

Darshan University

A Project Report on

**“College Management System”**

Under the subject

**Software Engineering (2101CS503)**

B. Tech, Semester – IV

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name: Keval Dhandhukiya | Enrollment No.: 23010101064 |
| Academic Year  (2024-2025) | |
| Internal Guide  Prof. Rajkumar 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** **(2101CS503)** 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 (Rajkumar Gondaliya**)** and that no part of any of report has been directly copied from any students’ reports, without providing due reference.

(Keval Dhandhukiya)

Student’s Signature

Date: 03-03-2025

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

**CERTIFICATE**

This is to certify that the SRS on **“College Management System” has** been satisfactorily prepared by **Keval Dhandhukiya** (23010101064) under my guidance in the fulfillment of the course **Software Engineering (2101CS503)** work during the academic year 2024-2025.

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

**Acknowledgement**

I wish to express my sincere gratitude to my project guide Prof. **Rajkumar Gondaliya** 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

**Keval Dhandhukiya**

**ABSTRACT**

The College Management System (UMS) is a digital solution designed to automate and streamline the daily operations of a college. This system transforms traditional college management into an efficient, internet-based application with user-specific logins for students, faculty and administrators.

Key features include student course enrollment, attendance tracking, grade viewing, resource access, and schedule management. Faculty can manage course assignments, monitor student performance, and communicate with stakeholders. Administrators can oversee student and faculty records, financial transactions, hostel accommodations, inventory, and generate various reports.

The UMS enhances operational efficiency and reduces human effort by offering capabilities beyond conventional systems. It ensures data integrity and user privacy, providing a seamless and user-friendly experience for all college stakeholders. The main purpose of this system is to reduce human efforts and improve the management of college operations.

**Table of Contents**

[List of Figures I](#_Toc177483493)

[List of Tables II](#_Toc177483494)

[1 Introduction 1](#_Toc177483495)

[1.1 Product perspective 1](#_Toc177483496)

[1.2 Product features 1](#_Toc177483497)

[**1.2.1** There are **three** different **users** who will be using this product: 1](#_Toc177483498)

[1.3 Functional Requirement 2](#_Toc177483499)

[**1.3.1** The features that are required for each user in the College Management System are: 2](#_Toc177483500)

[**1.3.2** **Student:** 2](#_Toc177483501)

[**1.3.3** **Admin**: 2](#_Toc177483502)

[**1.3.4** **Faculty**: 3](#_Toc177483503)

[1.4 Non-Functional Requirement 5](#_Toc177483504)

[**1.4.1** **Performance:** 5](#_Toc177483505)

[**1.4.2** **Security:** 5](#_Toc177483506)

[**1.4.3** **Reliability:** 5](#_Toc177483507)

[**1.4.4** **Usability:**. 5](#_Toc177483508)

[**1.4.5** **Scalability**: 5](#_Toc177483509)

[2 Design and Implementation Constraints 6](#_Toc177483510)

[2.1 Use case diagram 6](#_Toc177483511)

[2.2 Activity diagram and Swimlane diagram 7](#_Toc177483512)

[2.3 Sequence diagram 9](#_Toc177483513)

[2.4 State diagram 10](#_Toc177483514)

[Class diagram 11](#_Toc177483515)

[2.5 Data flow diagram 12](#_Toc177483516)

[**2.5.1** Context diagram (level-0) 12](#_Toc177483517)

[**2.5.2** DFD Level-1 13](#_Toc177483518)

[3 External interface requirement (Screens) 14](#_Toc177483519)

[3.1 Screen-1: Attendence Screen 14](#_Toc177483520)

[3.3 Screen-3: Fees Payment Screen 18](#_Toc177483521)

[3.4 screen-4: Exam Screen 20](#_Toc177483522)

[4 Database design 22](#_Toc177483523)

[4.1 List of Tables 22](#_Toc177483524)

[5 Stories and Scenario 25](#_Toc177483525)

[5.1 Story-1: Add New Course 25](#_Toc177483526)

[**5.1.1** Scenario# S1.1 25](#_Toc177483527)

[**5.1.2** Scenario# S1.2 25](#_Toc177483528)

[**5.1.3** Scenario# S1.3 25](#_Toc177483529)

[5.2 Story-2: Enroll Student in a Course 26](#_Toc177483530)

[5.3 Story-3: Manage due date for borrowed book 26](#_Toc177483531)

[5.4 Story-4: Generate Student Transcript 27](#_Toc177483532)

[5.5 Story-5: Schedule University Events 28](#_Toc177483533)

[6 Test cases 30](#_Toc177483534)

[7 References 35](#_Toc177483535)

# List of Figures

[Figure 2.1‑1 Use case diagram for College Management System 6](#_Toc146706848)

[Figure 2.2‑1 Activity diagram for fees payment system](#_Toc146706849) 7

[Figure 2.2‑2 Swimlane diagram for fess payment system 8](#_Toc146706850)

[Figure 2.3‑1 Sequence diagram for fees payment system 9](#_Toc146706851)

[Figure 2.4‑1 State diagram of Book 10](#_Toc146706852)

[Figure 2.5‑1 Class diagram for College Management System 11](#_Toc146706854)

[Figure 2.6‑1 Context diagram for College Management System 12](#_Toc146706855)

[Figure 2.6‑2 DFD level-1 for student registration system 13](#_Toc146706856)

