

Darshan University

A Project Report on

**“Hostel Management System”**

Under the subject

**Software Engineering (2101CS503)**

B. Tech, Semester – VI

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name: Renish Hirani | Enrollment No.: 22010101068 |
| Academic Year  (2023-2024) | |
| Internal Guide  Prof. XYZ  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** **(2101CS503)** for entitled **“Hostel 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 (**Guide name)** and that no part of any of report has been directly copied from any students’ reports, without providing due reference.

(Renish Hirani)

Student’s Signature

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

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

**CERTIFICATE**

This is to certify that the SRS on **“Hostel Management System” has** been satisfactorily prepared by **Renish Hirani** (**22010101068**) under my guidance in the fulfillment of the course **Software Engineering (2101CS503)** work during the academic year 2023-2024.

|  |  |  |
| --- | --- | --- |
| Internal Guide  Prof., Guide Name  Darshan University |  | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

**Acknowledgement**

I wish to express my sincere gratitude to my project guide Prof. **Guide name** and 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

**Renish Hirani**

**ABSTRACT**

This Project “ **HOSTEL MANAGEMENT INFORMATION SYSTEM**” targeted for the College Hostel integrates the transaction management of the Hostel for better control and timely response. Thiseliminates time delay and paper transactions being marked. The warden is provided with a bettercontrol over the transactions like adding the details of new students in the hostel, modifying the details of the students, deleting the students, viewing the students details in the Hostel. This project’s main motto is to reduce the effort of Wardens and provide better service to the students.The goal of this project is to develop a system for the computerization of the Hostel. The commontransactions of the hostel includes the maintenance of mess bills, information about students in thehostel, enrolling of new students and their payments and dues etc are stored into the databases andreports are generated according to the user requirements.

**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

### Student/Hosteler

* Login : Students should be able to login themselves into the hostel management system and manage their profiles.
* Room Allocation: The system should facilitate the allocation and assignment of rooms to students.
* Fees Payment: Students should be able to view their hostel fees, make payments, and receive invoices electronically.
* Maintenance Request And Tracking: Student should be able to submit maintenance requests and track their status
* Community Notice: Students should have access to hostel announcements, notices, and community events.
* Complaints : Students should be able to submit complaints and suggestions through the system.
* Feedback: Students should be able to submit feedback through the system.
* Security And Access Control: The system should ensure secure access to hostel facilities and personal information.
* Calendar And Event Booking: Students should be able to view and book hostel facilities for events or activities.
* Health And Wellness Resources: Students should have access to health and wellness resources and information through the system.
* Transportation Service Information: Students should be able to access information about transportation services available near the hostel.
* Notification Preferences and Alerts: Students should be able to customize their notification preferences and receive alerts related to hostel activities.
* Lost And Found: Students should have a mechanism to report lost items and track their status.

### Parents

* Parent Registration and Access: Parents should be able to register themselves and obtain access to the hostel management system.
* Student Information and Profile Viewing: Parents should be able to view their child's hostel profile and personal information.
* Room Allocation and Roommate Information: Parents should have access to information about their child's room allocation and roommate.
* Fee Payment and Financial Transactions: Parents should be able to view hostel fees, make payments, and receive financial statements.
* Emergency Contact and Notification Settings: Parents should be able to update emergency contact information and set notification preferences.
* Health and Medical Records Access: Parents should have access to their child's health and medical records stored in the hostel management system.
* Communication with Hostel Staff: Parents should be able to communicate with hostel administrators or staff members.
* Events and Activities Participation: Parents should be informed about hostel events, activities, and participation opportunities.
* Safety and Security Measures: Parents should have access to information about hostel safety and security measures.
* Educational Support and Resources: Parents should have access to educational resources and support services available through the hostel.
* Community Engagement and Updates: Parents should receive updates and newsletters about hostel community news and developments.
* Transportation and Travel Information: Parents should have access to information about transportation options and travel arrangements near the hostel.

### Admin

* User Manage: Administrators should be able to manage user accounts and permissions.
* Room Allocation and Management: Administrators should manage room allocation, assignments, and occupancy.
* Fees: Administrators should manage hostel fees, payment schedules, and financial transactions.
* Reporting and Analytics: Administrators should access reports and analytics on hostel occupancy, financial performance, and operational metrics.
* Complaints and Grievance Handling: Administrators should handle complaints, feedback, and grievances from students and parents.
* Security and Access Control: Administrators should manage security measures and access control within the hostel premises.
* Event and Facility Booking: Administrators should manage bookings for hostel facilities and coordinate events.
* Health and Safety Compliance: Administrators should ensure compliance with health and safety regulations within the hostel premises.
* Training and Staff Development: Administrators should manage training programs and professional development for hostel staff.
* Transportation Services Coordination: Administrators should coordinate transportation services for students living in the hostel.
* Document Management and Archiving: Administrators should manage documentation and archives related to hostel operations.
* Quality Assurance and Continuous Improvement: Administrators should implement quality assurance measures and initiatives for hostel services.
* Scalability and Customization: The HMS should be scalable and customizable to accommodate growth and meet specific institutional requirements.

### Warden

* Dashboard and Overview: Admin should have secure login credentials and role-based access to the system.
* Room Allocation: Ability to assign rooms, update allocations, and handle room transfers.
* Students Registration and Records: Capture student details (name, contact, course) and maintain records.
* Billing and Payments: Generate bills, track payments, and manage financial transactions.
* Attendance Tracking: Record student check-ins, check-outs, and attendance.
* Complaint Manage: Receive, assign and track complaints from students.
* Inventory Manage: Maintain Stock of hostel supplies(bedding, toiletries etc.)
* Maintenance Requests: Handle maintenance requests(plumbing, electrical etc.)
* Housekeeping schedules: Set Cleaning Schedules For Rooms And Common Areas.
* Security And Access Control: Manage access cards, CCTV monitoring, and security protocols.
* Emergency Procedure: Define evacuation plans, emergency contacts, and protocols.
* Event Manage: Plan And Organize Hostel events (workshops, celebration etc)
* Document Manage: Store And Manage Important Documents(contracts, policies).
* System Customization And Setting: Configure System Parameters(tax rates, room types).

