cases development, raised up new points of view concerned to the prototype's design. In addition, Post testing comments writing by the tester, presents good observation concerning to the prototype's usability, comments and recommendations.

14.1.1 Testers comments point outs

This is a summary of tester's comments and point outs. It was not possible to write all of them, since there are many redundant comments.

- Complains related to Log in and registering interface been in the same page
- Recovery account not possible
- Recovery password not possible
- Registering does not mention that E-mail will be the username during the log in process.
- Log out button should be more visible and easily to locate, "as a user, I am used that "log out button" is in the right corner.
- Log out button not visible.
- Discomfort with the location of picture object button and idea button object.
- A tip tool missing
- It is not possible to locate which button has been pressed from the menu, since there is not a clear mark for it, "Make clearer the button clicked"
- Colour's buttons in web site URL:http//ideaplatform.net tend to confuse at the first time.
- Log out not found.

- How make a collage, no clear, tool tip needed.
- Creating collages intuitiveness is not good enough.
- Create tab from main menu, tend to confuse with the menu beneath it, Idea, Pictures and Collages object tags
- Take time to see the link created between objects.

14.1.2 Other observations

- Right click event over an object; when a right-click action over and object, a grey window popups providing several options. If the object is taken back to the left panel, the grey window with optional options should close automatically after moving the object to the left panel.
- The word remove from the right click option over the objects, might mislead to the user. Delete would be a better word
- The application does not save the objects' position. The prototype does not save the object's position. In addition, the name "save position" for this action, is not telling entirely it does, "lock position objects", a name more related to the action.
- There is no reference of two objects linked, when one of them is hidden
- Error, Draw line : TypeError: C is null popups. When a linked object is hide And another object is dropped to the working area, the error popup
- Two objects linked, one of them is hided and the second still on the working space and click one. An error message comes up: Error C is null. Code should be verified
- A collage of object needs to follow different steps. This is, already, an error concerned to usability inconsistency on interface design.

14.1.3 Statistics

Statistical representation of the after testing questioner, Rate is done from 1 to 5, 1 the lowest grade and 5 the higher, percentage result over all the participants. See questioner in Appendix 2 pag.28

This website is easy to use.

10 % of the users – 2 grade

10 % of the users - 3 ;

20 % of the users -5 ,

60 % of the users -4

I am able to find what I need quickly on this website.

10 % of the users - 1;

40 % of the users - 3;

20 % of the users - 5;

40 % of the users - 4

I enjoy using the website.

10 % of the users - 1;

20 % of the users - 4;

20 % of the users - 5;

50 % of the users - 3

It is easy to navigate within the website.

10 % of the users - 1;

10 % of the users - 5;

20 % of the users - 2;

60 % of the users -3

This website keeps the promises it makes to me.

10 % of the users - 1;

10 % of the users - 5;

40 % of the users - 4;

40 % of the users - 3

I can count on the information I get on this website.

10 % of the users - 1;

30 % of the users - 3;

60 % of the users - 4

I feel confident conducting business with this website.

10 % of the users - 1;

30 % of the users - 3;

60 % of the users - 4

The information on this website is valuable.

10 % of the users - 1;

10 % of the users - 2;

10 % of the users - 5;

70 % of the users - 4

How likely are you to recommend this website to a friend or colleague?

20 % of the users - 1;

20 % of the users - 3;

30 % of the users - 4;

30 % of the users - 5

I will likely visit this website in the future

10 % of the users - 3;

20 % of the users - 1;

20 % of the users - 5;

40 % of the users - 5

I find the website to be attractive.

10 % of the users - 2;

10 % of the users - 4;

20 % of the users - 5;

60 % of the users – 3

The website has a clean and simple presentation.

10 % of the users - 2;

20 % of the users - 4;

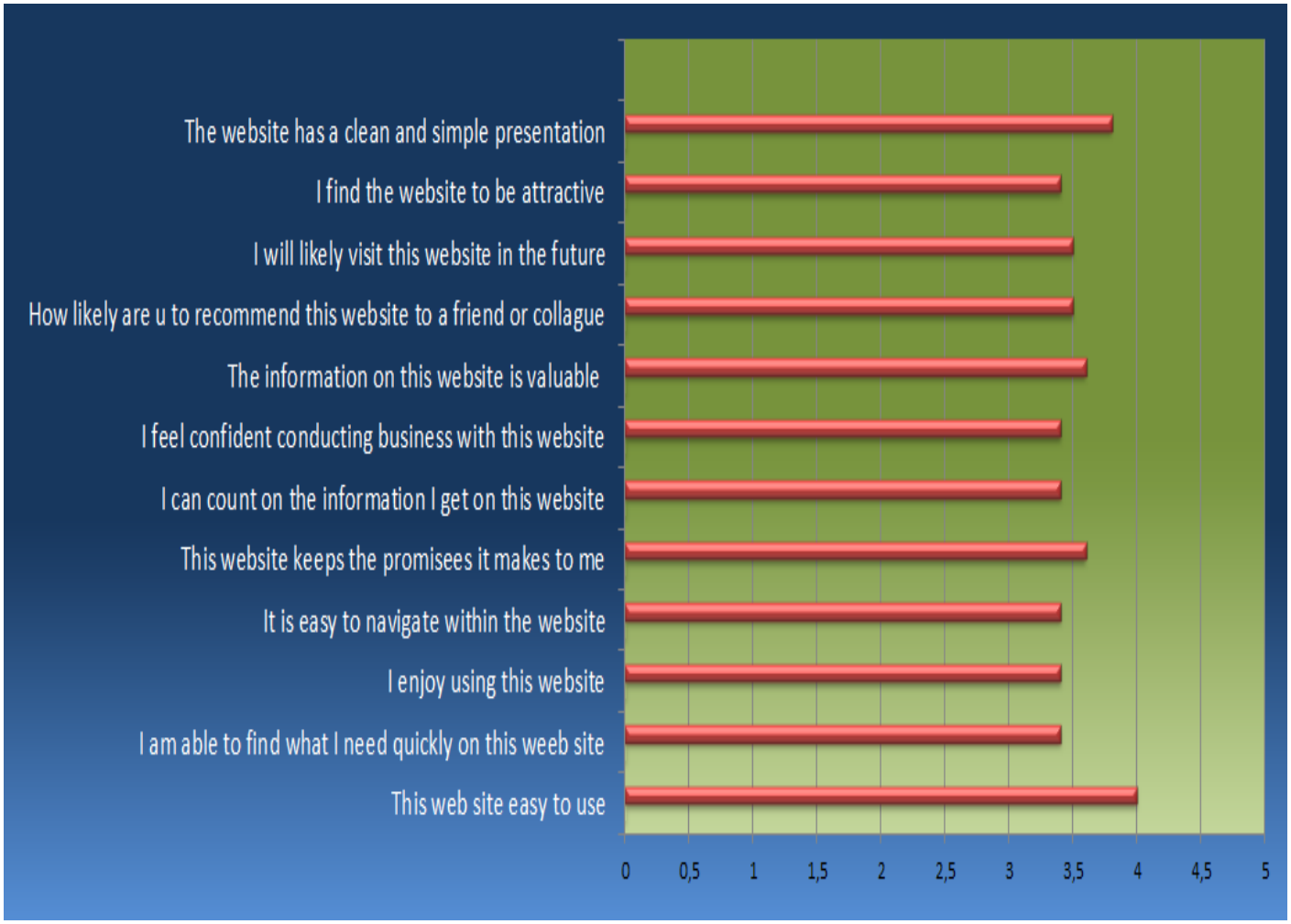
30 % of the users - 5;

40 % of the users - 3

14.1.4 After testing questioner average

Graphical average of the after testing questioner over the ten testers per question
The graphic shows the average rate of the web site by the after testing questioner, as it is evident the question were rated above 50 % of the scale rate, which is a positive reflection of the user's interaction satisfaction.

