

Darshan University

A Project Report on

**“Project Title”**

Under the subject

**Software Engineering (2301CS405)**

B. Tech, Semester – IV

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name: | Enrollment No.: |
| Academic Year  (2024-2025) | |
| Internal Guide  Prof. R. B. Gondaliya  Darshan University | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**DECLARATION**

We hereby declare that the SRS, submitted along with the **Software Engineering** **(2301CS405)** for entitled **“Project Title”** submitted in partial fulfilment for the Semester-5 of **Bachelor Technology (B. Tech)** in **Computer Science and Engineering (CSE)** Departmentto Darshan University, Rajkot, is a record of the work carried out at **Darshan University, Rajkot** under the supervision of R. B. Gondaliya and that no part of any of report has been directly copied from any students’ reports, without providing due reference.

(Student name)

Student’s Signature

Date: \_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**CERTIFICATE**

This is to certify that the SRS on **“Project Title” has** been satisfactorily prepared by **Student name** (**Enrollment**) under my guidance in the fulfillment of the course **Software Engineering (2301CS405)** work during the academic year 2024-2025.

|  |  |  |
| --- | --- | --- |
| Internal Guide  Prof. R. B. Gondaliya  Darshan University |  | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

**Acknowledgement**

I wish to express my sincere gratitude to my project guide Prof. R. B. Gondaliyaand all the faculty members for helping me through my project by giving me the necessary suggestions and advices along with their valuable co- ordination in completing this work.

I also thank my parents, friends and all the members of the family for their precious support and encouragement which they had provided in completion of my work. In addition to that, I would also like to mention the Darshan University personals who gave me the permission to use and experience the valuable resources required for the project from the University premises.

Thus, in conclusion to the above said, I once again thank the faculties and members of **Darshan University** for their valuable support in completion of the project.

Thanking You

**Student Name**

**ABSTRACT**

Library management system is a system which aims in developing a computerized system to maintain all the daily work of library. This system will act as a tool to transfer traditional library into digital library. This project has many features which are generally not available in general library system like facility of user login, books search on one click etc. It also has a facility of admin login through which the admin can monitor the whole system.  This system will be designed with the basic features such as librarian can add/view/update/delete books and students' details in it. It has also a facility where student after logging in their accounts can see list of books issued and its issue date and return date and also they can request the librarian to add new books by filling the book request form. The librarian after logging into his account i.e. admin account can generate various reports such as student report , issue report, teacher report and book report.

Main purpose of this system is to reduce human efforts as much as possible.

**Table of Contents**

[List of Figures I](#_Toc146706805)

[List of Tables II](#_Toc146706806)

[1 Introduction 1](#_Toc146706807)

[1.1 Product perspective 1](#_Toc146706808)

[1.2 Product features 1](#_Toc146706809)

[1.2.1 There are three different users who will be using this product: 1](#_Toc146706810)

[1.2.2 The features that are required for the Librarian are: 1](#_Toc146706811)

[1.2.3 The features that are required for the Member are: 1](#_Toc146706812)

[1.3 Functional Requirement 1](#_Toc146706813)

[1.3.1 Librarian 1](#_Toc146706814)

[1.3.2 Member 2](#_Toc146706815)

[1.4 Non-Functional Requirement 2](#_Toc146706816)

[1.4.1 Usability: 2](#_Toc146706817)

[1.4.2 Accuracy: 2](#_Toc146706818)

[1.4.3 Availability: 2](#_Toc146706819)

[1.4.4 Maintainability: 2](#_Toc146706820)

[2 Design and Implementation Constraints 3](#_Toc146706821)

[2.1 Use case diagram 3](#_Toc146706822)

[2.2 Activity diagram and Swimlane diagram 4](#_Toc146706823)

[2.3 Sequence diagram 5](#_Toc146706824)

[2.4 State diagram 6](#_Toc146706825)

[2.5 Class diagram 7](#_Toc146706826)

[2.6 Data flow diagram 8](#_Toc146706827)

