**Middle East Technical University**

**Department of Computer Engineering**

*`A Unified News Exchange Server `*

Test Specification Plan

Goncagül DEMİRDİZEN

Hilal KARAMAN

Ali Anıl SINACI

Ferhat ŞAHİNKAYA

# "NewsAgent"

by

i$T€ Yazılım

Spring, 2006

## TABLE OF CONTENTS

# 1 Introduction

## *1.1 Goals and Objectives*

NewsAgent is a Unified News Exchange Server with different Access methods which has to be developed and implemented in a limited period of time. In this period, we have to implement and integrate different modules which are together functioning properly and consistently. Errors and bugs in software projects especially in the projects implemented in a limited time are inevitable. However, our main goal is to release an error-free and correctly functioning final product meeting the project requirements. As a result of this, testing is one of the most important parts of our process. Testing is always a challenging process, however it helps to deliver usable systems. We believe that by following our testing plan and strategy, we will be able to detect and correct the bugs in our product and satisfy the requirements and needs.

## *1.2 Scope of Test Specification Plan*

Testing is a continuous period in a software development process from beginning to the end and in this report we examine the testing plan and strategy for our project NewsAgent. This document includes some of the testing procedures we have applied until now and mainly focuses on the testing plan and tasks to be applied in approximately one month period after the completion of the implementation. We also present some test scenarios for main functionalities of our project and explain the responsibilities of every team member in testing procedure. We have prepared a testing plan according to our project specifications and at the end of the document we present a testing schedule.

## *1.3 Major Constraints*

### 1.3.1 Time

We have to complete our project implementation with testing, debugging and removing the bugs and errors in a month's time and therefore, time is one of the main constraints for testing process of NewsAgent. In order to complete the project in time after a successful testing period, we have to follow our testing Schedule.

### 1.3.2 Staff

Our project group consists of 4 people who are responsible for both development and testing of the product. Therefore, staff is also a major constraint for testing process of NewsAgent.

### 1.3.3 Platform Independency

From the beginning of the project we aimed our product to be platform independent which means NewsAgent should run at both Windows and Linux platforms. In order to test this feature we plan to apply tests on both of the platforms and therefore, platform independency is another constraint for testing process of NewsAgent

