MIDDLE EAST TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

CENG491 FALL 2006

SENIOR DESIGN PROJECT

TEST SPECIFICATION REPORT

MULTIWAY

by

SANZATU YAZILIM

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Objectives

Our project Multiway is a message exchange system being composed of different servers which respond to users from different access sources and a core providing interaction between them. So, there are many different modules to be implemented in a limited time interval. Not only we have to implement the individual modules, but also we have to make all the modules work in accordance to form the whole system. Developing a software project in a limited time is a tough work and prone to be defected. At this point testing becomes one of the most important issues to detect any disorder from the beginning and correct those disorders in time.

We will review and check our modules to reach the following objectives:

- ✓ Demonstrating that our product is error-free,
- ✓ Finding errors and correcting them in time,
- ✓ Ensuring that the customer requirements are met,
- ✓ Ensuring that the product provides all of the necessary functionalities,
- ✓ Ensuring the performance of the product.

In the progress of our project Multiway, the testing phase is the last important step before our product goes to the customer. By applying our testing plan on our project, we will achieve the above goals and produce a product with high quality.

1.2 Statement of Scope

Actually the testing is present in every phase of implementation, that is, every member is responsible from the code of the module she implemented and tests whether any error or bug exists in her code. So, the first tester of every code is the person who generated it. After the codes are combined to build the whole system, we will check whether the system works correctly together with all different modules. These constitute unit and integration test phases of our testing plan. These phases and also the validation

and higher order tests we will apply are explained under the corresponding section of the document in detail.

At the end of the document, we also mention the tools we will use throughout testing phases and explain them clearly. In addition to those, the duties of each member in testing process are clarified and the schedule for testing is given.

1.3 Major Constraints

1.3.1 Time

Since our final release is approximately one month later from now, time is the main constraint for us. We created a schedule for the testing phase which is given in the last section. By this way, we provide the deadlines of each specific work beforehand. If we manage to obey the deadlines according to schedule, our project Multiway is released after proper testing and in time.

1.3.2 Staff

The project is being developed by a team of five people. Since, these five people are working on both development and testing, staff is one of the major constraints for the test.

2. TESTING STRATEGY

2.1 Unit Testing

Our system is composed of lots of servers interacting with each other and each of these servers is implemented as a separate module. According to our unit testing strategy, every developer should test the module that she has implemented. We build test scenarios and the implementer(s) of each module has to apply them to our Web module, Newsreader module, Mail module, RSS module and System Administration module in both black-box and white-box manner.

While applying unit tests, we make sure that testing will be useful and it is not time consuming. We also make sure no bug is left for each module after unit testing is done

because it would be very difficult to fix it in subsequent steps of testing job due to high interactivity of our modules with each other.

We determined some milestones for each module to be finished. They will be tested after their implementation is complete. We already applied unit testing to some of our modules which are finished, namely Web, RSS and System Administration Module. However in some cases, in order to test some modules we have to use some other modules. This is mostly the case for Newsreader module. Because we use INN's file system to keep the messages, Newsreader module must have been implemented before all other modules to be tested.

We apply white box testing to our modules as soon as their implementation is over. We are expecting that the implementation details will not have been forgotten, so testing will be more efficient. Furthermore, we will have realized errors before they propagate to later phases of implementation, since they will require greater amount of configuration management afterwards. As our white box testing strategy, we try to monitor the module during execution and check if the module runs as expected.

Black box testing is used as a complementary strategy since it is useful in discovering unexpected behavior rapidly. For black-box testing, we are going to observe the input that is provided to the module and the output or the effect produced as a result. General inputs and outputs of the modules are presented below for each module.

- *Web Access Module:*

For this module, the inputs will be user login information, user commands (such as post/read articles), subjects to be searched and the XML file seeing option. The outputs of this module should be the requested newsgroups and articles which are taken from NNTP server, the search results and XML files generated by RSS Module. The kept log info is the effect produced as a result.

- *Newsreader Access Module:*

For this module, the inputs will be user login and general information and the posted messages to the NNTP server file system. Newsgroups that user has read/write permission to and their messages will be sent to user as an output.