[2.6.1 Context diagram (level-0) 8](#_Toc146706828)

[2.6.2 DFD Level-1 8](#_Toc146706829)

[2.6.3 DFD Level-2 9](#_Toc146706830)

[3 External interface requirement (Screens) 10](#_Toc146706831)

[3.1 Screen-1: Registration Form 10](#_Toc146706832)

[3.2 Screen-2: Login Form 11](#_Toc146706833)

[3.3 Screen-3: Add borrower book 12](#_Toc146706834)

[4 Database design 13](#_Toc146706835)

[4.1 List of Tables 13](#_Toc146706836)

[5 Stories and Scenario 14](#_Toc146706837)

[5.1 Story-1: Add New Book in Library Catalogue 14](#_Toc146706838)

[5.1.1 Scenario# S1.1 14](#_Toc146706839)

[5.1.2 Scenario# S1.2 14](#_Toc146706840)

[5.1.3 Scenario# S1.3 14](#_Toc146706841)

[5.2 Story-2: Search Book 15](#_Toc146706842)

[5.3 Story-3: Manage due date for borrowed book 15](#_Toc146706843)

[5.4 Story-4: Renew book 15](#_Toc146706844)

[5.5 Story-5: Generate a report on book usage and availability 15](#_Toc146706845)

[6 Test cases 16](#_Toc146706846)

[7 References 20](#_Toc146706847)

# List of Figures

[Figure 2.1‑1 Use case diagram for library management system 3](#_Toc146706848)

[Figure 2.2‑1 Activity diagram for Book Issue 4](#_Toc146706849)

[Figure 2.2‑2 Swimlane diagram for Book Issue 4](#_Toc146706850)

[Figure 2.3‑1 Sequence diagram for Book Issue 5](#_Toc146706851)

[Figure 2.4‑1 State diagram of Book 6](#_Toc146706852)

[Figure 2.4‑2 State diagram for Librarian 6](#_Toc146706853)

[Figure 2.5‑1 Class diagram for Library management system 7](#_Toc146706854)

[Figure 2.6‑1 Context diagram for Library management system 8](#_Toc146706855)

[Figure 2.6‑2 DFD level-1 for Library management system 8](#_Toc146706856)

[Figure 2.6‑3 DFD level-2 for Issue book 9](#_Toc146706857)

[Figure 3.1‑1 Screen-1: Registration Form 10](#_Toc146706858)

[Figure 3.2‑1 Screen-2: Login Form 11](#_Toc146706859)

[Figure 3.3‑1 Screen-3: Add borrower book 12](#_Toc146706860)

# List of Tables

[Table 3.1‑1 Screen element of Registration form 10](#_Toc146706861)

[Table 3.2‑1 Screen element of Login form 11](#_Toc146706862)

[Table 3.3‑1 Screen element of Add borrower book 12](#_Toc146706863)

[Table 4.1‑1 Table: Book 13](#_Toc146706864)

[Table 4.1‑2 Table: Borrower 13](#_Toc146706865)

[Table 4.1‑3 Table: Staff 13](#_Toc146706866)

[Table 4.1‑4 Table: Student 13](#_Toc146706867)

# Introduction

## Product perspective

This project is basically updating the manual library system into an internet-based application so that the users can know the details of their account, availability of books etc. It is a multi-user version and can take care of all the fundamental functions of a Library like Cataloguing, Circulation, Accessioning and Housekeeping. It can satisfactorily cater to all the basic functions of a small library.

## Product features

### There are three different users who will be using this product:

* Librarian who will be acting as the administrator.
* Member who will be accessing the library.
* Guest who will request for membership.

### The features that are required for the Librarian are:

* Control the movement of books and other material and avoid losing the same.
* Search if you have a specific book in your collection based on the title, author etc.
* Print the spine labels for the book.
* Find what a specific person has borrowed from you. It offers the following modules Cataloguing, Circulation, and Queries.
* Can issue a book to the student.
* Can view the list of books available in each category.
* Can take the book returned from students.
* Add books and their information of the books to the database.
* Edit the information of the existing books.
* Can check the report of the issued Books.
* Can access all the accounts of the students.