The final web site rate was a 3.3 based on the after testing questioner



14.1.5 Recommendations

- It would be better to show the actual link line between objects after user login, if there are already linked objects.
- Let know to user that a just created object can be found at the object menu at the left panel.
- Login layout should be attractive for the user, missing logo and colors, that could give and identity to the prototype. Such as, the one made at the URL:http//ideaplatform.net web site.
- Make clearer how to get remove the collages, ideas objects and pictures objects from the working space and bring then back to the left panel.
- Visibilities' tag option from main menu should be hiding, when login
- Use different colors to make difference between the different objects types.
- Use color to make the interface more attractive to user.
- Separate tabs menu from the already created objects menu
- Tool tip, requested by testers.

14.1.6 Summary

Ideaclick prototype base on the test results requires a new interface design since most of the comment result from the test cases and comment and after testing questioner are related to the design of it, more precisely to the button's location on the interface.

60% of the testers agree that the web site is easy to use base on a 4 points grade scale of 5. On the opposite side, 60 % of the tester agrees that web site is unattractive with a 3 points grade scale; 40 % find difficult finding what they wanted on the web site with a 4 points grade scale.

The higher grades were given to the website's information value question, where a 70 % of the testers agree with a 4 points grade. They found the prototype very valuable and innovative.

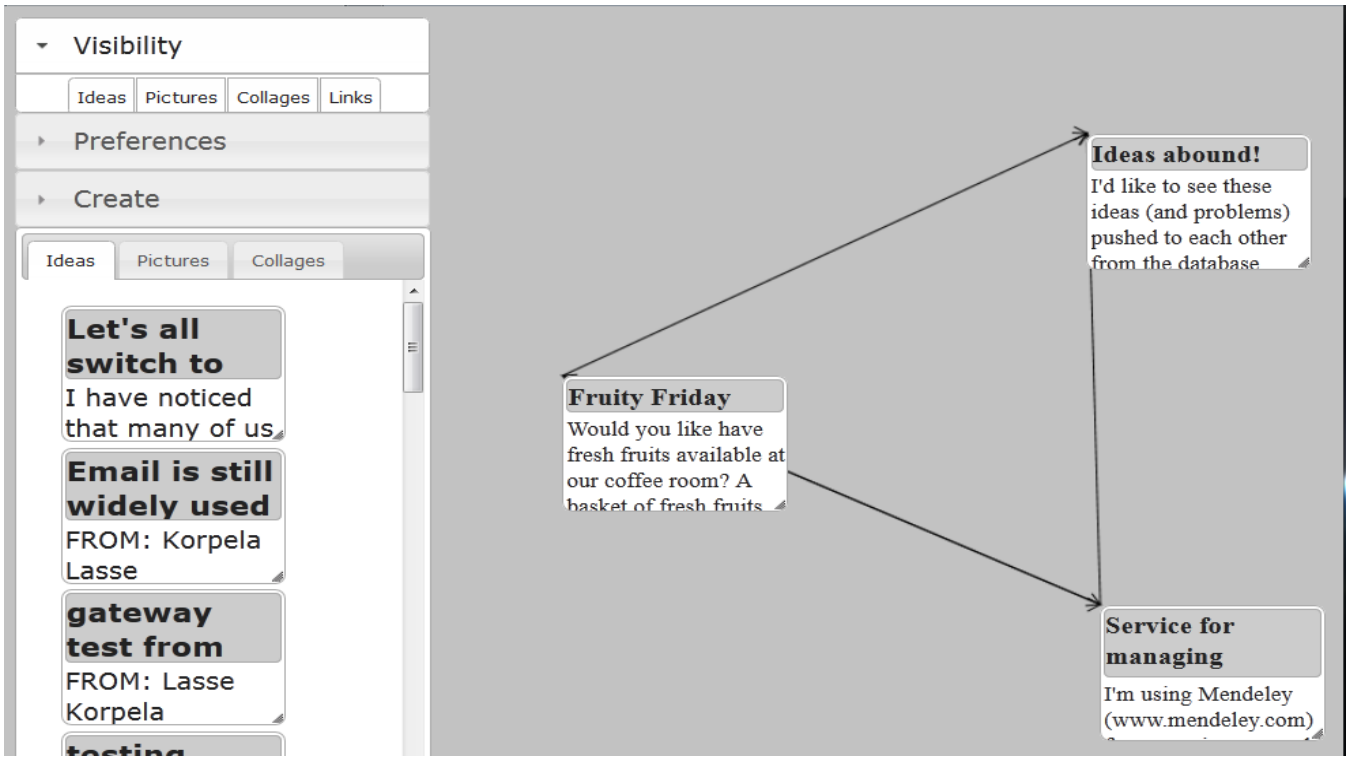
Comments and recommendation were related interface design, more specifically button positioning; buttons are not logically and as usually located. Just to mention a simple but very important one," Logout button ", normally situated on the uppers corners on the screen.
Menus in the prototype are situated to close each ether. Menu, create objects, objects visibility and preferences management respect to the menu; idea, objects and collage. 80 % of the testers disagree on the menu's location. These are Interface intuitiveness related and design, which may cause major problem in terms of the user's application uses in the future.
Performance issues are present in the application due to its prototype stage, these are related to its inconsistency, such as buttons location, objects right-click option, labels naming.
A 60 % agree that the colours used in the prototype should be different. "Interface colours are not attractive, friendly" "interface looks to cold", comments quotations. Brighter or warmers colours may create a more attractive user interface, therefore user's wiliness to use the application can increase. This are related to friendliness user issues, very important to pay attention in term of usability matters.

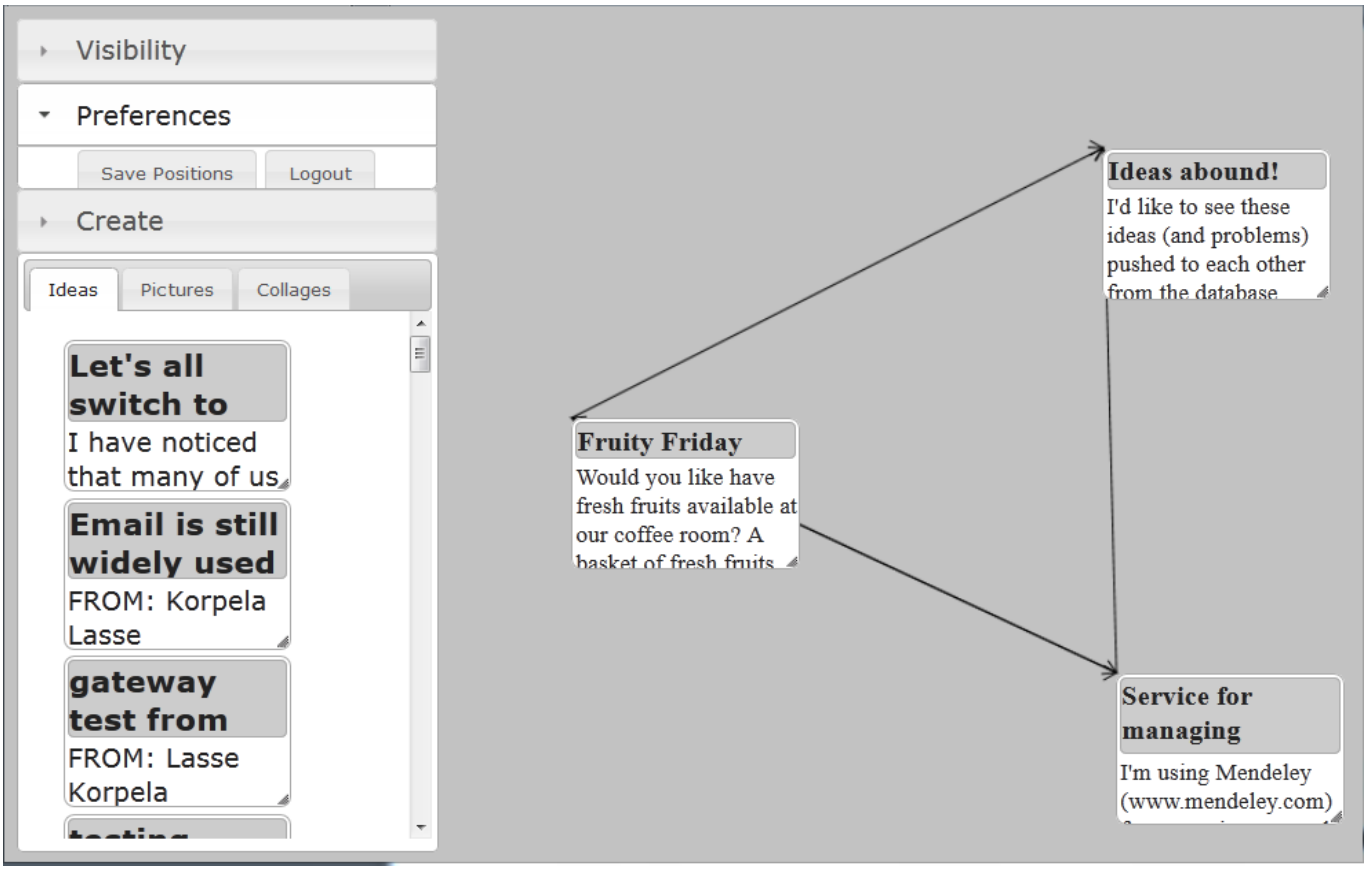
During actual test execution, tester look relax and comfortable. Their comments concerning to the prototype were very positive, no major problems understanding the purpose to use the prototype.

