
American International University-Bangladesh (AIUB)
Department of Computer Science
Faculty of Science &Technology (FST)
Summer 22 23

Section: A
Software Quality Assurance and Testing

# Library Management System

A Report submitted By



| SN | Student Name | Student ID |
|---|---|---|
| 1 | SUNJIDA NOURIN SHATHI | 20-42597-1 |
| 2 | MIRZA MOHIUDDIN | 20-43598-1 |
| 3 | ADRITA RAHMAN BUSHRA | 20-43367-1 |



**Checked By Industry Personnel**

Name:

Designation:

Company:

Sign:

Date:

# Software Test Plan

for

# <Library Management System>

Version 1.0 approved

Prepared by<Sunjida Nourin Shathi, Mirza Mohiuddin, Adrita Rahman Bushra>

<American International University-Bangladesh>
<16 August, 2023>

**Checked By Industry Personnel**

Name:

Designation:

Company:

Sign:

Date:

# Table of Contents



# Revision History

| Revision | Date | Updated by | Update Comments |
|---|---|---|---|
| 0.1 | 05/08/2023 | Sunjida Nourin Shathi | First Draft |
| 0.2 | 07/08/2023 | Adrita Rahman Bushra | Second Draft |
| 0.3 | 08/08/2023 | Mirza Mohiuddin | Third Draft |
| 0.4 | 11/08/2023 | Adrita Rahman Bushra | Fourth Draft |
| 0.5 | 13/08/2023 | Sunjida Nourin Shathi | Fifth Draft |
| 0.6 | 15/08/2023 | Mirza Mohiuddin | Final Draft |

# 1. TEST PLAN IDENTIFIER: RS-MTP01.3

# 2. REFERENCES


- Knop, M., Mueller, M., & Niehaves, B. (2021). Investigating the use of telemedicine for digitally mediated delegation in team-based primary care: Mixed methods study. Journal of medical Internet research, 23(8), e28151.
- Software Quality and Testing Course PowerPoint Slides.
- Anderson, R. B., White, L. E., & Patel, S. K. (2023). Enhancing User Experience in Library Management Systems through Responsive Design. Library Technology, 15(3), 67-82.


# 3. INTRODUCTION

## Background to the Problem

In our university's library, there is a notable absence of a modern and efficient system for managing and accessing the library's resources. Currently, students and library staff face challenges in effectively maintaining and organizing the extensive collection of books available. The absence of a centralized digital platform makes it laborious for both students and staff to obtain crucial information, such as a comprehensive booklist and the available quantity of each book.

Presently, the process of discovering and locating desired books relies heavily on manual methods, consuming valuable time and resources. Students are required to engage in time-consuming searches through physical catalog cards or laboriously peruse the shelves. This not only hampers the efficiency of the library's operations but also poses an inconvenience for students who seek to explore and access specific books. It will be beneficial for students to use a self-centered system while discovering his/her desired books.

## Solution to the Problem

We are dedicated to revolutionizing our university's library experience through the creation of an innovative Library Management System. This cutting-edge solution will empower students with easy access to the library's extensive booklist and real-time availability, ensuring informed decisions on their selections. Through a user-centric interface, students can autonomously borrow and return books using the self-service system, eliminating manual processes. If a student delay to return the borrow book a punishment amount automatically will be added for the student after the deadline.

Existing software solutions offer various avenues for achieving these goals. Solutions like Koha, an open-source Integrated Library System (ILS) etc. Evaluating these options will help us tailor the perfect system that seamlessly aligns with our university's unique requirements and enhances the library experience for our students.

# 4. REQUIRMENT SPECIFICATION

## 4.1 System Features

**1. System Sign Up**

Functional Requirements:

1.1. Users can access the Sign Up page from the home screen.
1.2. Users must provide a unique username, valid email address, and a strong password.
1.3. The system validates the email address format and ensures that the username is not already taken.
1.4. Upon successful submission, a confirmation email is sent to the user for verification.
1.5. Users can verify their email by clicking on the verification link in the email.
1.6. After email verification, users are registered and can proceed to log in.

Priority: High
Pre-conditions: None

**2. System Login**

Functional Requirements:

2.1. Users can access the Login page from the home screen.
2.2. Users must provide their registered email and password.
2.3. The system validates the entered credentials against the database.
2.4. Upon successful login, users are directed to the User Interface.

Priority: High
Pre-conditions: User must be registered and have a verified email address.

**3. Forgot Password**

Functional Requirements:

3.1. Users can access the "Forgot Password" link on the Login page.
3.2. Users must enter their registered email address for password recovery.
3.3. The system sends a password reset link to the user's email.
3.4. Users can click the link to reset their password and set a new one.

Priority: Medium
Pre-conditions: User must have a registered and verified email address.

**4. View User Interface**

Functional Requirements:

4.1. After successful login, users are directed to the User Interface.
4.2. The User Interface displays a dashboard with relevant information and options.
4.3. Users can view their borrowed books, fines, and account details.
4.4. Users can access the Book Category section and other library-related features.

Priority: High
Pre-conditions: User must be logged in.

**5. View Book Category**

Functional Requirements:

5.1. Users can navigate to the Book Category section from the User Interface.
5.2. The system displays a list of available book categories.
5.3. Users can select a specific category to view books within that category.
5.4. Each book entry includes details such as title, author, and availability status.

5.5. Users can check the quantity of available books in each category.

Priority: Medium
Pre-conditions: User must be logged in.

**6. Change Password**

Functional Requirements:

6.1. Users can access the Change Password section from their profile settings.
6.2. Users must enter their current password and the new desired password.
6.3. The system verifies the current password before allowing a password change.
6.4. After successful validation, the password is updated.

Priority: High
Pre-conditions: User must be logged in.

**7. Delete Book**

Functional Requirements:

7.1. Admin users can access the Delete Book section from the administrative dashboard.
7.2. Admin users can search for a book by title, author, or other criteria.
7.3. The system displays a list of matching books with options for deletion.
7.4. Admin users can select a book for deletion, and the system prompts for confirmation.
7.5. Upon confirmation, the book is removed from the database.

Priority: High
Pre-conditions: User must have administrative privileges.

