

Darshan University

A Project Report on

**“College Management System”**

Under the subject

**Software Engineering (2301CS405)**

B. Tech, Semester – IV

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name: Het Bhalani | Enrollment No.: 23010101019 |
| 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 **“College 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.

Het Bhalani

Student’s Signature

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

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

**CERTIFICATE**

This is to certify that the SRS on **“College Management System” has** been satisfactorily prepared by **Het Bhalani** (**23010101019**) 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

**Het Bhalani**

**ABSTRACT**

A College Management System (CMS) is an integrated software solution designed to automate and streamline administrative and academic functions of a college or university. It centralizes activities like student registration, course enrollment, faculty management, attendance filling, scheduling. The system offers role-based access for students, faculty, and visitors each with tailored interfaces and permissions.Students can register for courses, track performance, view grades, and communicate with instructors. Faculty can manage course content, attendance, assignments, and engage with students. Administrators manage courses, schedules, resources, admissions, fee collections, and financial reports.The CMS reduces manual paperwork, minimizes scheduling conflicts, and centralizes academic and administrative data. It ensures data accuracy, and real-time communication about schedules, grades, and announcements. The system is scalable, supporting growth in students, courses, and faculty, while integrating new features. Security is prioritized through role-based access control, encryption, and multi-factor authentication.With modular architecture, the CMS is flexible, offering high availability and easy integration with third-party systems. It ultimately enhances the experience for students, faculty, and staff by improving operational efficiency and communication.

**Table of Contents**

[List of Figures I](#_Toc186218366)

[List of Tables II](#_Toc186218367)

[1 Introduction 1](#_Toc186218368)

[1.1 Product perspective 1](#_Toc186218369)

[1.2 Product features 1](#_Toc186218370)

[1.2.1 Different user roles: 1](#_Toc186218371)

[1.2.2 The features that are required for the Librarian are: 1](#_Toc186218372)

[1.3 Functional Requirement 1](#_Toc186218373)

[1.3.1 User Authentication and Role Assignment 1](#_Toc186218374)

[1.3.2 Student feature 1](#_Toc186218375)

[1.3.3 Faculty feature 1](#_Toc186218376)

[1.3.4 Course feature 2](#_Toc186218377)

[1.3.5 Examination and Results 2](#_Toc186218378)

[1.3.6 Attendance Tracking 2](#_Toc186218379)

[1.3.7 Fee Handling 2](#_Toc186218380)

[1.3.8 Notifications and Alerts 2](#_Toc186218381)

[1.3.9 Library Features 2](#_Toc186218382)

[1.3.10 Hostel Features 2](#_Toc186218383)

[1.3.11 Logs and Reports 2](#_Toc186218384)

[1.3.12 Learning Modules 2](#_Toc186218385)

[1.3.13 Communication Tools 3](#_Toc186218386)

[1.4 Non-Functional Requirements 3](#_Toc186218387)

[1.4.1 Performance 3](#_Toc186218388)

[1.4.2 Scalability 3](#_Toc186218389)

[1.4.3 Security 3](#_Toc186218390)

[1.4.4 Usability 3](#_Toc186218391)

[1.4.5 Availability 3](#_Toc186218392)

[2 Design and Implementation Constraints 4](#_Toc186218393)

[2.1 Use case diagram 4](#_Toc186218394)

[2.2 Activity diagram and Swimlane diagram 5](#_Toc186218395)

[2.3 Sequence diagram 7](#_Toc186218396)

[2.4 State diagram 9](#_Toc186218397)

[2.5 Class diagram 10](#_Toc186218398)

[2.6 Data flow diagram 11](#_Toc186218399)

[2.6.1 Context diagram (level-0) 11](#_Toc186218400)

[2.6.2 DFD Level-1 11](#_Toc186218401)

[2.6.3 DFD Level-2 12](#_Toc186218402)

[3 External interface requirement (Screens) 13](#_Toc186218403)

[3.1 Screen-1: Registration Form 13](#_Toc186218404)

[3.2 Screen-2: Login Form 14](#_Toc186218405)

[3.3 Screen-3: Add borrower book 15](#_Toc186218406)

[4 Database design 16](#_Toc186218407)

[4.1 List of Tables 16](#_Toc186218408)

[5 Stories and Scenario 17](#_Toc186218409)

[5.1 Story-1: Add New Book in Library Catalogue 17](#_Toc186218410)