## Non-Functional Requirement

### Usability:

* The user interface shall be intuitive and user-friendly, requiring minimal training for hostel staff to use effectively.

### Scalability:

* It should be capable of handling an increase in the number of hostels and students without requiring major architectural changes.

### Availability:

* The system should be available 99.99% of the time to ensure uninterrupted access for users.

### Reliability:

### Data integrity should be maintained with a backup and recovery mechanism to ensure minimal data loss in case of system failure.

### Accessibility:

### The system should be accessible from different devices (desktops, tablets, mobile phones) and compatible with popular web browsers.

### Localization:

* It should support multiple languages and currencies to accommodate international students and staff.

### **Response Time**:

* The average response time for user actions (e.g., booking rooms, updating records) should be less than 2 seconds.

### **Backup and Recovery**:

* Regular automated backups of data should be performed with a tested recovery procedure to ensure quick restoration of services in case of data loss.

### User Support:

* There should be a dedicated helpdesk or support team to assist users with system-related queries and issues.

### **Backup and Recovery**:

* Regular automated backups of data should be performed with a tested recovery procedure to ensure quick restoration of services in case of data loss.

# Design and Implementation Constraints

## Use case diagram

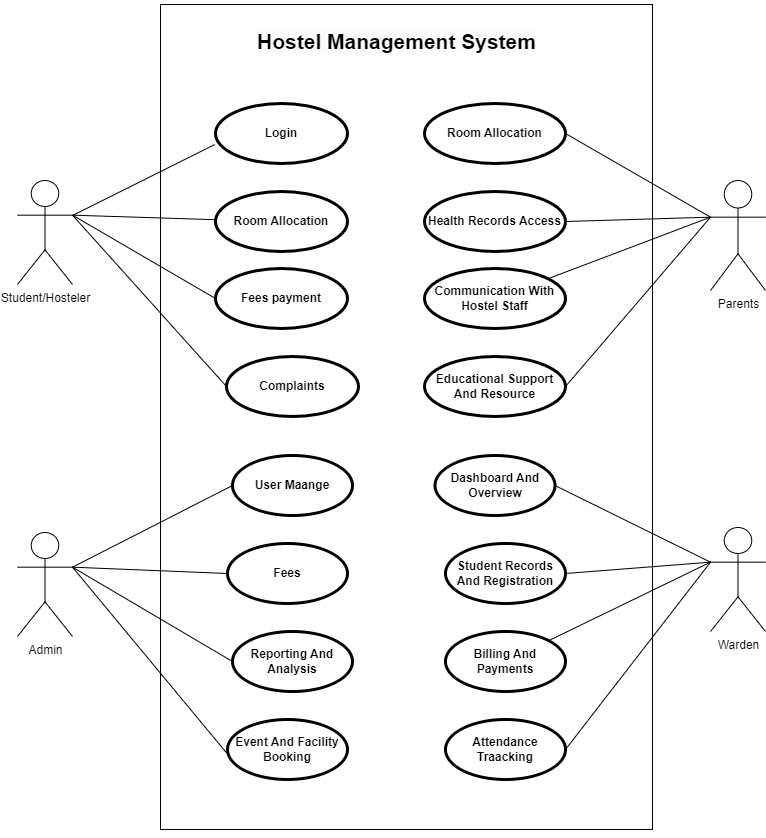


Figure 2.1‑1 Use case diagram for Hostel management system

## Activity diagram and Swimlane diagram

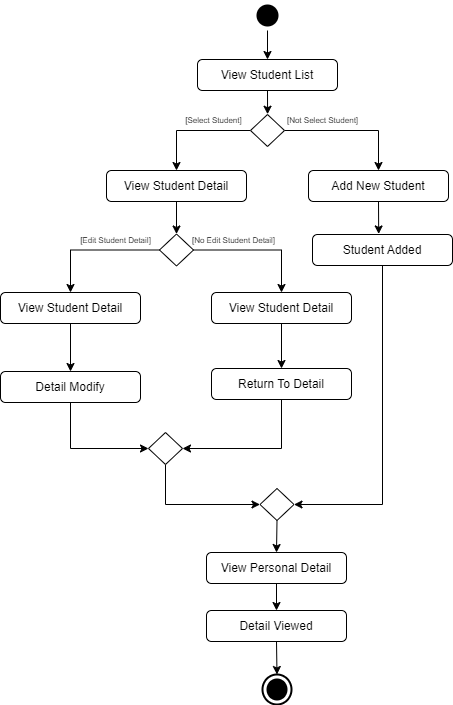


Figure 2.2‑1 Activity diagram for View Student List