**8. Delete Profile**

Functional Requirements:

8.1. Users can access the Delete Profile section from their profile settings.
8.2. The system displays a warning message about the consequences of profile deletion.
8.3. Users must confirm their intent to delete the profile.
8.4. After confirmation, the user's profile is deleted, including borrowing history and fines.

Priority: Medium
Pre-conditions: User must be logged in.

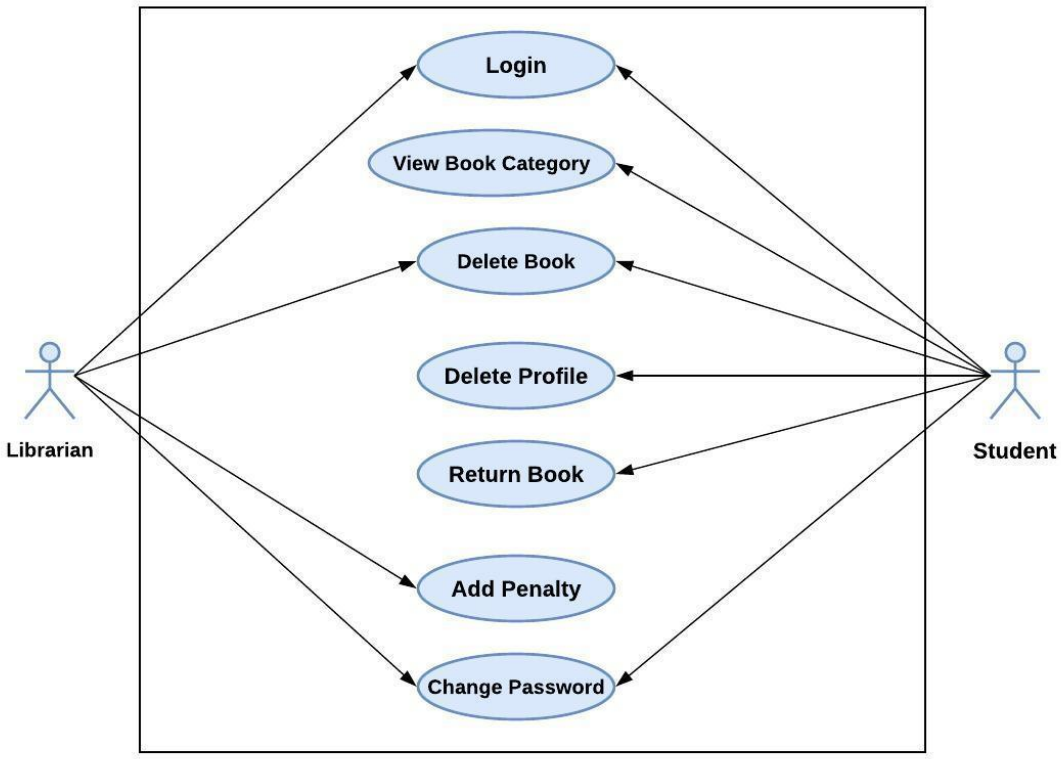



Figure 01: Use Case Diagram

## 4.2 System Quality Attributes

1. **Usability:** This can be estimated with regards to usability. The system ought to be easy to understand. An everyday customer of our very own ought to have the alternative to post about his/her gadgets on a regular of 5 minutes and a restrict of 7 minutes. Our framework highlights are not difficult to advance as route is particularly basic. It's Easy for a new or uncommon user to determine out how to utilize the framework.

2. **RELIABILITY AND CORRECTNESS:** No software is free from bug. Our software offers the actual output what user wants; its correctness has been ensured. Our software does not crash randomly as it has been tested and no false output is generated so It’s more reliable to use for our users.

3. **Modularity**: Each system should be developed according to our modules. Our system consists of many modules and integrated to make it a complete system. So, we can easily identify the bug in any module and then we need to fix only that particular module we don't have to worry about other modules so it makes the work easier for our tester. In addition to this we can add new features to our system as it is built in modules.

4. **Maintainability:** This means how without problems the maintenance team can perform

their work. The major task of our maintenance team is to fix bugs, add something new or exchange some features. One of our maintenance programmers will be able to exchange any function with 24-man hours or much less of development effort.

5. **Efficiency:** Main system quality features. Any given task in the system is measured in terms of time required to complete it. The efficiency of the application will be large.

6. **TESTABILITY:** It means how easily the testing team can perform their work. Testability is highly dependent on modularity and since our system is built module wise the tester will go easy; they don't have to test every module to fix bugs.

7. **FLEXIBILITY:** The effort required to change an operational program. A maintenance programmer with at least one year of experience supporting this product should be able to provide a new copy of output for the product, including code changes and testing, with no more than 2 hours of labor.

