

MIDDLE EAST TECHNICAL UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT

TEST MANAGEMENT PLAN

AQUT

Anatolian QUalified Technology

Mehmet Ali Özkeskin 1395375

Mustafa Çöçelli 1394840

Uğur Irmak 1347558

Şevket Dokgöz 1347384

INDEX

1. INTRODUCTION.....	3
1.1 Goals and Objectives.....	3
2. TESTING STRATEGY.....	4
2.1. Unit Testing.....	4
2.2. Integration Testing.....	5
2.3. System Testing	5
2.4. Performance Testing.....	5
3. TESTING PROCEDURE.....	5
3.1. Unit Testing.....	5
3.1.1 Messaging Unit.....	5
3.1.2 Interface Unit.....	6
3.1.3 Tag Unit.....	6
3.1.4 Note Unit.....	7
3.1.5 Database Unit.....	7
3.2. Integration Testing.....	8
3.3. System Testing	9
3.4. Performance Testing	9

1. INTRODUCTION

This section gives a general overview of the test specification for an extension integrated into Mozilla Firefox, Xilent.

1.1 Goals and Objectives

Xilent software aims to socialize the web users without any extra effort while they are surfing on the internet.

The software is basically a Mozilla Firefox extension which adds a left side bar to the browser; serving web 2.0 technologies to the visited pages which are 'note leaving' and 'tagging', plus providing an instant messaging environment among the page visitors basically.

While the functionalities of Xilent are so persuasive and effective the testing mechanism must also guarantee to achieve those goals by considering all the cases in an efficient environment with effective tests. In this challenging process we will ensure the software will function correctly and deliver error-free and fast serving. The purpose of the testing process is to satisfy Xilent's goals with the ability of countering any difficulties that may impact the development and future performance of the software.

In our testing strategy we will focus deeply on the three major parts of the software which are 'note leaving', 'tagging', 'instant messaging' and besides the other software which combines those modules and serves the related interfaces to the user.

Xilent testing strategy will examine following cases:

For 'note leaving' and 'tagging':

- functioning on any kind of webpage
- permission settings of the leavers
- size, position and color settings of the leavers
- updating and deleting by the leavers
- viewing settings of the viewers
- prohibited adjustments of viewers
- User page control

- examination of high load
- running time

For 'instant messaging':

- functioning on any kind of webpage
- page-visitor control
- message sending performance and cases
- message receiving performance
- chatting in high load
- running time

Besides those; the friendship mechanism, registering, logging in, site discovery, dynamic content of side-bar and user pages will be tested with all possible cases as well.

2. TESTING STRATEGY

In this section, we'll give a general explanation of the testing procedures we'll use in Xilent's testing. We mostly tested the product during development, but also as a final step in the project we will do a general testing. In the testing of the Xilent, we will follow a way that includes several steps. These are, in the order of the completion, Unit Testing, Integration Testing, System Testing and Performance Testing. In the next section, we will explain the testing procedures for the each components of the Xilent.

2.1 Unit Testing

Unit testing is the testing of the each individual components of the project separately to see whether application will suffer from errors arising from these components. This testing should be handled before the integration testing. It should be certain that errors arising in integration testing are not about individual components. Moreover, there will be one scenario to test each component. It is a white-box type of testing.

2.2 Integration Testing

After the unit testing of the each individual component is completed, it is time for the integration testing to see whether all components are working together properly. This type of testing is crucial to see whether the product is working bodily and answer the specifications of the Final Design Report.

2.3 System Testing

It is done to see whether the system is compatible with the final version of the product. Since it is a black-box type of testing, testing should be aware of the infrastructure of the product.

2.4 Performance Testing

Performance Testing is the testing conducted to evaluate the compliance of a system or component with specified performance requirements. In this part, it is aimed to test the application with large number of users, note and tag in the system.

3. TESTING PROCEDURE

In the previous section, Testing Strategy, the testing procedures that we'll apply were shortly described. In this section, you'll see which testing procedure will be used in which function/ module of our product.

3.1 Unit Testing

3.1.1 Messaging Unit

In messaging unit, there will be a jabber server handling with instant messaging between users. With this testing, it is good chance to see whether jabber server working properly. The purposes of the messaging unit are:

- To control whether the instant messaging between specified clients is succeed.
- To control whether a user could communicate more than one user without trouble

The aim of the messaging testing is:

- Providing uninterrupted and proper messaging service to users.

