TEST SUITE DESIGN

LIBRERIA

LIBRARY INFORMATION SYSTEM

Group Number : 52

Group members:

- Ashrujit Ghoshal (14CS10060)
- Sayan Ghosh (14CS10061)

Contents

1 Introduction

This document is a high-level overview defining testing strategy for the Library Information System (which we have named Libreria). Its objective is to communicate project-wide quality standards and procedures. It portrays a snapshot of the project as of the end of the planning phase. This document will address the different standards that will apply to the unit, integration and system testing of the specified application. Testing criteria under the white box, black box, and system-testing paradigm will be utilized. This paradigm will include, but is not limited to, the testing criteria, methods, and test cases of the overall design. Throughout the testing process the test documentation specifications described in the IEEE Standard 829-1983 for Software Test Documentation will be applied.

2 Test Objective

The objective of the test plan to find and report the bugs which are present in the software and to see how much of the features expected in the SRS document have been covered.Although exhaustive testing is not possible to be done in this case but still a broad range of tests will be done. The following will be performed by this software : creation of user, clerk and addition of book to the library, deletion of user and clerk, disposal of a book, issuing and returning of a book by a user, reserving a book already issued, searching of a book if present in the library and checking of book overdue and fine calculation.

3 Process Overview

The following set of operations briefly state the processes which need to be done in order to test the software.

1. First identification of the requirements is a must for which we have to refer to the SRS document.
2. Identify the test we need to perform for each module
3. Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit.
4. Identify the expected results for each test
5. Document the above details and perform the tests
6. Successful unit testing is required before the unit is eligible for component integration/system testing.
7. Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered, its possible cause, and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis
8. Test documents and reports shall be submitted. Any specifications to be reviewed, revised, or updated shall be handled immediately

4 Run Time Tests

4.1 Input Tests on units

We test the software by giving some valid and invalid test inputs. This is done in the following areas :

- **During login into the system:**
  We test by giving valid as well as invalid username and password combinations.Also text boxes may be left blank to see how the software reacts to a blank input
- **During creation of user :**
  During creation of a user of the library invalid inputs are given to test how the software reacts to it.
  For example in the phone number field one may give a textual input or a alphanumeric input.
- **During creation of clerk :**
  During creation of a clerk of the library invalid inputs are given to test how the software reacts to it.
  For example in the phone number field one may give a textual input or a alphanumeric input.
- **During creation of book :**
  During creation of book valid as well as invalid inputs are given to the fields with some constraints like integer or a range constraint.
  For example, one may give invalid inputs in the price and the rack number fields.Also the year field needs testing
- **During searching of a book :**
  Valid and invalid book names are to be given to see how the software responds to this particular input.

4.2 Black box tests

- Test the login feature of a librarian by giving valid librarian user name and password.
- Test user creation by giving valid entries in the respective fields
- Test clerk creation by giving valid entries in the respective fields
- Test book addition to the library system by giving valid entries in the respective fields.
- Test issue and return of a book by a valid user on a valid book
- Test reservation of a book on a book issued by a user and now being reserved by another valid user.
- Test deletion of a clerk and and deletion of an user by the librarian.
- Test the notification system of the librarian by issuing an overdue notice to some valid user and issuing a book disposal notification to a clerk
- Test if the notifications are properly received and acted upon by a user and also by a clerk.
- Test the disposal of a book by a clerk for which the librarian has issued a notification.

4.3 White box tests

- Try to login into the system via a valid username and password combination and also via an invalid username and password combination. Check the system response in both the cases.
- Check creation of user and clerk and also the addition of a book into the library system by giving invalid and blank entries to each of the fields and check and document the response and the result of the system.
- Search for a book present in the library and also for a book not present in the library.
- Search for a book already issued and see if the details shown are valid or not.
- Try to reserve for a book which is present on shelf and document the output or the response of the software.
- Check the disposal of a book not being notified by the librarian to be deleted.

4.4 Mutation testing

- Test by altering the data types of the various fields of the database system.
- Test by deletion of certain parameters in the database or the classes to see if the functionalities of the software are all satisfied in this smaller set of parameters
- Test by changing certain constraints during data comparison or some other procedures to see if the task can be accomplished in some other sophisticated way.

4.5 Performance tests

This test was conducted to evaluate the fulfillment of a system with specified performance requirements. It was done using black-box testing method. Following things were tested:

- Adding large number of products to the database to see how much time it takes to retrieve them from the database
- Calculating the issue statistics for a very large collection of books in the library and check the system time taken

Note : For all the above test the loading time of the software is not considered.