## 4.3 System Interface


Figure 02: Home Page

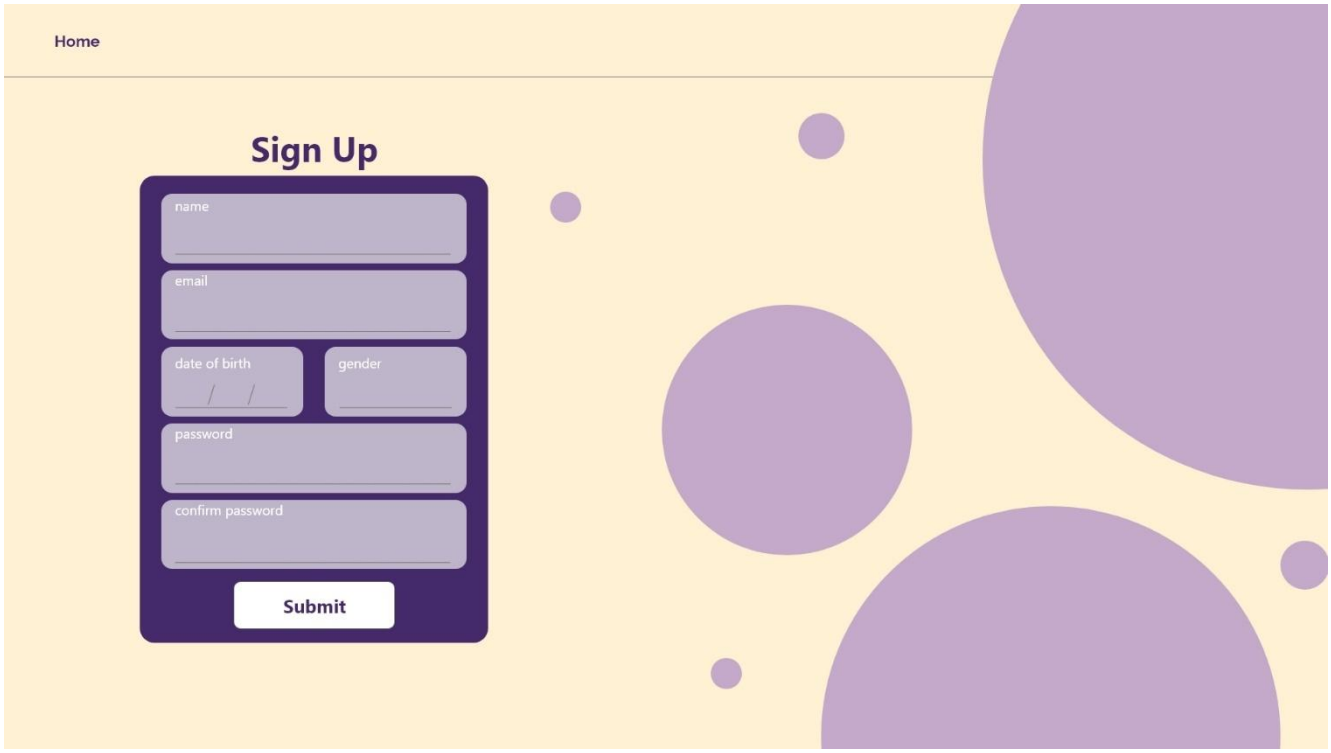



Figure 03: Sign up Interface

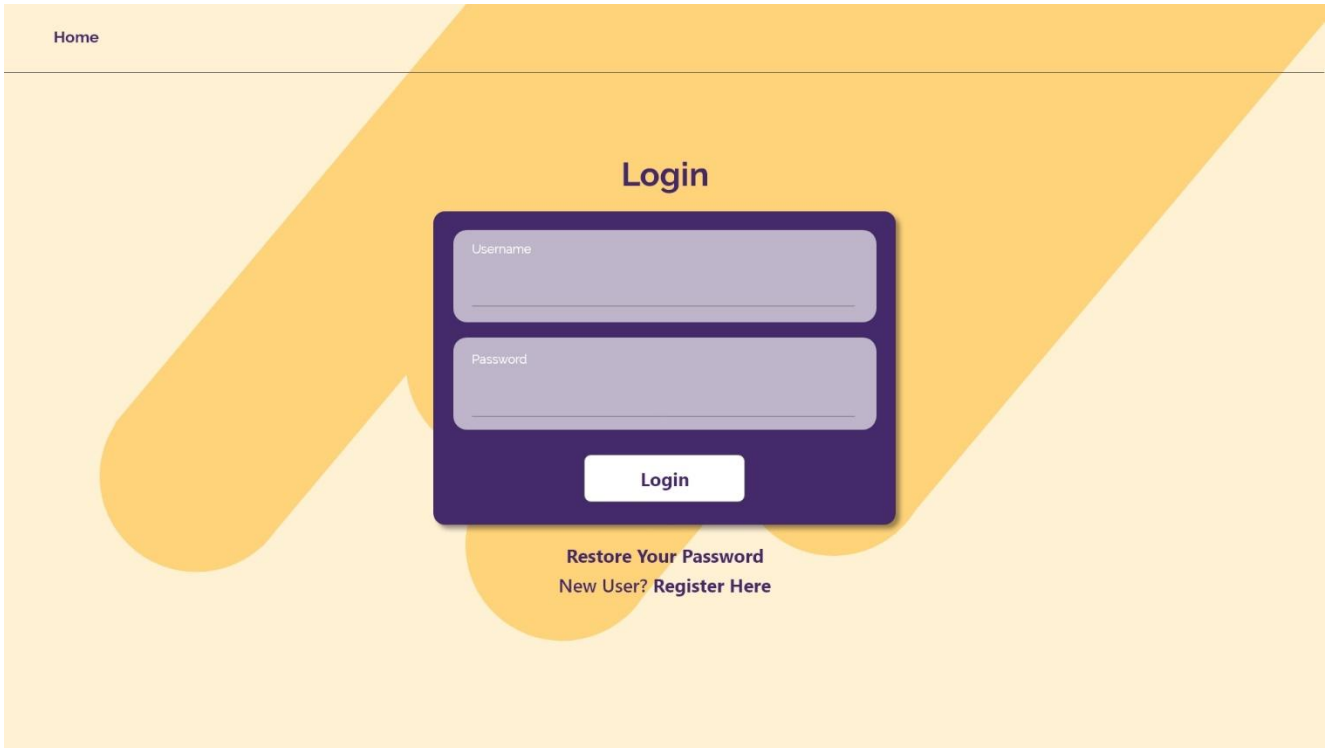



Figure 04: Login Interface

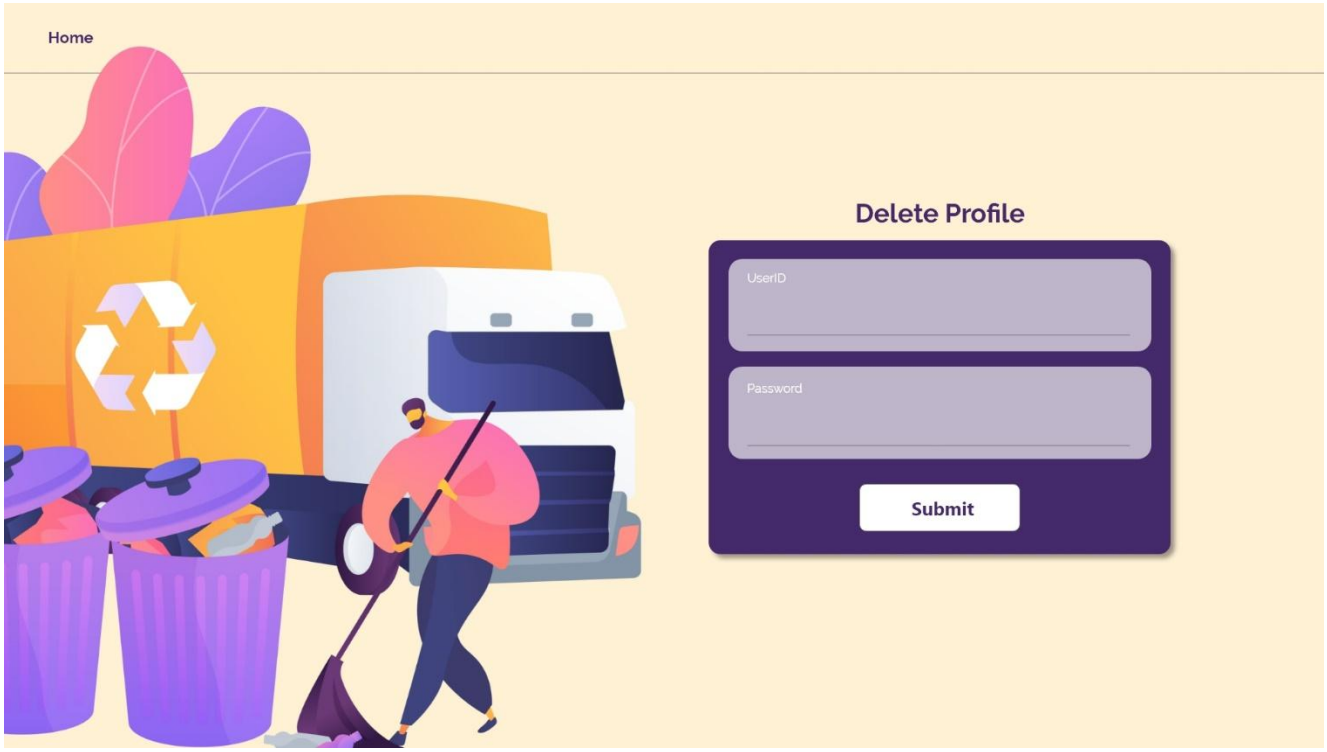



Figure 05: Delete Profile Interface


Figure 06: User Information Interface

