**1 Introduction**

**1.1 Intention**

This test report is used to record the bugs in the test. It will help us to perfect our Course Resource Sharing Subsystem, guide us to test the subsystem, and try to provide better user experience for our customer.

This report is aimed at the following people: the developers of Course Resource Sharing Subsystem, the managers of the system, the tester of the system, and other people related to the project.

**1.2 The background**

The Teaching Service System is based on the university network to provide service for the teaching activities. The system is composed of 7 subsystems, and our team implements the fourth subsystem, the Course Resourse Sharing Subsystem.

This subsystem is mainly responsible for sharing and utilization of course resource between teachers and students and for students to upload their homework or laboratory reports.

a. Project name

Course Resource Sharing Subsystem

b. Task originator

The TA. of the software engineering course, Xinyu Wang

c. Developer

Students of the software engineering course in Zhejiang University

d. Users

Students and instructors in universities

e. Implementation of the computer network

A local area network consisted of several PC

**1.3 Defination**

PC: Personal Computer, a computer that is used by one person at a time.

DB: Database, a collection of data which contains the information of users and resources.

Load Testing: test the performance when we overload the system to find and fix the bugs in designing or to show the load capacity.

Postgres SQL Server: A database management system.

Python: A possible programming language to interface between IPTables and Postgres.

Apache: An open source Web server.

JDBC: A possible Java-based interface between IPTables and the Database.

Gateway: Bridges the gap between the internet and a local network.

**1.4 Reference**

a. Roger, S. P. (2008). *Software engineering: A practitioner’s approach(sixth edition).* China Machine Press

b. Guojun Hu, *Testing report of the trading client in stock trading system*

c. Syntax\_Error team *,Design Report*

d. Syntax\_Error team, *Software Requirement Specification*

**2 Testing Outline**

**2.1 Software Specification**

This is the Course Resourse Sharing Subsystem, to perfect our program, we have some changes in these modules, and finally there are 12 modules in this subsystem. Here are the modules:

ConnectDB Module, Resource Management Prepare Module,

Personal Resource Management Module, Resource Management Module,

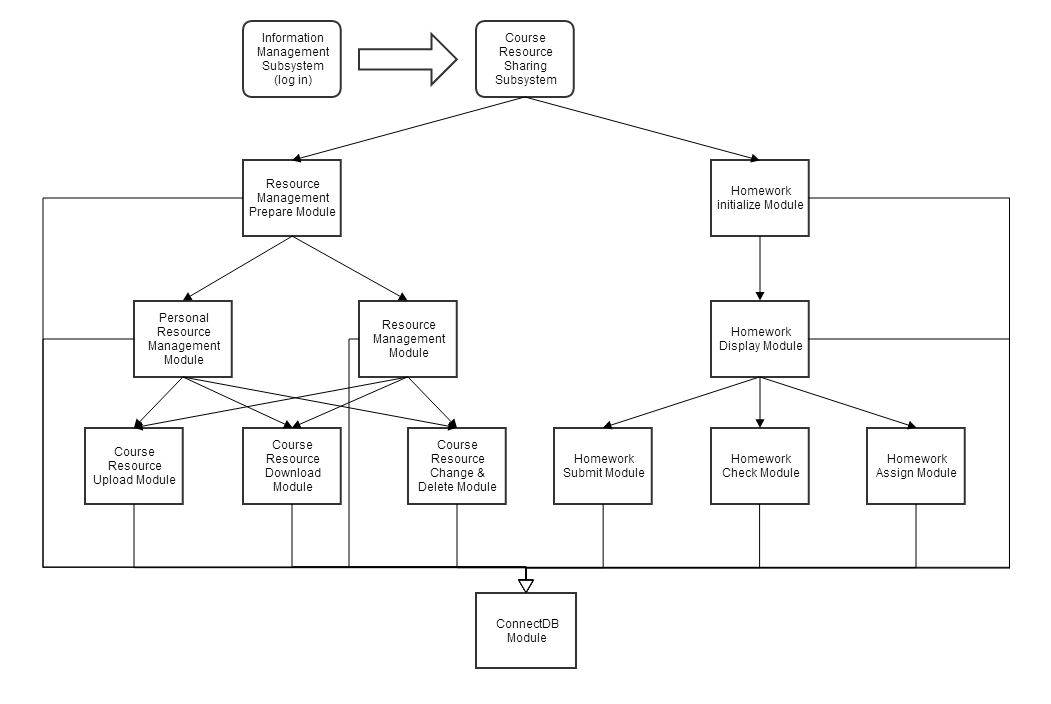
Course Resource Upload Module, Course Resource Download Module,

Course Resource Change & Delete Module, Homework initialize Module,

Homework Display Module, Homework Submit Module,

Homework Check Module, Homework Assign Module

The relationship of these modules and other subsystems can be shown in the following diagram:



Relationship Diagram

For each module, the main input and possible output are given in the following chart:

|  |  |  |
| --- | --- | --- |
| Module | Input | Output |
| ConnectDB Module | When any module link to the function | Successfully connect to the database. |
| Resource Management  Prepare Module | The information of user. (from session) | Continue the number of resources under the user’s file and the user’s courses. |
| Personal Resource  Management Module | The basic information of user. (from session) | Continue with the icon button of each resource which is upload by the user in the font-end page. |
| Resource Management  Module | The basic information of user. (from session) | Continue with the icon button of each resource grouped by courses in the font-end page. |
| Course Resource  Upload Module | Upload request from the user. | File was successfully upload in the server and the database is updated. |
| Course Resource  Download Module | Download request from the user. | The corresponding file. |
| Course Resource  Change & Delete Module | Resource information updating request. | The database is updated according to the input data. |
| Homework Initialize  Module | The information of user. (From session) | Count the number of homework under the courses which the users have. |
| Homework Display  Module | The basic information of user. (from session)  The id of course. (from the click action) | Gather the information of the homework. Then list the raw information and icon of homework on the right page. After clicking on the icon, show the specific information of the homework. |
| Homework Submit  Module | Submit the homework by uploading a file | File was successfully upload in the server and the database is updated. |
| Homework Check  Module | Download request from the user. | The corresponding file. |
| Homework Assign  Module | Upload a new assignment. | The database is updated successfully. |

**2.2 Test informaion**

The test includes following things: Function test, boundary test, load test and test of interface to other subsystem.

The test information is in the following chart:

|  |  |  |  |
| --- | --- | --- | --- |
| The Name of Test | Aim | Information | Different Changes |
| Function Test | Test the function of each module | Test each module as the request in *Software Requirement Specification.* | Nothing Changed |
| Boundary Test | Test the reaction when the system deal with boundary | Input some boundary data, and record the output and result of the system. | Nothing Changed |
| Load Test | Test the performance when we overload the system | Use the test software in Microsoft Visual Studio 2012, and use virtual users to run the system. | Nothing Changed |
| Test of Interface to Other Subsystem | Test the interface to other subsystem and show if it work perfect. | Run this subsystem and record the changes in the database. | Nothing Changed |

**2.3 Test One: Function Test**

**2.3.1 Scheduling**

|  |  |  |
| --- | --- | --- |
| Stage | Information | Date |
| The first stage (training stage) | Testers read the *Software Requirement Specification* of this subsystem, and make sure they are clear with the function of each module. | June 15th |
| The second stage (preparing stage) | Testers write the examples of testing according the function of each module, and prepare the test data. | June 16th |
| The third stage (testing stage) | Testers test the subsystem with the examples and data, and try to find some disadvantages and bugs. | June 17th |
| The fourth stage (adjusting stage) | The programmers and coders fix the problems according the test result. | June 18th |

**2.3.2 System Request**

a. OS: Windows Vista or newer

b. Browser: Microsoft Internet Explorer 10 or newer; Safari 7.0.3; Chrome 34.0.1847.137 or newer

c. Test Software: Microsoft Visual Studio 2012

c. PHP: PHP 5.0

d. Database management : MySQL

e. Web Server: Apache

**2.3.3 Test Information**

This subsystem includes 12 modules, and the input and output of each module are shown in the following chart:

|  |  |  |
| --- | --- | --- |
| Module | Input | Output |
| ConnectDB Module | No input | Successfully connect to the database. |
| Resource Management  Prepare Module | No input | Continue the number of resources under the user’s file and the user’s courses. |
| Personal Resource  Management Module | No input | Continue with the icon button of each resource which is upload by the user in the font-end page. |
| Resource Management  Module | No input | Continue with the icon button of each resource grouped by courses in the font-end page. |
| Course Resource  Upload Module | Upload request from the user. | File was successfully upload in the server and the database is updated. |
| Course Resource  Download Module | Download request from the user. | The corresponding file or an error report. |
| Course Resource  Change & Delete Module | Resource information updating request. | The database is updated according to the input data. |
| Homework Initialize  Module | No input | Count the number of homework under the courses which the users have. |
| Homework Display  Module | The id of course. (from the click action) | Gather the information of the homework. Then list the raw information and icon of homework on the right page. After clicking on the icon, show the specific information of the homework. |
| Homework Submit  Module | Submit the homework by uploading a file | File was successfully upload in the server and the database is updated, or or an error report when trying to access a unavailable homework. |
| Homework Check  Module | Download request from the user. | The corresponding file or an error report without login. |
| Homework Assign  Module | Upload a new assignment. | The database is updated successfully or an error report when input with wrong format or invalid characters. |

**2.4 Test Two: Boundary Test**

**2.4.1 Scheduling**

|  |  |  |
| --- | --- | --- |
| Stage | Information | Date |
| The first stage (training stage) | Testers read the *Software Requirement Specification* of this subsystem, and make sure they are clear with the boundary of each data needing input. | June 15th |
| The second stage (preparing stage) | Testers write the examples of testing according the boundary of each data, and prepare the test data. | June 16th |
| The third stage (testing stage) | Testers have a boundary with the examples and data, and try to find some disadvantages and bugs. | June 17th |
| The fourth stage (adjusting stage) | The programmers and coders fix the problems according the test result. | June 18th |