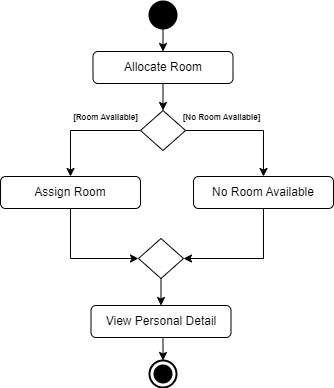


Figure 2.2‑1 Activity diagram for Allocate Room

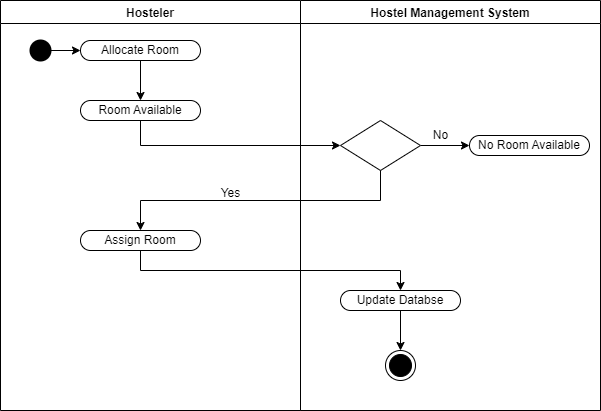


Figure 2.2‑2 Swimlane diagram for Room Available

## Sequence diagram

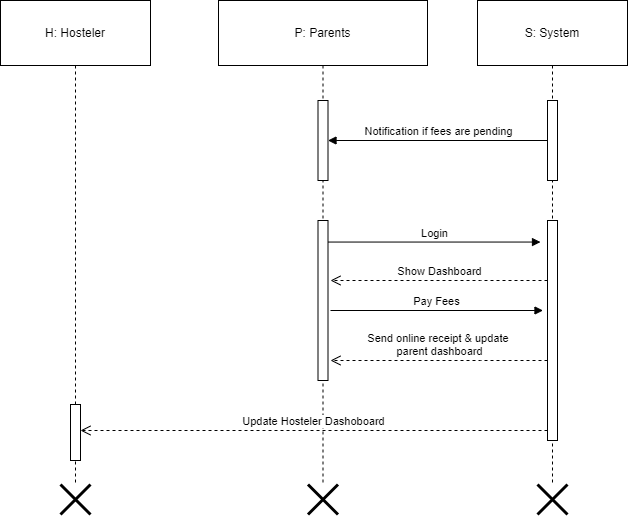


Figure 2.3‑1 Sequence diagram for Fees Payment

## 

## 

Figure 2.3‑2 Sequence diagram for Leave Appliacation

## State diagram

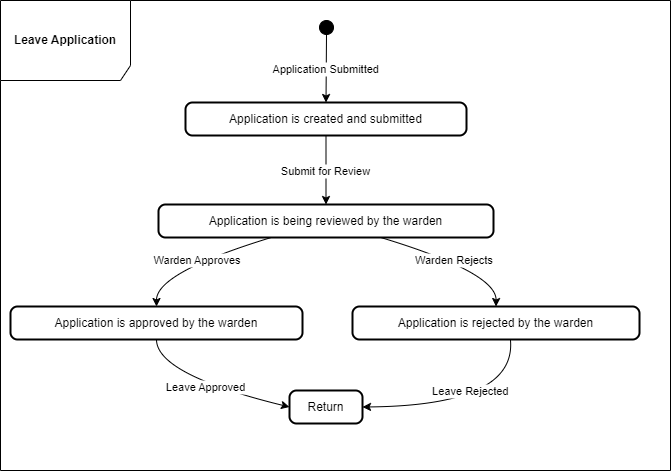


Figure 2.4‑1 State diagram of Leave Application

## Class diagram

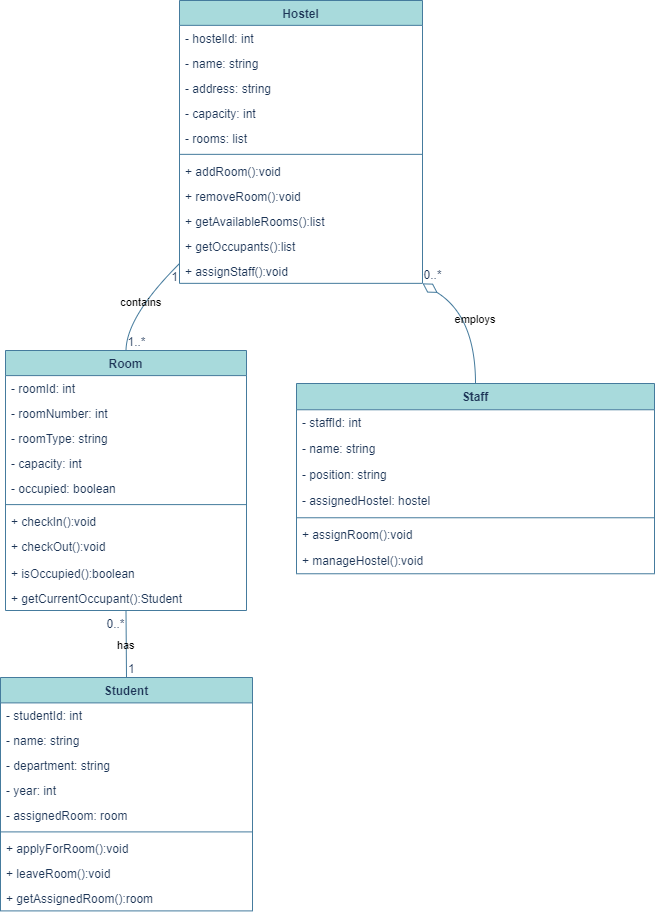


Figure 2.5‑1 Class diagram for Hostel management system

## Data flow diagram

### Context diagram (level-0)

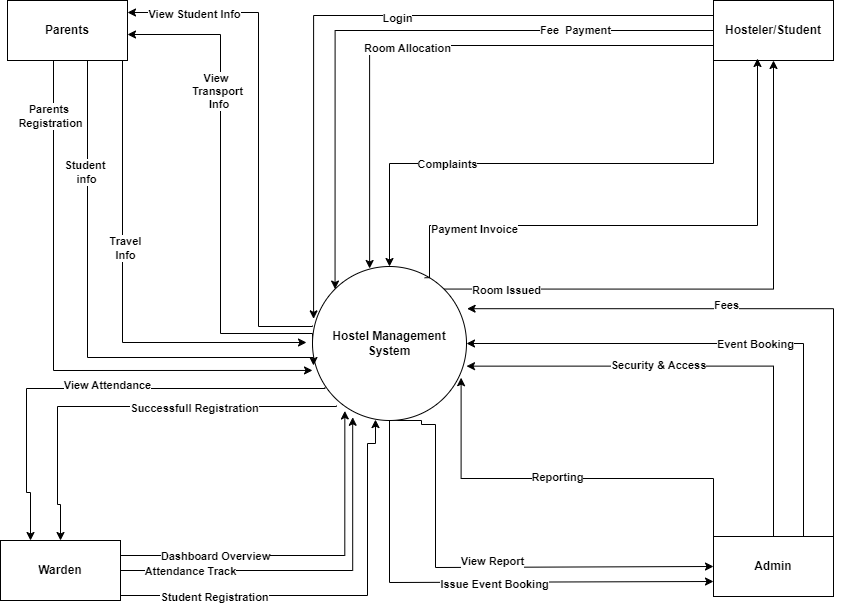


Figure 2.6‑1 Context diagram for Hostel management system

### DFD Level-1

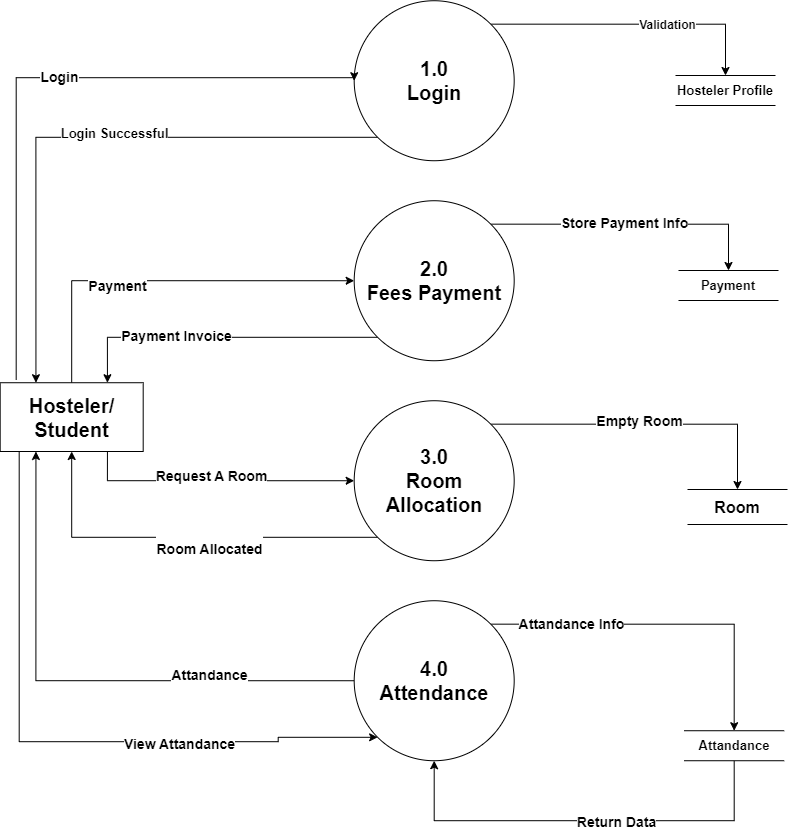


Figure 2.6‑2 DFD level-1 for Hostel management system