### The features that are required for the Member are:

* Can view the different categories of books available in the library.
* Can view the List of books available in each category.
* Can own an account in the library.
* Can view the books issued to him.
* Can put a request for a new book.

## Functional Requirement

### Librarian

* Add Article: New entries must be entered in database
* Update Article: Any changes in articles should be updated in case of update3.
* Delete Article: Wrong entry must be removed from system
* Membership management: To maintain a detailed database of the members. The system records the name, ID and password of each user. The system helps in ascertaining the track record of the member.
* Bar-coding: To give specific identification to each book. All books, old and new, are bar-coded on the basis of title, author, topic and date of publishing.
* Barcode scanning: To read the barcode easily using RFID sensors. The database is automatically updated when books are scanned while issuing or returning.
* Search function: to enable both the librarian and the members to search the catalog of books in the library. The search functions can be filtered to the need of each user.
* Check out Article: To issue any article must be checked out
* Check In article: After receiving any article system will renter article by Checking
* Reserve Article: This use case is used to reserve any book with the name of librarian, it can be pledged.
* Set user Permission: From this user case Librarian can give permission categorically, also enabling/disabling of user permission can be set through this use case

### Member

* Authentication: User must be authenticated before accessing system
* Search Article: User can search any article
* Request Article: After successful searching member mark this book as requested article
* Check Account: This use case is used to check account details

## Non-Functional Requirement

### Usability:

* The UI should be simple enough for everyone to understand and get the relevant information without any special training. Different languages can be provided based on the requirements.

### Accuracy:

* The data stored about the books and the fines calculated should be correct, consistent, and reliable.

### Availability:

* The System should be available for the duration when the library operates and must be recovered within an hour or less if it fails. The system should respond to the requests within two seconds or less.

### Maintainability:

* The software should be easily maintainable and adding new features and making changes to the software must be as simple as possible. In addition to this, the software must also be portable.

