

Darshan University

A Project Report on

**“Auction Management System”**

Under the subject

**Software Engineering (2301CS405)**

B. Tech, Semester – IV

Computer Science & Engineering Department

Submitted By

Student Name: Manav Kotecha A. Enrollment No.:23010101145

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 **“Auction Management System”** 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.

Manav Kotecha A.

M.A.K

Student’s Signature

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

**Computer Science & Engineering**

**Department**

**Darshan University**

**CERTIFICATE**

This is to certify that the SRS on **“Auction Management System” has** been satisfactorily prepared by **Manav Kotecha (23010101145)** under my guidance in the fulfillment of the course **Software Engineering (2301CS405)** work during the academic year 2024-2025.

Internal Guide Dean-DIET

Prof. R. B. Gondaliya Dr. Gopi Sanghani

Darshan University 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

**Manav Kotecha**

**ABSTRACT**

The **Auction Management System** is designed to develop a computerized platform that simplifies and automates the management of auction processes. This system acts as a tool to transition traditional auction practices into a modern, digital format. The project includes features that go beyond basic auction management, such as user registration, secure login, real-time bidding, and comprehensive auction tracking.

The system provides separate modules for participants and administrators. Participants can register, browse auction listings, place bids, and monitor their bidding status. The platform also includes features such as instant notifications, bid history, and secure payment processing. Additionally, users can request specific items to be included in upcoming auctions through a request form.

The admin module allows auction organizers to manage the entire system. Admins can add, update, or remove auction items, set bidding rules, monitor ongoing auctions, and handle user disputes. The system can also generate detailed reports, including auction performance reports, user participation reports, and revenue summaries.

The primary purpose of this system is to reduce manual efforts, enhance accessibility, and ensure a secure, efficient, and transparent auction experience. It is designed to cater to a wide range of industries, including art, real estate, automobiles, and more, while delivering a user-friendly and scalable solution.

**Table of Contents**

List of Figures ....................................................................................................................................................... I

List of Tables ....................................................................................................................................................... II

[1 Introduction 1](#_Toc59017)

[1.1 Product perspective 1](#_Toc59018)

[1.2 Product features 1](#_Toc59019)

[1.3 Functional Requirement 1](#_Toc59020)

[**1.2.1 Admin**: 1](#_Toc59021)

[**1.2.2 Buyer** 2](#_Toc59022)

[1.4 Non-Functional Requirement 2](#_Toc59023)

[**1.4.1** Usability: 2](#_Toc59024)

[**1.4.2** Accuracy: 2](#_Toc59025)

[**1.4.3** Availability: 2](#_Toc59026)

[**1.4.4** Maintainability: 3](#_Toc59027)

[2 Design and Implementation Constraints 4](#_Toc59028)

[2.1 Use case diagram 4](#_Toc59029)

[2.2 Activity diagram and Swimlane diagram 5](#_Toc59030)

[2.3 Sequence diagram 8](#_Toc59031)

[2.4 State diagram 9](#_Toc59032)

[2.5 Class diagram 10](#_Toc59033)

[2.6 Data flow diagram 11](#_Toc59034)

[**2.6.1** Context diagram (level-0) 11](#_Toc59035)

[**2.6.2** DFD Level-1 11](#_Toc59036)

[**2.6.3** DFD Level-2 12](#_Toc59037)

[3 External interface requirement (Screens) 13](#_Toc59038)

[3.1 Screen-1: Registration Form 13](#_Toc59039)

[3.2 Screen-2: Login Form 14](#_Toc59040)

[3.3 Screen-3: Add borrower book 15](#_Toc59041)

[4 Database design 16](#_Toc59042)

[4.1 List of Tables 16](#_Toc59043)

[5 Stories and Scenario 17](#_Toc59044)

[5.1 Story-1: Add New Book in Library Catalogue 17](#_Toc59045)

[**5.1.1** Scenario# S1.1 17](#_Toc59046)

[**5.1.2** Scenario# S1.2 17](#_Toc59047)

[**5.1.3** Scenario# S1.3 17](#_Toc59048)

[5.2 Story-2: Search Book 18](#_Toc59049)

[5.3 Story-3: Manage due date for borrowed book 18](#_Toc59050)

[5.4 Story-4: Renew book 18](#_Toc59051)

[5.5 Story-5: Generate a report on book usage and availability 18](#_Toc59052)

[6 Test cases 19](#_Toc59053)

[7 References 23](#_Toc59054)

**List of Figures**

Figure 2.1-1 Usecase Diagram for Auction Management System...................................................................... 4

Figure 2.2-1 Activity Diagram for Buyer ............................................................................................................. 5

Figure 2.2-2 Activity Diagram for Admin ............................................................................................................ 6

Figure 2.2-3 Swimlane Diagram for Buyer ......................................................................................................... 7

Figure 2.3-1 Sequence diagram for Book Issue .................................................................................................. 8

Figure 2.4-1 State diagram of Book .................................................................................................................... 9

Figure 2.4-2 State diagram for Librarian ........................................................................................................... 9

Figure 2.5-1 Class diagram for Library management system ........................................................................... 10

Figure 2.6-1 Context diagram for Library management system ...................................................................... 11

Figure 2.6-2 DFD level-1 for Library management system ............................................................................... 11

Figure 2.6-3 DFD level-2 for Issue book ........................................................................................................... 12

Figure 3.1-1 Screen-1: Registration Form ........................................................................................................ 13

Figure 3.2-1 Screen-2: Login Form ................................................................................................................... 14

Figure 3.3-1 Screen-3: Add borrower book ...................................................................................................... 15

**I**

**List of Tables**

Table 3.1-1 Screen element of Registration form ............................................................................................ 13

Table 3.2-1 Screen element of Login form ....................................................................................................... 14

Table 3.3-1 Screen element of Add borrower book ......................................................................................... 15

Table 4.1-1 Table: Book .................................................................................................................................... 16

Table 4.1-2 Table: Borrower ............................................................................................................................. 16

Table 4.1-3 Table: Staff ..................................................................................................................................... 16

Table 4.1-4 Table: Student ............................................................................................................................... 16

# 1 Introduction

## 1.1 Product perspective

This product is designed to handle the essential functions of an auction, including item cataloging, bid monitoring, user management, and report generation. It is particularly suitable for industries such as art, real estate, automobiles, and collectibles, while being scalable for larger or smaller auction events.

## 1.2 Product features

**1.2.1 Who can use this system**:

* **Auctioneer** – The administrator who is responsible for managing the auctions.
* **Bidder** – Participates in auctions by placing bids on available items.
* **Guest** – Can browse auctions.

**1.2.2 Features for the Auctioneer:**

* Add, update, or delete auction listings.
* Set auction parameters like starting bid, reserve price, auction duration.
* View and manage bids placed on items in real-time.
* Approve or reject user registrations.
* View bidder details.
* Send notifications to users about new auctions, updates, or alerts.
* Generate auction performance reports, revenue summaries, and user participation details.
* Ensure secure transactions.

**1.2.3 Features for the Bidder:**

* Register and log in to the system.
* View and update personal information.
* Show active auctions by category.
* Place bids on items.
* Receive notifications about bid status.
* View bid history .
* Submit requests for specific items to be included in future auctions.

**1.2.4 Features for the Guest:**

* Show available auctions and item details.