# External interface requirement (Screens)

## Screen-1: Painting Home page

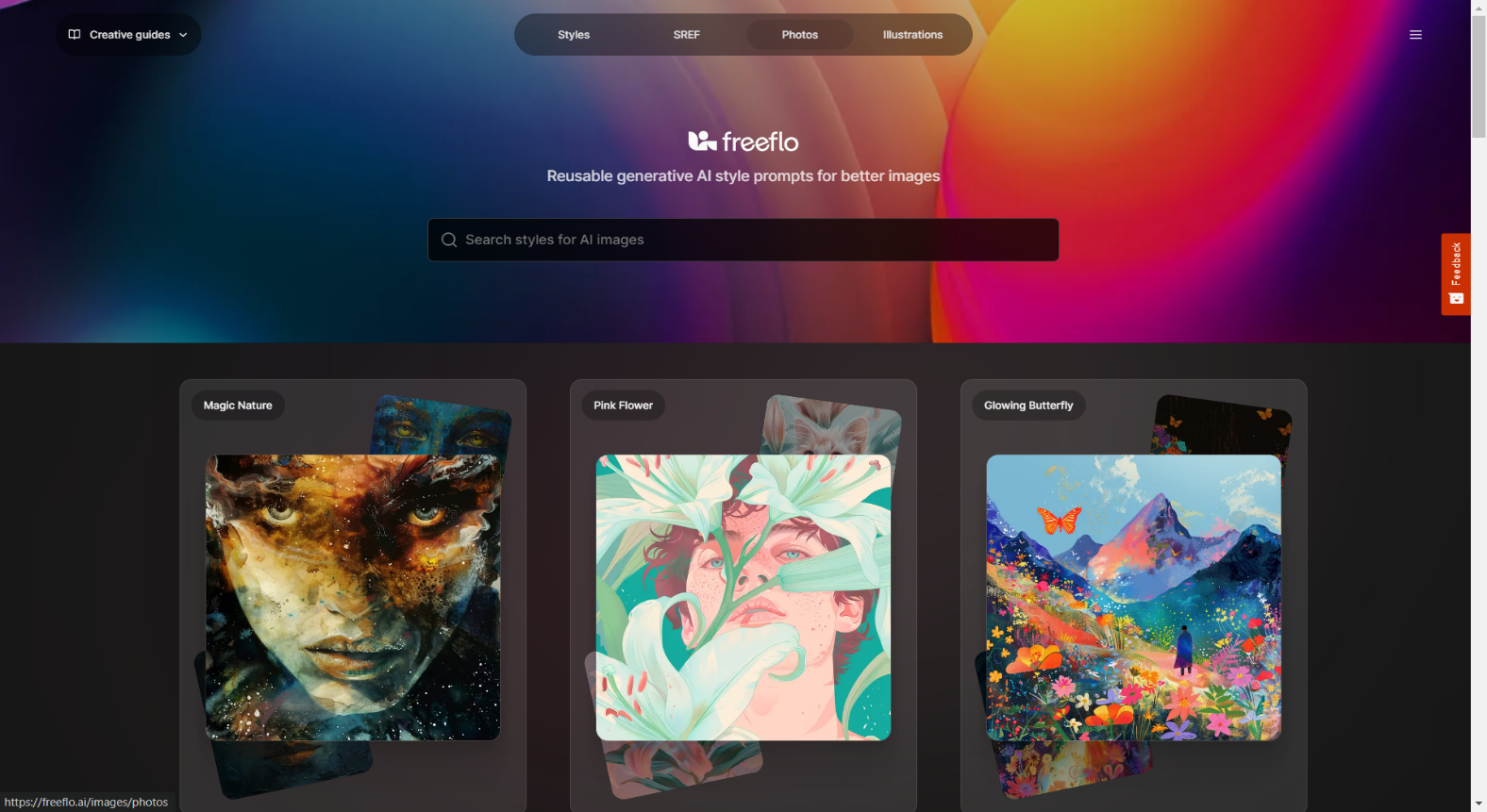


Figure 3.1‑1 Screen-1: Painting Home page

**Purpose:** This Screen Will Be Display Art Web Home Page.

Table 3.1‑1 Screen element of Painting Home Page

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Search Bar | Textbox | M | 1 | Search To Any Card |
| 2 | Card | Card | M | N | Display The Product |
| 3 | Paint Image | Image | M | 1 | Paint Image |
| 4 | Title Of Card | Text | M | 1 | Show To Card Title |
| 5 | Sidebar | Button | M | 1 | Open The Side Bar |
| 6 | Navbar | Link | M | N | Display Menu |

## Screen-2: Add Product Form

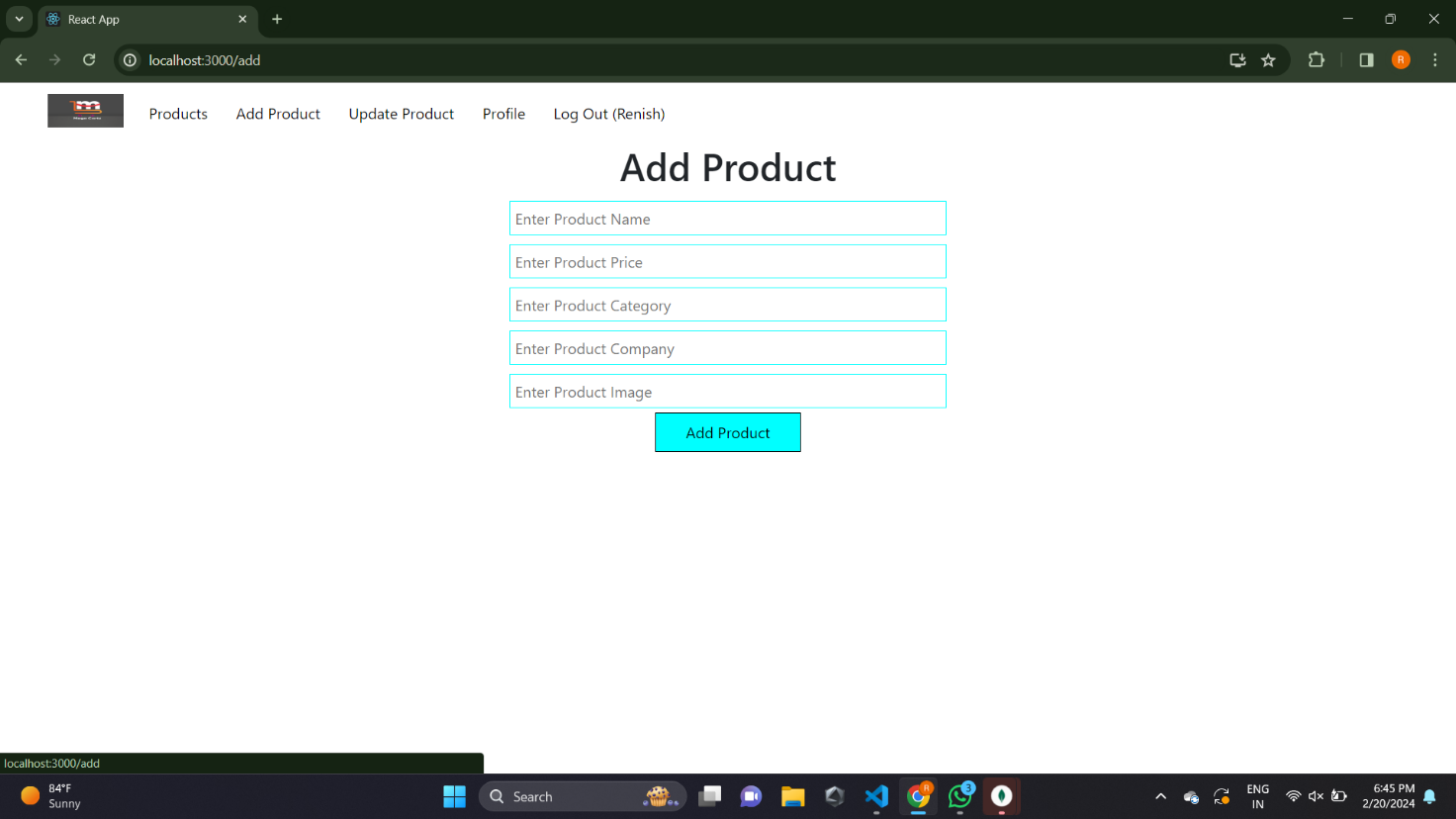


Figure 3.2‑1 Screen-2: Add Product Form

**Purpose:** This form will be used by the system’s Admin to Add records and features of the system. The Admin will input the correct combination of their to be able to Add Product to the system.