[Figure 3.1‑1 Screen-1: Marking attendence 14](#_Toc146706858)

[Figure 3.2‑1 Screen-2: Assignment 16](#_Toc146706859)

[Figure 3.3‑1 Screen-3: Fees payment 18](#_Toc146706860)

# List of Tables

[Table 3.1‑1 Screen element of Registration form 14](#_Toc146706861)

[Table 3.2‑1 Screen element of Login form 16](#_Toc146706862)

[Table 3.3‑1 Screen element of Add borrower book 18](#_Toc146706863)

[Table 4.1‑1 Table: Book 22](#_Toc146706864)

[Table 4.1‑2 Table: Borrower 22](#_Toc146706865)

[Table 4.1‑3 Table: Staff 23](#_Toc146706866)

[Table 4.1‑4 Table: Student 23](#_Toc146706867)

# Introduction

## Product perspective

This project aims to revolutionize the traditional, manual college management processes by transitioning to a comprehensive, internet-based application. The College Management System (UMS) will facilitate seamless interaction between students, faculty, and administration, enhancing overall efficiency and user experience.

## Product features

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

* **Student:** Students will be the primary users of the system. They can manage their course enrollments, track their attendance, view their grades, access course materials, and stay updated with university announcements.
* **Admin:** The administrative staff will oversee the overall management of the college operations. They will handle student and faculty records, financial management, hostel accommodations, inventory control, and generate comprehensive reports for better decision-making.
* **Faculty:** Faculty members will utilize the system to manage their course assignments, track student performance, handle attendance records, distribute resources, and communicate with students and administration.

## Functional Requirement

### The features that are required for each user in the College Management System are:

### **Student:**

* **Course Enrollment:** Students can enrol in courses and drop them within the registration period.
* **Attendance Viewing:** Students can view their attendance records for each course.
* **Class Schedule:** Students can view and download their class schedules.
* **Assignment Submission:** Students can submit assignments electronically and receive feedback.
* **Grade Viewing:** Students can view their grades for all completed courses.
* **Resource Access:** Students can access course materials, lecture notes, and additional resources.
* **Library Access:** Students can search the library catalog, borrow books, and check due dates.
* **Exam Schedule:** Students can view their upcoming exam schedules and locations.
* **Fee Payment:** Students can view and pay their tuition and other fees online.
* **Profile Management:** Students can update their personal information and academic details.
* **Communication Portal:** Students can send messages to faculty and receive notifications.
* **Club Registration:** Students can join college clubs and participate in extracurricular activities.
* **Internship/Job Portal:** Students can search and apply for internships and job opportunities.
* **Feedback Submission:** Students can provide feedback on courses and faculty.
* **Academic Calendar:** Students can view the academic calendar, including important dates and deadlines.

### **Admin:**

* **User Management:** Admin can create, update, and delete user accounts for students, parents, and faculty.
* **Fee Management:** Admin can manage tuition fees, payments, and generate financial reports.
* **Course Management:** Admin can create and update course details, schedules, and faculty assignments.
* **Attendance Reports:** Admin can generate attendance reports for all students.
* **Grade Reports:** Admin can generate and distribute grade reports for all courses.
* **Communication Management:** Admin can send notifications and messages to students, parents, and faculty.
* **Resource Allocation:** Admin can allocate resources such as classrooms, laboratories, and equipment.
* **Library Management:** Admin can manage the library catalog, borrowing records, and dues.
* **Profile Management:** Admin can update user profiles and access permissions.
* **Event Coordination:** Admin can coordinate and manage college events and activities.
* **Academic Calendar:** Admin can update and maintain the academic calendar.
* **Exam Management:** Admin can schedule exams, assign invigilators, and manage exam logistics.
* **Report Generation:** Admin can generate various academic and administrative reports.
* **Support Services:** Admin can manage support services like counselling, health services, and student welfare.
* **System Maintenance:** Admin can perform regular maintenance and updates to the College Management System.

### **Faculty**:

* **Grade Submission:** Faculty can enter and update student grades for their courses.
* **Exam Scheduling:** Faculty can schedule exams and manage exam logistics.
* **Course Material Upload:** Faculty can upload lecture notes, assignments, and other course materials.
* **Attendance Management:** Faculty can mark and update student attendance records.
* **Assignment Management:** Faculty can create, distribute, and grade assignments.
* **Student Communication:** Faculty can communicate with students individually or in groups.
* **Resource Access:** Faculty can access academic resources and research materials.
* **Profile Management:** Faculty can update their professional and personal information.
* **Feedback Collection:** Faculty can collect and review feedback from students on their courses.
* **Timetable Management:** Faculty can view and manage their teaching schedules.
* **Research Management:** Faculty can manage their research projects and publications.
* **Library Access:** Faculty can access library resources and manage their borrowing.
* **Committee Participation:** Faculty can participate in and manage academic and administrative committees.
* **Counselling and Mentoring:** Faculty can provide academic counselling and mentoring to students.
* **Event Management:** Faculty can organize and manage academic and extracurricular events.

## Non-Functional Requirement

### **Performance:** The system should handle multiple user requests efficiently, ensuring quick response times and minimal latency, even during peak usage periods.

### **Security:** Data transmission should be encrypted, user authentication must be robust, and access controls should be in place to prevent unauthorized access to sensitive information.

### **Reliability:** The system should be highly available with minimal downtime, regular data backups should prevent data loss, and error handling mechanisms should prevent system crashes.