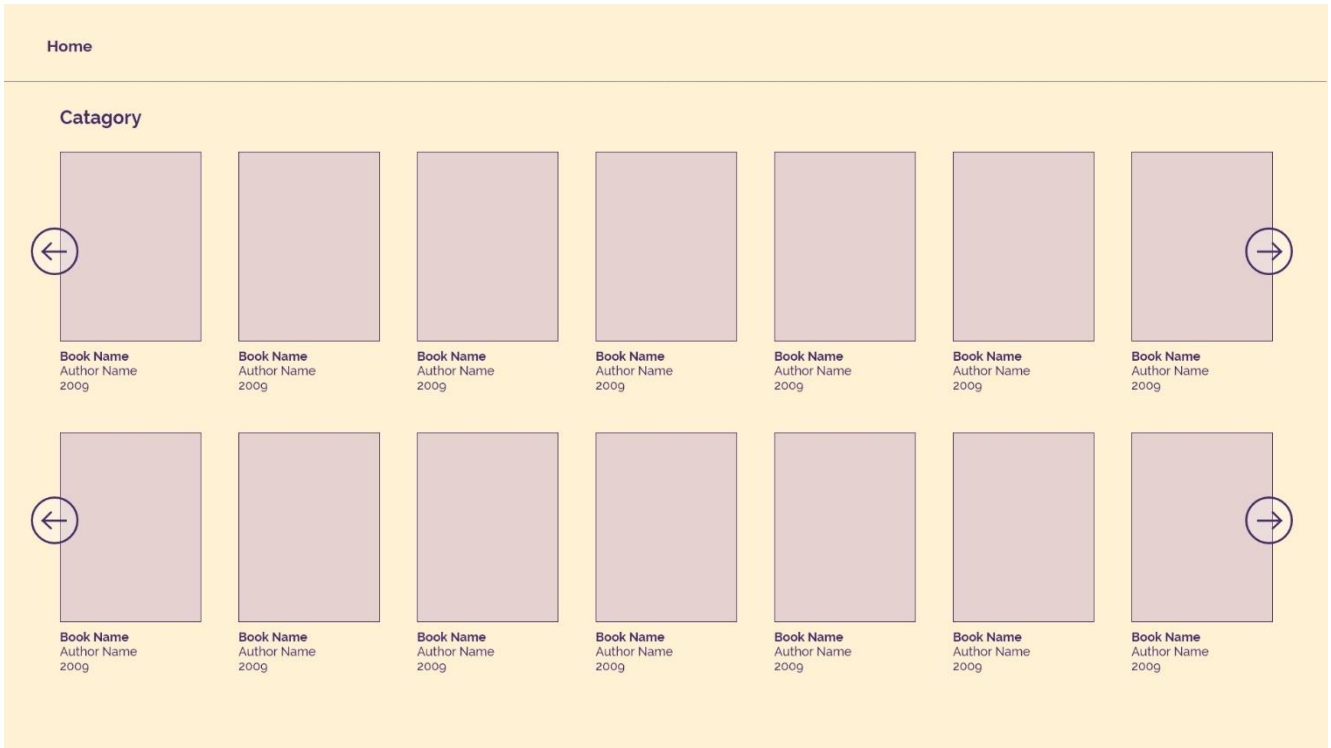



Figure 07: Book Category Interface

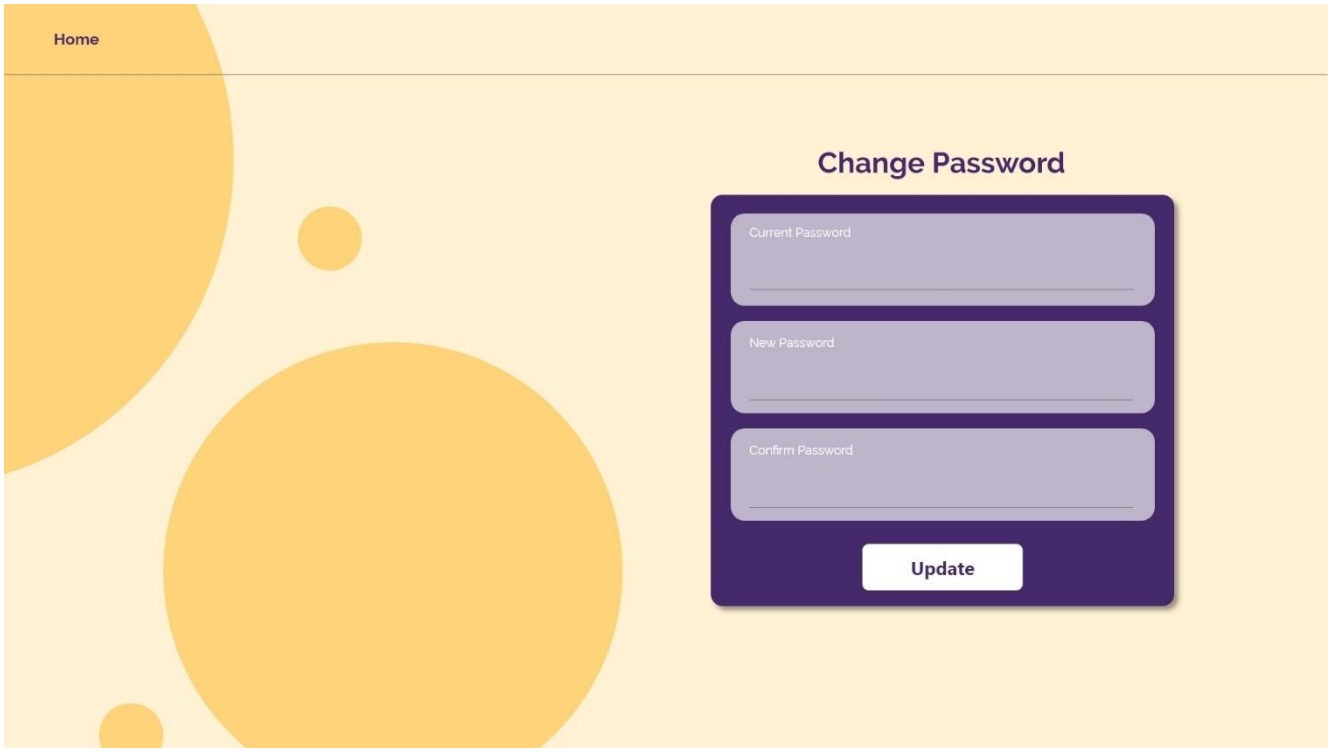



Figure 08: Change Password Interface

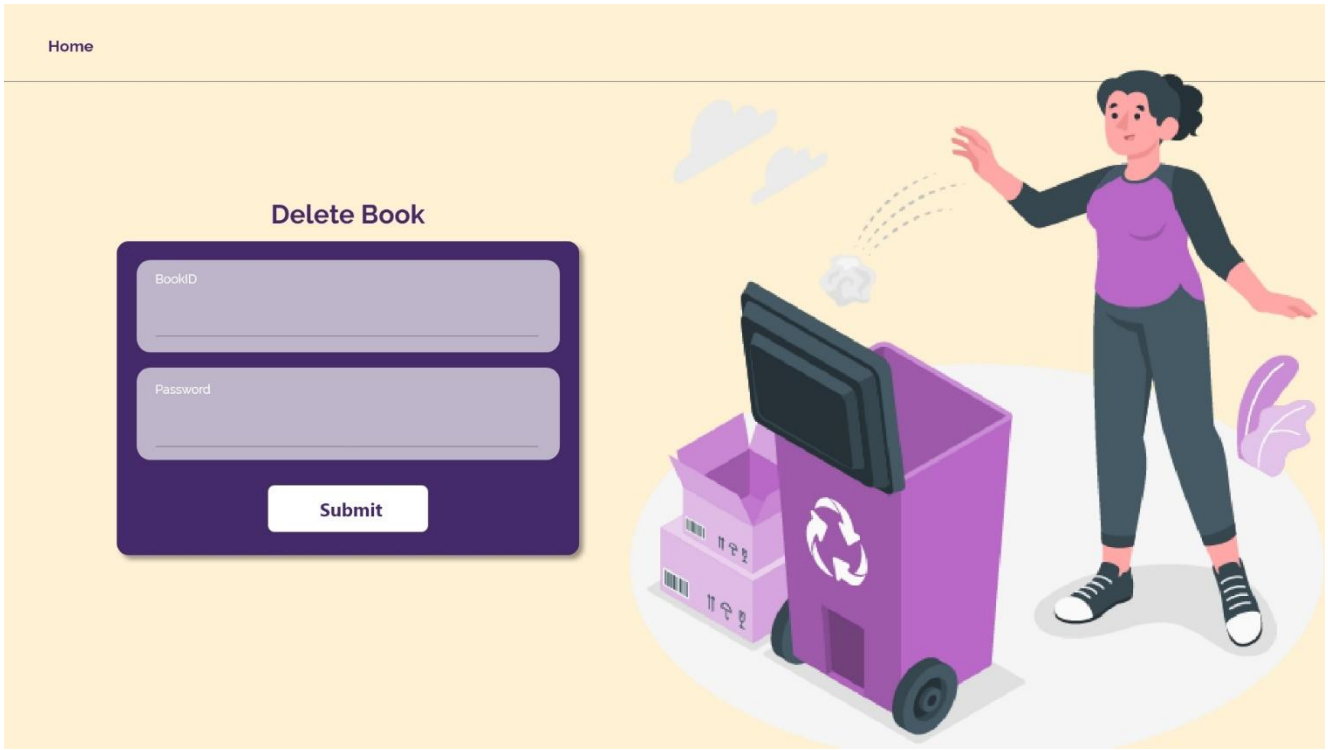



Figure 09: Delete Book Interface

## 4.4 Project Requirements

**Effort Estimation:**

Our project is to develop an application named “Library Management system”.
Development Time = 1 Months
Required number of peoples = 4

**Budget Estimation:**

Duration in weeks = 1*4 = 4 weeks
Office days = 5 days
Working hours = 8 Hours

So, Per week working hours is = (5*8) hours = 40 hours
So Total Working hours is = (40*4) hours
= 160 hours.