Table 3.2‑1 Screen element of Add Product Form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Add Product | Button | M | 1 | Add To product |
| 2 | Navbar | Link | M | N | To Display A Menu |
| 3 | Logo | Image | M | 1 | Showing To Web Logo |
| 4 | Enter Product Name | Textbox | M | 1 | Enter The Product Name |
| 5 | Enter Product Price | Textbox | M | 1 | Enter Product Price |
| 6 | Enter Product Category | Textbox | M | 1 | Enter product Category |
| 7 | Enter product Company | Textbox | M | 1 | Enter Product Company Name |
| 8 | Enter Product Image | Textbox | M | 1 | Enter Product Image |

## Screen-3: Product List Page

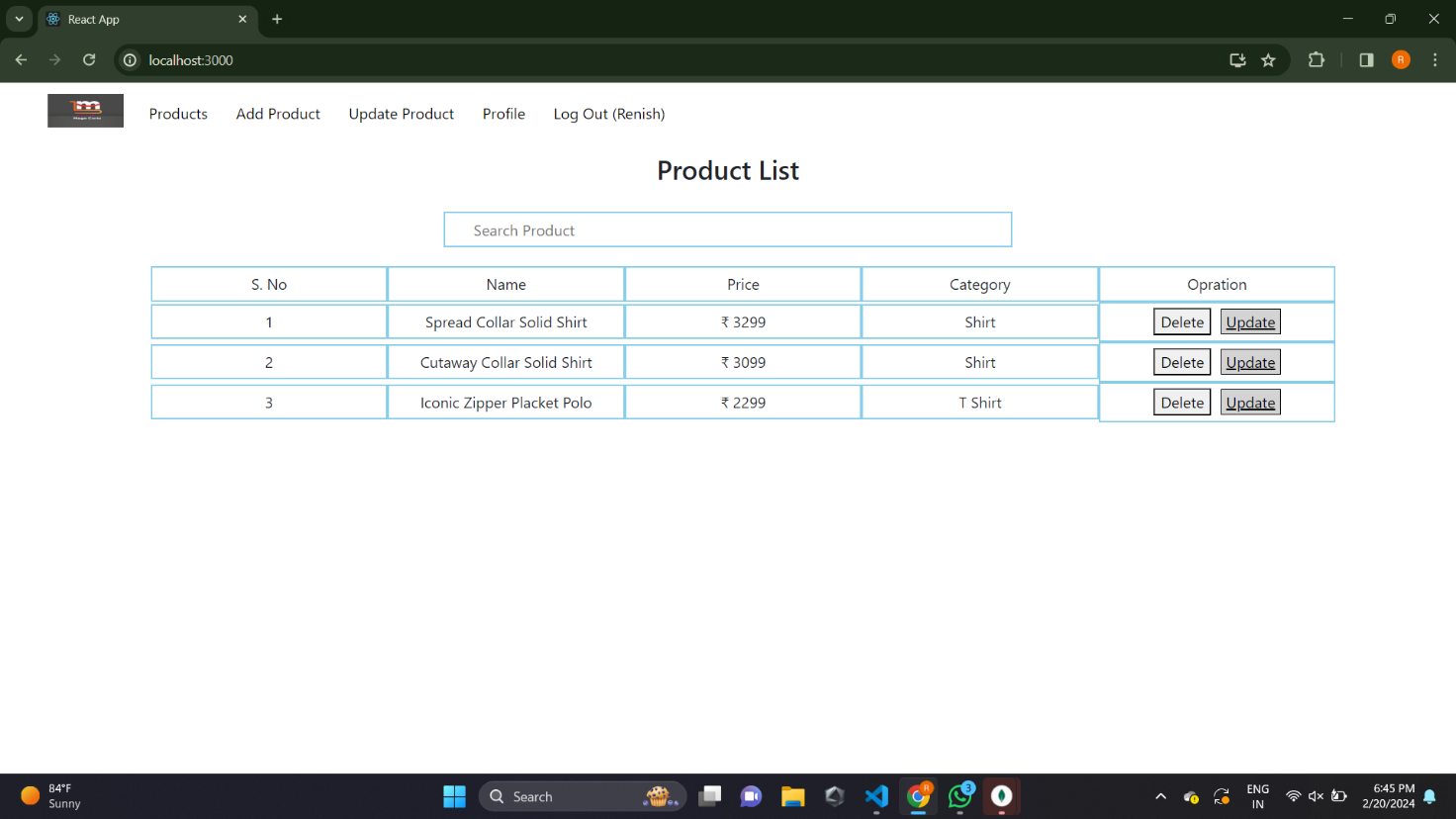


Figure 3.3‑1 Screen-3: Product List Page

**Purpose:** This Screen Will Be Display Product List.

Table 3.3‑1 Screen element of Product List Page

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Delete | Button | M | 1 | Delete A Product |
| 2 | Update | Button | M | 1 | Update A Product |
| 3 | Navbar | Link | M | N | To Display A Menu |
| 4 | Logo | Image | M | 1 | Showing To Web Logo |
| 5 | Search Product | Textbox | M | 1 | Search The Product Name |
| 6 | Sr No. | Text | M | 1 | Display Serial Number |
| 7 | Name | Text | M | 1 | Display Product Name |
| 8 | Price | Text | M | 1 | Display Product Price |
| 9 | Category | Text | M | 1 | Display Product Category |
| 10 | Product List | Text | O | 1 | Title Of Page |