# Design and Implementation Constraints

## Use case diagram

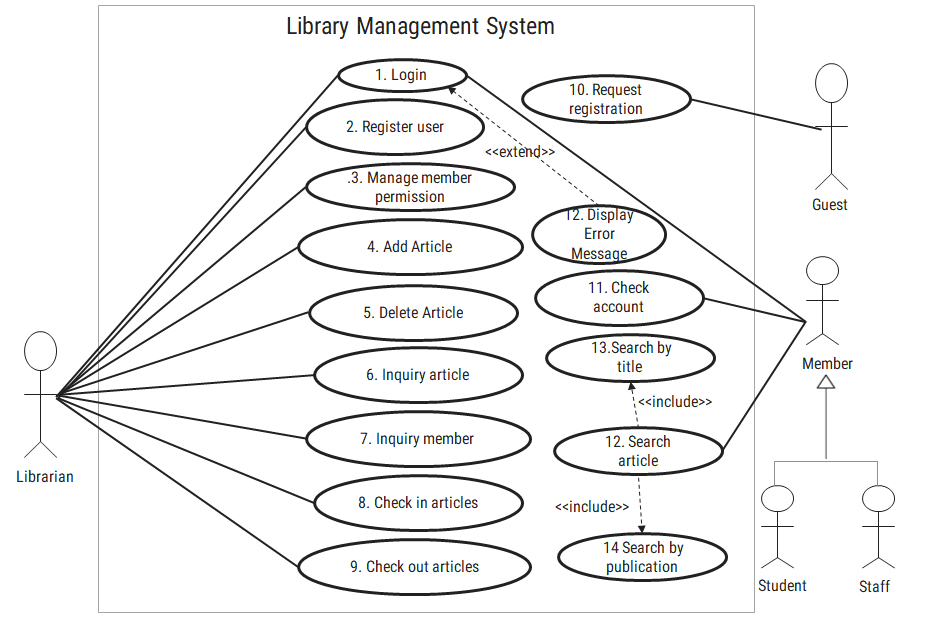


Figure 2.1‑1 Use case diagram for library management system

## Activity diagram and Swimlane diagram

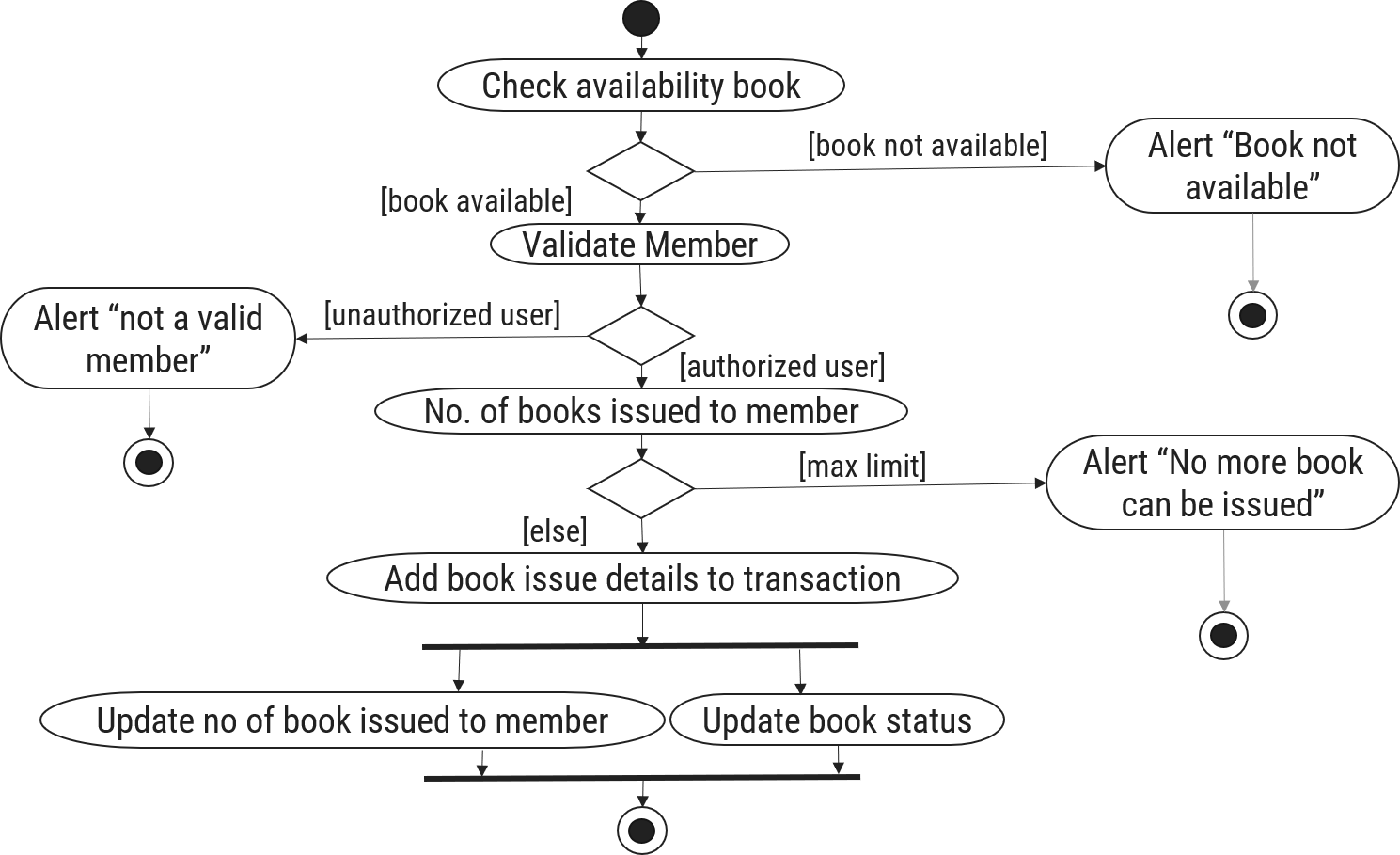


Figure 2.2‑1 Activity diagram for Book Issue

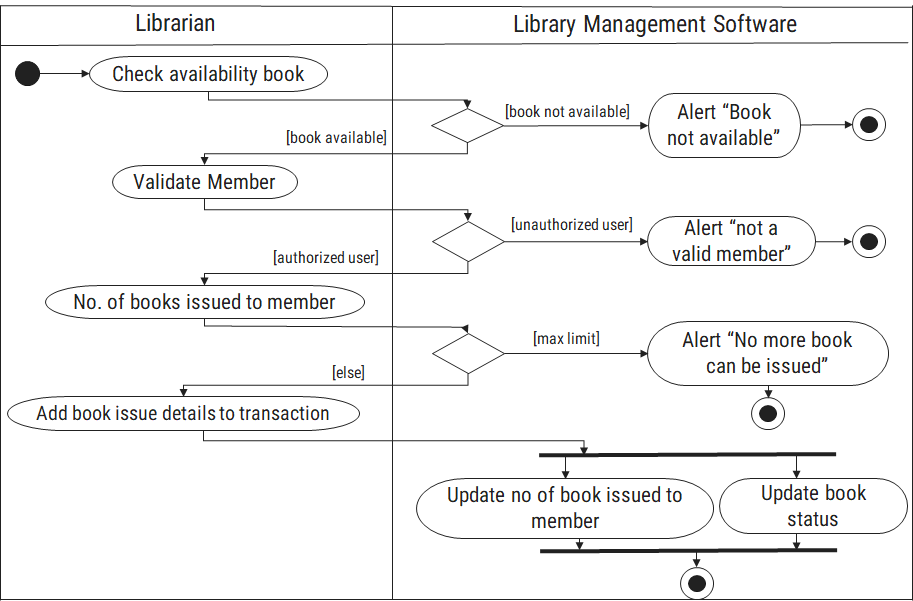


Figure 2.2‑2 Swimlane diagram for Book Issue

## Sequence diagram



Figure 2.3‑1 Sequence diagram for Book Issue

## State diagram



Figure 2.4‑1 State diagram of Book



Figure 2.4‑2 State diagram for Librarian

## Class diagram



Figure 2.5‑1 Class diagram for Library management system

## Data flow diagram

### Context diagram (level-0)



Figure 2.6‑1 Context diagram for Library management system

### DFD Level-1



Figure 2.6‑2 DFD level-1 for Library management system

### DFD Level-2



Figure 2.6‑3 DFD level-2 for Issue book

# External interface requirement (Screens)

## Screen-1: Registration Form



Figure 3.1‑1 Screen-1: Registration Form

**Purpose:** This form will allow the target end-users to register in the system. To register , the following information will be encoded in the system.

Table 3.1‑1 Screen element of Registration form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Username | Textbox | M | 1 | Username field should be editable and accept the Username. |
| 2 | Email | Textbox | M | 1 | Email field should be editable and accept the email with proper format. |
| 3 | Password | Password | M | 1 | Password field should be editable and accept the password and display as star or dot. |
| 4 | Retype password | Password | M | 1 | Retype Password field should be editable and accept the password and display as star or dot. |
| 5 | I agree to the terms | Checkbox | M | 1 | Checkbox for accepting terms and condition. |
| 6 | I already have a membership | Link | ------ | ------ | Link for navigate to login page for registered user. |
| 7 | Register | Button | ------ | ------ | Register is a button for store the entered data into database. |

## Screen-2: Login Form



Figure 3.2‑1 Screen-2: Login Form

**Purpose:** This form will be used by the system’s users to access records and features of the system. The users will input the correct combination of their username and password to be able to login to the system.