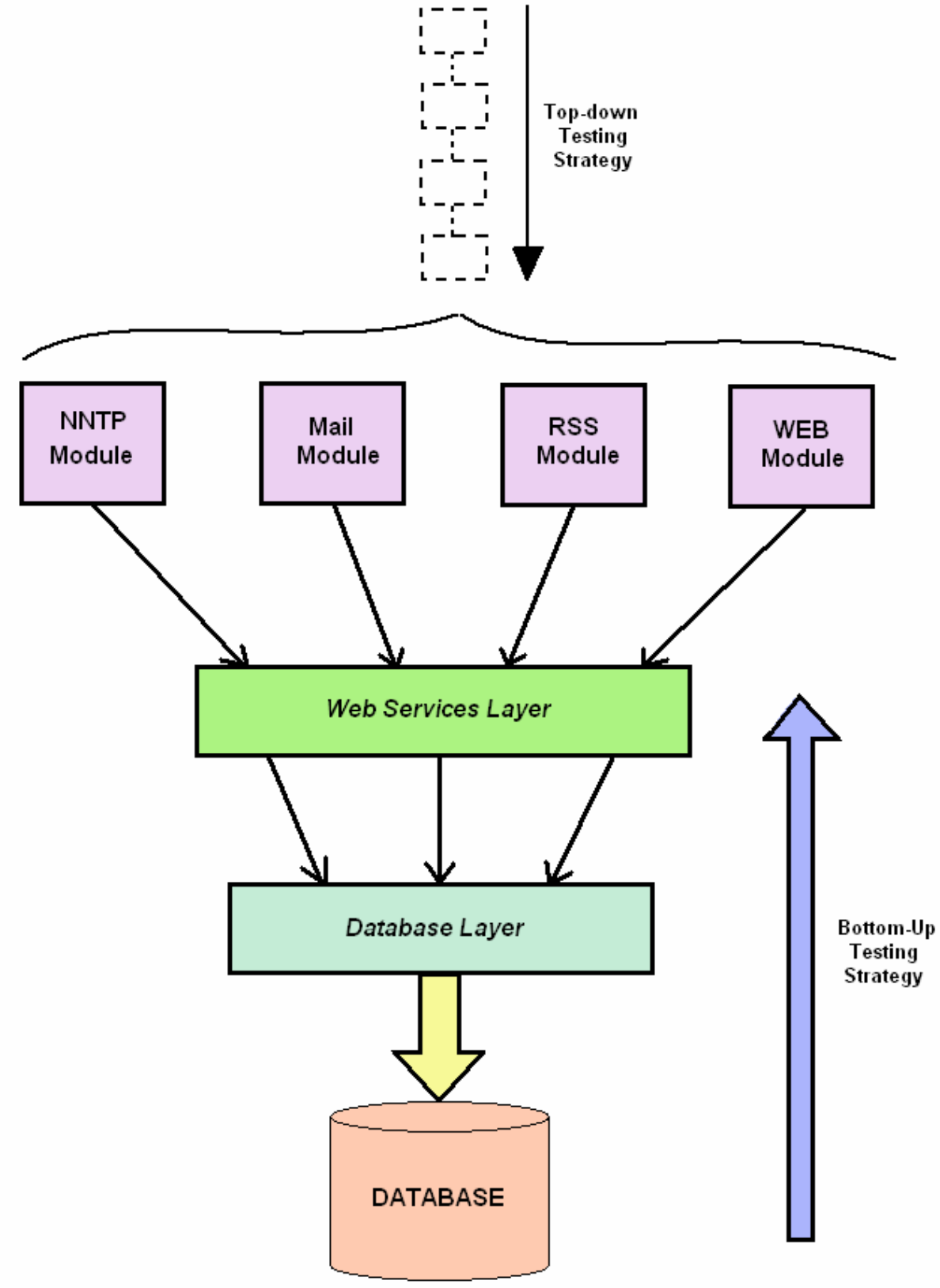
# 2 Testing Plan and Strategy

## *2.1 Testing Plan*

Our aim is to find errors, bugs and apply a good testing strategy that has a high probability of finding an error. We also want to make sure that there are no defects in the product. After we have generated the source code, we are going to test our product to identify the errors and remove them before delivery to the customer. Our goal is to detect and correct as many bugs as possible early in our software development cycle. In order to acquire this we have to design a series of test cases that have a high likelihood of finding errors.

## *2.2 Testing Strategy*

Since NewsAgent has different layers and modules, testing strategy differs for each subpart of the product. We present a testing schema below, which will briefly explain our testing strategy. In general, we follow a bottom-up strategy for testing. Therefore, we have started from database layer as shown in the schema. For this layer, we have applied unit tests in order to check performance and correctness of our database queries. We tested our retrievals, insertions and modifications. Testing of this part is very important since each web service and its methods use the data returned from database layer and insert data into database through this layer. Any error in this layer can cause many problems in above layers since they construct the NewsAgent core. After testing database layer, we passed to web services layer. Any operation in NewsAgent is handled by web services. So testing this part is another important issue in testing the product. For testing our web services, we deployed each of them separately and invoked related methods. We have checked whether each web service works correctly. Up to now, most of the NewsAgent core testing is finished. For other modules, developers of each module applied the first testing procedures to them and after implementation is completed, we will test these modules deeply; NNTP Module, Mail Module, RSS/Atom Module and Web Module. While testing these modules, we will follow a different strategy which is top-down testing strategy.




**Testing Strategy of NewsAgent**

# 3 Testing Procedure

## 3.1 Unit Testing

In the unit test case we are testing the separate modules of the software. White box testing is used where each module or component of the software is tested individually. By this type of testing we have advantages as mentioned below.

i) As the knowledge of internal coding structure is prerequisite, it becomes very easy to find out which type of input/data can help in testing the application effectively.
ii) The other advantage of white box testing is that it helps in optimizing the code
iii) It helps in removing the extra lines of code, which can bring in hidden defects.