To conclude, Ideaclick testing plan, over all objective, it is to create additional data and information base on the usable perspective for VISCI Tools project. Therefore, Idea-Click project's responsible can make a decision over the project's next step.

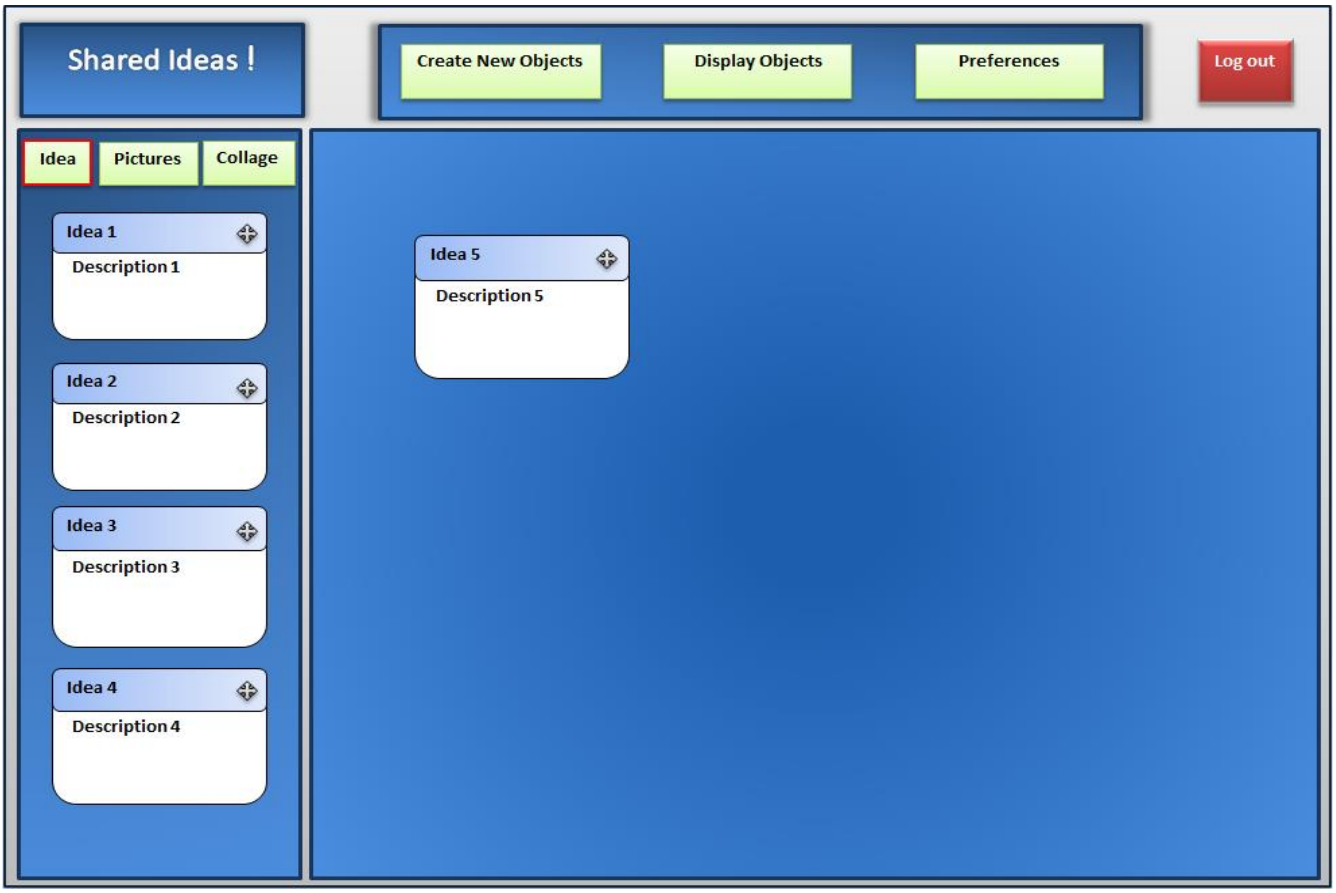
14.1.7 Mockups

Actual View

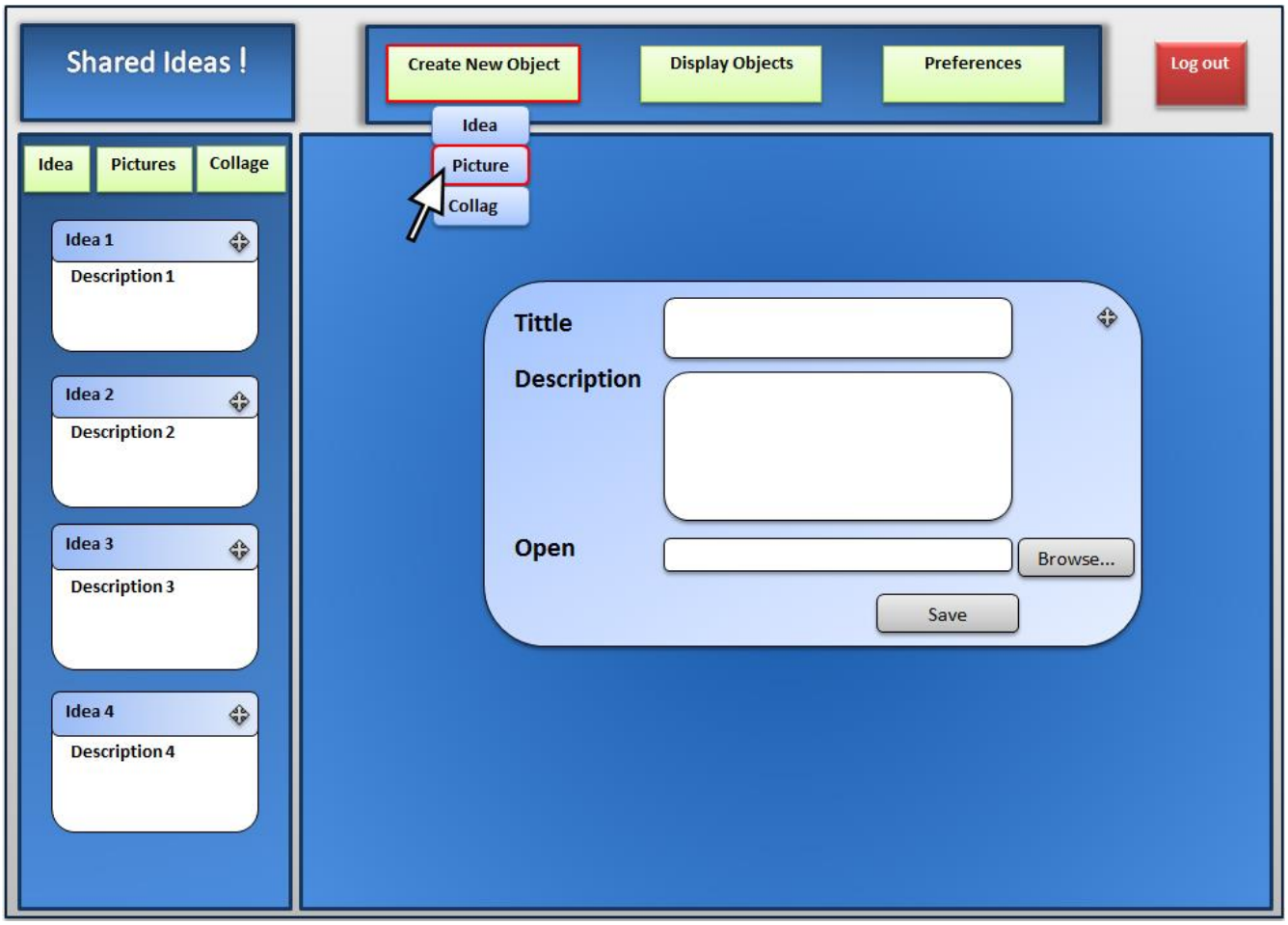




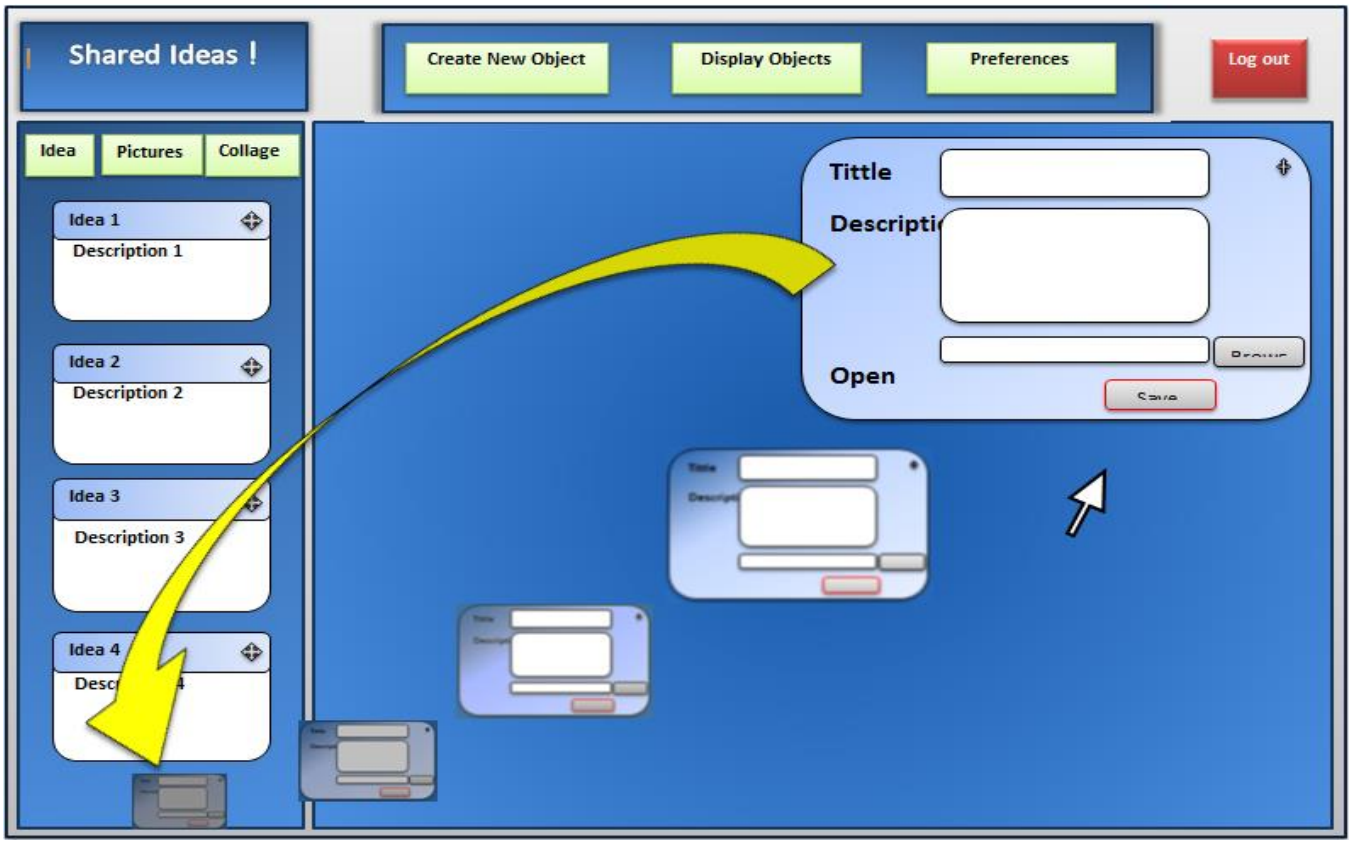
Mockup suggested



Menu distribution, Colours, Button selection



Visual effect of the object moving to the side bar, this let know the user where the object can be located and it is save, Not the yellow arrow.



References

Affordable Usability, User Experience Driven Web Design (2011)
URL: http://www.affordableusability.com/usability/
Access: 2 Nov 2012

Creating uses cases, Ken worthy, E. (1997). Use case modeling: Capturing user requirements. URL: http://www.usability.gov/methods/design_site/usecases.html
Access: 28 Oct 2012

Foraker Labs, Usabilityfirst, usability methods (2002-2012)
URL: http://www.usabilityfirst.com/usability-methods/
Access: 15 Oct 2012

Heuristic Evaluations, Molich, R. and Nielsen, J. (1990)., Improving a human- computer dialogue, Communications of the ACM, 33(3), 338-348 - Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics, CHI'94 Conference Proceedings.
URL:http://www.usability.gov/methods/test_refine/heuristic.html

Sauro, J. 2011. Measuring usability, 10 Essential Usability Metrics,
URL: http://www.measuringusability.com/blog/essential-metrics.php
Access: 20 Oct 2012

Performancetesting, Baseline testing (2009).
ULR: http://www.performancetesting.co.za/Baseline%20Testing.htm
Access: 1 Sept 2012

Software Testing Help, Testing Plan Sample: software testing and quality assurance templates (2007)., URL: http://www.softwaretestinghelp.com/test-plan-sample-softwaretesting-and-quality-assurance-templates/ Access: 10 Sept 2012