[5.1.1 Scenario# S1.1 17](#_Toc186218411)

[5.1.2 Scenario# S1.2 17](#_Toc186218412)

[5.1.3 Scenario# S1.3 17](#_Toc186218413)

[5.2 Story-2: Search Book 18](#_Toc186218414)

[5.3 Story-3: Manage due date for borrowed book 18](#_Toc186218415)

[5.4 Story-4: Renew book 18](#_Toc186218416)

[5.5 Story-5: Generate a report on book usage and availability 18](#_Toc186218417)

[6 Test cases 19](#_Toc186218418)

[7 References 23](#_Toc186218419)

# List of Figures

[Figure 2.1‑1 Use case diagram for library management system 4](#_Toc146706848)

[Figure 2.2‑1 Activity diagram for Book Issue 5](#_Toc146706849)

[Figure 2.2‑2 Swimlane diagram for Book Issue 7](#_Toc146706850)

[Figure 2.3‑1 Sequence diagram for Book Issue 8](#_Toc146706851)

[Figure 2.4‑1 State diagram of Book 9](#_Toc146706852)

[Figure 2.4‑2 State diagram for Librarian 9](#_Toc146706853)

[Figure 2.5‑1 Class diagram for Library management system 10](#_Toc146706854)

[Figure 2.6‑1 Context diagram for Library management system 11](#_Toc146706855)

[Figure 2.6‑2 DFD level-1 for Library management system 11](#_Toc146706856)

[Figure 2.6‑3 DFD level-2 for Issue book 12](#_Toc146706857)

[Figure 3.1‑1 Screen-1: Registration Form 13](#_Toc146706858)

[Figure 3.2‑1 Screen-2: Login Form 14](#_Toc146706859)

[Figure 3.3‑1 Screen-3: Add borrower book 15](#_Toc146706860)

# List of Tables

[Table 3.1‑1 Screen element of Registration form 13](#_Toc146706861)

[Table 3.2‑1 Screen element of Login form 14](#_Toc146706862)

[Table 3.3‑1 Screen element of Add borrower book 15](#_Toc146706863)

[Table 4.1‑1 Table: Book 16](#_Toc146706864)

[Table 4.1‑2 Table: Borrower 16](#_Toc146706865)

[Table 4.1‑3 Table: Staff 16](#_Toc146706866)

[Table 4.1‑4 Table: Student 16](#_Toc146706867)

# Introduction

## Product perspective

This project is about upgrading the manual college management system to an internet-based application that integrates various academic and administrative functions into one unified platform. It aims to simplify operations like student management, faculty scheduling, and course handling, catering to the comprehensive needs of a college.

## Product features

### Different user roles:

* **Students**: Access grades, attendance, schedules, and notifications.
* **Faculty**: Manage schedules, evaluate assignments, and interact with students.
* **Administrators**: Handle admissions, payroll, and resource allocations.
* **Parents**: Monitor student progress and access fee-related details.
* **Institutional Management**: Generate analytical reports for decision-making.

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

* Automation of administrative workflows.
* Centralized data storage for academic, operational, and financial information.
* Role-based data access.
* Transparent records management for accountability.

## Functional Requirement

### **Student**

* View profile
* Access timetable
* Give test
* Check test marks
* Download result
* Check attendance

### **Faculty**

* Fill attendance
* Add subject ppt
* Upload marks of test
* Access timetable
* Generate attendance report

### **Teaching Assistant**

* Access assigned course material
* Give grades to student
* Fill attendance
* Update attendance
* Check payment receipt

### **Admin**

* CRUD students
* Assign courses to faculty
* Upload fee receipt
* Upload course material
* Upload important announcements

### **Visitor**

* Download brochures
* Browse general info of collage
* Check admission process
* Check course fees
* Contact collage

## Non-Functional Requirements

### Performance

* The system shall support concurrent access by at least 500 users without performance degradation.

### Scalability

* The system shall be designed to scale with increasing user numbers and expanding institutional needs.

### Security

* The system shall use encrypted data storage and transmission to ensure the security of sensitive information.

### Usability

* The system shall have an intuitive user interface, ensuring accessibility for users with varying technical expertise.

### Availability

* The system shall ensure 99.9% uptime, with minimal interruptions for maintenance.