We are carrying out unit testing in order to check if the particular module or unit of code is working fine. The Unit Testing comes at the very basic level as it is carried out when a unit of the code is developed or a particular functionality is built.

All team members have their own test files to test whether an implemented function or class is working properly or not. If a problem is identified during testing process, the problem is specified in a detailed manner after a debugging process and the code is modified according to the needs and this process is applied up to all tests on a function or a class are satisfied. In addition to these, even if tests are satisfied, different data is controlled at some specified points. If there is no problem about the data at these specified points, the class or function is labeled as correct, however if this is not the case, the tests are applied again and new breakpoints are determined to observe data.

We are looking for entry and exit conditions of the data. We make sure that all the components work without any troubles. The test primarily is carried out by the programmer who designed and implemented the module. Lead tester then carry out test on the modules to finalize the testing. Up to now we have tested each of our classes and functions according to this unit test strategy and when a new class is implemented these specified steps are processed again.

## 3.2 Integration Testing

In this testing period we look for any signs of the collision between our software components and those of the clients. We want to make sure there is no confusion among the application on the network when they are running simultaneously.

As we know, integration testing is testing of combined parts of an application to determine if they function together correctly. The 'parts' can be code modules, individual applications,

client and server applications on a network, etc. And this type of testing is especially relevant to client/server and distributed systems. We are carefully looking for any sort of collision between several different applications.

In addition to these, for NewsAgent, integration test is one of the most important processes during development and testing. Because there are different modules that access to the same database in coordination and system should be stable at any instant. RSS and SMTP modules should be adapted to the system comfortably and there should be no inconsistency in the system when these modules are integrated. For this reason, we are testing the stability of the system after the implementation of each functionality in any of the modules. For instance, when post article functionality in NNTP module is implemented, we have tested if the posted article is added to the related feeds in RSS modules, mails are sent to related users (who want mail from the newsgroup that the article is posted) and if the posted article can be read by a web user. 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 Security Testing

Testing the security of a news server is really a key point and also testing is an inevitable feature of NewsAgent. Since NewsAgent may be used in workplaces or foundations where security of data is the most important issue, security should be handled carefully. NewsAgent will use SSL for handling security issues. SSL provides data encryption which will be used in transmission of passwords. Also, newsgroups and articles should not be accessed by users who have not right to access them. Security testing will be done by controlling the flow of data in different modules of NewsAgent and will be useful for finding out any security holes.

## 3.4 Validation Testing

Validation testing, which is also named as system testing, is a testing procedure that measures an aspect of an implementation's behaviour against an expectation. By the help of validation testing, we understand whether the software meets the expectations or not. We have divided our validation testing procedure into two parts; Requirements Validation, Design and Data Flow Testing.

### 3.4.1 Requirements Validation

In the light of the requirements stated in requirement analysis report, we will test whether our system meets these requirements. For this purpose, we will follow black-box testing strategy. The general requirements to be tested are stated below:

- ***General Design of WebModule:*** This part includes testing the web module to see whether the links and relations between pages are designed properly. For this purpose, we will produce some use scenarios and according to these, for instance, we will test whether any unnecessary pages are visited or not.
- ***General Functionality of Web Module:*** In this testing, we will test all functionalities of the web module in order to find out any inconsistency or incorrect functions. Some of these functionalities are;
    - Identifying a user and direct the user to the web page according to his/her user group.
    - Retireving newsgroups that are accessible by the user, retrieving correct articles for the newsgroup and display them in a tree-like structure.
    - Managing subscriptions, unsunscriptions and mail options.
    - For the administration part, giving permission to all deletion and insertion processes and maintaining correct results.
- ***Coordination of Nntp Module and Web Module:*** For the main purpose of the project, nntp module and web module should be well integrated. We will make any possible change on the database in web module, and check whether the changes are reflected in the nntp module, and vice versa. Because, data should be consistent and data loss should not be allowed.
- ***Coordination of Smtp Module and Web/Nntp Modules:*** According to the mail options those are adjusted in the web module, smtp module sends e-mail to the users. Articles posted via nntp module or web module invoke these functions. Therefore, smtp module should be in coordination with web module and nntp module. For this purpose, we will subscribe to some newsgroups for e-mail with mailing options 1, 2 and 3. Then examine whether instant, daily and weekly mail options work properly by posting articles via nntp and web.
- ***Coordination of Rss/Atom Module and Web/Nntp Modules:*** Rss/Atom module is invoked when a new article is posted through web or nntp module. Therefore, when a new article is posted, no matter from which module, it should be appended to the