Table 3.2‑1 Screen element of Login form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Username | Textbox | M | 1 | Username field should be editable and accept the Username. |
| 2 | Password | Password | M | 1 | Password field should be editable and accept the password and display as star or dot. |
| 3 | Remember Me | Checkbox | M | 1 | Saving login credentials through remember me checkbox |
| 4 | I forgot my password | Link | ------ | ------ | Link for navigate to Forgot password page for allows users to recover password. |
| 5 | Register a new membership | Link | ------ | ------ | Link for navigate to membership registration. |
| 6 | Sign in | Button | ------ | ------ | Login button navigates to another page even if valid login credentials. |

## Screen-3: Add borrower book



Figure 3.3‑1 Screen-3: Add borrower book

**Purpose:** This module will allow the system administrator to add, edit, update or delete borrowers of book. The admin can add borrower information and manage it.

Table 3.3‑1 Screen element of Add borrower book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Borrower ID | Textbox | M | 1 | Borrower ID field should be editable and fill autogenerated Borrower ID. |
| 2 | Student ID | Textbox | M | 1 | Student ID field should be editable and accept Student barcode id. |
| 3 | First Name | Textbox | M | 1 | First Name field should be editable and accept only string as a borrower first name. |
| 4 | Middle Name | Textbox | O | 1 | Middle Name field should be editable and accept only string as a borrower middle name. |
| 5 | Last Name | Textbox | M | 1 | Last Name field should be editable and accept only string as a borrower last name. |
| 6 | Contact | Textbox | M | 1 | Contact field should be editable and accept only 10-digit mobile number. |
| 7 | Grade Id | Textbox | O | 1 | Auto fetch detail from the database if proper student id is entered. |
| 8 | Submit | Button | ----- | ----- | Submit is a button for store the entered data into database. |
| 9 | Cancel | Button | ----- | ----- | Cancel is a button for close add borrower dialog. |

# Database design

## List of Tables

* Book
* Borrower
* Student
* Staff

Table 4.1‑1 Table: Book

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| BookID | int | NN | PK (Auto Increment) |  |
| BookName | varchar(100) | NN |  |  |
| ISBN | varchar(100) | AN |  |  |
| Publication\_year | int | AN |  |  |
| Language | varchar(50) | AN |  |  |

Table 4.1‑2 Table: Borrower

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| BorrowerID | int | NN | PK (Auto Increment) |  |
| BookID | varchar(100) | NN | FK | Reference of Book Table |
| BorrowedFromDate | DateTime | AN |  |  |
| BorrowedToDate | DateTime | AN |  |  |
| ActualReturnDate | DateTime | AN |  |  |
| IssuedBy | int | NN | FK | Reference of Student and Staff Table |

Table 4.1‑3 Table: Staff

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| StaffID | int | NN | PK (Auto Increment) |  |
| StaffName | varchar(100) | AN |  |  |
| IsAdmin | Boolean | AN |  |  |
| Designation | varchar(100) | AN |  |  |

Table 4.1‑4 Table: Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| StudentID | int | NN | PK (Auto Increment) |  |
| StudentName | varchar(100) | AN |  |  |
| Gender | varchar(100) | AN |  |  |
| DOB | DateTime | AN |  |  |
| Department | varchar(100) | AN |  |  |
| Contact | number(10,0) | AN |  |  |

# Stories and Scenario

## Story-1: Add New Book in Library Catalogue

|  |  |  |
| --- | --- | --- |
| *Story # S1* | : | As an Admin,  I want to add a new product to the store inventory  So that customers can view and purchase it. |
| Priority | **:** | High |
| Estimate | **:** | XL |
| Reason | **:** | Adding new products to the inventory is essential to keeping the store’s offerings current and attractive to potential customers. |