### **Usability:** The interface should be intuitive and easy to use, supporting various devices and screen sizes, with accessibility features for users with disabilities.

### **Scalability**: The system architecture should be able to handle increasing user and data loads, with strategies in place for horizontal and vertical scaling and load balancing to ensure optimal performance.

# Design and Implementation Constraints

## Use case diagram

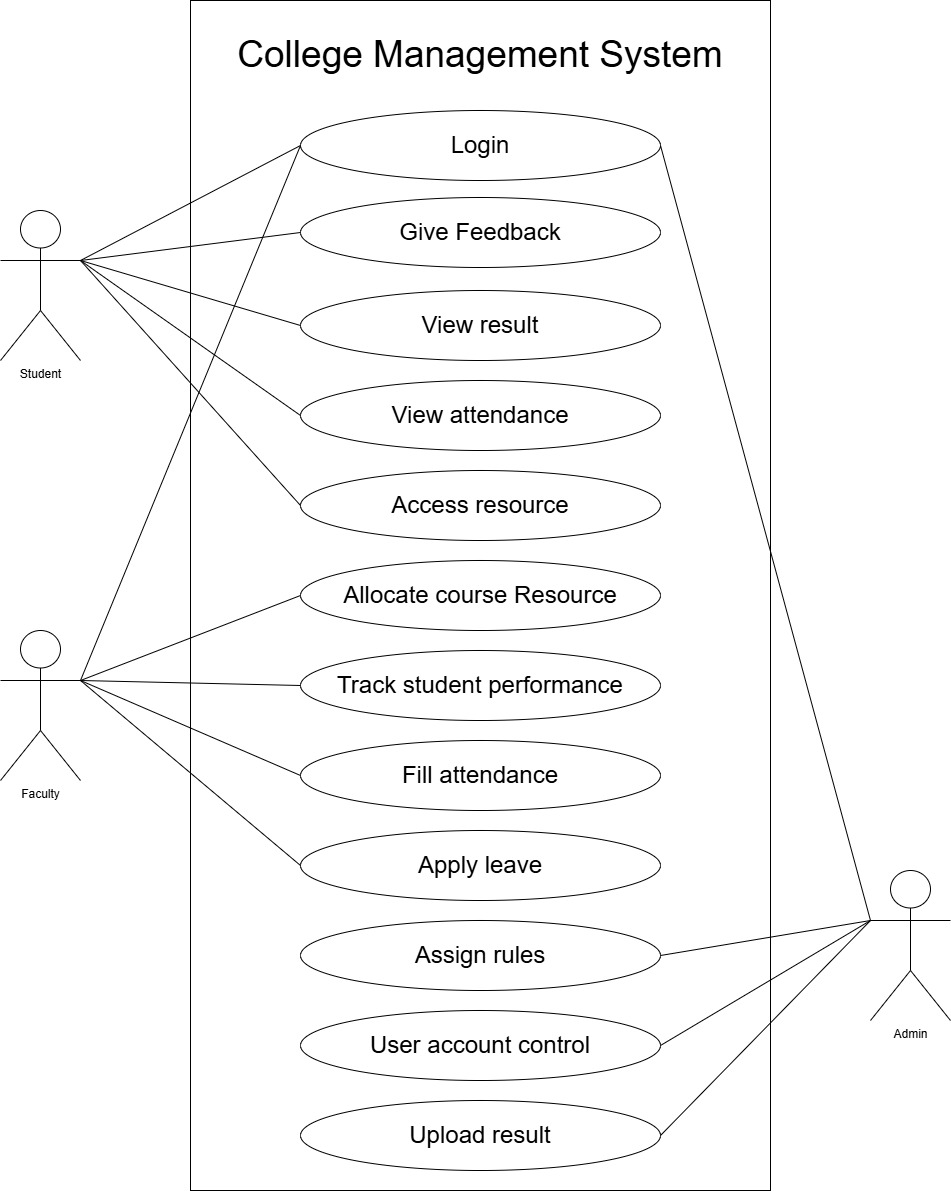


Figure ‑ Use case diagram for College Management System

## Activity diagram and Swimlane diagram

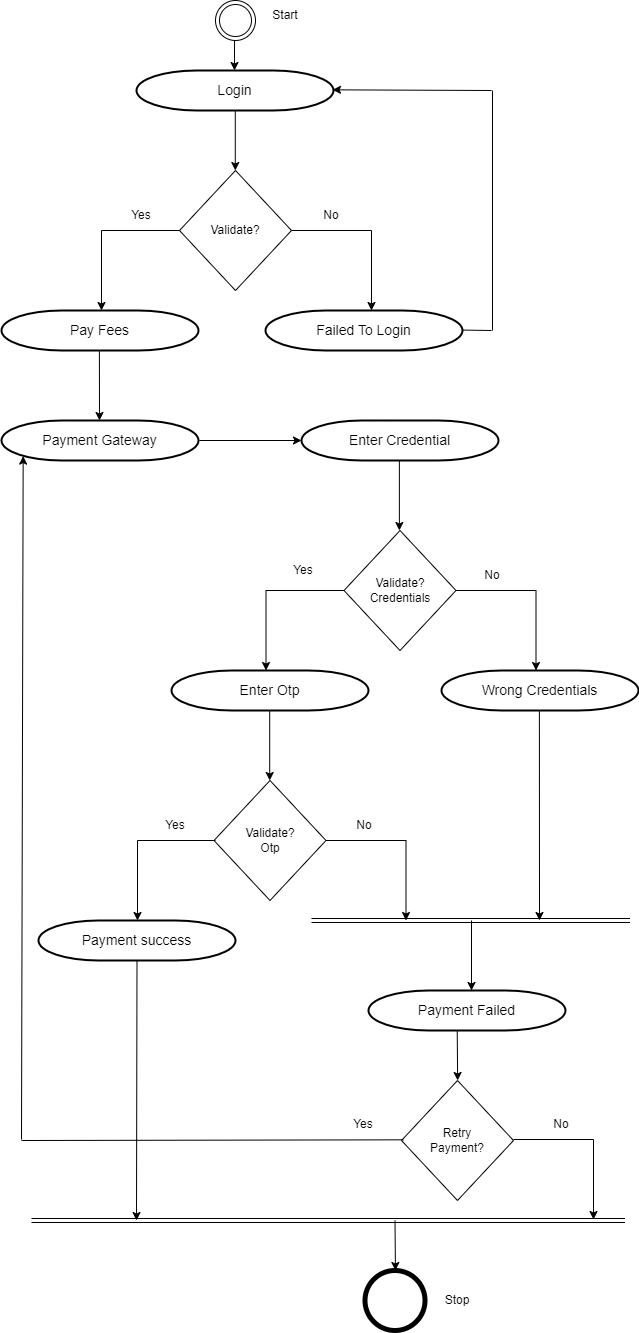


Figure 2.2‑ Activity Diagram for Fees Payment System

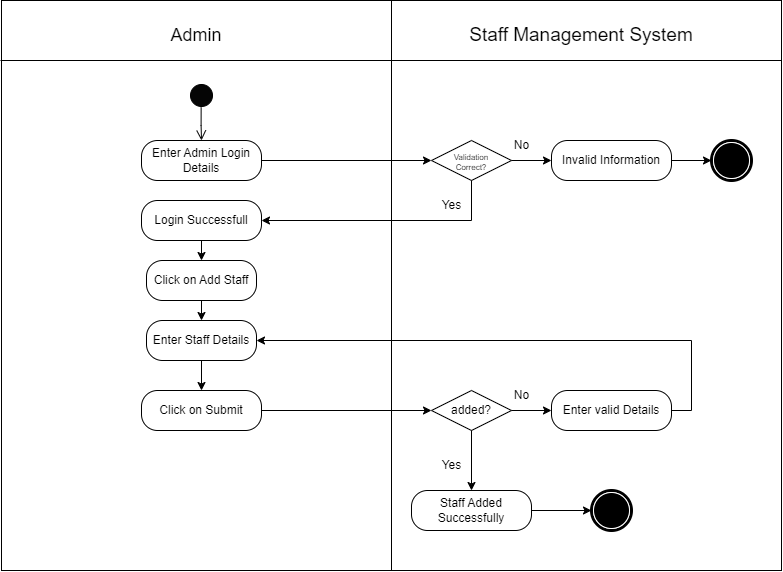


Figure ‑ Swimlane diagram for add staff

## Sequence diagram

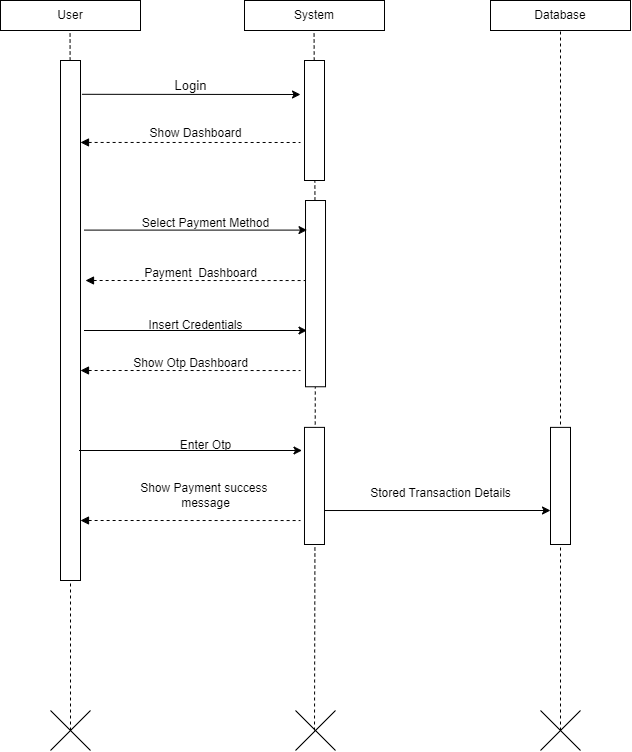


Figure ‑ Sequence diagram for Fees Payment System