# Design and Implementation Constraints

## Usecase diagram

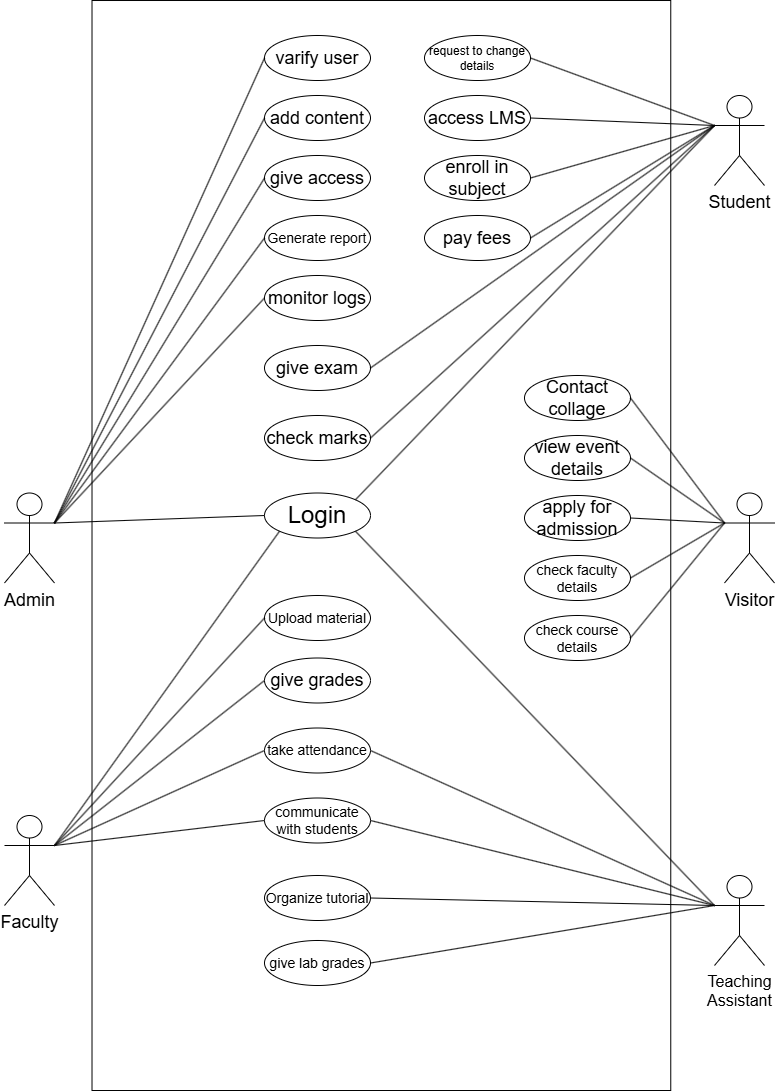


Figure 0‑1 Use case diagram for collage management system

## Activity diagram and Swimlane diagram

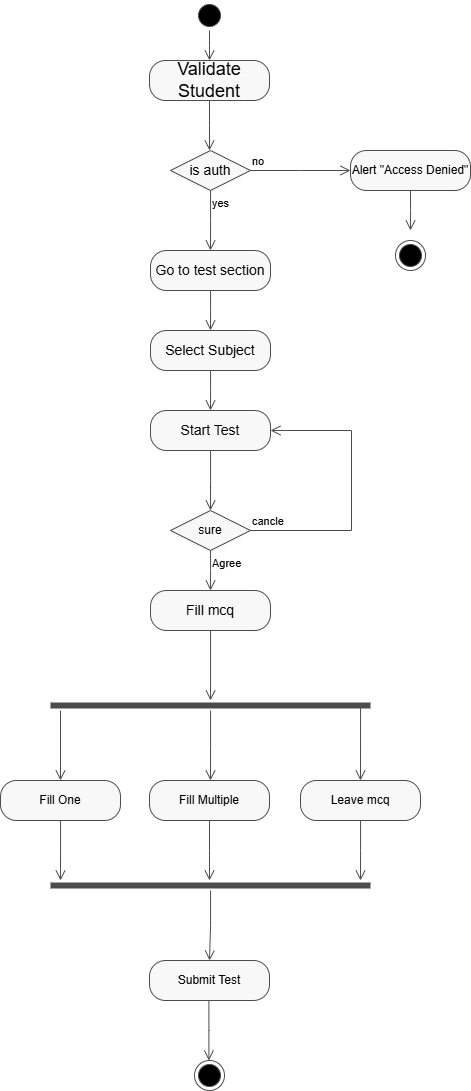


Figure 0‑1 Activity diagram for MCQ test

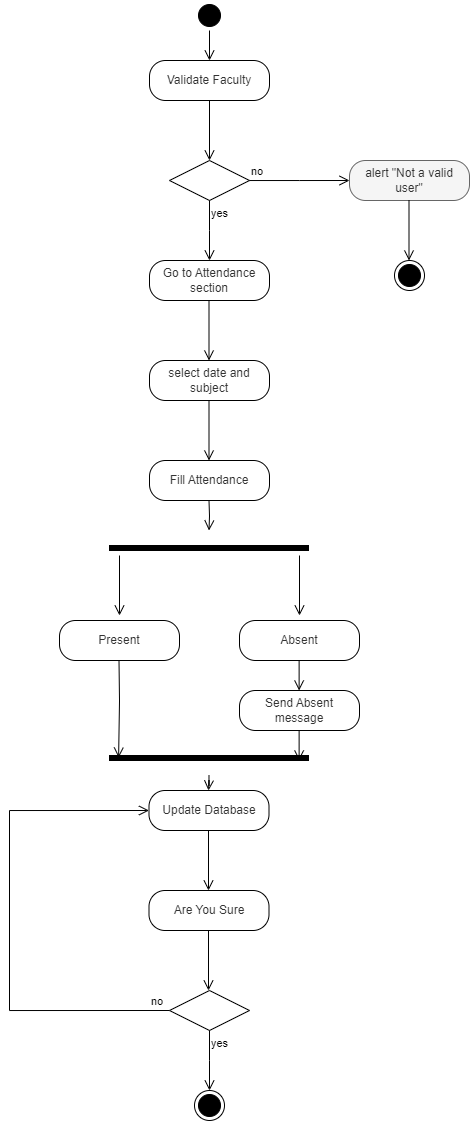


Figure 0‑2 Activity diagram for Atendance

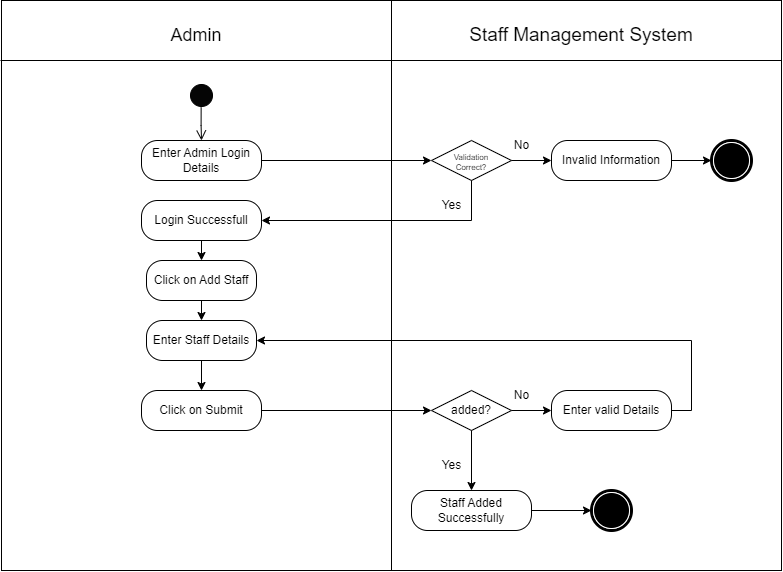


Figure 0‑2 Swimlane diagram for Adding staff

## Sequence diagram



Figure 0‑1 Sequence diagram for Book Issue

## State diagram



Figure 0‑1 State diagram of Book



Figure 0‑2 State diagram for Librarian

## Class diagram



Figure 0‑1 Class diagram for Library management system

## Data flow diagram

### Context diagram (level-0)



Figure 1‑1 Context diagram for Library management system

### DFD Level-1



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

