TEST SPECIFICATION REPORT

Spring 2005-2006

Ahmet	**Tolga Kılınç**	**1297944**
Berkan	**Kurtoğlu**	**1297993**
Hüseyin	**Özgür Batur**	**1297530**
Kerim	**Korkmaz**	**1347681**
Kadir	**Polat**	**1250687**

TABLE OF CONTENTS

1. INTRODUCTION

1.1 STATEMENT OF PURPOSE

Unfortunately every software has bugs.It becomes impossible to design error free software when the scope of the project gets large.”Mind Gate” is a Application Level Gateway for Web Access Control and Accounting Project which has to be implemented in a short period of time. In that short period of time the we have to design a detailed tets specification in order to minimize the errors and bugs.

1.2 TESTING SCOPE

Unit testing: Put in to practice for only the modules of the project.

Integration testing: Synchronization of the modules will be tested.

Validation testing: Requirement validation test will be performed.

Performance testing: Speed concerned tests will be performed.

2. TEST STRATEGY

Testing process of the project includes many test specifications such as:

2.1 UNIT TESTING

Each program module is tested individually, independent from the other modules in the system before the integration. It will be tested that if the module facilities work properly according to the module design or not.

2.2 Integration Testing

After being satisfied that individual program modules are working correctly and meet the desired objectives, we integrate the modules into a working system.After combining modules, functional testing was applied in order to guarantee if each module does its functionalities as required. Synchronization of the modules and the data flow between each module is tested.

2.3 Validation Testing

In that method of the test we will investigate whether functionalities and the specifications of the software requirement document are satisfied by the product.
We will perform the black box testing while testing all the components of the project together.By preparing some input data we will check whether we get the desired results from the software.Comparing the results obtained from the tests and the expectations we will have some ideas about the validation of the software.Problems occured during the tests will be recorded in a deficiency list in order to handle in a later release.

2.4 Performance Testing

MindGate works on web control, so there will be lots of requests and responses coming through every instance. Because of this MindGate should perform efficiently and meet the demands of the users while satisfying the security issues.

3. TEST PROCEDURE

3.1 Unit Testing

3.1.1 Filtering Unit

In Filtering Unit there will be a proxy server, which will run in the client side of the local area network, handling all http requests coming from clients and all responses to these requests coming from internet.

First, we will test session of user who login system or not. We will use different client and try to retrieve some web pages. We will request for pages without open session and after open session. If session is not opened by filtering unit, proxy should not send web page and send a login page to client. If session is opened before requesting web page, web page should be retrieved and filter according to filtering policy. In addition, we will test log files according to successive session according to user id and client machine's ip.

Second, proxy server runs trade based. Therefore, we will test it by requesting web pages from different clients at the same time. Some of these clients have session or not and others session are expired. This testing phase will show us whether threads of proxy working well or not.

3.1.2 Architecture Management Unit

In Architecture Management Unit there will be a Web Server which will contain Admin Interface, Error Pages, Login Page, and User Feedback forms. This server will run as a part of the MindGate and will serve its contents as needed; it can be accessed via internet anywhere around the world.

First, we will test sending web server's content to user is working correctly or not. Web server contents such as Login Page or Error Page is sent to client when user tries to enter a blocked web site or has not open session. We will try to retrieve blocked web site and in this situation if web server works correctly, we will take error page.

Second, administrator uses architecture management unit to configure MindGate. Therefore, administrator does some queries and configurations. We test this configuration according to white box testing. We will make configuration and that test it whether it is done or not.

3.1.3 Database Management Unit

In Database Management Unit there will be an SQL Server, which will used to store User Related Data of the MindGate. These data will be used for login and administrative purposes.

We will test all tables individually according to queries and insertion calls of categorizer module, filtering module and architecture management module. We will create all tables and all test cases will be applied to these tables. In this phase, we only test SQL server and it calls. If there is a problem in SQL server, it affects all modules' operation badly. Therefore, by regarding other module, we will test database management unit separately and carefully.

In database module, there are Lucene index files. MindGate Database Controller Unit basically uses Lucene to indexing and searching documents in memory. All indexed files are held in the fast-access memory and not on a slower hard disk. Therefore, we will test all Lucene indexer and searcher class individually. If there is a searching error or indexing error, we will find it by controlling all execution results of indexer and searcher class.

3.1.4 Categorizer Unit

In Categorizer Unit, there will be a Document Similarity Calculator Module. It will be implemented using Indexer and Searcher classes.