### Scenario# S1.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.1* | : | Adding a New Product with Valid Information |
| Prerequisite | **:** | Admin is logged in to the online shopping platform's admin panel. |
| Acceptance Criteria | **:** | **Given:** The Admin is on the "Add New Product" page in the inventory management section. Valid product information (name, description, price, category, and stock level) is available.  **When:**  The Admin clicks the "Add New Product" button  The Admin enters the valid product details (name, description, price, category, stock quantity)  The Admin uploads a product image  The Admin clicks the "Save" button to add the product to the store inventory.  **Then** the system successfully adds the product to the inventory, and the Admin receives a confirmation message with the product's ID. |

### Scenario# S1.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.2* | : | Adding a New Product with Missing Information |
| Prerequisite | **:** | Admin is logged in to the online shopping platform's admin panel. |
| Acceptance Criteria | **:** | **Given:** The Admin is on the "Add New Product" page and attempts to add a new product, but some required fields (e.g., product name, price, or category) are left empty.  **When:**  The Admin clicks the "Add New Product" button  The Admin enters incomplete product details (leaving fields like name, description, price, or stock quantity empty)  **Then** the system displays an error message indicating the missing required fields (e.g., "Please fill in all required fields: name, price, and category.") |

## Story-2: Product Search and Filter

|  |  |  |
| --- | --- | --- |
| *Story # S2* | : | As a customer,  I want to search for products and filter results,  So that I can easily find the items I'm looking for and compare them. |
| Priority | **:** | High |
| Estimate | **:** | M |
| Reason | **:** | Search and filtering features improve the user experience by making product discovery faster and more efficient. |

### Scenario# S2.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S2.1* | : | Searching for a Product by Name |
| Prerequisite | **:** | The user is on the platform's home page. |
| Acceptance Criteria | **:** | **Given:** The user is on the home page with a search bar available.  **When:**  The user types the product name (e.g., "wireless headphones") in the search bar.  The user presses the "Search" button.  **Then** the system displays a list of products matching the search term, including names, images, and prices. |

### Scenario# S2.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S2.2* | : | Adding a New Product with Missing Information |
| Prerequisite | **:** | Admin is logged in to the online shopping platform's admin panel. |
| Acceptance Criteria | **:** | **Given:** The Admin is on the "Add New Product" page and attempts to add a new product, but some required fields (e.g., product name, price, or category) are left empty.  **When:**  The Admin clicks the "Add New Product" button.  The Admin enters incomplete product details (leaving fields like name, description, price, or stock quantity empty).  **Then** the system displays an error message indicating the missing required fields (e.g., "Please fill in all required fields: name, price, and category.") |

## Story-3: Track Order Status

|  |  |  |
| --- | --- | --- |
| *Story # S3* | : | As a Customer,  I want to track the status of my order  So that I can know when my products will be delivered. |
| Priority | **:** | Medium |
| Estimate | **:** | M |
| Reason | **:** | Order tracking provides transparency to customers and helps improve their experience by giving them visibility over the shipping process. |

### Scenario# S3.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S3.1* | : | Tracking an Order with Active Shipment |
| Prerequisite | **:** | The customer is logged into their account and has at least one order with an active shipment. |
| Acceptance Criteria | **:** | **Given:** The customer is logged into their account and navigates to the "My Orders" page.  **When:**  The customer selects an order with a tracking number.  The system fetches real-time shipment data from the carrier service.  **Then** the system displays a tracking status with the latest shipping information, including the current location of the package and estimated delivery date. |

### Scenario# S3.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S3.2* | : | Tracking an Order with Delayed Shipment |
| Prerequisite | **:** | The customer is logged into their account and has an order with a delayed shipment. |
| Acceptance Criteria | **:** | **Given:** The customer is logged into their account and navigates to the "My Orders" page.  **When:**  The customer selects an order with delayed shipment.  **Then** the system displays a delayed shipment notification along with an explanation (e.g., "Delayed due to weather conditions") and a new estimated delivery date. |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | EMI Calculator | Test Designed by: | P. U. Jadeja |
| Module Name: | **Login** | **Test Designed date:** | 01-10-2023 |
| Release Version: | **1.0** | **Test Executed by:** | **R. B. Gondaliya** |
|  |  | **Test Execution date:** | 15-01-2023 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pre-condition: Web application should be accessible | | | | |
| Test Case ID | **Test Title** | **Test Type** | **Description** | **Test Case ID** |
| TC\_001 | Login to web application with valid credential | Functional | Login to Library management system web application through valid credential | TC\_001 |
| TC\_002 | Login to web application with invalid credential | Functional | Login to Library management system web application through invalid credential | TC\_002 |
| TC\_003 | Varify login page elements | GUI | varify that all elements are availabe on login page | TC\_003 |