### DFD Level-2



Figure 1‑3 DFD level-2 for Issue book

# External interface requirement (Screens)

## Screen-1: Student Profile

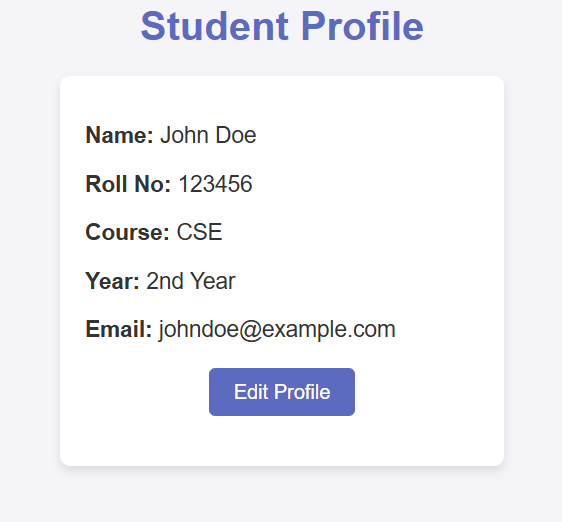


Figure 1‑1 Screen-1: Student Profile

**Purpose:** This screen allows students to view and update their personal information, including name, contact details, and enrolled courses. It helps maintain accurate student records within the system.

Table 1‑1 Screen element of Student Profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Name | Text | M | 1 | Student's full name, editable field. |
| 2 | Roll Number | Text | M | 1 | Unique student roll number, read-only field. |
| 3 | Course | Text | M | 1 | Select enrolled course. |
| 4 | Email | Text | M | 1 | Student email, must follow email format. |
| 5 | Save | Button | - | - | Saves student profile changes. |

## Screen-2: Fill Attendance

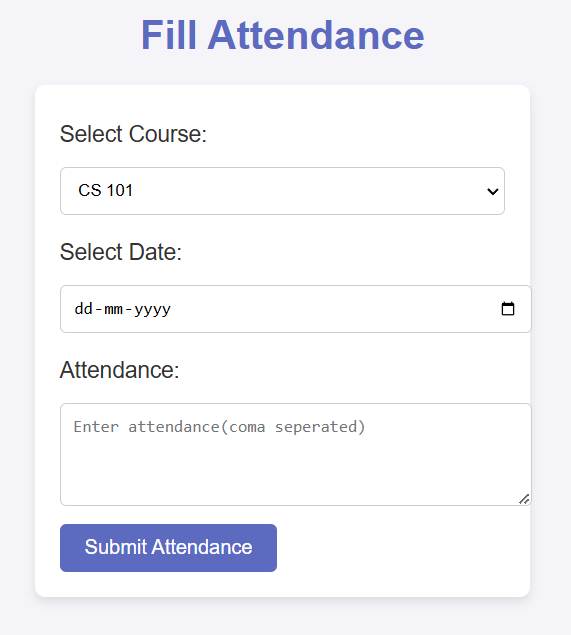