## Screen-4: Add Member Page

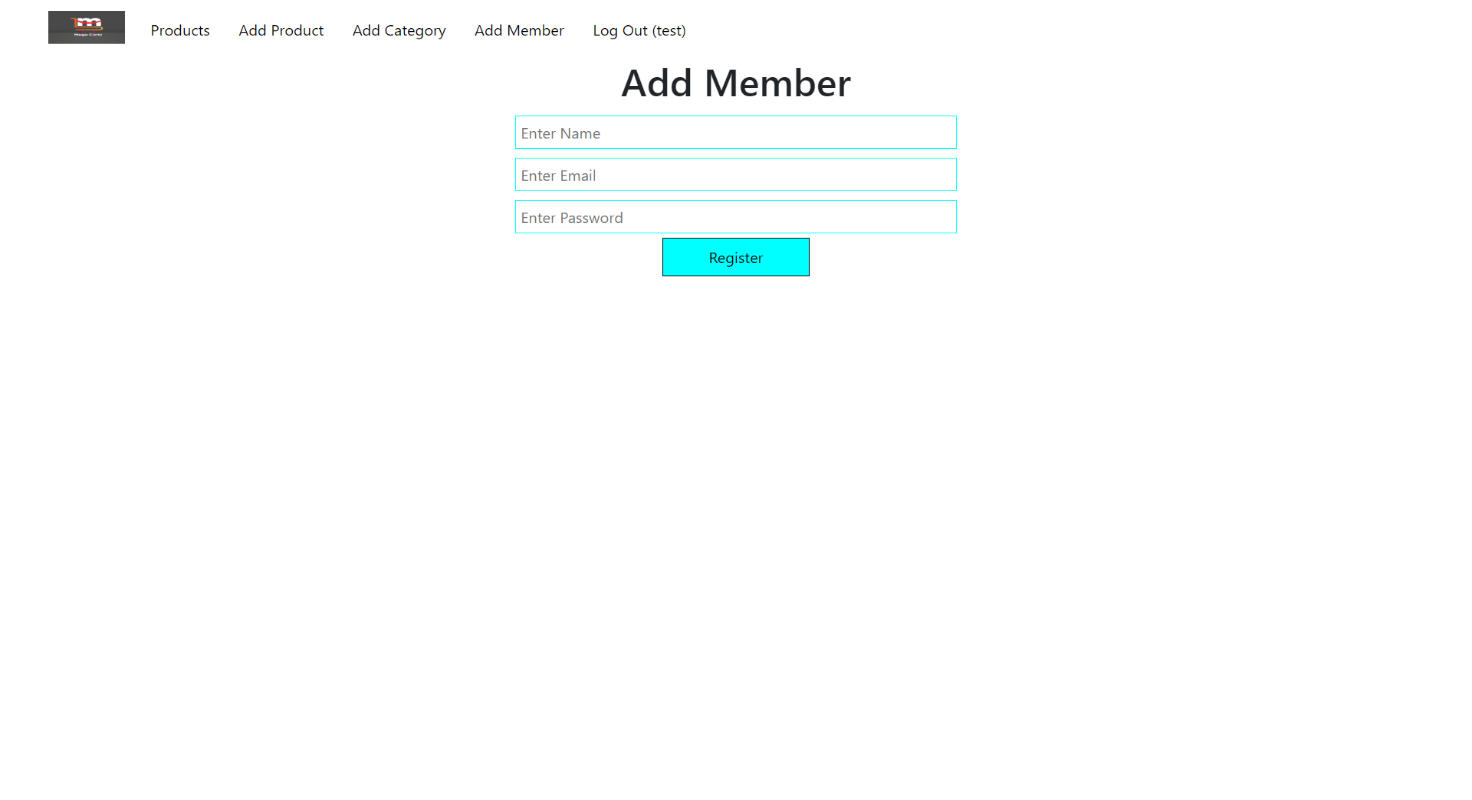


Figure 3.2‑1 Screen-2: Add Member Form

**Purpose:** This form will be used by the system’s Admin to Add Member and features of the system. The Admin will input the correct combination of their to be able to Add member to the system.

Table 3.2‑1 Screen element of Add Product Form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Register | Button | M | 1 | Add To Member |
| 2 | Navbar | Link | M | N | To Display A Menu |
| 3 | Logo | Image | M | 1 | Showing To Web Logo |
| 4 | Enter Name | Textbox | M | 1 | Enter Product Name |
| 5 | Enter Email | Textbox | M | 1 | Enter Email |
| 6 | Enter Password | Textbox | M | 1 | Enter Password |
| 7 | Add Member | Text | O | 1 | Title The Page |

