Hall Management System

Software Test Plan

Prepared by:

1) Subham Ghosh

Roll No.: *11CS10060*
and

2) Sourav Sarkar

Roll No.: *11CS30037*

Problem No.: *19*

1. INTRODUCTION

1.1 Objectives

The software test plan is aimed at verifying the functionality and correct working of every aspect and part of the software. The software should not cease to work correctly when it has been exposed to stress and all corner cases should be handled effectively.

1.2 Testing Strategy

Testing is the process of analyzing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item.

Specific test plan components include:

- Purpose for this level of test,

- Management and technical approach,

- Pass / Fail criteria,

- Hardware/ software requirements,

- Reported bugs.

1.3 Scope

Testing will be performed at several points in the life cycle as the product is constructed. Testing is a very 'dependent' activity. As a result, test planning is a continuing activity performed throughout the system development life cycle. Test plans must be developed for each level of product testing.

2. TESTS

Features to be tested:-

All the use cases are thoroughly tested by white box testing and black box testing. The databases are checked after every use case and if they do not match with the expected outcome, they are reported to be failed.

Features not to be tested:-

- iText library for generating pdf file
- JDBC MySQL connector

2.1 Unit Testing Each unit has been tested for consistency by adding main() functions to each class and passing hardcoded data to the constructors of the class. The corresponding **class files have been enclosed** and the screenshots have been included in the Test Suite Results document.

2.2 Integration Testing

All of the above units has been integrated together and tested for consistency. The dataflow from one unit to another has been verified.

2.3 Interface Testing

The software has been tested on machines with different configurations (for example computers with different operating systems like Windows or Linux) and the functionality of the interface has been verified.

2.4 Security Testing Password validation system has been implemented for each user and checks are done at appropriate points of time to ensure consistency. Whenever any transaction is made in any account (be it salary/ mess etc.) the details of the transaction are stored in our logs and these logs show up in our account statement when the warden exercise his use case to generate account statements. This minimizes the possibility of financial fraud.

2.5 Recovery Testing

The file "logindata.txt" stores the details of the database where all the data is stored. Hence whenever any query is made this file is used and hence this software is consistent in its functionality whenever it is restarted. This logindata.txt file is generated when Run Setup is executed.

2.6 Performance Testing

Each of the use cases are tested with respect to their response times and are implemented in such a way that the response time is as small as possible. To achieve the number of SQL queries that are made while carrying out a particular use case is at most two. Furthermore only one query suffices at all places except for the use case of view complaints exercised by the warden. In this use case the first query is for retrieving all the complaints which have not been forwarded to the HMC yet by the warden. The second query is made for matching the hall code of the complaint and the warden's hall code, if they match the complaint is displayed. This is the only place where two SQL queries are made and hence the use case with the maximum delay in response time.

2.7 Use Case Testing

The following use cases were checked to cover all **Corner Cases** as far as possible:-

- If there are no complaints added by any student and the warden of the HMC tries to view complaints, NO COMPLAINTS is displayed.
- If a non-numerical value is entered in a textbox which expects a numerical value (for example salary) a message box is displayed appropriately.
- If passwords and usernames do not match appropriate messages are displayed
- If the HMS database does not exist in MySQL then no user can login to the software.
- Clicking the next button in the complaints window when the end of the list has been reached results in a message box being popped up which shows that the end of the list has been reached. Clicking the back button at the start of the list produces a similar result.

- When a hall's clerk logs in, if there are no workers recruited by the hall, NO WORKERS RECRUITED is shown and the clerk cannot give attendance to anyone.
- When a mess manager logs in, if no student has been allotted that particular hall yet, NO STUDENTS is displayed as the mess manager cannot exercise his option of setting the mess charge for a student.
- When a new worker, student or hall is added it is ensured that the ID that is being entered is unique, otherwise an appropriate message is displayed.
- While making any transaction, it is checked whether there is enough money left in the respective account. If not it is not possible to carry out the transaction.

2.8 Alpha Testing

The software was distributed among a group of 10 people in our Hall of Residence and they were made to test its functionality.

2.9 Stress Testing

The software was subjected to simultaneous logins from different machines and the functionality of the software was tested in this situation. The response times were observed and was ensured that they were not very much.

3. PASS / FAIL CRITERIA

3.1 Suspension Criteria
The test is considered suspended if any of the following is encountered:-

- The software crashes • It

results in wrong output

3.2 Approval Criteria

The test is considered suspended if any of the following is encountered:-
- The correct results are obtained
- The software does not much time to achieve the aforementioned criteria.

4. ENVIRONMENTAL REQUIREMENTS

4.1 Hardware
- Working internet connection
- MySQL should be installed on an appropriate server so that it can handle simultaneous request from various users.

4.2 Software
- Operating System:- Windows/ Linux
- Java should be installed (preferably JDK 6)
- iText and JDBC libraries should be included
- MySQL should be installed

5. BUGS REPORTED

- Checking for negative salary amounts has not been done and if negative amounts are entered then money gets credit (instead of getting debit) to the account

- Even if the dues of the student are zero it is possible for the student to pay his dues and receipts for zero amount are generated.
- While allotting a room to student according to the student's preference (single / double room) it would have been easier to check the availability of single or double rooms by maintaining respective fields in the hall data table. Instead a search is done, every time a student is admitted, through the entire room list to find any available room. This has an effect on the performance of the software (number of queries remain same but the search time is linear instead of being constant).