## 1.3 Functional Requirement

### **1.3.1 Admin**:

* **Add Auction Item:** Admin can add new items for auction by entering all required details into the system.
* **Update Auction Item:** Admin can modify the details of existing auction items.
* **Delete Auction Item:** Admin can remove any incorrect auction item from the system.
* **User Management:** Admin can manage user accounts.
* **Auction Monitoring:** Admin can monitor ongoing auctions and review bidding activities in real-time.
* **Set Auction Parameters:** Admin can define rules like minimum bid increments, auction start and end times, and reserve prices.
* **Reports and Analytic:** Admin can generate detailed reports on auction performance, user activity, and revenue.
* **Search Functionality:** Admin can search for auction items.
* **Set User Permissions:** Admin can assign roles or permissions to users, categorizing them as buyers, sellers.
* **Notification Management:** Admin can send updates or alerts to users about auctions, changes, or reminders.

### **1.2.2 Buyer**

* **Authentication:** Buyers must authenticate using a username and password before accessing the system.
* **Browse Auctions:** Buyers can browse active auctions with filters like item category or base price.
* **Place Bids:** Buyers can place bids on items and monitor bid status in real-time.
* **Bid History:** Buyers can view their bidding history, including won and lost auctions.
* **Search Auction Items:** Buyers can search for specific auction items using keywords, categories, or other criteria.
* **Notifications:** Buyers receive notifications for important updates.
* **Payment:** Buyers can complete payments for won auctions through secure payment .
* **Account Management:** Buyers can view and update their account details, such as personal information and payment methods.

**1.2.3**  **Seller**

* **Authentication:** Sellers must log in using their credentials to access their account.
* **Add Auction Item:** Sellers can add items for auction by entering details like item description and base price.
* **Update Auction Item:** Sellers can modify the details of their items, such as description or base price, before the auction begins.
* **Remove Auction Item:** Sellers can cancel or remove an item from the auction list.
* **View Bidding Activity:** Sellers can monitor bids placed on their items in real-time.
* **Set Auction Parameters:** Sellers can define auction rules for their items including minimum bid increments.
* **Search Functionality:** Sellers can search their listed items or auction history .
* **View Reports:** Sellers can access performance reports for their auctioned items, including total bids and final selling price.
* **Notifications:** Sellers receive updates about auction status, bids received, and payment confirmations.
* **Payment Management:** Sellers can view and manage payment records for their sold items.

## 1.4 Non-Functional Requirement

### **1.4.1** 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.

### **1.4.2** Accuracy:

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

### **1.4.3** Availability:

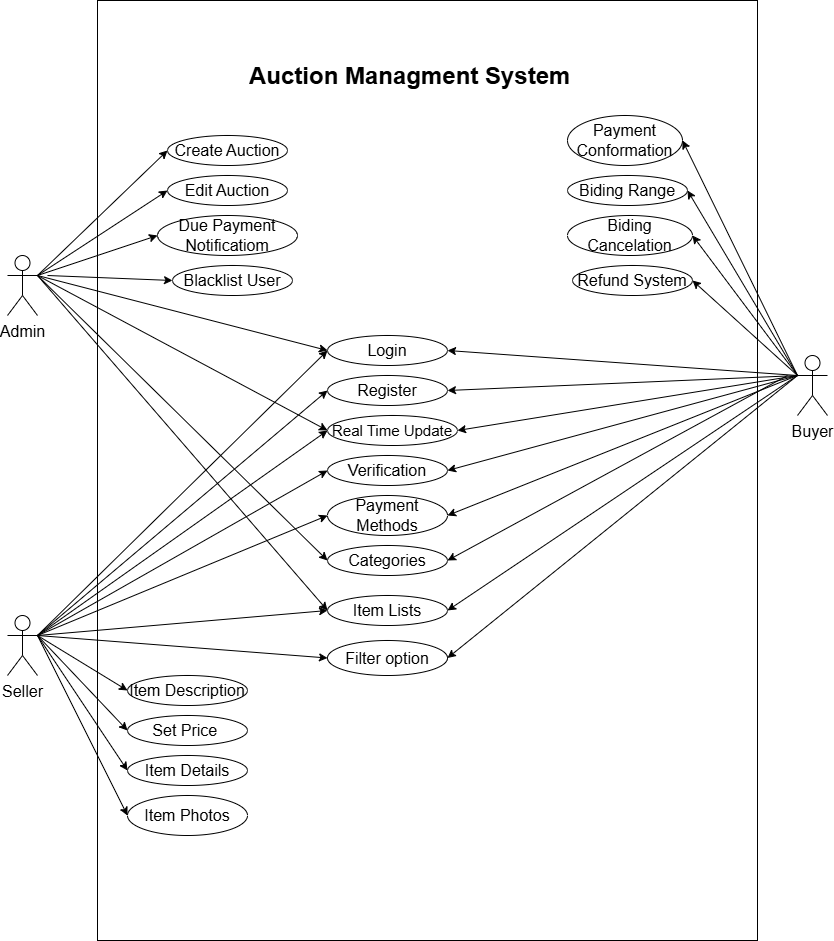
• The System should be available for the duration when the library operates and must be recovered within an hour or less if it fails. The system should respond to the requests within two seconds or less.

### **1.4.4** 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.

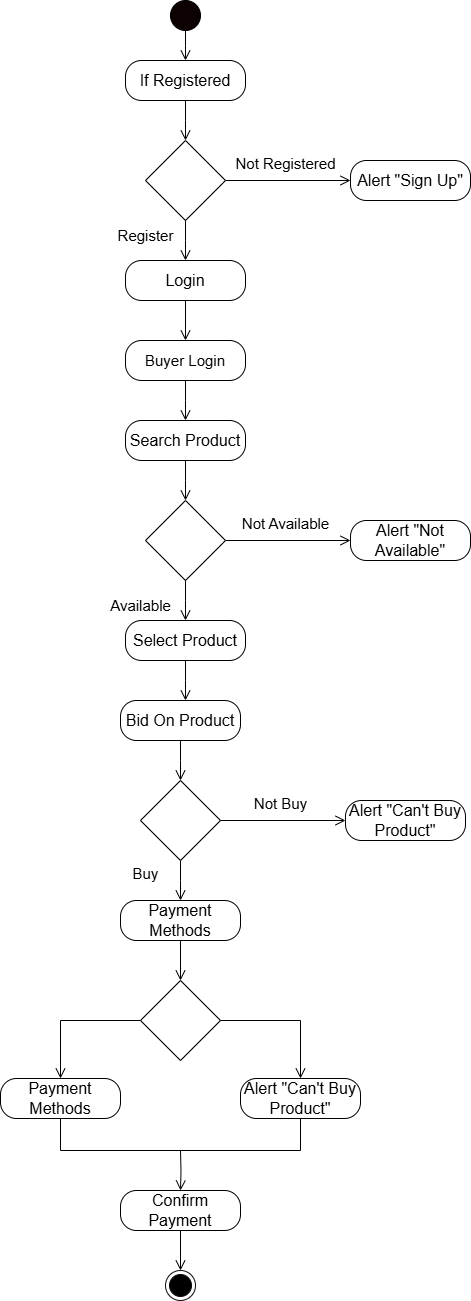
# 2 Design and Implementation Constraints

