Library Management System

Software Testing Specifications

Palak Singhal 16CO129
Sharanya Kamath 16CO140

Date: 13/04/2018

Table of Contents

1. Test Plan Description

The Library Management Application is made up of many features which allow the user to search for books present in the library and issue them. Each feature has a number of use cases and each of those use cases can be tested in a lot of different ways. We plan to discuss all the features implemented along with uses cases for all of them and how we plan to test all those use cases. We also analyze risk in our use cases and brief our testing team.

Features	**Use Cases**	**Components**
Login	● Login by User ● Login by Admin	Cancelhold.java LoginAdmin.java LoginCancel.java MainActivity.java Cancelhold.xml Activity_main.xml Login.xml Loginadmin.xml User.java MySQLiteHelper.java
Register	● Registration of a new user	CreateClass.java Createaccount.xml User.java MySQLiteHelper.java MainActivity.java
Book	● Addition of a book to the database ● Search of a book by the user ● Issue book by the user	AddBook.java Book.java BookResults.java IssueBook.java MySQLiteHelper.java Addbook.xml Issuebook.xml Inventory.java Inventory.xml
User	● Admin can view all registered users with their login date & time, and recently used date & time.	User.java Transaction.java Log.java Transaction.xml

2. Introduction

This document is the Software Testing Specification report of Library Management Application created by undergraduate students of Computer Science and Engineering branch of National Institute of Technology, Karnataka.

SCOPE

The Library Management system has been developed to tackle the problems faced by the current manual system in libraries. This software is set to eliminate, and in some cases reduce the hardships faced by the existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed by the user to use this system. Thus, it is user friendly. It can lead to error-free, fast and secure management system. It can assist the user to concentrate on their other activities, than concentrate on record keeping. Thus it will help the organization in better utilization of resources.

The plan contains a list and brief description of the use cases to be tested and the software components associated with each test case. The plan also provides a schedule for the testing and the assignment of team members to their respective testing tasks. The process for documenting resolving software errors and/or anomalies that are found during the testing is also specified. The test specification includes a list of the features to be tested for each of the use cases, the description each test case needed to fully test the use case, and the test procedures, or steps, necessary to execute each of the test cases.

3. Test Items

These are things we intend to test within the scope of this test plan. Essentially a list of what is to be tested.

- GUI
- Database
- Basic functions :
  - Add a student
  - View student details
  - Add a book
  - Delete a book
  - Issue a book
  - Search book
  - View book details

4. Features to be tested

This section, focusing on the functional aspects of testing, identifies all features and combinations of features to be tested.

- Components developed in house
- The components developed by this organization will be tested.
- Components developed by outsource vendors
- Components outsourced to be developed specifically for this project where this test team has primary responsibility to test and validate those components will be tested.
- Components outsourced to be developed specifically for this project where the outsourced vendor is responsible for development as well as testing will not be tested as components by the component testers. Rather, the test results from the vendor will be reviewed by the test leads, and if they pass review, the component will be tested in-house starting with integration test.

5. Features not to be tested

- 3rd party and Off-The-Shelf components
- It is assumed that 3rd party components were evaluated and the pros and cons properly weighed before choosing that component with our software. The interfaces to those components will be tested, but not the functionality or performance those components. This includes any 3rd party websites and GPS devices or software.
- Infrastructure components will also not be tested.

- ❏ The actual database software utilized SQLite is assumed to work as designed and will not be directly tested for functionality. Performance tests will be done during system test with respect to GUI response time that will involve the database. However, no testing will be done directly against the database.
- ❏ The internet/WIFI backbone will be utilized during testing, however, no tests will be written executed to directly test the communications backbone.

6. Testing Approach

We used the SRS to prepare the design test cases and their procedures. These tests were designed to verify the documentation previously listed.

The following testing approaches were accounted for:

1. Unit Testing:

Definition : Test smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation.

Participants/ Tested by : Developers

Methodology : Used for the Database test, records in each table, Basic function test, add a student, add a book, Network test

2. System and Integration Testing:

Definition : Integration testing is the phase in software testing in which individual software modules are combined and tested as a group.

System integration testing (SIT) is a high-level software testing process in which testers verify that all related systems maintain data integrity and can operate in coordination with other systems in the same environment.

Participants/ Tested by : System Tester

Methodology : It is used for the Database test, Queries for insert, update, delete the records.