Usability.org, Usability Methods,
URL http://www.usability.gov/methods/index.html
Access: 1 Jul 2012

Usability Test Plan Template DOC, Usability.org, Usability Test Plan,
URL:http:// www.usability.gov/templates/docs/u-test_plan_template.doc Access: 1 Jul 2012

Irramann, O. 2011.Virtual Collaboration tools and the front end of an innovation process. pp. 1

15 Appendix

15.1 Apex 1

ASQ, After Scenario Questionnaire

ASQ 1

Overall, I am satisfied with the ease of completing the task in the scenario ?							
Strongly Agree						Strongly Disagree	Not applicable
1	2	3	4	5	6	7	N/A
comments:							

ASQ 2

Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario ?							
Strongly Agree						Strongly Disagree	Not applicable
1	2	3	4	5	6	7	N/A
comments:							

ASQ 3

Overall, I am satisfied with the support information (pre-documentation) when completing the tasks ?							
Strongly Agree						Strongly Disagree	Not applicable
1	2	3	4	5	6	7	N/A
Comments:							

15.2 Apex 2

SUPR-Q, the Standardized Universal Percentile Rank Questionnaire

The SUPR-Q is a Rating Scale to Measure perceptions of Usability, Trust, Credibility, Appearance and Loyalty for Websites

	Strongly Disagree 1	**2**	**3**	**4**	**Strongly Agree 5**
This website is easy to use.					
I am able to find what I need quickly on this website.					
I enjoy using the website.					
It is easy to navigate within the website.					
I feel comfortable purchasing from this website.					
This website keeps the promises it makes to me.					
I can count on the information I get on this website.					
I feel confident conducting business with this website.					
The information on this website is valuable.					
How likely are you to recommend this website to a friend or colleague					

I will likely visit this website in the future.					
I find the website to be attractive.					
The website has a clean and simple presentation.					