Figure 2‑1 Screen-2: fill attendance

**Purpose:** Faculty members can mark student attendance for their respective courses using this screen. It ensures proper record-keeping and enables tracking of student participation.

Table 2‑1 Screen element of fill Attendance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Course Name | Dropdown | M | 1 | Select course to mark attendance. |
| 2 | Date | Date Picker | M | 1 | Select attendance date. |
| 3 | Student List | TextBox | M | N | List of students. |
| 4 | Submit | Button | - | - | Saves attendance data. |

## Screen-3: View Assigned Course Material

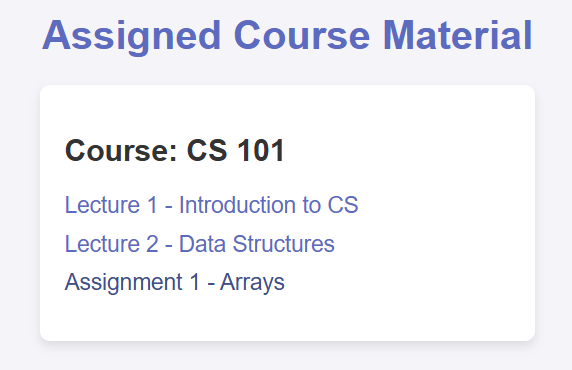


Figure 3‑1 Screen-3: View Assigned Course Material

**Purpose:** Teaching assistants can access course materials assigned to them, including lecture notes and assignments. This helps them assist faculty and students effectively.