Developer salary is = 800 Taka

Total developers Salary = (800*160) Taka

= 128,000Taka

| Expanse | Amount | Total Amount |
|---|---|---|
| Salary for 4 developers | | 512,000 Taka |
| 1 month’s office rent | 1*8000 | 8,000 Taka |
| Electricity and other costs | | 10,000 Taka |
| 1 months Maintenance cost | 1*10*1200 | 12,000 Taka |
| Travel Cost | 1*2000 | 2,000 Taka |
| Total Cost | | 544,000 Taka |
| 15% of total cost(profit) | | 81,600 Taka |
| Now total budget is | | 625,600 Taka |

## 5. FEATURES NOT TO BE TESTED

The following is a list of the areas that will not be specifically addressed. All testing in these areas will be indirect as a result of other testing efforts. For example:

**Users can contact Librarian**: In our web application there is a feature where a user can contact with librarian. This feature is not to be tested in this release of the software.

**Automatic fine added**: In this system where a user can borrow a book from the library. There is a fixed time to returning the borrowing book so if a user returns the borrowing book after fixed time, he/she will be fined a significant amount for the delay. This feature is not to be tested in this release of the software.

**Librarian Can contact system admin**: This feature was also not been tested because of the same reason of user can contact librarian feature, due to low risk.

**User Can add own location information:** We have a feature in our system where a user can add own location but it is not mandatory at the first place and due to a tight schedule to follow so we opt out this feature to be tested of this version of the software.

# 6. TESTING APPROACH

## 6.1 Testing Levels

SYSTEM/INTEGRATION After completing the test of all the smallest part, the system must be connected together. The 100% tested all parts of the units are converted into a module where another test is done. It's called the integration testing. After adding all modules together, the regression testing is done to the units but in here, the main purpose is to test the interface of the system that all the connections are working perfectly or not. Also, it can be called data flow test.

- **Unit Testing:** Unit testing is the most common part of a testing where a small part of the project is going to be tested by the developers. After developing a small part of the project, this small part or portion is tested to get understand the system is working properlyor not. In here test personal should find out the correctness of the inaccessible code.

- **Integration Testing:** Then in the second phase we will do the integration. In this test we will ensure that all software modules are logically integrated and tested as a group. Our project consists of several software modules written by four programmers. The purpose of this level of testing is to find errors in the way different software modules interact when integrated in this step, we will follow the "bottom-up integration" approach.

- **System Testing:** Then we will do system testing. Through system testing we will checkfull-featured, fully integrated systems. Then we will verify if it meets all the requirements. Black-box testing falls under this condition. So, at this level, we will follow the "Black Box Testing" technique.

- **Acceptance Testing:** The last phase of our testing is acceptance testing. We will perform this check to check the acceptability of our products. This test will becompleted to check if any errors have been missed throughout the functional testing phase. At this level, we will follow the "Black Box Testing" technique. After that, we can run unit tests again. The customer will do ACCEPTANCE testing with the help of the test manager and development team leader. After the System/Integration test is completed, the acceptance test will run in parallel with the existing manual ZIP/FAX process for one month.

## 6.1 Test Tools

The only test tools to be used are the standard AS/400 provided utilities and commands.

- The Program Development Manager (PDM) is used as the source version configuration management tool in conjunction with the in-house check-in/check-out control utility. The check-in/out utility is part of each developer's standard AS/400 access menu. Only the standard AS/400 provided utilities and commands are to be used as test tools. Therefore, there is no explicit mention of a testing tool name other than the standard utilities and commands provided by the AS/400 platform.

## 6.2 Meetings

The testing team will meet every week and judge the system quality and preparing for the next step. The team leader will meet the developer and testers timely and check the progress of the system. Then the project manager will check the overall process of the system time to time. These two meetings are arranged on different weeks. Emergency meeting are called as needed for emergency situation.

# 7. TEST CASES

## 7.1 Log in

This Test case will Test the website login page by verifying valid username and password

| | | | | |
|---|---|---|---|---|
| Project Name: Library Management System | | | Test Designed by: Sunjida Nourin Shathi | |
| Test Case ID: TC_001 | | | Test Designed date: 05/08/2023 | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: Sunjida Nourin Shathi | |
| Module Name: Login | | | Test Execution date: 06/08/2023 | |
| Test Title: Verifying login with valid username, password | | | | |
| Description: Test the website login page | | | | |
| Precondition (If any): Users must type valid username & password | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Go to the website.<br>2. Click the "Login" button.<br>3. Enter username<br>4. Enter password.<br>5. Click "Login" button | Username:Shathi_45<br><br>Password:<br>Shathi445 | Users should Login to the website | As expected, | Pass |
| Post Condition: User is validated with database and successfully registered into the application. The account session details are logged in the database. | | | | |

### 7.2 Registration:

This Test case will check the registration with valid username, email, passwordand date of birth, gender, Password and Confirm Password.

<table>
<tr><td colspan="3">Project Name: Library Management System</td><td colspan="2">Test Designed by: Mirza Mohiuddin</td></tr>
<tr><td colspan="3">Test Case ID: TC_002</td><td colspan="2">Test Designed date: 05/08/2023</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High</td><td colspan="2">Test Executed by: Mirza Mohiuddin</td></tr>
<tr><td colspan="3">Module Name: Registration</td><td colspan="2">Test Execution date: 06/08/2023</td></tr>
<tr><td colspan="5">Test Title: Registration with valid username, email, DOB,Gender and password.</td></tr>
<tr><td colspan="5">Description: Test the website registration.</td></tr>
<tr><td colspan="5">Precondition (If any): User must fill up all the input field.</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Go to the website.<br>2. Click the “registration” button<br>3. Enter all valid information.<br>4. Click “Submit” button</td><td>Username: Mirza_45<br>Gmail: mirzainfo.44@gmail.com<br>Date of birth: 19/11/1996<br>Gender: male<br>Password: mirza4455</td><td>Users should registration to the website</td><td>As expected,</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: User is validated with database and successfully register an account. Theaccount details are stored in the database</td></tr>
</table>