## 2.1 Use case diagram

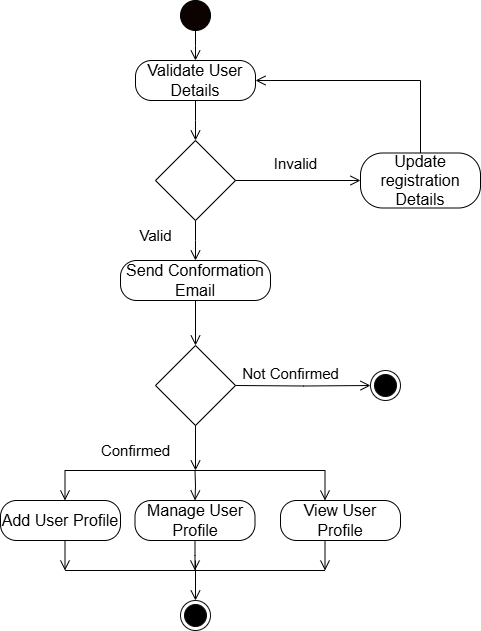


*Figure 2.1-1 Usecase Diagram for Auction Management System*

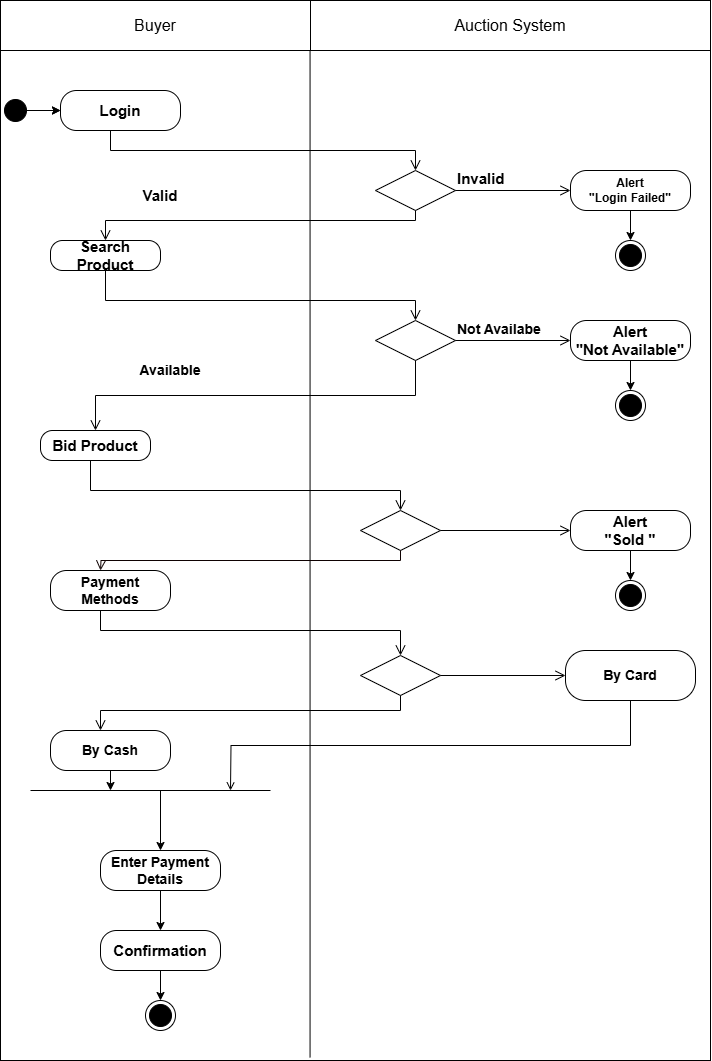
## 2.2 Activity diagram and Swimlane diagram