- *Mail Transfer Module:*

For this module, the inputs will be the emails coming from users and the posted messages in the NNTP server file system. The outputs of this module should be the messages which were converted from coming emails. Sending emails to users will be the effect produced as a result.

- *RSS Module:*

For this module, the inputs will be the articles in NNTP server file system and users' comments on RSS feeds. The outputs will be the generated XML files and messages to be kept in the file system.

- *System Management Module:*

All the admin configuration commands will be the input of this module. The changes in the configuration settings and in the database will be the effect produced as a result.

2.2 Integration Testing

When we are confident that the all individual modules are working well, we will begin the integration test of the modules. There may be cases such that the module works correctly when it is independent from the system but it may not produce the expected results when it is integrated to the system. To make sure that all of the modules work together correctly we have to define a testing strategy for the integration of modules.

In our project Multiway, the modules are not independent from each other. They have to interact properly to form the whole system. For example, e-mail news which is sent to the system via SMTP server implemented in our mail module should be kept in our newsreader module's file system or messages coming via the web would be sent to users as email. Therefore, integration testing is the most important step of our testing strategy.

In integration test we are using bottom up modeling. The reason why we chose bottom up testing is that it is easy to combine the little pieces. Modules that have been tested by unit testing will be added to the system one by one and it will be tested whether whole system works correctly after the addition of the new module. Furthermore, we will

be testing whether the added module still works correctly and does its job in accordance with the already integrated and tested ones.

We already tested Web, RSS and Newsreader modules' interaction for the implemented parts. We continuously integrate newly implemented parts.

2.3 Validation Testing

We also include validation testing in our test procedures in order to understand whether our product meets the requirements and complies with the expectations of the customers. We have divided the validation test strategy and procedure into two subgroups as the following.

2.3.1 Requirements Validation

Requirements validation tests will be held in a black-box manner. In this type of testing, we will be making use of our analysis report that we have generated according to the customers' expectations and requirements. In our report, we have mentioned the functionalities of the system as follows:

- ✓ Access to the system platform via HTTP and NNTP.
- ✓ Automatic email posting from system to system members.
- ✓ System members being capable of posting messages to the groups via sending email.
- ✓ RSS feed support.
- ✓ Spam filtering for received messages and received emails.
- ✓ Search functionality among the messages.
- ✓ System authentication and high security.

During the requirements validation, we will be testing whether all of the above functionalities are provided by Multiway.

Until now, we were able to test the validity of the requirements which were included in implemented parts of some modules. In a one-month-time, we will be completing the implementation of Multiway and testing the whole system in terms of requirements validation.

2.3.2 Design Validation

We aim implementing Multiway nearly fully consistent with its final design. For this purpose, during the implementation of each module and class we make use of our final design report.

In our final design report, we explained data design and interface design of our system in detail. We make sure that after the implementation, the data flow of the system and the interfaces supplied to the users via different access sources are consistent with the ones at the step of final design.

Until now, the implemented modules have been validated in terms of design. We will also be testing the design validation of the newly implemented modules in the following times.

2.4 Higher Order Testing

Besides unit, integration and validation testing, we will perform some higher order testing in order to make our system a bug-free and high quality one. Since our product will be accesses by lots of different users from many different sources at the same time, performance and stress tests gain more importance.

2.4.1 Performance Testing

Our system may have high number of users. As a consequence, the performance issues will be considerable. Thus, by the performance tests, it is aimed to discover how the system will respond to the user needs. First, performance tests will be applied to each module separately. After the integration of the modules is completed, whole system will also be tested. By these performance tests, it is aimed to find out if extra measures are needed to enhance the performance.

2.4.2 Stress Testing

We will test our program for abnormal situations. When there are too many tasks to be completed, the program should not crash and handle the situation. We will send so many requests concurrently from so many users, and the program should not be locked.

We will check whether our system properly serve many users from all the access sources, namely web, newsreaders, email and rss for reading, posting and also searching messages at the same time.