## State diagram

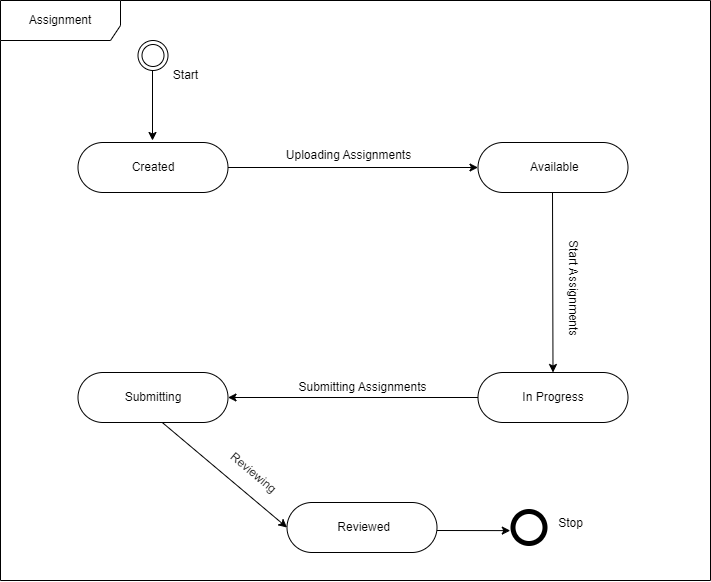


Figure ‑ State diagram of assignment

## Class diagram

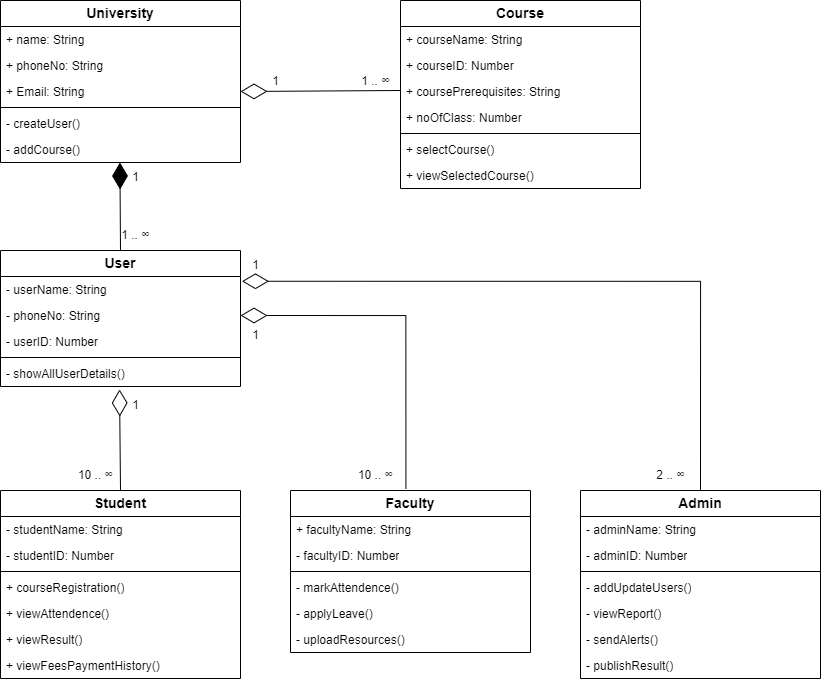


Figure ‑ Class diagram for College Management System

## Data flow diagram

### Context diagram (level-0)

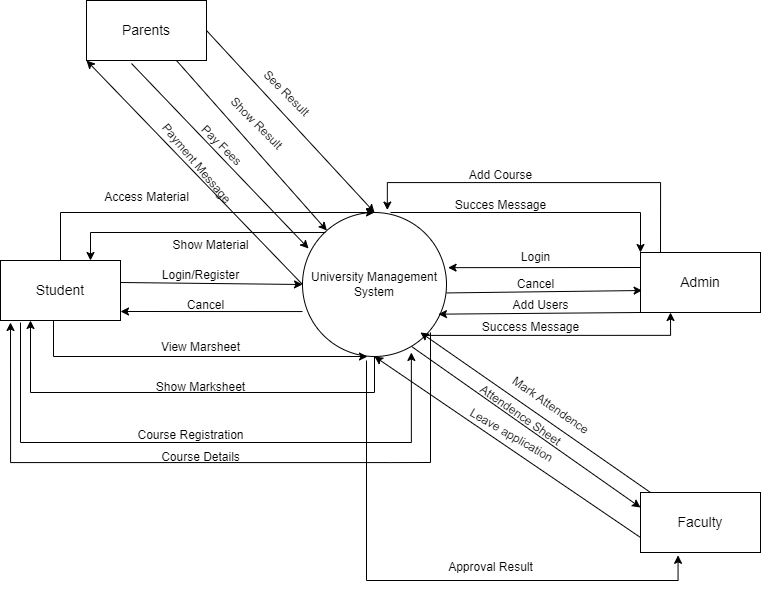


Figure ‑ Context diagram for College Management System

### DFD Level-1

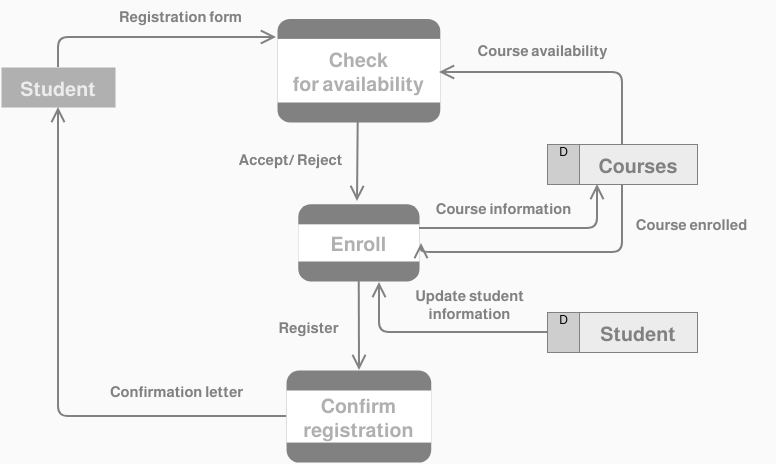


Figure ‑ DFD level-1 for Student Registration system

# External interface requirement (Screens)

## Screen-1: Attendance Screen

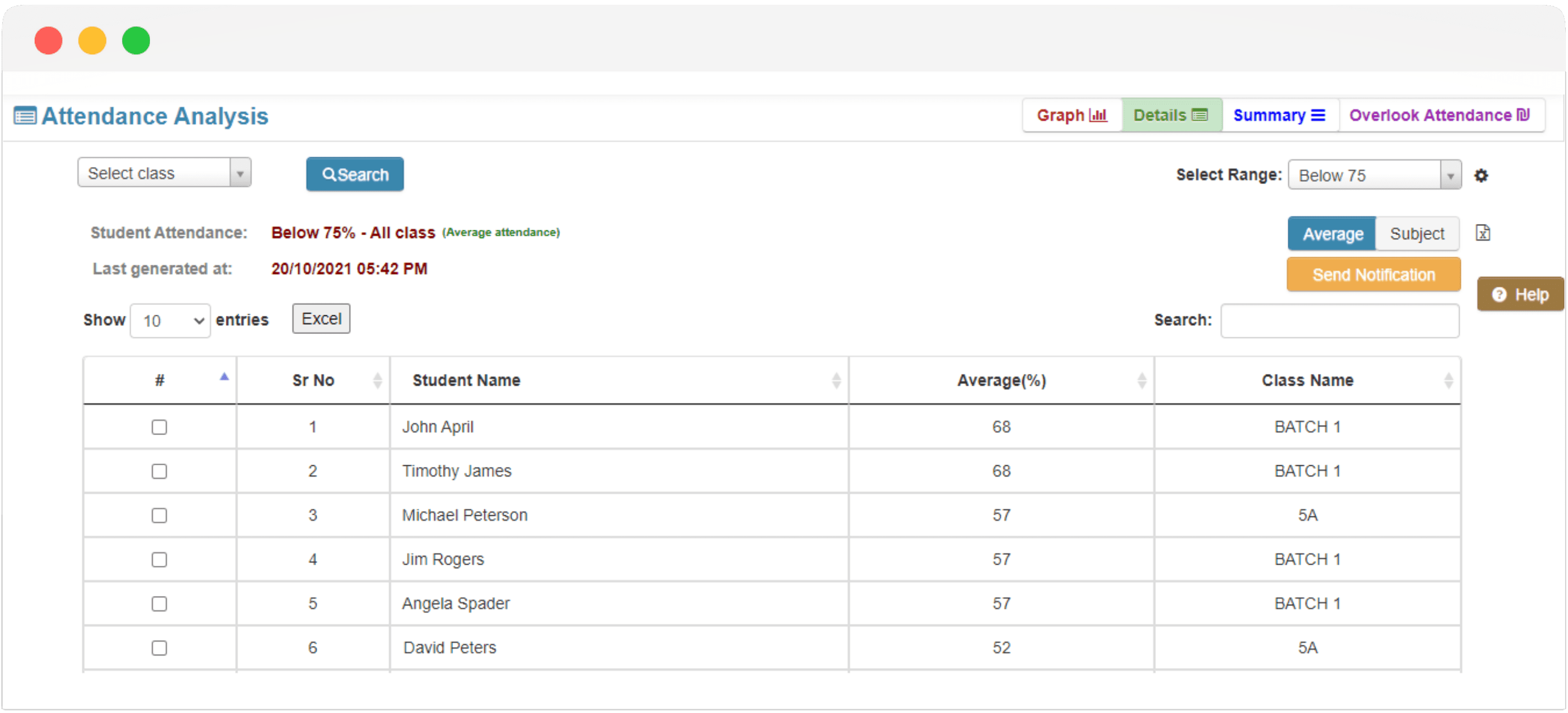


Figure ‑ Screen-1: Marking attendance

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

Table ‑ Screen element of Attendance Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Select Class | Dropdown | M | 1 | Dropdown to select the class. |
| 2 | Search | Button | M | 1 | Button to initiate the search based on selected criteria. |