*Figure 2.2-1 Activity Diagram for Buyer*

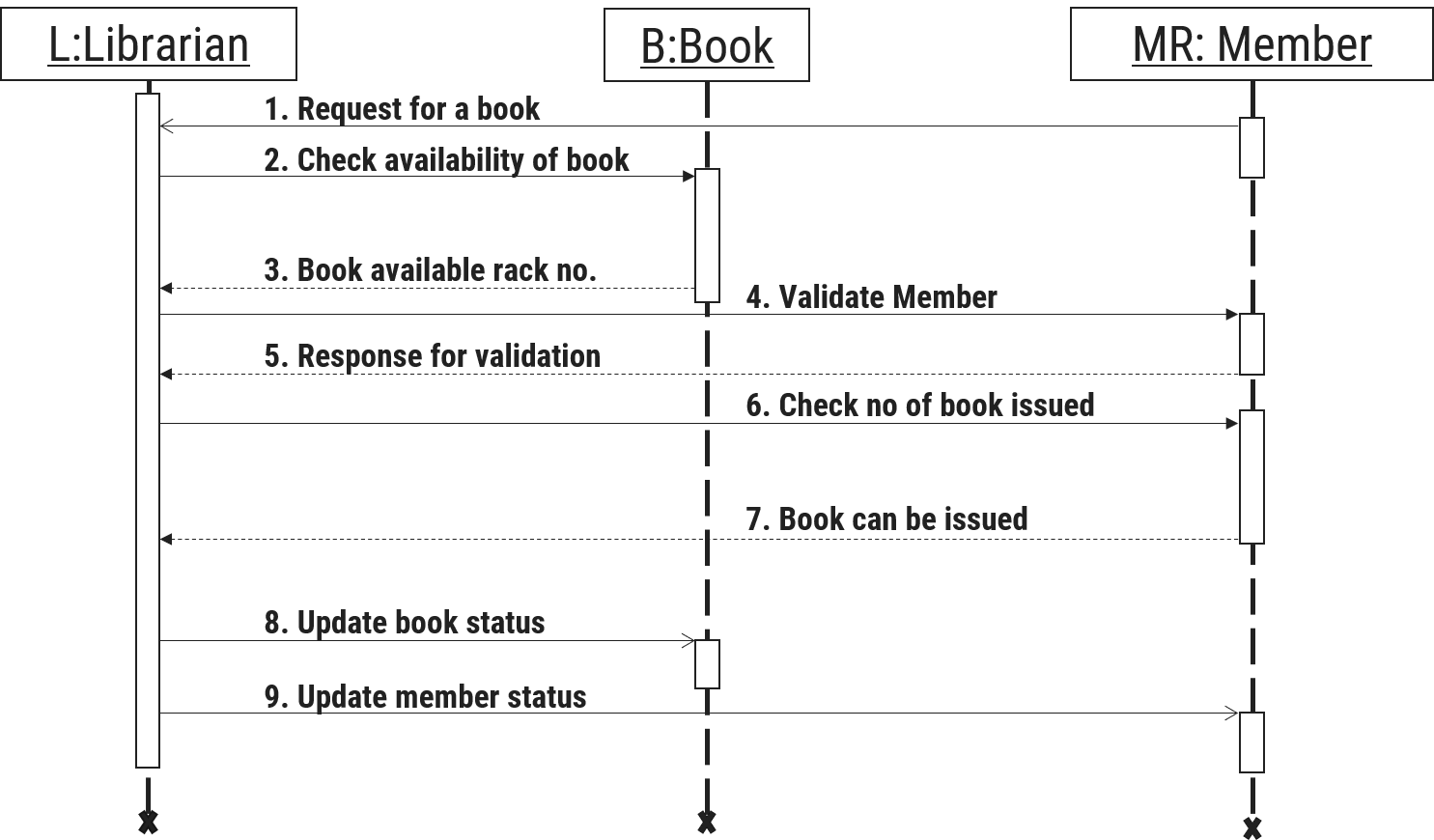


*Figure 2.2-2 Activity Diagram for Admin*



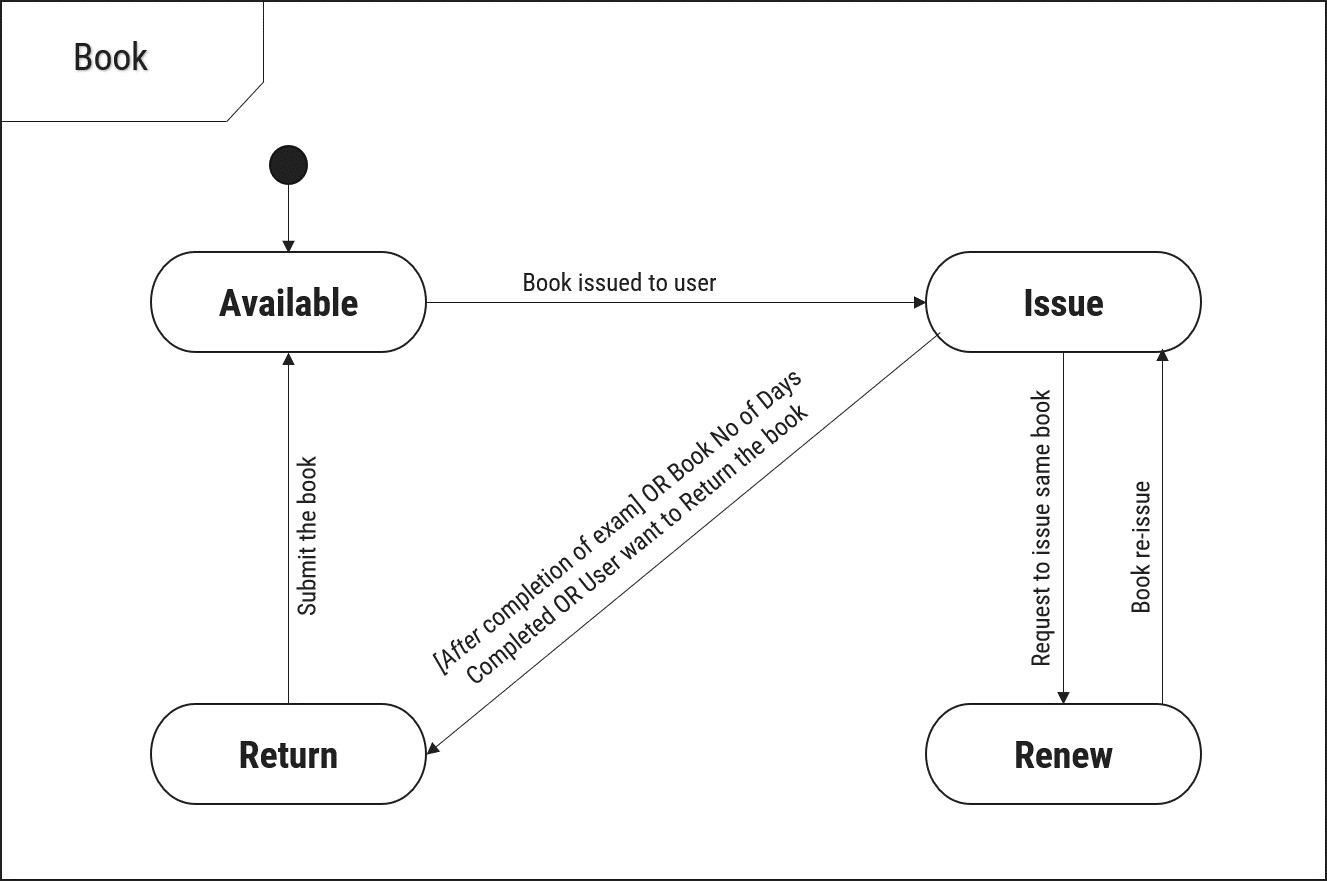
*Figure 2.2-3 Swimlane Diagram for Buyer*