HAAGA-HELIA
University of Applied Sciences

IdeaClick Prototype

Web Application Security Testing Plan

Manuel Bacso

Security Testing Plan

DP in Business Information Technology

24 April 2013

Table of Contents

Abbreviation	
HTTP	Hypertext Transfer Protocol
XML	Extensible Markup Language
API	Application Programming Interface
HTML	Hypertext Markup Language
DoS	Denial of Service
OWASP	*Open Web Application Security Project*
SQL	Structured Query Language
IP	*Internet Protocol*
OS	*Operating System*
URL	*Uniform Resource Locator*
MD5	*Message-Digest Algorithm*
PHP	*PHP: Hypertext Preprocessor*

16 Introduction

Software security is an idea implemented to protect software against malicious attacks and other hacker risks so that the software continues to function correctly under such potential risks. Security is necessary to provide integrity, authentication and availability.

- Integrity is an extent which is intended to determine if the information sent to the receiver is correct.
- Authentication comprises of confirming the identity of a person/program, making sure that information can be trusted.
- Authorization determines whether a request should be allowed to perform a certain operation.

Any compromise to integrity, authentication and availability makes software insecure. Software systems can be attacked to steal information, monitor content, introduce vulnerabilities and damage the behavior of software. Malware can cause DoS (denial of service) or crash the system itself.

SQL injections use malicious SQL code to retrieve or modify important information from database servers. SQL injections can be used to bypass login credentials. Occasionally, if bad security measures have been applied, SQL injections can cause hackers to fetch important information from a database or delete all important data from a database. In most cases passwords are the preferred choice for retrieval. (Techopedia.)

16.1 Objectives and Scope

The Objectives of this test plan is to identify and ensure the reliability of the IdeaClick prototype by searching for security vulnerabilities and exploits. These tests will be conducted based on guidelines provided by a recognized source of Web Application Security best practices, OWASP (The Open Web Application Security Project).

OWASP is a non-profit community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted. All of its tools

and documents are free for anyone interested in improving their application security (OWASP. 2013.)

Due to OWASP's Testing Guide being so thoroughly defined other Educations or Organizations Guidelines have not been included. Some although have been chosen to aid the selection of the most feasible OWASP Test cases for the IdeaClick prototype.

SANS is a cooperative research and education organization, focusing mainly on Computer Security Training & Certifications. These educations findings provide good guidance to selectively identify the most feasible Test Cases from the OWASP Testing Guide.

The Scope of the test plan will be to define and implement test cases based on OWASP Security Checklist, which will be used to write reports on security vulnerabilities of the IdeaClick prototype. Selected high risk issues will be analyzed and reported.

The testing will be divided into two parts where feasible guidelines for this test plan have been selected. The first phase involves information gathering, for the user to get to know the application and manually search for vulnerabilities. The second phase involves the testing of predefined tasks.

OWASP Application Security Checklist will provide a guideline for the testing. Since this Checklist covers all possible vulnerabilities Web Applications may have, only a few important Tasks will be selected, based on what is most feasible for the selected prototype. Risks that have already been identified and documented prior to this test won't be tested. Risks that don't apply to this prototype, due to e.g. different technologies being used, won't be tested either.

16.2 Methodology

Methodologies are industry guidelines on how tests should be carried out. There is no optimal path which to take when testing for security vulnerabilities. Every application is different and therefore requires custom test cases. However, even without following the OWASP guidelines word for word, they can help to ensure that the application is being tested thoroughly.

OWASP's Testing methods are used to cover a wide area of vulnerabilities in a web application. It focuses on all possible security risks an application may have. Although the OWASP Guide provides a thorough approach on testing an application, it is impossible to cover every aspect, and in many cases the OWASP Guide may be too thorough to be a feasible testing plan. Therefore only the most important test plans will be chosen.

The OWASP Testing Methodology is based on black box testing, this means that the tester does not know anything about the application before the Testing. He may although receive a quick introduction onsite prior to the testing phase.

The test is divided into 2 phases, the Passive mode, and the Active mode.

In the Passive mode the tester focuses on gathering as much information as possible and familiarizing himself with the Application. By the end of this phase, the tester should understand how the system works and possible entry points the system may have. The tester may also use third party tools, such as an HTTP proxy which monitors incoming and outgoing requests.

For example, the tester could find the following:

- https://ideaplatform.net/core/include/forms/login.php

This may indicate an authentication form in which the application requests a username and a password.

The following parameters represent two access points (gates) to the application:

- http:// ideaplatform.net /ideaclick/?a=1&b=2

Here are two entry points to the application which need to be tested. Each entry point may be a possible source of attack for a hacker. Identifying these entry points help in the process of the active phase.

The active phase consists of the testing phase, where the tester now uses all the gathered data in the passive phase. This phase is split up into two major categories, Au-

thentication Testing and Session Management Testing. More detailed specifications can be found in the Test Case Definitions section.

This testing plan uses black-box testing, below is a comparison between black-box and white-box testing by Stuyyard, D. and Pinto, M. (2011, 702).

> This involves attacking the application from the outside and monitoring its inputs and outputs, with no prior knowledge of its inner workings. In contrast, a white-box approach involves looking inside the application's internals, with full access to design documentation, source code, and other materials.
>
> Performing a white-box code review can be a highly effective way to discover vulnerabilities within an application. With access to source code, it is often possible to quickly locate problems that would be extremely difficult or time-consuming to detect using only black-box techniques. For example, a backdoor password that grants access to any user account may be easy to identify by reading the code but nearly impossible to detect using a password-guessing attack.
>
> However, code review usually is not an effective substitute for black-box testing. Of course, in one sense, all the vulnerabilities in an application are "in the source code," so it must in principle be possible to locate all those vulnerabilities via code review. However, much vulnerability can be discovered more quickly and efficiently using black-box methods.
>
> In most situations, black-box and white-box techniques can complement and enhance each other.

16.3 Participants and Environment

There will be 3 participants testing the application based on the selected test cases from the OWASP Checklist. These participants must have prior knowledge to IT security and preferably some insight on what OWASP is all about. However, they shouldn't have any prior knowledge of the IdeaPlatform and its prototypes. An observer will also be present at all times, writing notes as the tester progresses with the testing. Every session will be video recorded and analyzed afterwards for better and more accurate results.

All tests will be conducted on the same environment. Every tester will be provided with a laptop running the Windows OS and will be using only the Firefox Browser to conduct all tests. The motive for this specific environment is that most computer users are familiar with the Windows OS and the platform itself has been only designed and tested on the Firefox Browser. Due to the platform only being a prototype, the management has decided that testing different browsers wouldn't be feasible.

The tester may use any tool necessary to test, assuming that the setup of the tool does not require much time. One tool which will be provided to the testers is OWASP's WebScarab. This tool allows for users to intercept requests and is very useful when attempting to hack into an application due to its ability to easily change the parameters of the request.

Firefox is one of the most used browsers along with Internet Explorer and Chrome. The Firefox browser has been released in 2002 and is currently on its 20th stable release version. This bowser supports many plugins, also called extensions, which can be very useful when testing a Web Application. Here are some useful extensions:

- Firebug allows for easy analyzing and live editing of HTML, JavaScript and CSS code. Also provides error messages if content fails to load or other errors exist in the code.
- FoxyProxy enables flexible management of the browser's proxy configuration, allowing quick switching, setting of different proxies for different URLs, and so on.
- LiveHTTPHeaders lets you modify requests and responses and replay individual requests.
- PrefBar allows you to enable and disable cookies, allowing quick access control checks, as well as switching between different proxies, clearing the cache, and switching the browser's user agent.
- Wappalyzer uncovers technologies in use on the current page, showing an icon for each one found in the URL bar.
  (Stuyyard, D. and Pinto, M. 2011, 750.)