| 3 | Select Range | Dropdown | M | 1 | Dropdown to select the attendance range. |
| 4 | Average | Button | M | 1 | Button to show average attendance. |
| 5 | Subject | Button | M | 1 | Button to show subject-wise attendance. |
| 6 | Send Notification | Button | M | 1 | Button to send notifications to selected students. |
| 7 | Help | Button | M | 1 | Button to get help or instructions. |
| 8 | Entries Show | Dropdown | M | 1 | Dropdown to select the number of entries to show per page. |
| 9 | Excel | Button | M | 1 | Button to export the data to an Excel file. |
| 10 | Search (table filter) | Text | M | 1 | Text field to filter table entries. |
| 11 | Checkbox (select all) | Checkbox | M | 1 | Checkbox to select all entries in the table. |
| 12 | Checkbox (individual rows) | Checkbox | M | 1 | Checkbox to select individual entries in the table. |
| 13 | Sr No | Column | O | 1 | Column showing the serial number of the students. |
| 14 | Student Name | Column | O | 1 | Column showing the names of the students. |
| 15 | Average (%) | Column | O | 1 | Column showing the average attendance percentage of the students. |
| 16 | Class Name | Column | O | 1 | Column showing the class names of the students. |
| 17 | Graph | Link | O | 1 | Link to navigate to the graph view of the attendance data. |
| 18 | Details | Link | O | 1 | Link to navigate to the detailed view of the attendance data. |
| 19 | Summary | Link | O | 1 | Link to navigate to the summary view of the attendance data. |
| 20 | Overlook Attendance | Link | O | 1 | Link to navigate to the overlook attendance view. |
| 21 | Last generated at | Text | O | 1 | Text showing the timestamp when the attendance data was last generated. |
| 22 | Student Attendance Status | Text | O | 1 | Text showing the current status of student attendance (e.g. 75%) |