## 2.3 Sequence diagram

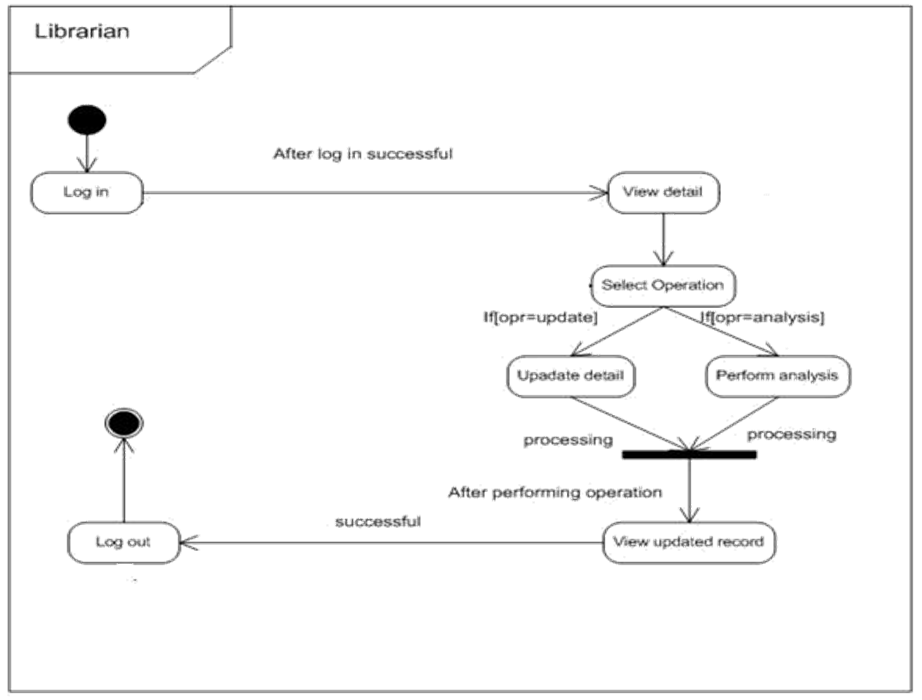


*Figure 2.3-1 Sequence diagram for Book Issue*

## 2.4 State diagram

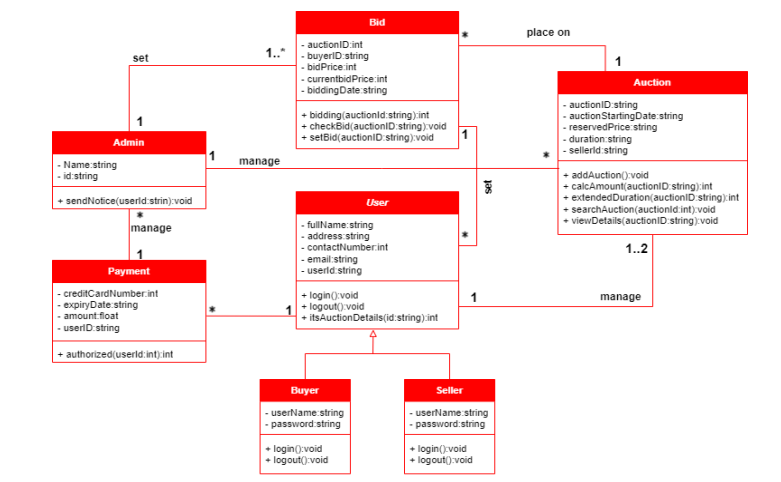


*Figure 2.4-1 State diagram of Book*



*Figure 2.4-2 State diagram for Librarian*

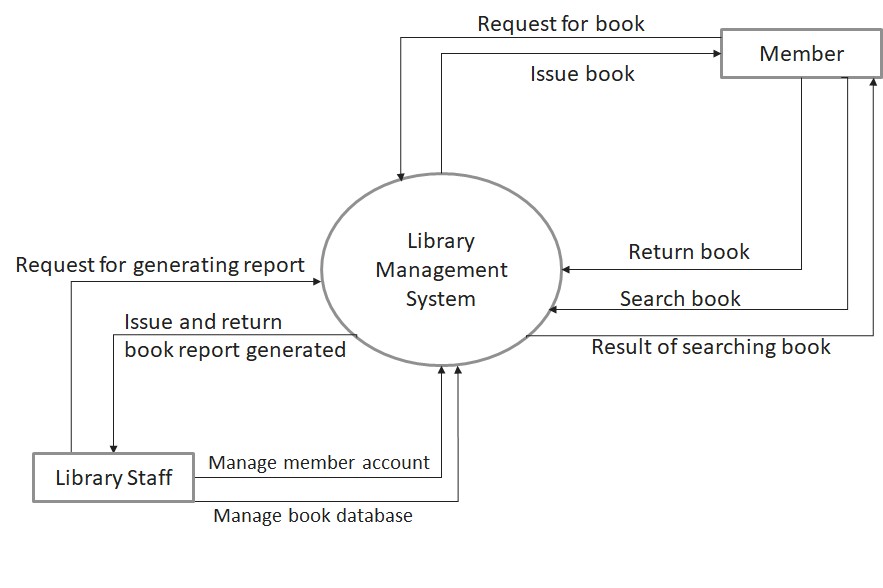
## 2.5 Class diagram



*Figure 2.5-1 Class diagram for Auction management system*

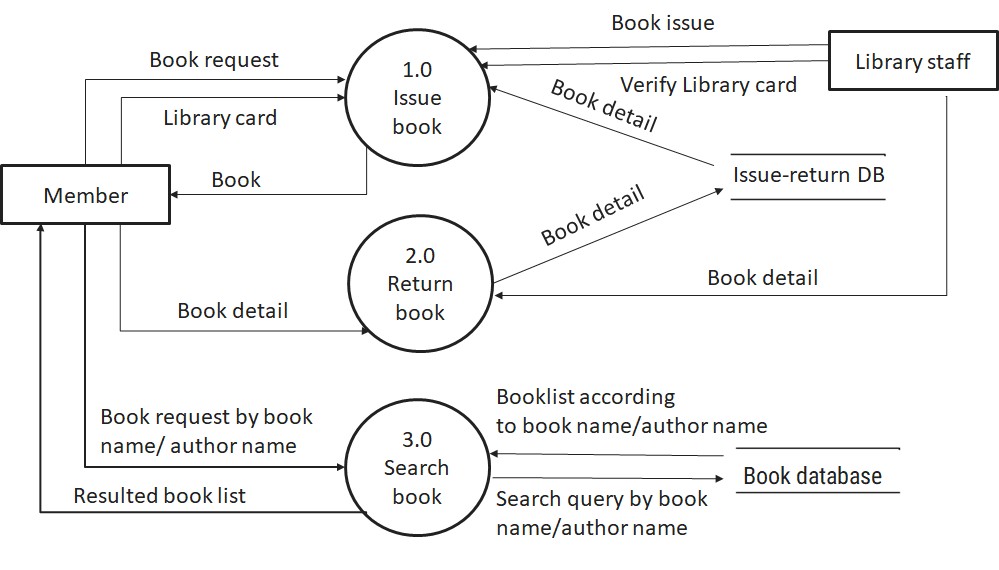
## 2.6 Data flow diagram

### **2.6.1** Context diagram (level-0)



*Figure 2.6-1 Context diagram for Library management system*

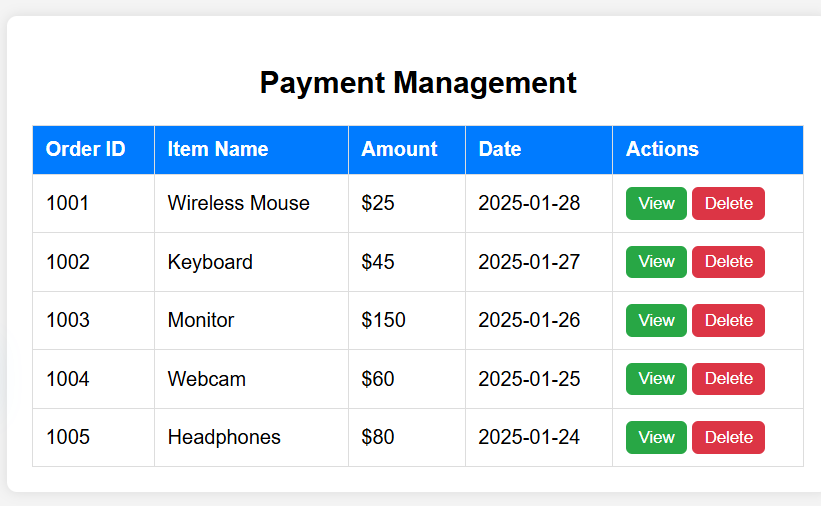
### **2.6.2** DFD Level-1



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

# 3 External interface requirement (Screens)

## 3.1 Screen-1: Payment Managment



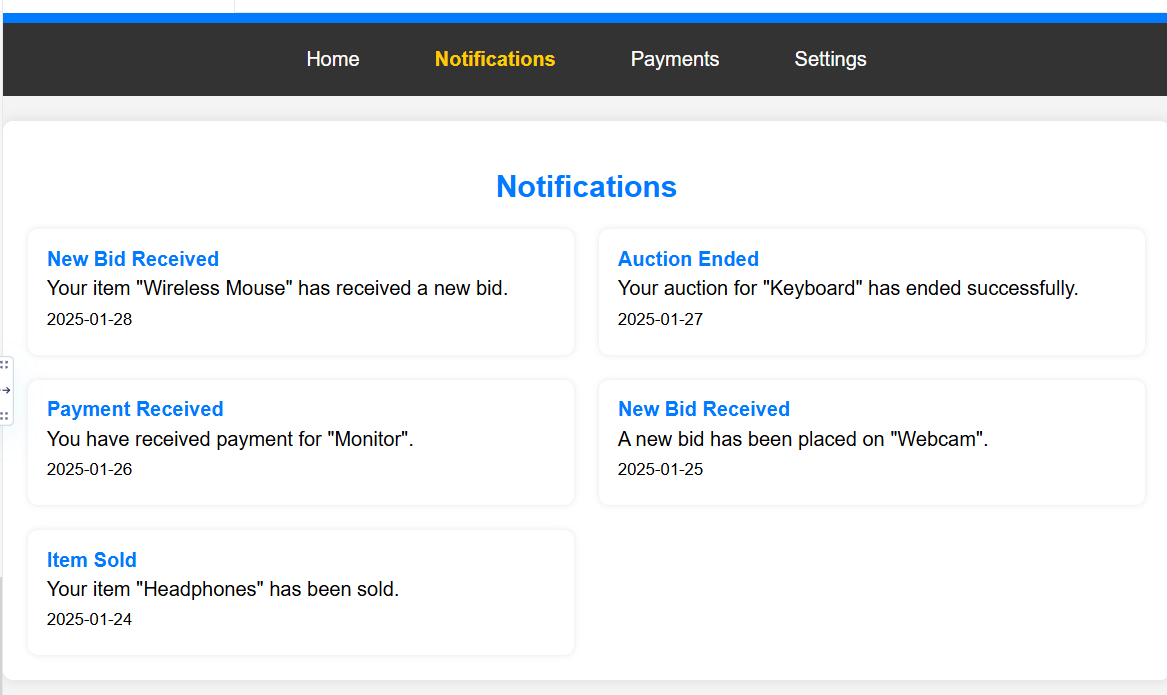
*Figure 3.1-1 Screen-1: Payment*

**Purpose: Allows users to view, manage, and delete their payment transactions in an organized manner.**

*Table 3.2-1 Screen element of Item List*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | **Payment Management Title** | **Text** | **M** | **1** | **Displays the title 'Payment Management' indicating the section for managing transactions.** |
| **2** | **Order ID Column** | **Text** | **M** | **N** | **Displays unique order IDs for each transaction.** |
| **3** | **Item Name Column** | **Text** | **M** | **N** | **Lists the item names associated with each transaction.** |
| **4** | **Amount Column** | **Text** | **M** | **N** | **Shows the amount associated with each transaction.** |
| **5** | **Date Column** | **Text** | **M** | **N** | **Displays the date of each transaction.** |
| **6** | **Actions Column** | **Button** | **M** | **N** | **Provides 'View' and 'Delete' buttons for each transaction.** |

## 3.2 Screen-2: Notification

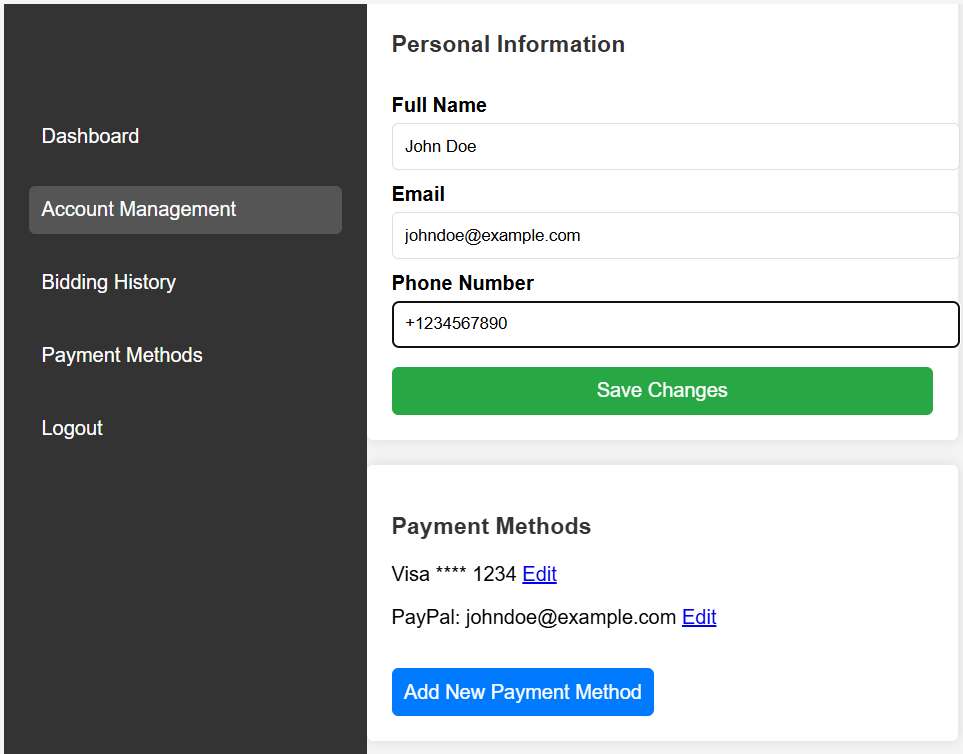


*Figure 3.2-1 Screen-2: Notification*