WebScarab is a web application security testing tool developed by OWASP. This tool is written in Java allowing it to run on cross-platforms. Web Scarab acts as a proxy, intercepting all requests made by the browser, allowing the user to modify any request made and therefore changing the conditions of the request (Wikipedia. 2013.)

Each participant will be briefed with all necessary information before the testing. This phase will give the user time to get to know the prototype before starting to test the application, giving him/her a quick insight on what the application is all about.

The training will include a quick introduction to the platform and the prototype, as well as some time for the tester to familiarize.

16.4 Procedure

A specific procedure will be used to conduct the testing phase, below is a description by Omar Gutierrez (2013, 6), this comes from the usability testing document which was written at the same time as this one.

> Every user will have to follow certain steps to prepare for the two test phases. Some of these steps will be arranged in advance to reduce the tester preparation time.
>
> The participants will make carry on with this testing plan at Aalto's University facilities, Espoo.
>
> A single laptop will be given to every tester in order to access the IdeaClick prototype. In addition, the observers and facilitators are presented to the tester. Their responsibilities during the testing development are also described to testers. The testers' tasks execution will be followed up by the observers situated in the same room. There are no devices that may interfere; the only participants during the testing are the observers and facilitator.
>
> The testers are committed to go through the entire testing process, until they finish all the test cases. There is no room for the tester to leave the testing procedure. Testers are informed of this prior to their start.
>
> The facilitator will give a short description of the web application IdeaClick, including the uses and purpose of it. There are no specific descriptions regarding the IdeaClick button events, design, web page location, or information that possibly could make the testers navigation easier. The facilitator is not entitled to give further information that might enhance or change the outcome of the testing phase.
>
> The Facilitator will handout a printed copy of the testing cases to the testers, the facilitator will ask the tester to read the testing case out loud to ensure that he is going through the entire test cases and making sure that he knows the amount of time that he has in order to complete all test cases. When the tester feels ready, he may begin to execute the series of steps described in

the test cases. Observers ask the tester to speak out his thoughts related to the test case while he executes it. The Facilitator may also ask the tester if there are any questions or doubts related to the testing proceeds.

The observer will write down the process of the tester entering the data, any comment that the tester could make, as well as facial or emotional expression and body language.

After each test is completed, the observer will hand out a blank page to the tester so he can write any comment or thoughts from the testing procedure and its execution. Finally the facilitator will ask to tester to answer an after-test questioner.

The Facilitator will dismiss the tester.

An individual may play the role of Facilitator and Observer during a single session.

Facilitator

- Presents an initial idea of the product to be tested
- Explains the purpose of the testing to the participant
- Assists the tester during the test process
- Provides assistance to tester during the test case process development
- Handout use cases to tester
- Handout after-test questioner to participant
- Knows how to use the IdeaClick prototype

Test Observers

- Only present as an observer
- Takes notes if necessary
- Silent to tester
- Knows how to use the IdeaClick prototype
- Can act as facilitator

Tester

- Execute test cases
- Brief interview after questionnaire
- Write down thoughts regarding IdeaClick interface.

- No prior knowledge of the IdeaClick prototype other than what is provided during the training

17 Test Case Definitions

The test case definitions are based on the OWASP testing guide. It will follow four simple steps, information gathering, authentication, session management and data validation.

17.1 Information Gathering

Information gathering allows the user to identify how the application works. This procedure gives the tester the understanding he needs over the application, before he begins his testing. This is an important step of penetration testing.

17.1.1 Manually explore the site

In this phase the user is in charge of gathering as much information as possible about the application. The user may use any tools he likes to get to know the application. The purpose is for the user to familiarize himself and understand the application and its functionality. When this is done, the user will have a much easier understanding of the actual test cases provided to him.

17.1.2 Identify application entry points

Identifying entry points to the application allows the user to identify areas which should be investigated further. These entry points may give the attacked the possibility of hacking into the application.

This is the point in which the tester gets a good understanding of how the user/browser communicates with the application. HTTP requests made when sending information through the application are easy ways of identifying if the developer may be sending data through the application which was not intended to be seen by the user. These HTTP requests (GET and POST requests) can easily be extracted by using a proxy (such as OWASP's WebScarab) or a browser plug-in.

Requests:

Web requests can be made in two ways, a GET request or a POST request. GET requests data to be transferred from a specified source, while POST submits data to a specified source. The submitted data can then be processed on the server. Post requests are never cached so they are used primarily to send sensitive data such as login credentials.

Important things to remember:

- Identify where GETs are used and where POSTs are used.
- Identify all parameters used in a GET and POST request. Get request parameters can easily be identified in the URL
- Within the POST request, pay special attention to any hidden parameters. Forms may have hidden fields which are sent as well in POST requests. These are not visible to the client but can be identified in the request.
- Identify all the parameters of the query string, some parameters may contain several sub-parameters split up by a dot (.) or comma (,)
- A special note when it comes to identifying multiple parameters in one string or within a POST request is that some or all of the parameters will be needed to execute your attacks. You need to identify all of the parameters (even if encoded or encrypted) and identify which ones are processed by the application. Later sections of the guide will identify how to test these parameters, at this point, just make sure you identify each one of them.
- Identify possible headers that may have been included and are not considered default headers. (OWASP Testing Guide. 2008, 57.)

Responses:

In the responses it is important to look at what data is being transferred back. Sometimes transferred data is stored as cookies, here we need to identify where these cookies are stored. This data can later be modified.

This example shows a GET request that would purchase an item from an online shopping application:

GET Request: https://x.x.x.x/searchitems.php?category=plants&filter=asc

Cookie: SESSIONID=BLDFWAEL76G

Result Expected:

It is important to test all parameters sent, such as: category, filter, and the Cookie. These parameters contain sensitive information. The server should act accordingly when attacks are made.

17.2 Authentication

Authentication is an important step in an application which is supposed to keep any data hidden to a certain group of users. Authentication is the act of establishing something is authentic, meaning a user is confirming his identity to access a certain part of the application. The authentication process consists of verifying the user's digital identity in the application, an example of this procedure is logging into an application. This test identifies how the user is identified on the system and whether it is possible to circumvent this procedure allowing unauthorized access.

17.2.1 Test for authentication bypass

Many applications use login procedures to authenticate that the user has the right to access the requested information. What is important here is that every single request that is made by the user is authenticated; this also means that every site that is requested should be authenticated.

Some applications focus on authenticating the initial pages requested by users, but forget to authenticate hidden pages, which may be identified by an attacker by searching for HTTP requests which may lack this authentication procedure.

In addition to this, it is often possible to bypass authentication measures by tampering with requests and tricking the application into thinking that we're already authenticated. This can be accomplished either by modifying the given URL parameter or by manipulating the form or by counterfeiting sessions.

There are several methods to bypass the authentication schema in use by a web application:

Direct page request (forced browsing), can be achieved if an application only uses access control in the login page If a user directly requests access to a different page, which may not verify the credentials of the user, the attacker may bypass authentication.

Parameter Modification, this is successfully achieved if the application uses fixed values after a successful login, the attacker may modify the parameters, which allow him to gain access. Example: http://www.site.com/page.asp?authenticated=yes

Session ID Prediction Most web applications handle authentication using session identification Tokens. If these sessions ID's are predictable due to them increasing linearly over time, the attacker could gain access by guessing a valid session ID.

SQL Injection is a widely known technique of attacking web applications since it is easy to use and can directly be inserted into the user or password box in the login page of an application. Here is an example if a user would enter an SQL injection script into the password field of a login page: " Select id from Users where userid='100' and password='anything' or 'x' = 'x' "

17.3 Session Management

Each application that communicates with the internet is most likely to use session management to handle interactions between the application/user and the server. Sessions allow authenticated users to remain authenticated and without the necessity of validating their credentials again which keeping a secure environment. An identification token is sent to the user, this token is used to validate each request made to the application. This token is usually called a Session ID or Cookie.