3. Performance and Stress Testing:

Definition : Determine how a system performs in terms of responsiveness and stability under a particular workload.

Stress testing tries to break the system under test by overwhelming its resources or by taking resources away from it

Participants/ Tested by : Tester

Methodology : It is used for the Database test, records in each table, Basic function test, Network test

4.User Acceptance Testing:

Definition : Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies

Participants/ Tested by : Users / End Users

Methodology : It is used for Whole System Test

5. Automated Regression Testing:

Definition : Uncover new errors, or regressions, in existing functionality after changes have been made to a system, such as functional enhancements, patches or configuration changes.

Participants/ Tested by : Tester

Methodology : It is used for Whole System Test

7. Item Pass / Fail Criteria

Test acceptance criteria:

- ❑ Approved Functional Specification document, Use case documents must be available prior to start of Test design phase.
- ❑ Test cases approved and signed-off prior to start of Test execution
- ❑ Development completed, unit tested with pass status and results shared to Testing team to avoid duplicate defects
- ❑ Test environment with application installed, configured and ready to use state

GUI test

Pass criteria:

- ❖ Librarians could use this GUI to interface with the backend library database without any difficulties

  Result: pass

Database test

Pass criteria:

- ❖ Results of all basic and advanced operations are normal

  Result: pass

Basic function test

Add a student

Pass criteria:

- Each student should have following attributes: Student ID (unique), Name, Username, Email Address

  Result: pass

- The retrieved student information by viewing student details should contain the above four attributes.

  Result: pass

Add a book

Pass criteria:

- Each book shall have following attributes: Call Number, ISBN, Title, Author name.

  Result: pass

- The retrieved book information should contain the four attributes.

  Result: pass

Delete book

Pass Criteria:

- The book can be deleted only if no user has issued it.

  Result: partially pass.

  When no user has issued it, pass. When there are user having issued it, did not pass

- If book were deleted, it would not appear in further search queries.

  Result: pass

Search for book

Pass criteria:

- The product shall let Librarian query books' detail information.

  Result: pass

- The search results would produce a list of books with following details: ISBN number, Title, Author, Quantity.

Result: pass

- The display would also provide the number of copies which is available for issue

  Result: pass

- A book may have more than one copy. But every copy with the same ISBN number should have same detail information.

  Result: pass

View student detail

Pass criteria:

- The detail view should show: Name, User name, ID,Email Address, Date/Time of Joining, Date/Time of most recent use.

  Result: Pass

8. Suspension / Resumption Criteria

- Any test suspended before completion will be abandoned and resumed from the first step of the test.
- Any preconditions for that test will be re-established before the test is resumed.
- If a test cannot be completed due to continuous application failure, that will be noted in the report for that test case.

9. Test Deliverables

S.No	Deliverable Name	Author	Reviewer
1	Test Plan	Test Lead	Project Manager/Business Analyst
2	Functional Test Cases	Test Team	Business Analyst's Signoff
3	Logging Defects	Test Team	Test Lead/Programming Lead

4	Daily reports	Test Team/Test Lead	Test Lead/Project Manager
5	Test Closure Report	Test Lead	Project Manager

10. Testing Tasks

1. Scope and estimate

As an agile tester, you'll help estimate the scope and size of the testing effort for each user story. The estimated effort for testing is part of the overall estimation for the size of the user story, which can't be marked as "done" until it passes all the tests. After each sprint, your team will review and update the estimates of upcoming user stories based on the team's experience from the previous sprint and re-plan upcoming sprints based on the new estimates, which should be improving over time.

2. Assess testability

You'll be involved in the design of the software by working closely with developers to assess and advise on testability aspects. You'll also be looking at concerns such as whether software testing can be automated, whether components can be tested independently from the rest of the package, and how much information is written to the log files.

3. Design and execute test cases

On an agile project, everyone on the team plays a role in testing. Each team member might have their own specialty, but everyone is responsible for delivering the team's user stories at the end of the sprint. The team will be writing functional, performance, and automated unit tests, as well as creating scripts to automatically deploy code into test environments and execute the tests. As a tester on the team, you'll be helping to design and execute automated and manual tests, including exploratory testing. As testing is infused throughout the development process, you'll become involved in testing at the component and API level, as well as at the end-to-end and feature level. You'll also be testing those nonfunctional requirements teams sometimes refer to as the "ilities": security, reliability, maintainability, scalability, usability, and so on.