**Purpose: Keeps users informed about their auction activities, including bids, payments, and auction status.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | **Notifications Title** | **Text** | **M** | **1** | **Displays the section title 'Notifications' for alert messages.** |
| **2** | **Notification Cards** | **Text** | **M** | **N** | **Displays various notifications related to bids, auctions, and payments.** |
| **3** | **Navigation Bar** | **Link** | **M** | **1** | **Provides links to Home, Notifications, Payments, and Settings.** |

## 3.3 Screen-3: Account Management

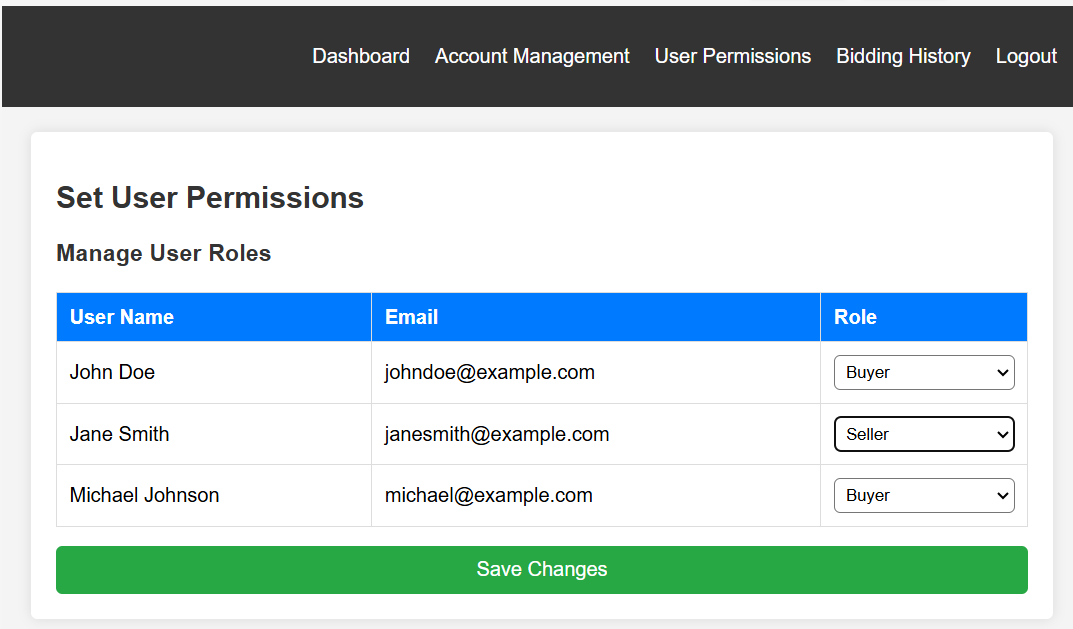


*Figure 3.3-1 Screen-3: Item Details*

**Purpose: Allows users to update their personal information and manage payment methods.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | **Account Management Title** | **Text** | **M** | **1** | **Displays 'Personal Information' for managing user details.** |
| **2** | **Full Name Field** | **Input Box** | **M** | **1** | **Allows users to enter/edit their full name.** |
| **3** | **Email Field** | **Input Box** | **M** | **1** | **Displays the user's email and allows editing.** |
| **4** | **Phone Number Field** | **Input Box** | **M** | **1** | **Allows users to enter/edit their phone number.** |
| **5** | **Save Changes Button** | **Button** | **M** | **1** | **Saves updated user information.** |
| **6** | **Payment Methods Title** | **Text** | **M** | **1** | **Displays 'Payment Methods' section.** |
| **7** | **Saved Payment Methods** | **Text** | **M** | **N** | **Lists saved payment methods like Visa and PayPal.** |
| **8** | **Edit Payment Method Button** | **Button** | **M** | **N** | **Allows users to modify their payment methods.** |
| **9** | **Add Payment Method Button** | **Button** | **M** | **1** | **Enables users to add a new payment method.** |
| **10** | **Sidebar Menu** | **Link** | **M** | **1** | **Provides navigation to Dashboard, Account Management, Bidding History, Payment Methods, and Logout.** |