We will test Categorizer Unit by sending a web page to module and then we will check it's desicion about this web page. Module first check its static list whether web site has been categorized before or not. If it is not in list, module categorize web page and add it static list. We will test this static list by searcher and indexer classes. We will also test this checking process. Static lists will be constracted by us and module desicion will test according to this static lists. And also, we will give categorizer randomly choosen web page and its result about this page is checked by hands.Therefore it's percent of achievement will be found.

3.2 Integration Testing

When we are satisfied that individual program modules are working correctly and meet established objectives, we will combine the modules into a working system.

While doing unit test, we will integrate some of modules because without other modules we will not be able to run another modules such as Filtering Unit and Architecture Management Unit. But just after completing this phase, we will integrate all of the modules.

We will test all case which explained in Unit Testing phase, will be done in Integration Testing phase.

After combining modules, functional testing will be applied in order to guarantee if each module does its functionalities as required.

3.3 Validation Testing

· **Reliable filtering**

This is one of the most important requirements of MindGate Software, because making wrong decisions while blocking the content will lead many big problems. It will slowdown the work in an organization and will affect the employees' psychology. In a school it will effect students desires of use internet in an negative way. So Software should has a really strong mechanisms about making right decisions on different contents.

· **Sophisticated Categorization**

Software will used mainly for filtering purposes and this can be achieved by using different methods. But accomplishing the filtering task by using categorization will result in extra useful data which is collected over a time period. This categorized data can be used many areas and tools in the future such as, which needs a categorized unstructured data to statistical researches, or complex data mining.

· **Blocking harmfully categorized content in a flexible way.**

Software such as MindGate which includes some machine decision mechanisms in it are heavily error prone. Therefore blocking mechanisms which are expected from these products should be very flexible in their nature. There will be many wrong decisions which results in blocking wrong content which is clear according to predefined policies, so to fix these errors very good response mechanisms should be implemented.

· Feedback from users about filtering.

Getting feedback from the users and managers of the organizations about filtering policies and wrong decisions done by MindGate, other important requirements of MindGate. This should be considered in all units of the software because it consists of human-human and human-computer interactions. Feedbacks can be sent to Administrator of the system or directly to the Software and can be logged for future uses.

· Updating lists using remote sources from time to time.

This requirement is about sharing information collected in different organizations which are about web content and categorization to improve the performance of filtering done by MindGate. In different LANs accessed pages and contents in the internet will also be different so by extending the capabilities of the application as accessing and using these different analyzed and categorized content will result in less error-prone decisions and very fast response times from MindGate.

· **Network traffic controlling.**

In LAN internet traffic controlling mechanisms should be used. Because LAN has limited resources for internet access, so these resources should be maintained and controlled very effectively, to achieve a high performance.

· Not slowing down the network traffic.

While Software such as MindGate are performing content filtering task, they will use the LAN's internet resources so this leads some slowdown in LAN internet traffic. This slowdown results in many unwanted situations which users complain a lot. According to this facts in MindGate these requirement will be taken in account seriously.

To test these requirements we will make unit testing and see whether the communication between modules are correct, whether system behaves as expected. Finally the requirements that are not correctly implemented, validated, will be revised and corrected.

3.4 Performance Testing

After all the modules are completed and integrated, we will do the performance testing. When there are more than one user connected to the system, time spent carrying out certain functions of the system is assessed.

MindGate performace will be test when there are more than one user , and proxy server response time will be evaluated. And also, categorizer unit will be tested when there are a lot of requests. This phase of testing will be done in department labs. In this way, amount of maximum user in MindGate will be evaluted.

4. TEST SCHEDULE

The following is the schedule for the testing plan that is presented in this report:

Test Plan Delivery

(Deadline) 23.04.2006

Unit Test and Integration Tests

23.04.2006 - 7.05.2006

Module testing of each module is done individually by the team member responsible from that module.

- Berkan is responsible for the GUI module and testing.
- Tolga is responsible for the Connection Control module and testing.
- Özgür is responsible for the Categorizer module and testing.
- Kadir and Kerim are responsible for the lucene database module and testing.

Every team member will be responsible for the Integration tests.

Validation Tests

7.05.2006 - 14.05.2006

Kadir , Kerim and Berkan will be responsible for the validation tests.

Performance Tests

14.05.2006 - 20.05.2006

Özgür and Tolga will be resbonsible for the performance tests.

Results (i.e. Bugs) Tracing and Correction:

20.05.2006 – (Deadline)