17.3.1 Identify how session management is handled

As stated above, sessions provide continuous authentication of each application without the user having to repeatedly provide credentials for authentication, this is called session management. Session Management eases the users' interactions with the application, but leaves it more vulnerable to exploits by attackers. Here it is necessary to

identify whether the Session ID's and cookies of the application are securely handled and cannot be hijacked by any attacker.

Cookies and Session ID's are sent to the client once he has sent his credentials and authenticated himself, after this the client will sent this token to the server with each request made until the token expires or is destroyed by the application.

HTTP is a stateless protocol, this means that the server treats each request as an individual and is in no way related to any other request. Sessions allow the client to provide a state to a stateless protocol. This token is a unique identifier for the user and may be renewed several times during a single session. (OWASP Testing Guide. 2008, 146.)

Any web application where the user has data attached to his login, such as ideas in the ideaClick prototype, requires session management. The website the user is interacting with needs to continuously keep track of the items the user has created, for this the application must keep track of the identity of the user.

Unfortunately these cookies are vulnerable to attacks if not issued and managed correctly. Attackers may try to hijack these sessions by stealing another users session ID. In most cases the attacker will do the following steps to try to hijack a session:

- cookie collection: collecting sufficient cookies which can later be analyzed
- cookie reverse engineering: analyzing these cookie samples to identify the generation algorithm
- cookie manipulation: forging a valid cookie in order to perform an attack on the application. (OWASP .2013.)

A different way of hijacking an application is by overflowing a cookie, this is done by overflowing the memory on the server, to interfere with its intended actions and attempt to inject malicious code during the process. (OWASP. 2013.)

17.4 Data Validation Testing

Bad input validation leads to the most common security vulnerabilities in web applications. Developers must ensure that all input by the user is validated to ensure that no

malicious code is being executed. Many of these weaknesses are exploited by SQL Injection, code injection and cross site scripting. Here the tester must search for exploits in form inputs, by using different methods of injecting malicious code, such as SQL and Code injection.

17.4.1 SQL Injection

SQL Injection is a widely known technique of attacking web applications since it is easy to use and can directly be inserted into the user or password box in the login page of an application. Here is an example if a user would enter an SQL injection script into the password field of a login page: " Select id from Users where userid='100' and password='anything' or 'x' = 'x' ".

Here the tester must find out if it is possible to gain knowledge of secure data in the database by retrieving important data. To make things easier for the tester, he will receive information of the type of SQL server running and what engine the Database is using, this will allow the user to rule out unnecessary attempts.

17.4.2 Code Injection

In code injection the attacker attempts to insert a code that will not necessarily be executed immediately, but may be executed by the server when, e.g. a page request is made. This is particularly effective on forums. If the attacker manages to successfully insert malicious code into a comment, this code is being executed when every visitor sees that particular comment, making every visitor vulnerable without even realizing it. This can allow the hacker to retrieve e.g. the Session ID of the user and perform session hijacking, by impersonating the other user.

17.5 Excluded

There exist many more ways to test an application for security vulnerabilities. This plan only focuses on any security vulnerabilities that are directly related to the IdeaClick prototype and have not yet been identified prior to this test plan. Many security risks may exist in the back end of the application, the so called "core" of the application. These out of scope tests will be excluded.

18 Test Cases

18.1 Task 1, <Information Gathering – Manually explore the site>

Goal/Output:	To gather as much information about the application by manually exploring the site
Inputs:	Credentials username: test@user.com password: testuser
Assumptions	The user will have a good understanding of the application and may identify possible vulnerabilities.
Steps:	1. Open the application under www.ideaplatform.net/ideaclick 2. Start exploring
Time for expert:	15 min
Instructions for user:	No tools should be used during this phase. Only the IdeaClick prototype needs to be tested. All other interfaces, as well as the backbone are not part of this test. The login page should also be explored.
Notes:	This is a way to understand the application. Not to break it.

18.2 Task 2, <Information Gathering – Identify application entry points>

Goal/Output:	To gather as much information about the application
Inputs:	Credentials username: test@user.com password: testuser
Assumptions	The user will have a good understanding of the application and may identify possible vulnerabilities that allow him unauthorized access to the application.

Steps:	1. Open the application under www.ideaplatform.net/ideaclick 2. Identify entry points to the application
Time for expert:	5 min
Instructions for user:	Any tools the tester wishes can be used. Only the IdeaClick prototype needs to be tested. All other interfaces, as well as the backbone are not part of this test. The login page should also be explored.
Notes:	This is a way to understand the application. Not to break it.

18.3 Task 3, <Test for authentication bypass>

Goal/Output:	Identify if there is a possibility of bypassing the authentication of the application.
Inputs:	-
Assumptions	The user requires valid credentials to log in to the application
Steps:	1. Open the application under www.ideaplatform.net/ideaclick 2. Log out of the application if necessary 3. Attempt to bypass authentication Possible methods: a. Direct page request b. Parameter Modification c. Session ID Prediction

	d. SQL Injection
Time for expert:	5 min
Instructions for user:	Any tools the tester wishes can be used.
Notes:	-

18.4 Task 4, <Attempt SQL Injection>

Goal/Output:	Identify if there is a possibility of injecting SQL code into forms and successfully retrieve sensitive information.
Inputs:	Credentials username: test@user.com password: testuser
Assumptions	The attacker is unable to retrieve any sensitive information. The page acts as it should and prevents the attacker from breaking in.
Steps:	1. Open the application under www.ideaplatform.net/ideaclick 2. Log on to the application 3. Attempt to insert SQL code into any input form available
Time for expert:	5 min
Instructions for user:	Attempt to insert malicious SQL code and retrieve sensitive information
Notes:	-

18.5 Task 5, <Attempt Code Injection>

Goal/Output:	Identify if there is a possibility of injecting Code into forms which may later be executed by the server.
Inputs:	Credentials username: test@user.com password: testuser
Assumptions	The application successfully parses the information and renders is harmless.
Steps:	1. Open the application under www.ideaplatform.net/ideaclick 2. Log on to the application 3. Attempt to insert malicious Code into any input form available
Time for expert:	5 min
Instructions for user:	Attempt to insert malicious code and retrieve sensitive information
Notes:	-

18.6 Task 6, <Information Gathering – Identify how Session Management is handled>

Goal/Output:	To gather as much information about the application
Inputs:	Credentials username: test@user.com password: testuser
Assumptions	The user will have a good understanding of the application and may identify possible vulnerabilities that would allow him to possibly hijack someone else's session.

Steps:	1. Open the application under www.ideaplatform.net/ideaclick 2. Identify how the application manages its sessions.
Time for expert:	5 min
Instructions for user:	Any tools the tester wishes can be used. Only the IdeaClick prototype needs to be tested. All other interfaces, as well as the backbone are not part of this test. The login page should also be explored.
Notes:	This is a way to understand the application. Not to break it.

18.7 Task 7, <Session Hijacking>

Goal/Output:	Identify if the tester is able to hijack a session and therefore impersonate another user.
Inputs:	-
Assumptions	The tester is unable to hijack the session and requires valid credentials to access the application
Steps:	1. Open the application under www.ideaplatform.net/ideaclick 2. Log out of the application if necessary 3. Attempt to hijack a session. Possible methods: a. Session ID manipulation
Time for expert:	10 min
Instructions for	Two additional users are logged into the application during this

user:	time. The tester can use any tools necessary.
Notes:	-

19 Results and Conclusion

The Goal of this testing plan is to determine the most important risks the IdeaClick prototype may have. These risks will be documented and analyzed further for assessment of likelihood of attack and severity if the attack is achieved. The overall severity will give a better understanding on where the application needs security improvement.

19.1 Reporting Results

The most important part of testing is the results that are determined by the tests. Therefore it is vital that the results are analyzed and reported as well as possible.