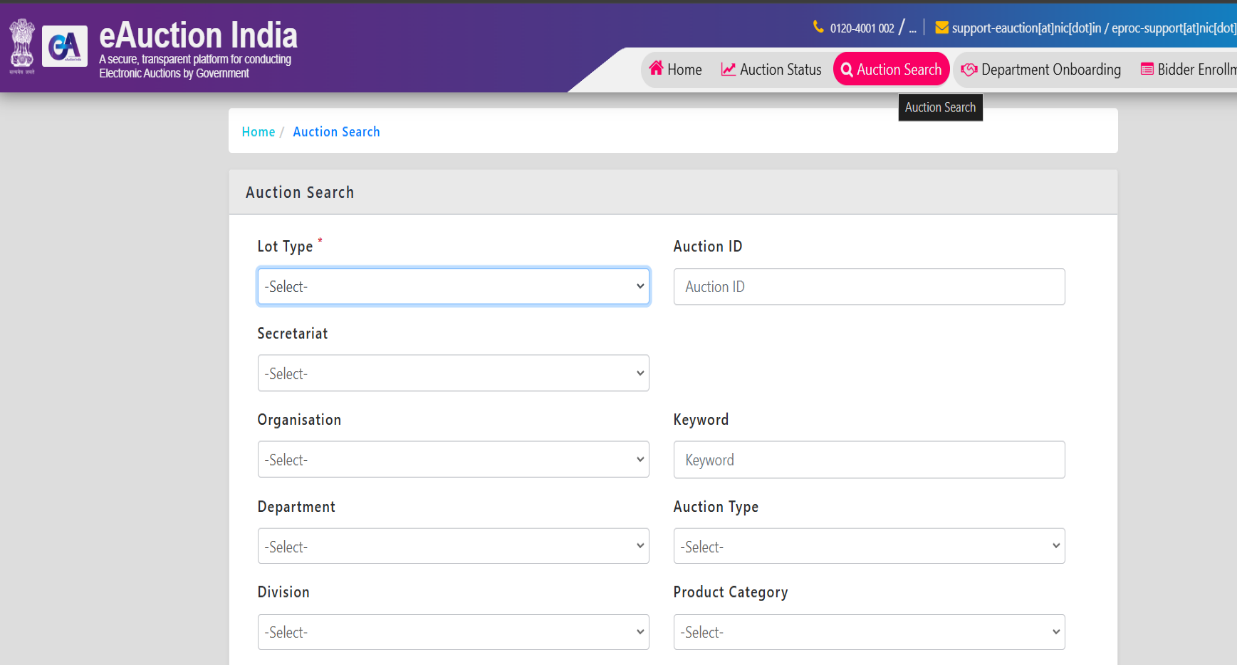
3.4 Screen-4: **Set User Permissions**



**Purpose: The 'Set User Permissions' screen allows administrators to manage user roles effectively. It provides a clear and intuitive interface for assigning or modifying user roles to control access levels.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | 1/N | **Description** |
| **1** | **Title** | **Text** | **M** | 1 | **Displays the heading "Set User Permissions."** |
| **2** | **User Table** | **Table** | **M** | 1 | **Displays a table listing user names, emails, and their roles.** |
| **3** | **Role Dropdown** | **Dropdown** | **M** | N | **Allows the admin to assign roles such as Buyer or Seller.** |
| **4** | **Save Changes Button** | **Button** | **M** | 1 | **Enables the admin to save the changes made to user roles.** |
| **5** | **Navbar** | **Link** | **M** | 1 | **Provides navigation links like Dashboard, Account Management, User Permissions, and Logout.** |

3.5 Screen-5: **Search Functionality**

****

**Purpose: The 'Auction Search' screen provides users with advanced search capabilities to find auction items based on various criteria like lot type, organization, and keywords. It streamlines the auction search process and ensures users can quickly locate relevant items.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | 1/N | **Description** |
| **1** | **Title** | **Text** | **M** | 1 | **Displays the title "Auction Search."** |
| **2** | **Lot Type Dropdown** | **Dropdown** | **M** | 1 | **Allows users to select the type of lot they are interested in.** |
| **3** | **Auction ID Field** | **Text** | **O** | 1 | **Enables users to search for specific auctions by ID.** |
| **4** | **Secretariat Dropdown** | **Dropdown** | **O** | N | **Filters auctions based on the selected secretariat.** |
| **5** | **Keyword Field** | **Text** | **O** | 1 | **Allows users to enter keywords for a more targeted search.** |
| **6** | **Organisation Dropdown** | **Dropdown** | **O** | N | **Filters auctions based on the selected organization.** |
| **7** | **Department Dropdown** | **Dropdown** | **O** | N | **Further refines the search by department.** |
| **8** | **Division Dropdown** | **Dropdown** | **O** | N | **Allows filtering by specific divisions.** |
| **9** | **Auction Type Dropdown** | **Dropdown** | **O** | N | **Filters auctions based on their type.** |
| **10** | **Product Category Dropdown** | **Dropdown** | **O** | N | **Enables users to select a product category for the search.** |

# 4 Database design

## 4.1 List of Tables

* Users
* Item
* Bids
* Categories
* Payments

*Table 4.1-1 Table: Users*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **UserID** | int | NN | PK (Auto Increment) |  |
| **UserName** | varchar(100) | NN |  |  |
| Password | varchar(50) | NN |  |  |
| Email | varchar(50) | NN |  |  |
| ContactNumber | int | NN |  |  |
| **Address** | varchar(100) | AN |  |  |

*Table 4.1-2 Table: Item*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **ItemID** | int | NN | PK (Auto Increment) |  |
| **ItemName** | varchar(100) | NN |  |  |
| Title | varchar(50) | AN |  |  |
| Description | varchar(100) | NN |  |  |
| **StartingPrice** | int | NN |  |  |
| **StartDate** | DateTime | NN |  |  |
| **EndDate** | DateTime | AN |  |  |
| **CategoryID** | int | AN | FK | Referencing of Category Table |

*Table 4.1-3 Table: Bids*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **BidID** | int | NN | PK (Auto Increment) |  |
| ItemId | int | NN |  | Referencing of Item Table |
| UserID | int | NN |  | Referencing of Users Table |
| BidAmmount | int | NN |  |  |
| **BidStart** | DateTime | AN | FK |  |