2.4.3 Alpha and Beta Testing

Every time we add a new feature to our software package, we run all SMTP, NNTP and HTTP servers and other applications of search, rss and administration to see the results. Moreover, we test the reliability and consistency with other features of this new feature. Consequently, our alpha testing is continued as we continue to add new features to our system. Our alpha testing schema includes testing the code using white box testing. Also, we plan to perform additional inspection by using black box testing.

However, as we intend to come up with a reliable software package we have to a second test, namely beta test, to see whether we have any bug or inconsistency which have not recognized before. For that purpose, we plan to release our product to a limited audience outside of our company so that further testing can ensure the product has few faults or bugs. With the feedback from our audience, we can both correct the faults and improve the program performance.

3 TESTING SCENARIO

For testing our system as a whole, we prepared a test scenario including functionalities of all modules. By the help of this scenario, we will identify missing parts of our system according to its required functionalities and usability.

3.1 Admin Test Scenario Description

Mr. Brown, who is the manager of Zengin Holding, wants a system which enables safe communication between members of the company and Multiway software package is installed to their servers. He takes the responsibility of being the administrator of the system. He enters the user list information to the system and the system automatically generates a unique password and account for each user. Mr. Brown opens a group named “Departments” and subgroups for different departments. While arranging the settings of the subgroups, he decides that some of these subgroups should have longer post duration

times while the messages in other groups need not be seen more than one day. Also some groups would need RSS feed while others not. In addition to these some departments were working on special topics and the posts in these departments' groups should have not been accessible to other departments' staff. Mr. Brown took all of these into account while opening the subgroups and regulated the settings of the groups in that way. At the end he recognized that he had made created unnecessary groups and had written some names of the staff wrong, so he removed some groups and edited some user information.

In those days, a subgroup of 5 people from Human Resources department was working on a special research and the administrator is asked to open a private subgroup to which only these 5 people could sign in. He fulfilled this request by using system admin interface of Multiway.

3.2 User Test Scenario Description

Mr. Junior Brown was working in Sales Department. After his company began using Multiway Communication System software, he became a member of the system automatically. He selected the groups he wanted to be visible in his interface. Now he is able to communicate with other company members, follow latest news about their group project, read some posts by his RSS reader. He accesses the system via web browsers or application programs such as Thunderbird, Outlook. He posts messages to the groups either with these ways or by just sending it as an e-mail.

4 TESTING TOOLS AND ENVIRONMENT

Multiway is reached by many methods, so it is tested by using a tool for each method as follows:

- ✓ Thunderbird or tin for testing NNTP server side of our system.
- ✓ An RSS reader software for testing working of RSS feeds.
- ✓ A web browser like Mozilla Firefox for testing web side of our system.
- ✓ Thunderbird or Microsoft Outlook for testing cases about e-mail side of system.

These tools can be found and used in different environments like Linux or Windows platform.

5 TEST RESOURCES AND STAFFING

Since Multiway can be also accessed by remote machines, we will use any machine having the testing tools as a resource.

The staffing strategy of our group for testing is as follows:

Instead of having a specialized tester in our team, we decided to make all the members to participate in the testing process and test the modules that they have implemented. Whole system testing is handled by all of the members.

6 TEST SCHEDULE

Task	Handled by	Start	Deadline
Test plan delivery	Safiye, Aslı	01.05.2007	06.05.2007
Unit testing of Web Access Module	Başak, Aslı	30.04.2007	20.05.2007
Unit Testing of RSS Module	Çiğdem, Efser, Safiye	30.04.2007	20.05.2007
Unit Testing of Newsreader Access Module	Başak, Efser	30.04.2007	25.05.2007
Unit Testing of System Management Module	Çiğdem, Efser, Safiye	30.04.2007	20.05.2007
Unit Testing of Mail Module	Çiğdem, Safiye, Aslı	02.05.2007	25.05.2007
Integration Testing	All group members	25.05.2007	27.05.2007
Validation Testing	All group members	27.05.2007	30.05.2007
Performance Testing	All group members	30.05.2007	02.06.2007
Stress Testing	All group members	02.06.2007	04.06.2007
Alpha Testing	All group members	04.06.2007	06.06.2007
Beta Testing	Audience	06.06.2007	08.06.2007