4. Automate

In an agile development environment, there are frequent small-functionality increments at the end of each sprint, which means the software is continually changing. The frequency of change makes the speed of regression testing incredibly important, because the code should be tested every time a change is committed. This means you need to automate your tests as much as possible—manual testing simply takes too long. Look for opportunities to automate tests and deployment scripts and develop test automation frameworks for your team and the rest of the agile release train.

5. Collaborate

You'll be working more closely with developers than ever before. If you find a defect, tell the developer, and let them use your system to debug so they can find and fix the problem as quickly as possible. Don't force them to set up their own system. You're working together on the same code and user story, with the same goal of providing working software at the end of the sprint.

You'll also sometimes need to help members of the team who require assistance in completing a user story that hasn't progressed as planned.

Keep in mind that collaboration in larger organizations differs from the ideal of having colocated teams in pure agile. In addition to the unavoidable interaction with enterprise bureaucracy and the need to work with non-agile teams, agile team members in large organizations often work with offshore colleagues. This introduces challenges such as different time zones, languages, and cultures. When you're colocated, much of the valuable information (and gossip) is exchanged by the water cooler or over a casual cup of coffee. Keep your off-site colleagues in the loop.

6. Verify fixes

OK. This doesn't sound new. Yesterday, you were working on verifying fixes, too. But these fixes might have been made weeks or months ago, and you could only test them when a formal build was delivered to QA. In agile though, the aim is to fix and verify bugs within the same sprint, because otherwise the tests won't pass, and the user story can't be considered "done."

In larger organizations, there might still be occasions when it's not possible to fix a defect in the same sprint, perhaps because you're working with another part of the organization that isn't aligned with your goals. Many organizations include an innovation and planning sprint, which gives you an opportunity to verify fixes later, but still within the set of sprints that make up the program increment.

7. Attend daily stand-up meetings

It's important to attend and contribute to the daily stand-up meetings. To be really effective, don't just talk about what you accomplished yesterday and what you're going to be doing today. The most important part of a daily stand-up meeting is sharing the obstacles that will prevent you from making progress as a tester on the team.

Focus on identifying and describing these obstacles to the rest of the team and enlist the help of the team to remove them. Eric Jacobson has a number of examples in his article on things a tester should say at a daily setup, including reviewing the steps to reproduce a specific defect, deciding which defects the developers should fix first because they're a barrier to testing the rest of the system, and so on.

8. Track different metrics

Yesterday, when you were part of a QA team, you followed metrics that were important to your organization, such as the status of requirements, number of reopened defects, etc. Today, you'll be looking at a new set of metrics that you'll need to track as part of an agile organization, such as sprint burndown, velocity, and release burndown.

9. Fail

In agile, failure is accepted, and the lessons learned from failures are shared with the team. Support from management to accommodate failure is critical to the success of agile in the enterprise.

11. Environmental Needs

This subsection contains the properties that are needed to test the Library Management application. An environment matching the following criteria can be used in tests, and will later be referred to as Acceptance Server.

- The environment will have a network connection. Preferably the environment will have a wireless connection to simulate the network environment under which a typical user will access the Library Management application.
- The environment will support Android applications.
- The environment will have API 19 or above.
- The environment will access a test application database, used for storing application usage statistics.

12. Responsibilities

The roles and responsibilities of each team member during the testing process are listed in the table below. These roles are intended to be fluid, and each member may assume various roles as needed during testing.

Name	**Role**	**Responsibilities**
Palak Singhal	Test Lead, Tester	Oversee and coordinate test process. Execute test procedures and augment automated tests.
Sharanya Kamath	Recorder, Tester	Record test results. Execute test procedures and augment automated tests.

13. Staffing and Training Needs

This section describes any specific training and staffing needs that are required to deliver the acceptance test plan. The Responsibilities section and Approach section of this document will affect the staffing and training needs.

The main benefit of this section is that it communicates the resource requirements and seeks the approval for these resources to support the delivery of the test plan i.e. the types and numbers of business users required for acceptance testing

Training needs

Consideration needs to be given to areas such as:

- Product training of test analysts on the application or system
- Training in test design techniques for business users involved in UAT
- Use of test execution and reporting tools for all users

It is important to remember to include the training that will be delivered by the test team to third parties, to facilitate use of shared test tools, for instance.