Table 3‑1 Screen element of View Assigned Course Material

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Course Name | Text | M | 1 | Select assigned course. |
| 2 | Material List | Link | M | N | Displays course materials as downloadable links. |
| 3 | Download | Button | - | - | Allows TA to download materials. |

## Screen-4: CRUD User

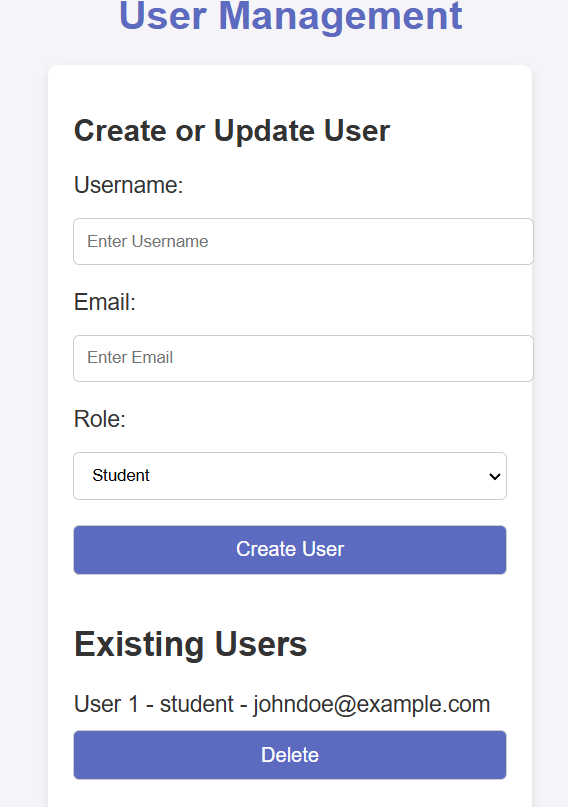


Figure 4‑1 Screen-4: CRUD User

**Purpose:** The admin can create, read, update, and delete user accounts, ensuring proper user management. This screen helps maintain system integrity by controlling user access.

Table 3‑1 Screen element of View Assigned Course Material

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | User Role | Dropdown | M | 1 | Select role (Student, Faculty, TA, etc.). |
| 2 | Name | Textbox | M | 1 | Enter user full name. |
| 3 | Email | Textbox | M | 1 | Enter user email. |
| 4 | Password | Password | M | 1 | Enter user password. |
| 5 | Add User | Button | - | - | Creates a new user. |
| 6 | Update User | Button | - | - | Updates existing user details. |
| 7 | Delete User | Button | - | - | Deletes user from system. |

## Screen-5: Visitor

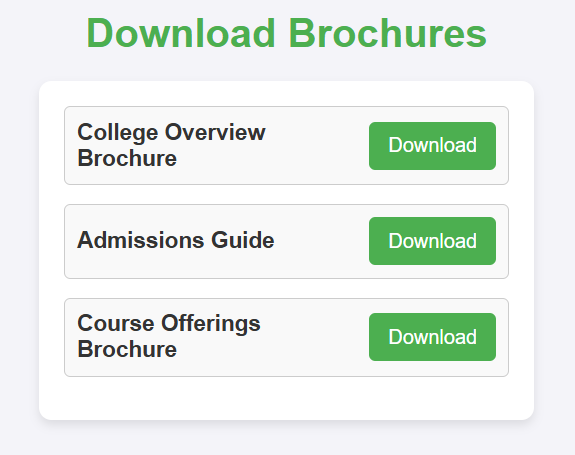


Figure 5‑1 Screen-5: Download Brochures

**Purpose:** Visitors can download informational brochures about the college, including course offerings and facilities. It serves as a resource for prospective students and parents.

Table 3‑1 Screen element of View Assigned Course Material

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Brochure List | List | M | N | Displays available brochures. |
| 2 | Download | Button | - | - | Allows visitor to download brochure. |

## List of Tables

* Book
* Borrower
* Student
* Staff

Table 1‑1 Table: Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| Emrollment | String | NN | PK |  |
| FirstName | varchar(100) | NN |  |  |
| LastName | varchar(100) | NN |  |  |
| SPI | int | NN |  |  |
| ContactNumber | varchar(10) | NN |  |  |
| Address | Varchar(100) | AN |  |  |
| Gender | Binary | NN |  |  |
| DOB | DateTime | NN |  |  |
| Department | Varchar(100) | NN |  |  |
| DepartmentID | int | NN | FK | Reference to Department Table |

Table 1‑2 Table: Course

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| CourseID | int | NN | PK (Auto Increment) |  |
| CourseName | varchar(100) | NN |  |  |
| CourseType | varchar(50) | AN |  | offline |
| TimeToComplete | Varchar(50) | AN |  |  |
| ProfName | Varchar(50) | AN |  |  |
| FacultyID | int | NN | FK | Reference to Faculty Table |