3.1.2 Interface Unit

In interface unit, there is a web server handling the display of the product. With this testing, it is good chance to see whether web server display the WebPages of the application properly. The purposes of the messaging unit are:

- To control the design of the WebPages are displayed in user's computer properly.
- To control whether web server could take the request of the user from interface of the application.
- To control whether interface unit direct the user correct pages after the actions of the user like login, signup.
- To control whether a user interacts with the interface unit.
- To control whether the interface of the application serves simplicity to users.
- To control whether the components in the interface are displayed by considering authorization control.

The aims of the interface unit testing are:

- Web server displays the WebPages as they are designed.
- Web server responds the request coming from users.
- A user could not access the components of the application he/she is not authorized to access.

3.1.3 Tag Unit

In tag unit, there are web server and database server to leave the tags the intended position of the WebPages and display them when visiting WebPages. The purposes of the tag unit are:

- To control whether the intended images and text on WebPages are tagged.
- To control whether the display of the WebPages deteriorate after tagging an image or text.

- To control whether the authorization of the tagged images and text can be set up.
- To control whether the tagged images and text are displayed in the users' computer

The aims of the tag unit are:

- Tag the image and text on WebPages.
- No deterioration after tagging in WebPages.
- Authorization control when leaving and viewing tagged information.

3.1.4 Note Unit

In note unit, there are web server and database server to leave notes the intended position of the WebPages and display them when visiting WebPages. The purposes of the note unit are:

- To control whether notes are left in WebPages.
- To control whether the display of the WebPages deteriorate after leaving notes
- To control whether the authorization of the left notes can be set up.
- To control whether the left notes are displayed in the users' computer

The aims of the tag unit are:

- Leave notes to WebPages.
- No deterioration after leaving notes in WebPages.
- Authorization control when leaving and viewing notes.

3.1.5 Database Unit

In database unit, there is MySQL server to store the data about the users, notes and tags. By means of the MySQL server, we are able to keep the data about the users, notes and tags in this server.

- To control whether the information about the users, notes and tags are stored in database server correctly.
- To control the whether the information about the users, notes and tags are loaded from database server correctly.
- To control whether irregular actions are prevented in database server like duplicate usernames.

The aims of the database unit testing are:

- The data about the users, tags and notes are stored to specified database tables in the database server.
- The data about the users, tags and notes are loaded from specified database tables in the database server.
- No permission for irregular actions that user can attempt.

3.2. Integration Testing

After testing all units/modules of the Xilent and seeing that all parts works as we expected, we will move to the integration testing. At first, we combined all modules, which are tested before. And we got an integrated product. After this process is passed, we will start integration testing. The purpose of the integration testing can be listed as in the below:

- To see if the individual units function correctly as a complete system.
- To see if the integrated units satisfy our design specifications.
- To see if any user experience any problems while doing his tasks on the news server from the beginning to the end.

Integration test is one of the most important phases during testing progress. Because there are different modules that access to the same server and database in coordination and system should be stable at any instant. For example, in Xilent, Messaging module should be adapted to the system comfortably since all users are in coordination with XMPP server by the help of the server and site discovery, online user, chat features are in relation with this part. So there should be no inconsistency in this module to work together with other features. Moreover, in the Note and Tag modules, every change in the tag/note content and theirs' positional change is reflected to database. By integration testing, stability is provided by discovering any collision between these two modules. If one of the modules is failed to satisfy the stability the related parts of the system are reviewed for specifying the reason of this fact. Then after the specification of the reason, the problem is solved by the interaction between the coders of the problematic modules.

3.3. System Testing

System testing is done for evaluating the system according to its functional requirements. It is completely a black-box method, so we will check inputs and outputs of the units. The purposes of the system testing are:

- To see if our system gives correct outputs to the inputs which satisfy system specifications
- To see if functional requirements are satisfied in the current status of the product

In Xilent, we can look for tag/note permission and viewing part for this testing. In the viewing part, user select an option of all/my friends/just mine in the side bar. The tag/note mechanism filters the user choices and shows them to user. In permission part, user who makes tagging/ leaving note selects the permission of his/her tag/note and this selection is taken into consideration for privacy issues. Moreover, other example to system testing, when user changes his/her current website, this change reflected on the side bar of the other users.

3.4. Performance Testing

After the system testing is completed successfully, it will guarantee that our application responds the specification of Final Design Report. Now, it is time for considering the efficiency issues about the application. The purposes of the performance testing are following:

- To control whether the application handles the request coming from user on time
- To control whether the application fail when lots of user enter the system.
- To control whether the application make service to users for long time without a trouble.

The aims of the performance testing are:

- Every request will be handled on time without any trouble.
- The system will have the capability of serving lots of user without any trouble.