### 7.3 Log out:

This Test case will Test that the logout option is working properly or not

| Project Name: Library Management System | | | Test Designed by: Adrita Rahman Bushra | |
|---|---|---|---|---|
| Test Case ID: TC_002 | | | Test Designed date: 05/08/2023 | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: Adrita Rahman Bushra | |
| Module Name: logout | | | Test Execution date: 06/08/2023 | |
| Test Title: Verifying logout option | | | | |
| Description: Test the website logout option | | | | |
| Precondition (If any): 1. Need account on this website<br>2. Need to be logged in | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. go to the site<br>2. log in to the site<br>3. Click the "Logout" button | User id:adrita_01<br>Password: Adrita@78 | Successful logout | As expected, | Pass |
| Post Condition: User goes back to the home page | | | | |

## 7.4 Change Password:

This Test case will check the change password feature is working accurately or not

| | | | | |
|---|---|---|---|---|
| Project Name: Library Management System | | | Test Designed by: Adrita Rahman Bushra | |
| Test Case ID: TC_004 | | | Test Designed date: : 05/08/2023 | |
| Test Priority (Low, Medium, High): High | | | Test Executed by:Mohiuddin Mirza | |
| Module Name: Update Password | | | Test Execution date: : 05/08/2023 | |
| Test Title: Can change password from the profile. | | | | |
| Description: Test password is update or not. | | | | |
| Precondition (If any): User Must have to login into profile. | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. go to the site<br>2. log in to the site<br>3. Input old password.<br>4. Input new password<br>5. Confirm new password<br>4. click “save” to change password | New Password=admus#11 | Successful update password | As expected, | Pass |
| Post Condition: The updated password is stored in the database | | | | |

### 7.5 Show books:

This Test case will check that the feature is showing Books with their corresponding detailsor not.

| Project Name: Library Management System | | | Test Designed by: Sunjida Nourin Shathi | |
|---|---|---|---|---|
| Test Case ID: TC_005 | | | Test Designed date: 05/08/2023 | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: Adrita Rahman Bushra | |
| Module Name: show Books | | | Test Execution date: 06/08/2023 | |
| Test Title: show all the books with their corresponding details | | | | |
| Description: Test all the books are visible or not | | | | |
| Precondition (If any): | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. go to the site<br>2. log in to the site<br>3. go to the books page | User id:shathi_01<br>Password:<br>Shathi@78 | Can see all the books with details | As expected, | Pass |
| Post Condition: N/A | | | | |

### 7.6 Book Status:

This feature shows the status of the book in borrow or not.

| Project Name: Library Management System | | | Test Designed by: Mirza Mohiuddin | |
|---|---|---|---|---|
| Test Case ID: TC_006 | | | Test Designed date: 05/08/2023 | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: Sunjida Nourin Shathi | |
| Module Name: Status | | | Test Execution date: 06/08/2023 | |
| Test Title: Show the book is under borrow or not | | | | |
| Description: Book Status | | | | |
| Precondition (If any): The book name should be available intothe book database | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Go to checkbook status<br>2. In search box, give a book name<br>3. Press check status | 1.Book Name: Teach yourself C++<br><br>2.Current Status: Borrowed | Show a message that book is under borrow | As expected, | Pass |
| Post Condition: N/A | | | | |

### 7.7 Add or remove a book:

This test is executed to add or remove a book from the central database viaadmin

| | | | | |
|---|---|---|---|---|
| Project Name: Library Management System | | | Test Designed by: Adrita Rahman Bushra | |
| Test Case ID: TC_007 | | | Test Designed date: 05/08/2023 | |
| Test Priority (Low, Medium, High): High | | | Test Executed by: Sunjida Nourin Shathi | |
| Module Name: Add Remove | | | Test Execution date: 05/08/2023 | |
| Test Title: Add or remove a book | | | | |
| Description: Book Status | | | | |
| Precondition (If any): Add and remove the book into centraldatabase | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Go to login<br><br>2. Give Id and passwordas a valid admin<br><br>3. Go to Add Book/ Remove book in central database<br><br>4. Type the book name and quantity to be added or removed<br><br>5. Press add or remove | 1. Book Name: Data<br>2. Communication and Network<br><br>3. Book Amount: 25<br><br>4. Press Add | The central database will be updated and show the add or removedbook from the central database page | As expected, | Pass |
| Post Condition: The login session is going to be stored into thecentral login database and the book adding time will be written at the book add column with time and date. | | | | |

### 7.8 Library Card issue:

It will generate a library card for the users

| Project Name: Library Management System | | | Test Designed by: Sunjida Nourin Shathi | |
|---|---|---|---|---|
| Test Case ID: TC_008 | | | Test Designed date: 05/08/2023 | |
| Test Priority (Low, Medium, High): Low | | | Test Executed by: Mirza Mohiuddin | |
| Module Name: Card | | | Test Execution date: 06/08/2023 | |
| Test Title: Card Generate | | | | |
| Description: An existing user will be applied for the card | | | | |
| Precondition (If any): The librarian must be a valid andusername must be inside the user database | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Go to Login<br>2. Login as librarian using valid name and Id<br>3. Press the card generate<br>4. Give the username into card generate form<br>5. Press Go | 1. Username: Mr mirza<br><br>2. Status in database: Valid | Card is issued anda date or card delivery inside the dashboard | As expected, | Pass |
| Post Condition: N/A | | | | |

### 7.9 Feedback:

User can give feedback or complain about the books or library stuff

| Project Name: Library Management System | | | Test Designed by: Mirza Mohiuddin | |
|---|---|---|---|---|
| Test Case ID: TC_009 | | | Test Designed date: 06/08/2023 | |
| Test Priority (Low, Medium, High): Low | | | Test Executed by: Adrita Rahman Bushra | |
| Module Name: Feedback | | | Test Execution date: 07/08/2023 | |
| Test Title: Feedback | | | | |
| Description: The user will give a feedback into the box | | | | |
| Precondition (If any): User must be validated | | | | |
| Test Steps | Test Data | Expected Results | Actual Results | Status (Pass/Fail) |
| 1. Log in as user<br>2. Go to feedback page<br>3.Write feedback | User Feedback: The books pages were torn apart | A complain will be generate into complain database which is accessible by the admin only | As expected, | Pass |
| Post Condition: N/A | | | | |