*Table 4.1-4 Table: Categories*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **CategoryID** | int | NN | PK (Auto Increment) |  |
| **CategoryName** | varchar(100) | NN |  |  |
| **Description** | varchar(100) | AN |  |  |

*Table 4.1-5 Table: Payments*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **PaymentID** | int | NN | PK (Auto Increment) |  |
| **UserID** | int | NN | FK | Referencing of Users Table |
| ItemID | int | NN | FK | Referencing of Items Table |
| PaymentAmount | int | NN |  |  |
| PaymentDate | DateTime | NN |  |  |
| **PaymentMethod** | varchar(50) | AN |  |  |

# 5 Stories and Scenario

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

|  |  |  |
| --- | --- | --- |
| *Story #* ***S1*** | **:** | **As a** Librarian,  **I want to** add a new book in library catalogue  **So that** everyone can easily find and borrow it. |
| **Priority** | **:** | High |
| **Estimate** | **:** | XL |
| **Reason** | **:** | The addition of a new book to the library catalogue is crucial for ensuring that the library's collection is up-to-date and accessible to everyone. |

### **5.1.1** Scenario# S1.1

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***S1.1*** | **:** | Adding a New Book with Valid Information |
| **Prerequisite** | **:** | Librarian is logged in to the Library management system. |
| **Acceptance**  **Criteria** | **:** | **Given:**  The Librarian is navigated to the library catalog management page. Valid book information, including title, author, ISBN, and other relevant details is added.  **When:**  The librarian selects the "Add New Book" option  And The librarian enters valid book details  The librarian clicks the "Save" button to add the book to the catalog.    **Then t**he system successfully adds the book to the catalog and the librarian receives a confirmation message with the book's identification number. |

### **5.1.2** Scenario# S1.2

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***S1.2*** | **:** | Adding a New Book with Invalid Information. |
| **Prerequisite** | **:** | The librarian is logged into the library management system. |
| **Acceptance**  **Criteria** | **:** | **Given:** The librarian is on the library catalogue management page  **When:** The librarian selects the "Add New Book" option and the librarian enters an incomplete or incorrect book details and librarian clicks the "Save" button to add the book to the catalogue.  **Then t**he system displays error messages for the incorrect or missing information and the book is not added to the catalogue. |

### **5.1.3** Scenario# S1.3

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***S1.3*** | **:** | Attempting to Add a Duplicate Book |
| **Prerequisite** | **:** | The librarian is logged into the library management system and the librarian is on the library catalogue management page |
| **Acceptance**  **Criteria** | **:** | **Given**: The book information, including title, author, ISBN, and other relevant details, is available and the book with the same ISBN is already in the catalogue.  **When**: User Clicks on “Add book” button. Enter a number of copies with the same book detail mentioned in the field. |
|  | | **Then**: Generate unique book id, barcode and spine label for various book of same title. |

## 5.2 Story-2: Search Book

|  |  |  |
| --- | --- | --- |
| *Story #* ***S2*** | : | **As a** Librarian or member,  **I want to** search for books by title, author, or keyword,  **So that** I can quickly find books that match my interests. |
| **Priority** | **:** | High |
| **Estimate** | **:** | M |
| **Reason** | **:** | Implementing a search functionality is essential for enhancing the user experience, as it allows librarian and member to efficiently discover and access the library's resources. |

## 5.3 Story-3: Manage due date for borrowed book

|  |  |  |
| --- | --- | --- |
| *Story #* ***S3*** | **:** | **As** Librarian,  **I want** to manage due dates for borrowed books  **So that** I can ensure that books are returned on time and avoid overdue fines. |
| **Priority** | **:** | High |
| **Estimate** | **:** | M |
| **Reason** | **:** | Proper due date management is crucial for maintaining the library's collection and ensuring that books are available for all members. |

## 5.4 Story-4: Renew book

|  |  |  |
| --- | --- | --- |
| *Story #* ***S3*** | **:** | **As** Librarian,  **I want** to renew a book that I have borrowed  **So that** I can extend my borrowing period if needed. |
| **Priority** | **:** | Medium |
| **Estimate** | **:** | M |
| **Reason** | **:** | Book renewal functionality is a convenience feature for librarian, allowing them to keep a book for an extended period if no one else has requested it. |

## 5.5 Story-5: Generate a report on book usage and availability

|  |  |  |
| --- | --- | --- |
| *Story #* ***S3*** | **:** | **As** Librarian,  **I want** to generate reports on library usage and book availability **So that** I can make informed decisions about library’s books. |
| **Priority** | **:** | Medium |
| **Estimate** | **:** | L |
| **Reason** | **:** | Reporting functionality helps librarians track the usage of library resources and make data-driven decisions to improve services and collections. |

# 6 Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Name:** | **EMI Calculator** | **Test Designed by:** | **P. U. Jadeja** |
| **Module Name:** | **Set User Permissions** | **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** | **Reference** |
| TC\_001 | Set User Permissions - Valid User | Functional | Verify that a valid user can set permissions successfully in the EMI Calculator module. | TC\_001 |
| TC\_002 | Set User Permissions - Admin Role | Functional | Verify that an admin user can set permissions for other users in the EMI Calculator module. | TC\_002 |

|  |  |
| --- | --- |
| 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 | Verify that sellers can access the Payment Management page | Payment Management page should be accessible | Site launched successfully | Pass |  | [https://accounts. google.com/Ser viceLogin](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 require d when human  action validati on | get captcha from image which is near by captcha  field |  |
|  |  |  |  |  | perfor m |  |  |
| 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 | Should be display an error message | Display an error  Username not found | Pass |  |  |  |
|  | invalid  Username and Valid Password | User not found |  |  |  |  |  |
| 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.googl e.com/ServiceLogin](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  “Usernam  e” text box on the  page load | Cursor is  focused  in  Usernam  e textbox | Cursor  focus in Username textbox | Pass |  |  |  |
| 4 | Verify that tab functionali  ty 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  placeholde  r | All text  fields have proper  placehold er | 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 |  |  |  |

# 7 References

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

**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 | Verify that the valid user can access the "Set User Permissions" page | The page should be accessible successfully | Page accessed successfully | Pass |  | User: Valid User |  |
| 2 | Verify that the valid user can select permissions from the available options | Permissions should be selectable without issues | Permissions selected successfully | Pass |  | Permissions: Admin |  |
| 3 | Verify that the user can save the selected permissions | Permissions should be saved successfully | Permissions saved successfully | Pass |  | Permissions saved: Admin, User |  |
| 4 | Verify that a success message is displayed after saving permissions | A success message should appear after saving | Success message displayed | Pass |  |  |  |

### Test Case Title: **Set User Permissions - Admin Role**

**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 | Verify that the admin user can access the "Set User Permissions" page | The page should be accessible successfully for admin | Page accessed successfully for admin | Pass |  | User: Admin |  |
| 2 | Verify that the admin user can assign permissions to other users | Permissions should be assignable to other users | Permissions assigned successfully | Pass |  | Assigned Permissions: User, Admin |  |
| 3 | Verify that the admin can save the selected permissions for other users | Permissions should be saved successfully for the selected users | Permissions saved successfully for other users | Pass |  | Permissions: Admin for User1, User for User2 |  |
| 4 | Verify that a success message is displayed after assigning permissions | Success message should appear after assignment | Success message displayed | Pass |  |  |  |