Staffing needs

It is also important to identify and communicate the team size and resources required to deliver the test plan i.e. the numbers and types of business users required for acceptance testing.

A description and distribution of tasks should be provided in high level terms, providing information on the numbers of individuals required for each role and stating if multiple roles will be combined for individuals. There should also be an indication when and for how long each resource will be required.

14. Testing Schedule

Activity	**Description**	**Exit Criteria**
Alpha Testing	All tests are executed and recorded according to specified test procedures.	All tests have been executed and reported. 4th-6th April
Alpha Defect Correction	Correct critical defects, and as many non-critical defects possible.	All critical defects have been corrected, and all non-critical defects have been

		corrected or the time limit for this activity has passed. 6th-8th April
Beta Testing	All tests are executed and recorded according to specified test procedures.	All tests have been executed and reported. 9th-11th April
Server Testing	The application doesn't require a server as it runs on local machine at present.	-
Beta Defect Correction	Correct critical defects, and as many non-critical defect as were possible.	All critical defects have been corrected,and all non-critical defects have been corrected or the time limit for this activity has passed. 11th April
Test Reporting	Use the results of testing to generate the test report.	The test report is complete. 12-13th April

15. Risks And Contingencies

This subsection contains a list of the possible risks that are most likely to affect the testing schedule and the ability to deliver the software according to schedule.

Risk	**Probability**	**Impact**	**Mitigation Plan**
SCHEDULE Testing schedule is tight. If the start of the testing is delayed due to design tasks, the test cannot be extended beyond the UAT scheduled start date.	High	High	The testing team can control the preparation tasks (in advance) and the early communication with involved parties. Some buffer has been added to the schedule for contingencies, although not as much as best practices advise.
RESOURCES Not enough resources,	Medium	High	Holidays and vacation have been estimated and built into the

resources on boarding too late (process takes around 15 days			schedule; deviations from the estimation could derive in delays in the testing.
DEFECTS Defects are found at a late stage of the cycle or at a late cycle; defects discovered late are most likely be due to unclear specifications and are time consuming to resolve.	Medium	High	Defect management plan is in place to ensure prompt communication and fixing of issues.
SCOPE Scope is completely defined	Medium	Medium	Scope is well defined but the changes are in the functionality are not yet finalized or keep on changing.
Natural disasters	Low	Medium	Teams and responsibilities have been spread to two different geographic areas. In a catastrophic event in one of the areas, there will resources in the other areas needed to continue (although at a slower pace) the testing activities
Non-availability of independent test environment and accessibility	Medium	High	Due to non availability of the environment, the schedule gets impacted and will lead to delayed start of test execution.
Delayed testing due to new issues	Medium	High	During testing, there is a good chance that some “new” defects may be identified and may become an issue that will take time to resolve. There are defects that can be raised during testing because of unclear document specification. These defects can yield to an issue that will need time to be resolved.

16. Testing Specifications

TEST RESULT: UNIT TESTING

LOGIN FORM:

Sr.No.	Test Case	Expected Result	Test Result
1.	Enter valid name and password & click on login button	Software should display profile window	Successful
2.	Enter invalid	Software should display error message.	Successful

BOOK ENTRY FORM:

Sr.No.	Test Case	Excepted Result	Test Result
1	On the click of ADD button	At first user have to fill all fields with proper data, if any error like entering text data instead of number or entering number instead of text, is found then it gives proper message otherwise adds record to the database.	Successful
2.	On the Click of SEARCH Button	Displays the Details of all books in the database and user can search for the book he wants.	Successful

USER ACCOUNT FORM:

SL.No	Test Case	Excepted Result	Test Result
1.	On the click of ADD button	At first user have to fill all fields with proper data , if any Error like entering text data instead of number or entering number instead of text..is found then it gives proper message otherwise Adds Record To the Database	Successful
2.	On the Click of Transaction log Button	Displays the Details of book for entered Register no. Otherwise gives proper Error message.	Successful
3.	On the Click of LOGOUT button	Logs out of profile window	Successful
4.	On the click of LOGIN button.	Logs into profile window if username and password exist in the database. Else error message is displayed.	Successful

BOOK ISSUE FORM:

SL.No	Test Case	Excepted Result	Test Result
1.	On the Click of ISSUE button	If the quantity of entered book is less than the quantity of book in database then book is issued and its quantity decreases.	Successful