**2.4.2 System Request**

a. OS: Windows Vista or newer

b. Browser: Microsoft Internet Explorer 10 or newer; Safari 7.0.3; Chrome 34.0.1847.137 or newer

c. Test Software: Microsoft Visual Studio 2012

c. PHP: PHP 5.0

d. Database management : MySQL

e. Web Server: Apache

**2.4.3 Test Information**

The boundary test of this subsystem is mainly about the name of file or resource and information of homework. These data need some format or limit in length or character, so the test of boundary is needed. If the system react in a wrong way, then there is problems to fix.

Our method of boundary test is to input some boundary data, and watch and record the changes in the client and database.

Here is the information of data which needs to be tested:

|  |  |  |  |
| --- | --- | --- | --- |
| The name of data | Data type | length | Description |
| Reso\_name | varchar | 30 | Resource name |
| File\_name | varchar | 30 | File name |
| hw\_title | varchar | 20 | Homework title |
| hw\_deadline | date | / | Homework deadline |
| hw\_requirement | varchar | 200 | Homework requirment |

**2.5 Test Three: Load Test**

**2.5.1 Scheduling**

|  |  |  |
| --- | --- | --- |
| Stage | Information | Date |
| The first stage (training stage) | Search for the information of load test, and find a software to do the load test. | June 15th |
| The second stage (preparing stage) | Testers learn to use the software, and prepare the test data. | June 16th |
| The third stage (testing stage) | Testers have a load test with the subsystem and have an a analysis to the test result. | June 17th |
| The fourth stage (adjusting stage) | The programmers and coders fix the problems according the test result. | June 18th |

**2.5.2 System Request**

a. OS: Windows Vista or newer

b. Browser: Microsoft Internet Explorer 10 or newer; Safari 7.0.3; Chrome 34.0.1847.137 or newer

c. Test Software: Microsoft Visual Studio 2012

c. PHP: PHP 5.0

d. Database management : MySQL

e. Web Server: Apache

**2.5.3 Test Information**

**2.5.3.1 About the load test**

The load test is trying to find a limit of the system or a point that the system cannot work normally so that we can know the max level of support that the system can give. In a word, the load test is going to find when the software cannot work as we except. The load test is mostly for a web app, especially a website. In this condition, the aim of load test is to find that if the system could handle with all kinds of requests by the users, which shows that if the system is healthy and strong. Another aim is to find the reaction of program when so many requests are made in a time. The usual load test includes: the time of receiving a huge size of data, the recovery time, the time of data import and export, and so on.

The load test use multithreading technology, and imitate the condition that many users open the website in a time. Then test and record the reaction of the server.

**2.5.2 Test tool**

This load test uses the tool in Microsoft Visual Studio 2012, imitating the condition that many users try to access the server in a time. For example, we can test the performance of the website in the full load case before submit it.

**2.5.3 Test training**

Before this test, the testers of load test need to know the information of the subsystem, and learn to use the tool and the method of analysis.

Please refer the information of the subsystem and the method of analysis in:

*Software Requirement Specification*

And the use of the tool:

http://msdn.microsoft.com

**2.6 Test Four: Test of Interface to Other Subsystem.**

**2.6.1 Introduction**

The Course Recourse Sharing Subsystem have connection with the Information Management Subsystem because the login module is deleted from this subsystem and the Information Management Subsystem will cover this module for the whole system. So the test of interface between this two subsystems is need.

**2.6.2 Scheduling**

|  |  |  |
| --- | --- | --- |
| Stage | Information | Date |
| The first stage (training stage) | Testers read the *Software Requirement Specification* of this subsystem, and make sure they are clear with the relationship of the two subsystems. | June 15th |
| The second stage (preparing stage) | Testers write the examples of testing according the function of each module, and prepare the test data. | June 16th |
| The third stage (testing stage) | Testers try to test the interface and find the bugs. | June 17th |
| The fourth stage (adjusting stage) | The programmers and coders fix the problems according the test result. | June 18th |

**2.6.3 System Request**

a. OS: Windows Vista or newer

b. Browser: Microsoft Internet Explorer 10 or newer; Safari 7.0.3; Chrome 34.0.1847.137 or newer

c. Test Software: Microsoft Visual Studio 2012

c. PHP: PHP 5.0

d. Database management : MySQL

e. Web Server: Apache

**2.6.4 Test Information**

The only thing we need to care is that if the operation can be made without the Information Management Subsystem. So try to do some requests that need logging in and see if we can get the result, or try to access the website or change the data in the database via the address bar without logging in. If such operation can be done, then we have to solve these problems. Also we need to test these cases when we have already logged in, and if we cannot get the right output, there would be another problem.