Table 1‑3 Table: Faculty

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| FacultyID | int | NN | PK (Auto Increment) |  |
| FirstName | varchar(100) | NN |  |  |
| LastName | Varchar(100) | AN |  |  |
| Designation | varchar(100) | AN |  |  |
| Salary | int | NN |  |  |
| Subject | Varcha(50) | NN |  |  |

Table 1‑4 Table: Department

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| DepartmentID | int | NN | PK (Auto Increment) |  |
| DepartmentName | varchar(100) | NN |  |  |
| Contact | varchar(100) | AN |  |  |
| Email | Varchar(100) | AN |  |  |
| Department | varchar(100) | AN |  |  |
| HOD | Varchar(50) | NN |  |  |

Table 1-5 Table: Exam

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Key & Constrains | Default Value & Description |
| ExamID | int | NN | PK |  |
| CourseID | int | NN | FK | Reference to Course Table |
| ExamDate | DateTime | NN |  |  |
| Duration | Time | AN |  |  |
| TotalMarks | int | NN |  |  |

# Stories and Scenario

## Story-1: Download Study Material

|  |  |  |
| --- | --- | --- |
| *Story # S1* | : | As a Student  I want to download PPTs form LMS |
| Priority | **:** | High |
| Estimate | **:** | XL |
| Reason | **:** | The study material is important for students |

### Scenario# S1.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.1* | : | Student Can Download PPT |
| Prerequisite | **:** | The student is logged into the college management system. |
|  |  |  |
| Acceptance Criteria | **:** | **Given**: The student is on the course materials page.  **When**: The student selects a PPT file and clicks the "Download" button.  **Then**: The system successfully downloads the PPT file to the student's device. |

### Scenario# S1.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.2* | : | Student Can Not Download PPT |
| Prerequisite | **:** | The student is not logged in or does not have permission to download the PPT. |
| Acceptance Criteria | **:** | **Given**: The student is on the course materials page.  **When**: The student attempts to download a PPT file without the necessary permissions.  **Then**: The system displays an error message stating that the student does not have the required permissions, and the download is not allowed |

## Story-2: Fill Attendance

|  |  |  |
| --- | --- | --- |
| *Story # S2* | : | As a Faculty I want to fill attendance of students |
| Priority | **:** | High |
| Estimate | **:** | XL |
| Reason | **:** | To fill students attendance so that they can check their attendance |

### Scenario# S2.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S2.1* | : | Faculty Cannot Fill Attendance |
| Prerequisite | **:** | The faculty member is not logged in or does not have permission to mark attendance. |
| Acceptance Criteria | **:** | **Given**: The faculty member is on the attendance management page.  **When**: The faculty attempts to mark attendance without the necessary permissions or while logged out.  **Then**: The system displays an error message stating that the faculty does not have the required permissions, and attendance is not recorded. |

### Scenario# S2.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S2.2* | : | Faculty Can Fill Attendance |
| Prerequisite | **:** | The faculty member is logged into the college management system. |
| Acceptance Criteria | **:** | **Given**: The faculty member is on the attendance management page.  **When**: The faculty selects a class and marks attendance for students, then clicks the "Submit" button.  **Then**: The system successfully saves the attendance records for that class. |

## Story-3: Fill Attendance

|  |  |  |
| --- | --- | --- |
| *Story # S3* | : | As a Student I want to fill test |
| Priority | **:** | High |
| Estimate | **:** | XL |
| Reason | **:** | To check ability of student in studies |

### Scenario# S3.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S3.1* | : | Student Can Give Test Online |
| Prerequisite | **:** | The student is logged into the online test portal. |
| Acceptance Criteria | **:** | **Given**: The student is on the test dashboard.  **When**: The student selects an available test and clicks the "Start Test" button.  **Then**: The system allows the student to attempt the test within the given time frame and submit responses successfully. |

### Scenario# S3.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S3.2* | : | Student Cannot Give Test Online |
| Prerequisite | **:** | The student is not logged in or the test is unavailable. |
| Acceptance Criteria | **:** | **Given**: The student is on the test dashboard.  **When**: The student attempts to start a test without logging in or after the test time has expired.  **Then**: The system displays an error message stating that the test cannot be accessed and prevents the student from proceeding. |

# 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