**3.2 Screen-2: Assignment Screen**

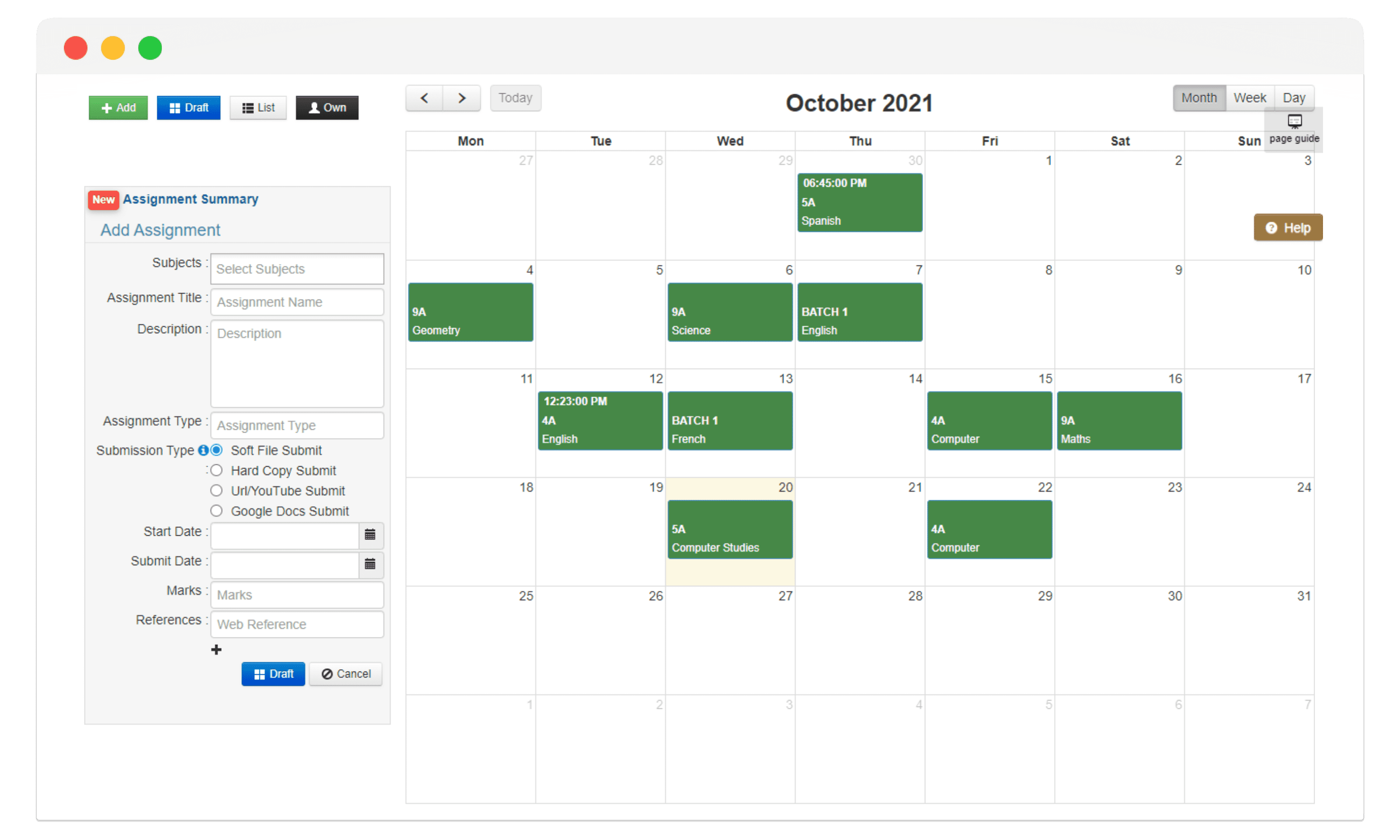


Figure ‑ Screen-2: Login Form

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

Table ‑ Screen element of Login form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Add Assignment Button | Button | M | 1 | Button to add a new assignment. |
| 2 | Draft Button | Button | M | 1 | Button to save the assignment as a draft. |
| 3 | List Button | Button | M | 1 | Button to view the list of assignments. |
| 4 | Own Button | Button | M | 1 | Button to view own assignments. |
| 5 | Subjects | Dropdown | M | 1 | Dropdown to select the subjects. |
| 6 | Assignment Title | Text | M | 1 | Text field to enter the assignment title. |
| 7 | Description | Text area | M | 1 | Text area to enter the assignment description. |
| 8 | Assignment Type | Dropdown | M | 1 | Dropdown to select the assignment type. |
| 9 | Submission Type - Soft File Submit | Radio Button | M | 1 | Radio button to select soft file submission type. |
| 10 | Submission Type - Hard Copy Submit | Radio Button | M | 1 | Radio button to select hard copy submission type. |
| 11 | Submission Type - URL/YouTube Submit | Radio Button | M | 1 | Radio button to select URL/YouTube submission type. |
| 12 | Submission Type - Google Docs Submit | Radio Button | M | 1 | Radio button to select Google Docs submission type. |