rss and atom feeds and rss/atom users should be able to access these articles.

### 3.4.2 Design Validation

Throughout the implementation process, we have tried to be consistent with our design. We implemented the database layer, web services layer, objects and all other required classes according to the classes explained in the final design report. There is no inconsistency in the report, so we expect not to have any inconsistency in the implementation. However, if there are any mistaken points, in order to detect them, we will use black-box testing and check whether the implemented classes are consistent with the functions and well interacted with the modules.

## *3.5 High-Order Testing*

Besides the tests mentioned above, we also need perform some higher order tests in order to release better versions of NewsAgent. High-order tests those we are planning to perform are performance test, stress test and alpha-beta test.

### 3.5.1 Performance Test

Since NewsAgent is a unified news server that is accessible by any number of clients at the same time, performance test is necessary to perform. For testing the performance of NewsAgent, we will login to the system through web module and connect via nntp module from many different computers with different accounts. The expected result is to respond user's expectations in a reasonable time. When performing this test, we should consider that accesses to the database are through web services. Therefore, the time to perform functions that the user requested may take a little bit more.

### 3.5.2 Stress Test

In a unified news server, there is usually a big number of articles. Therefore, NewsAgent should be able to deal with large amount of data. Performing stress test is a good way to check whether the system is durable and qualified enough to work well in the cases of big data load. For this purpose, we will post as many articles as possible and create many newsgroups, users who are subscribe to feeds and e-mail features of these newsgroups.

### 3.5.3 Alpha – Beta Test

After performing unit tests, integrating the modules and performing the other tests, NewsAgent will be ready to perform alpha and beta testing. We will first perform alpha

testing using white box testing method. Since alpha testing should be performed on developer's side, we are planning to perform this test with another NEWSUNI project group. After alpha test, we will perform beta testing. We will release our product to a limited number of possible users to ensure that the product has few faults or bugs.

# 4 Test Scenario

**Web Module Test Scenario:**

- The user logs into the system and is directed to the page according to his/her user group.

If the user is an administrator, administration menu is displayed. The following scenarios are possible for the administrator.

- Click create user/newsgroup/usergroup → Corresponding creation form is displayed → According to the values entered, creation of user/newsgroup/usergroup is performed or a failure message is displayed in some predefined cases.
- Click delete user/newsgroup/usergroup → Corresponding items are listed → The selected items are deleted.
- Click edit user/newsgroup/usergroup → Corresponding edition form is displayed → According to the values entered, edition of user/newsgroup/usergroup is performed or a failure message is displayed in some predefined cases.

If the user is not an administrator, user menu is displayed. The following scenarios are possible for the user.

- Follow Account link → User account information is displayed → User may request to edit the information → according to the values entered, account info is edited or a failure message is displayed.
- Follow News link → Subscribed newsgroups are displayed in the left frame, articles of a default newsgroup are displayed in the upper frame and the content of a default article is displayed in the lower frame.
- Follow Newsgroups link → Accessible newsgroups for the user group are displayed in the upper frame → The user checks / unchecks the related checkboxes and subscribes/unsubscribes to the newsgroups.
- Follow mail options link → Mail options for the newsgroups are displayed → User changes the mail options for any newsgroup → Mails are sent according to the new options.