## Screen-5: E-Commerce Home Page

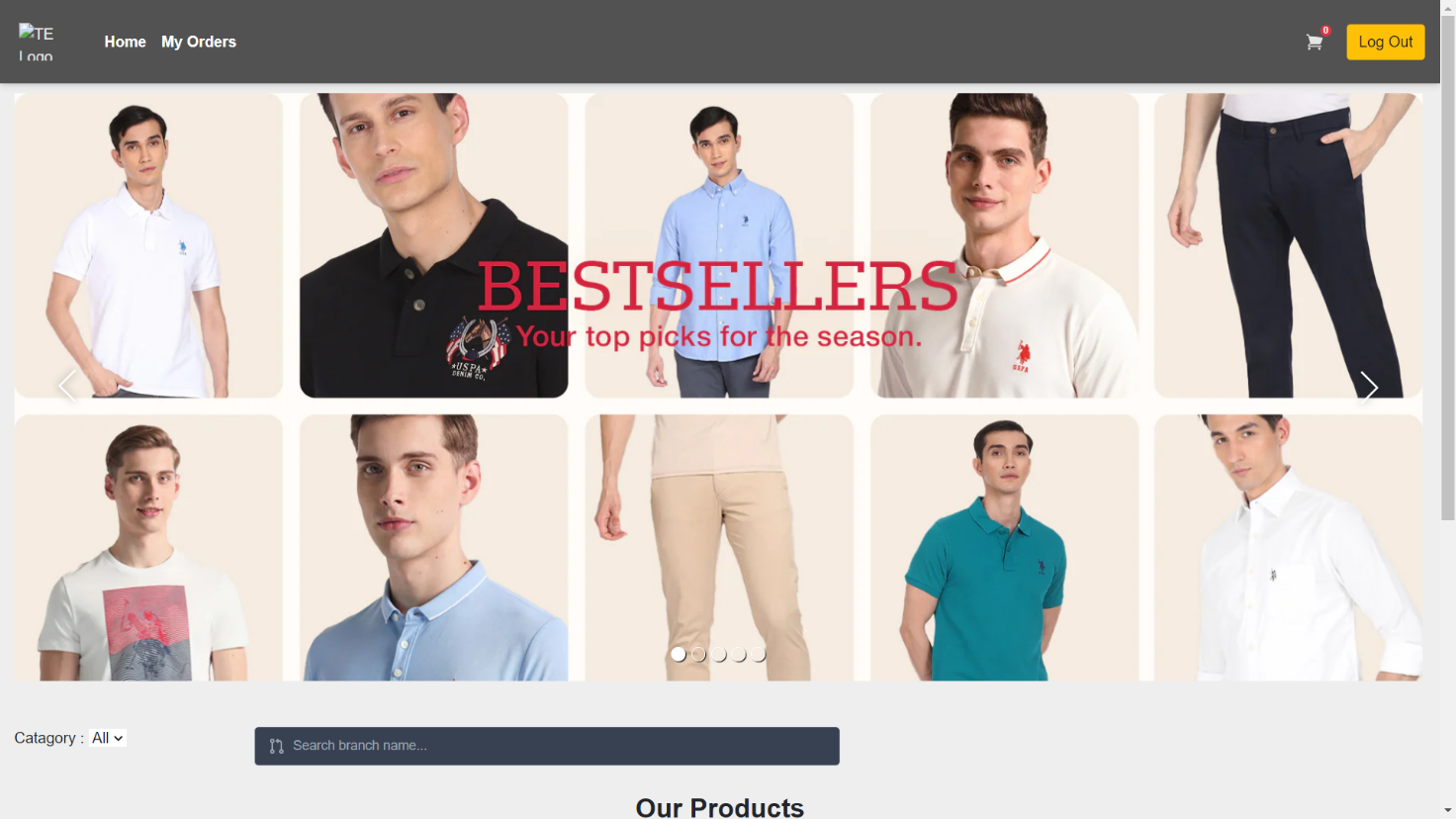


Figure 3.2‑1 Screen-2: E-Commerce Web Home Page

**Purpose:**

Table 3.2‑1 Screen element of E-Commerce Web Home Page

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Logout | Button | M | 1 | Log Out Screen |
| 2 | Navbar | Link | M | N | To Display Menu |
| 3 | Logo | Image | M | 1 | Showing To Web Logo |
| 4 | Cart | Button | M | 1 | Display Add Product Count |
| 5 | Search Bar | Textbox | M | 1 | Search Product Name |
| 6 | Category | Dropdown | M | 1 | Display All Category |
| 7 | Image | Image | O | N | Display Product Image |

# Database design

## List of Tables

* Admin
* Warden
* Hosteler
* Parents
* Room

Table 4.1‑1 Table: Admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| AdminID | int | NN | Primary Key | Auto Increment Start With 1 |
| UserName | varchar(100) | NN | - | - |
| Password | varchar(25) | NN |  | - |
| Email | varchar(50) | AN | Unique Key | Unique Email Address |
| PhoneNumber | varchar(10) | AN | Unique Key | Unique Phone Number |
| HostelerID | Int | NN | Foreign Key | Reference Of Hosteler Table |
| ParentID | Int | NN | Foreign Key | Reference Of Parents Table |
| WardenID | Int | NN | Foreign Key | Reference Of Warden Table |

Table 4.1‑2 Table: Hosteler

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| HostelerID | int | NN | Primary Key | Auto Increment Start With 1 |
| FullName | varchar(100) | NN | - | - |
| Gender | varchar(10) | NN | - | - |
| DateOfBirth | Datetime | NN | - | - |
| Email | varchar(50) | AN | Unique Key | Unique Email Address |
| PhoneNumber | varchar(10) | AN | Unique Key | Unique Phone Number |
| AdmissionDate | Datetime | NN | - | - |
| RoomID | Int | NN | Foreign Key | Reference Of Room Table |

Table 4.1‑3 Table: Parents

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| ParentID | int | NN | Primary Key | Auto Increment Start With 1 |
| FulllName | varchar(100) | NN | - | - |
| RelationShip | Varchar(50) | NN | - | Relationship to the hosteler (e.g., Father, Mother, Guardian) |
| HostelerID | int | NN | Foreign Key | Reference Of Hosteler Table |
| Email | varchar(50) | AN | Unique Key | Unique Email Address |
| PhoneNumber | varchar(10) | AN | Unique Key | Unique Phone Number |

Table 4.1‑4 Table: Warden

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| WardenID | int | NN | Primary Key | Auto Increment Start With 1 |
| FullName | varchar(100) | NN | - | - |
| Gender | varchar(100) | NA | - | - |
| Email | varchar(50) | AN | Unique Key | Unique Email Address |
| PhoneNumber | varchar(10) | AN | Unique Key | Unique Phone Number |

Table 4.1‑5 Table: Room

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| RoomID | int | NN | Primary Key | Auto Increment Start With 1 |
| RoomNumber | Int | NN | Unique Key | Unique Room Number |
| RoomType | varchar(100) | NN | - | Type of the room (e.g., Single, Double) |
| Capacity | Int | NN | - | Maximum number of occupants allowed |
| Availability | varchar(10) | NN | - | Current status of the room |

# Stories and Scenario

## 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. |

### 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. |

### 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. |

### 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. |

## 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. |

## 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. |

## 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. |

## 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. |

# 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