### 7.10 Remove a user:

This test case will check can we remove an existing user or not

<table>
<tr><td colspan="3">Project Name: Library Management System</td><td colspan="2">Test Designed by: Sunjida Nourin Shathi</td></tr>
<tr><td colspan="3">Test Case ID: TC_010</td><td colspan="2">Test Designed date: 06/08/2023</td></tr>
<tr><td colspan="3">Test Priority (Low, Medium, High): High</td><td colspan="2">Test Executed by: Adrita Rahman Bushra</td></tr>
<tr><td colspan="3">Module Name: Remove</td><td colspan="2">Test Execution date: 07/08/2023</td></tr>
<tr><td colspan="5">Test Title: Remove a user</td></tr>
<tr><td colspan="5">Description: The admin can remove a user</td></tr>
<tr><td colspan="5">Precondition (If any): The admin must have a valid account andthe user to be removed must be on the user database</td></tr>
<tr><td>Test Steps</td><td>Test Data</td><td>Expected Results</td><td>Actual Results</td><td>Status (Pass/Fail)</td></tr>
<tr><td>1. Log in as admin<br>2. Go to user information page<br>3. Search the username andID to be removed<br>4.Press remove</td><td>1. Username: Mr X<br>2. ID; 125478</td><td>The user will be removed from the user database</td><td>As expected,</td><td>Pass</td></tr>
<tr><td colspan="5">Post Condition: N/A</td></tr>
</table>

## 8. ITEM PASS/FAIL CRITERIA

Here we have implemented a total of 10 test cases. At first, when applied the test case to system 70% of the test cases were passed successfully and 30% were fail. The test cases were failing due to some query related issues on the database. When the test case was applied after solving query related problem, all the test cases are successfully passed.

## 9. TEST DELIVERABLES

- Test plan
- Test results documents
- Test summary
- Errors
- Bug report

## 10. STAFFING AND TRAINING NEEDS

This section shows how to staff the test jobs and prepare them for the work. Staffing is set for the duration of the project. It's realistic to assume that the vast majority of the staff will agree to do some testing. The following occupations are recognized:

Project Manager: Responsible for maturing the complete execution of the Web website. This includes creating requirements, managing the seller relationship, and overseeing the testing cycle.

Test Manager: Responsible for fostering the expert test strategy, examining the test deliverable, dealing with test cycles, collecting measurements and reporting progress to the Project Manager, and recommending when testing should be completed.

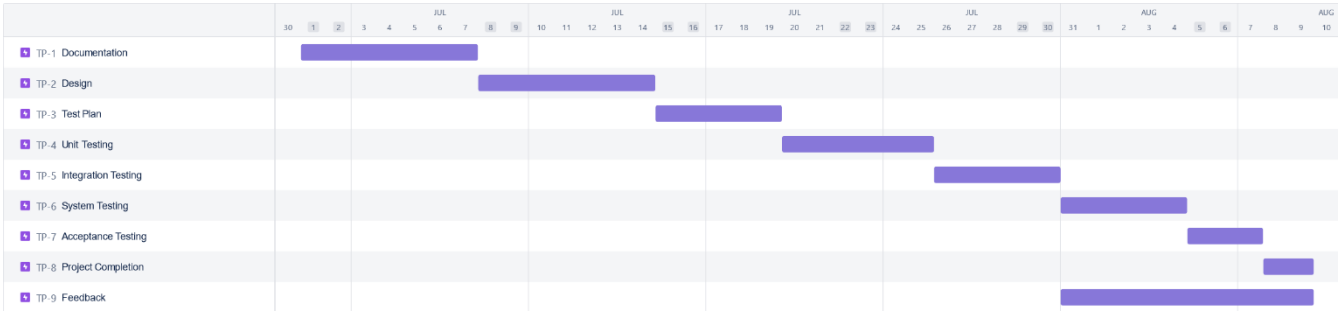
Test Engineer: Planning tests, creating test methods, creating test information, running tests, preparing occurrence reports, examining episodes, writing mechanized test strategies, and detailing measurements to the test administrator are all responsibilities of this position.

The test manager and test specialists should be familiar with the website development life cycle. Because this project is being developed in a traditional manner, this is a nonexclusive depiction of Staffing and Training requirements. As a result, the names of conscious people for each project aren't given.

## 11. RESPONSIBILITIES

| | TM | PM | Dev Team | Test Team | Client |
|---|---|---|---|---|---|
| Acceptance Test Documentation & Execution | X | X | | X | X |
| System Test Documentation & Execution | | | X | X | |
| Unit Test Documentation & Execution | | | X | X | |
| System Design Reviews | | | X | X | X |
| Details Design Reviews | | | | | |
| Test Procedures and rules | X | X | X | | |
| Screen & Report Prototype reviews | | | X | X | X |
| Change control regression testing | X | X | X | X | |

## 12. TESTING SCHEDULE



## 13. PLANNING RISKS AND CONTINGENCIES

**Risks Planning**

Technical, programmatic, and process risks are identified and categorized as part of software risk management, which then forms the basis of a plan that connects each to a mitigation approach. Throughout the project, the project manager keeps an eye on risk. If any do, a particular owner takes a mitigating step.

- Lack of encrypted data: Keep an eye on security and back up the data with highly encryption.
- Attempt unauthorized access: Consecutively three failed login attempts in an hour, the user will be restricted.
- Error in Functionalities: Regularly test the application and make a daily backup.
- Wrong SQL Command for Sensitive Data: Keep security scans and backups up to data.

**Contingency Planning**

A contingency plan in project management is a defined, actionable plan that is to be enacted if an identified risk becomes a reality. It is essentially a "Plan B", to be put in place when things go differently than expected.

- Network Failure: We will install two fiber optics connection from the different ISP as if one will be work as back up of another

## 14. APROVALS

| | |
|---|---|
| Project Sponsor | Boighor Library |
| Development Management | Sunjida Nourin Shathi |
| EDI Project Manager | Mirza Mohiuddin |
| RS Test Manager | Adrita Rahman Bushra |
| RS Development Team Manager | Mirza Mohiuddin |
| Reassigned Sales | Adrita Rahman Bushra |
| Order Entry EDI Team Manager | Sunjida Nourin Shathi |