- Click on a newsgroup name → Articles of the selected newsgroup are listed in the upper frame → User can click the subject of any article → Content of the article is displayed in the lower frame → User can click reply or new post → Article post form is displayed in the lower frame → According to the values entered, article is posted or a failure message is displayed. When the page is refreshed new article is displayed in the articles list.

**NNTP Module Test Scenario:**

- The user connects to the server via an NNTP client → User requests to list the newsgroups → User selects a newsgroup → Articles of that newsgroup are retrieved and displayed → The user selects and article and the content of the article is retrieved and displayed. → The user requests to post a new article or reply to an article → The new article is inserted to the database and displayed.

**RSS/Atom Module Test Scenario:**

- The user connects to server via an RSS/Atom reader → User subscribes to newsgroup feeds → Username and password is requested from the user → User enters username and password → If they are valid, thae articles of the subscribed newsgroups are expected to be sent and displayed.

**E-mail Module Test Scenario:**

- The user logs in via web interface → User adjusts mail receiving options → For any newsgroup, articles are expected to be received as e-mail according to the mail options set.
- A user sends an e-mail to our server → If the user is registered, e-mail is retrieved and parsed → E-mail is converted to an article → Article is expected to be inserted to the database. → If the sender is not a valid user, e-mail is expected to be rejected.

# 5 Record Keeping and Logs

As mentioned, testing is one of the most important processes during development of a project. However, some kind of record keeping mechanism should also be used for identifying bugs or state of the system at any instant. For NewsAgent, we have login and config log files for record keeping. LoginLog files keep all login data of the system. User who logged in to the system, date and time, loginIP are the data that will be added to the LoginLogs after a login process. Users are also able to access their own loginlogs to see when they are logged into the system, date and time of login operation. In addition to that, configuration log files keep

changes on any data of the system. For instance, when a user changes his/her username, configuration logs will be updated and any administrator will be able to retrieve these configurations on the database by using his administrator interface using web module. By keeping loginlogs and making it possible for users to access their own log files, users will be able to see if a record kept has a different loginIP when compared to user's usual loginIP. By this way, user can keep his/her data in secure and possible fraud can be prevented. Also, configuration logs are kept for similar reasons and administrators can access all logs for managing the system comfortably.

# 6 Staffing

The task distribution for testing process to the team members can be shown as follows:

| | |
|---|---|
| NewsAgent Core, NNTP Module and Integration Testing | **Ali Anıl Sınacı** |
| Web Module, E-mail Module, NNTP Module Testing | **Hilal Karaman** |
| Test Coordinator, Web module, RSS-Atom Module Testing | **Goncagül Demirdizen** |
| Bug Tracer, Web Module and Security Testing | **Ferhat Şahinkaya** |

# 7 Tools and Environment

We use the following tools and environment for testing process of NewsAgent.

- Eclipse: We use eclipse for compilation and debugging throughout the project development cycle.
- JUnit: We use JUnit Framework for unit tests for our independent modules.
- TCPMon: We use TCPMon to examine and monitor messages related to web services.
- Phoenix 2.3.0 (Avalon Framework): We use Phoenix for testing of NNTP module.
- Apache Tomcat 6.0: We use Apache Tomcat for running and testing of the servlets in Web module.

# 8 Testing Schedule

Our testing schedule can be shown as follows:

| Testing Tasks | Start Date | End Date |
|---|---|---|
| Test Specification Plan Delivery | 01.05.2007 | 06.05.2007 |
| Unit Tests and Integration Tests | 20.03.2007 | 15.05.2007 |
| Validation Tests | 15.05.2007 | 20.05.2007 |

| | | |
|---|---|---|
| Performance and Security Tests | 20.05.2007 | 25.05.2007 |
| Stress Tests | 25.05.2007 | 27.05.2007 |
| Alpha Tests | 27.05.2007 | 31.05.2007 |
| Beta Tests | 01.06.2007 | 05.06.2007 |
| Bug Tracing, Detection and Correction | 25.05.2007 | 11.06.2007 |