|  |  |
| --- | --- |
| **Test Case Title** | Login to web application with valid credential |
| **Test Type** | Functional |
| **Test Priority** | High |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **BUG ID** |
| 1 | Access Web application URL | The site launched properly | Site launched successfully | Pass |  | <https://accounts.google.com/ServiceLogin> |  |
| 2 | Enter valid Username in username field | Username field should be editable and accept the Username | Username input accepted | Pass |  | Username:  Rbgondaliya@gmail.com |  |
| 3 | Enter valid Password in Password field | Password field should be editable and accept the password and display as star or dot | Password input displayed in dot and accepted | pass |  | Password: rbgondaliya |  |
| 4 | Enter valid captcha code in captch field | Captch field should editable and accept captcha and captcha is case sensitive | Captcha input accepted | Pass | Step required when human action validation perform | get captcha from image which is near by captcha field |  |
| 5 | Click on login button | User should login into site and navigated to dashboard | User navigated to dashboard and username should br display in top of the right side. | pass |  |  |  |

|  |  |
| --- | --- |
| **Test Case Title** | Login to web application with invalid credential |
| **Test Type** | Functional |
| **Test Priority** | Medium |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Verify that User is not able to Login with invalid Username and invalid Password | Should be display an error message enter wrong username or password | Display an error of wrong username and password | Pass |  |  |  |
| 2 | Verify that User is not able to Login with Valid Username and invalid Password | Should be display an error message enter wrong password | Display an error of wrong password | Pass |  |  |  |
| 3 | Verify that User is not able to Login with invalid Username and Valid Password | Should be display an error message User not found | Display an error Username not found | Pass |  |  |  |
| 4 | Verify that User is not able to Login with blank Username or Password | Set required field validation message for Username and Password | Display an error of wrong username and password | Fail | Not performa a validation function fix it |  | Bug\_002 |

|  |  |
| --- | --- |
| **Test Case Title** | Varify login page elements |
| **Test Type** | GUI |
| **Test Priority** | Medium |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Launch application with the given url | The site launched properly | Site launched successfully | Pass |  | <https://accounts.google.com/ServiceLogin> |  |
| 2 | Verify that the login screen contains elements such as Username, Password, Sign in button, Remember password check box, Forgot password link, and Create an account link. | All listed control displayed properly on the page | Login page loaded successfully | Pass |  |  |  |
| 3 | Verify that cursor is focused on “Username” text box on the page load | Cursor is focused in Username textbox | Cursor focus in Username textbox | Pass |  |  |  |
| 4 | Verify that tab functionality is working properly or not | When tab pressed cursor move in next control | Cursor moving in next control | Pass |  |  |  |
| 5 | Verify that all the fields such as Username, Password has a valid placeholder | All text fields have proper placeholder | All text fields have proper placeholder | Pass |  |  |  |
| 6 | Verify that the labels float upward when the text field is in focus or filled (In case of floating label) | When field is focused or filled, label display on top of the filled | When field is focus or filled, label display on top of the filled | Pass | step required when fields with floating label |  |  |
| 7 | verify that forgot password link working properly | when click on forgot password load forgot passworg page | forgot password link not working | Fail |  |  |  |

# References

* http://www.w3schools.com/html/html\_intro.asp
* https://www.w3schools.com/php/default.asp
* https://www.javatpoint.com/uml