The results of the testers will be thoroughly analyzed and an estimation of the risk of each attack will be made. These risks will be separated into two parts, the likelihood of attack and the severity this attack may have on the application.

The risk likelihood and impact will be documented based on the OWASP Guide of valuing risks. A simple formula will calculate the risks "Risk = Likelihood * Impact".

Below is a table defining how to overall risk rating will be calculated

Overall Risk Severity				
Impact	High	Medium	High	Critical
	Medium	Low	Medium	High
	Low	Note	Low	Medium
		Low	Medium	High
	Likelihood			

19.2 Test Results

As any web application, the results clearly indicate vulnerabilities that may be exploited by users. Some of these issues are quite minimal and may require only little attention, while others may result in quite high risks for the application. Some of these issues include imitating another user as well as retrieving unauthorized data.

19.2.1 Executive Summary

There is clearly one issue which is easily visible and attracts a lot of attention. While many risks in the application are quite small, the ability to hijack another user's session is quite severe. It is vital for the system to be checked and the necessary repairs to be made.

The ability to retrieve hidden objects should be checked as well and fixed.

19.2.2 Technical Management Overview

The ability to hijack another session may easily be fixed by creating a user identification token which may not easily be guessed, such as a random set of longer digits.

Furthermore, additional authentication in the code is required to verify that a user is not able to retrieve hidden objects.

With the Overall Risk Severity chart we can split the results into three categories, starting with the most severe.

Luckily no test result has been identified as a Critical

High

Something that should immediately be addressed is the way a User's identification is stored on the clients' computer. Even though the system is using two identifiers, only one is in most cases used as real identification to the server. These User tokens are very simple to guess. Even though hijacking a session may sound as a very big red flag to a tester, this vulnerability has not been classified as critical due to the likelihood of at-

tacker identifying this flaw. Nevertheless this is still of very high importance and should be resolved as soon as possible.

Medium

An additional flaw in the system is the ability to retrieve objects based on their unique but simple identifier, this doesn't allow the attacker to do any harm to the system, but it does allow him to view even hidden objects, or objects that may have already been removed from the system and are kept as a backup. This flaw is classified as medium, since this may only happen while a user is logged in to the system.

Low

Testers have also identified that they are not automatically thrown out of the system after a longer period of inactivity. Although all users know to log out of a system after they have finished, it is not always the case. Should a user forget to close his browser window in an internet café, the next user could freely access the system even after a long period of inactivity on the computer.

19.2.3 Assessment Findings

This section provides more descriptive results, including methods on how to possibly handle and fix the issues found.

User Identification

The User identification is stored with a very simple Identifier as a cookie, userID = 2. This cookie is written by the JavaScript code. This can easily be fixed by changed the identification tokens to more complex digits. Long random strings can easily make the application more secure. One example would be to use encryption of the tokens with the addition of a SALT (key only known by the server to decrypt the string).

Encryption example:

Identifier of the User, e.g. 2

Salt: ideaclick

MD5 Encrypted identifier to be sent to client: d09eec78f709f77bb684a52de87769af

Object retrieval

So called InnoObjects may be retrieved even if they have already been removed from the application by typing the unique Identifier of the Object. These identifiers are very easy to guess. E.g. InnoObjectID = 176.

The same procedure as in the User token is recommended to drastically decrease the possibility of hijacking a session.

In addition, not only the userID should be used to validate the authenticity of the user. The application uses a PHP session variable which may also be used to ensure the users credentials are valid.

Session expiration

Even though this is not a life threatening matter, expiring a session could make some users life more easy. This is especially the case if they tend to leave their computer unattended for long periods of time without locking the computer-

Due to the sessions being handled by the PHP code, a simple line of code when the session is created will suffice to ensure someone attempting to access the system after a while is unable to do so.

PHP code to be added where "minutes_to_timeout" is the amount of minutes desired before the session times out after inactivity: ini_set('session.gc-maxlifetime', 60*minutes_to_timeout);

19.3 Conclusion

The IdeaClick prototype does have some security flaws that need to be looked at and fixed by a developer. Eventhough these issues do have some implications on the prototype, they are easily fixed. We should remember that IdeaClick is just a prototype and

not a tool. If these issues are fixed, this prototype is well on its way of becoming a platform.

References

Stuyyard, D. and Pinto, M. 2011. The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, Wile Publishing, Inc. Indianapolis, Indiana.

Gutierrez, O. 2013. IdeaClick Usability Testing Plan

OWASP Testing Guide. 2008.
URL:https://www.google.fi/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&ved=0CEQQFjAC&url=https%3A%2F%2Fwww.owasp.org%2Fimages%2F5%2F56%2FOWASP_Testing_Guide_v3.pdf&ei=_8d3Uf_pCsrEsgaJvYGoDg&usg=AFQjCNFL2d3b66xLQA66GarlswyumhXXSQ&sig2=KRjzXqs4nvdQdBcVlMQbyg&bvm=bv.45580626,d.Yms

OWASP. 2013. About The Open Web Application Security Project.
URL:https://www.owasp.org/index.php/About_The_Open_Web_Application_Security_Project

OWASP. 2013. Testing for Session Management Schema.
URL:https://www.owasp.org/index.php/Testing_for_Session_Management_Schema_(OWASP-SM-001)

Techopedia. Software Security.
URL: http://www.techopedia.com/definition/24866/software-security
Accessed: 5.12.2012

Wikipedia. 2013. WebScarab.
URL: http://en.wikipedia.org/wiki/WebScarab
Accessed: 9.2.2013

Appendices

Appendix 1 – Test Tasks Paper

Task: The task to be performed

Test performed: The user defined test. This should specify what the user has chosen to test

Finding: Description of the outcome of the test

Difficulty: Rating of how difficult the test was based on how long the test takes. 1 = Very easy. 5 = Very difficult test to perform

Severity: Rating of how severe the results were. 1 = No result found, 5 = Large quantity of sensitive data retrieved

Task	Test performed	Finding	Difficulty (1-5)	Severity (1-5)
Task 1 Information Gathering – Manually explore the site	Example: Identify login form	Uses encoded and secured transfer		
	Successful logout procedure	The session is terminated and user logged out		
	Insert new Idea, encoding	The data is encoded before transferring		
Task 2 Information Gathering – Identify application entry points	Insert new Idea, User Identification	The data is sent with a Session ID, identifying the user		
	Identify if hidden/removed objects can be retrieved	If the correct ID of the object is available, even a hidden object can be retrieved		

Task 3 Test for authentication bypass	Test if user can be imitated without valid credentials	Without logging in to the system, a user cannot be imitated		
	Test if a user can be imitated with valid credentials of another user	While logged in, another user can be imitated		
Task 4 Attempt SQL Injection	Attempt SQL injection on login screen	No successful login could be attempted		
	Attempt SQL injection while inserting Idea	No data could be retrieved and no data was harmed with SQL injection		
	Attempt SQL injection while retrieving selected objects	Additional objects can be retrieved if ID is known or guessed. SQL injection caused no harm		
Task 5 Attempt	Attempt Code injection on login	Authentication is denied, new credentials are re-		

Code Injection	screen	quested		
	Attempt Code injection while inserting Idea	Code is escaped and normal Text is visible without any code changes		
	Attempt Code injection while retrieving selected objects	Hidden objects may be retrieved by directly requesting the objects ID		
Task 6 Information Gathering – Identify how Session Management is handled	Identify if cookies are used	Cookies store Session ID and user ID, user ID's are predictable		
	Identify if cookies expire	Cookies expire after 15 minutes		
	Identify id Session ID expires	Session ID does not expire, at least not in a short time. User must log out of browser closed for Session to be closed		
Task 7 Session Hijacking	Identify if Session can be hijacked via Cookie	Cookie is stored with expiration timer. Sessions are handled by PHP Server		
	Identify if Session can be hi-	No random Session ID allowed access		

	jacked via random Session ID			
	Identify if Session can be hijacked by calculating Session ID	Session ID's are not predictable. Session ID is random and cannot be easily guessed		