| 13 | Start Date | Date Picker | M | 1 | Date picker to select the start date of the assignment. |
| 14 | Submit Date | Date Picker | M | 1 | Date picker to select the submission date of the assignment. |
| 15 | Marks | Text | M | 1 | Text field to enter the marks for the assignment. |
| 16 | References | Text | M | 1 | Text field to enter the web reference for the assignment. |
| 17 | Draft Button (Form) | Button | M | 1 | Button to save the assignment form as a draft. |
| 18 | Cancel Button (Form) | Button | M | 1 | Button to cancel the assignment form. |
| 19 | Calendar | Calendar | M | 1 | Calendar to view the assignments for the month. |
| 20 | Month View Button | Button | M | 1 | Button to switch to the month view of the calendar. |
| 21 | Week View Button | Button | M | 1 | Button to switch to the week view of the calendar. |
| 22 | Day View Button | Button | M | 1 | Button to switch to the day view of the calendar. |
| 23 | Page Guide Button | Button | M | 1 | Button to show the page guide. |
| 24 | Help Button | Button | M | 1 | Button to get help or instructions. |

## Screen-3: Fees Payment Screen

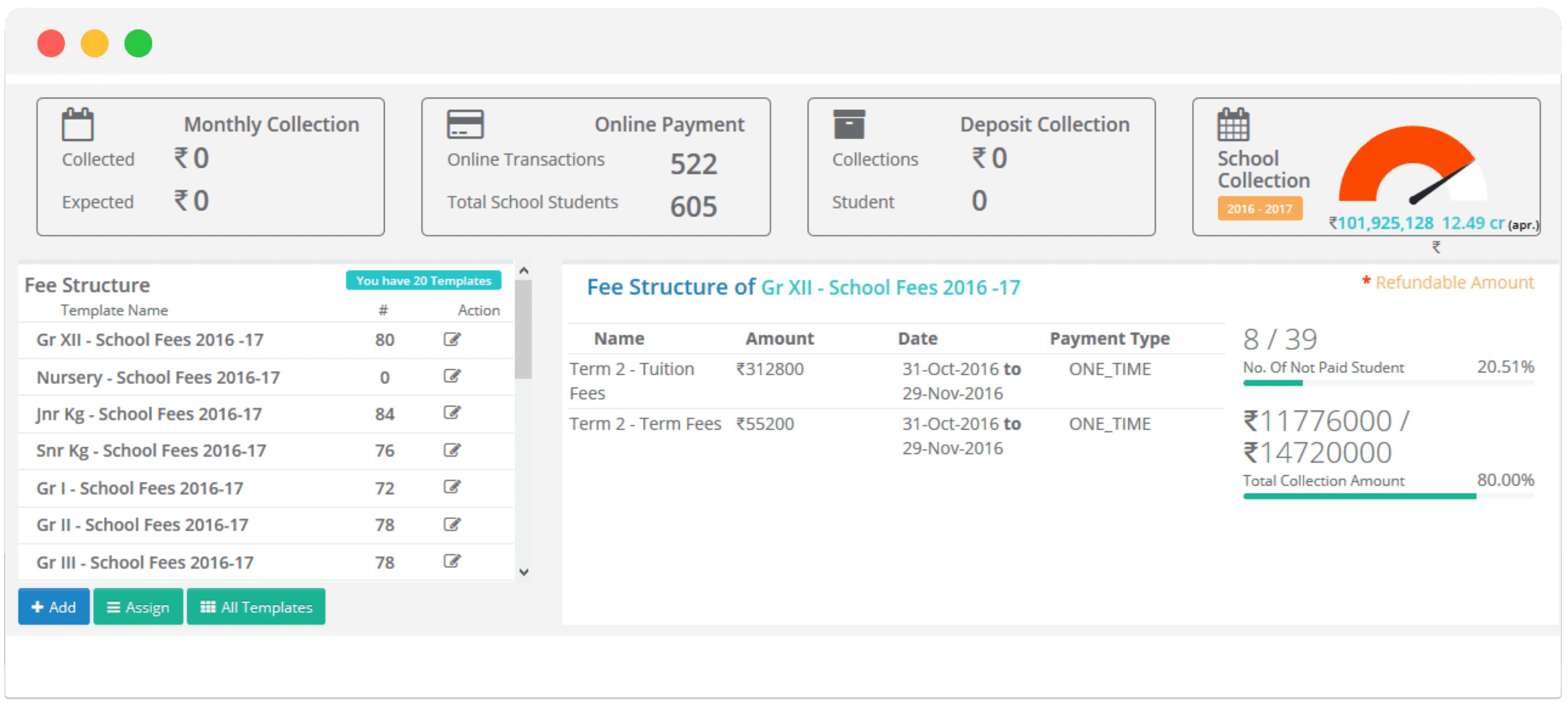


Figure ‑ Screen-3: Fees Payment Screen

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

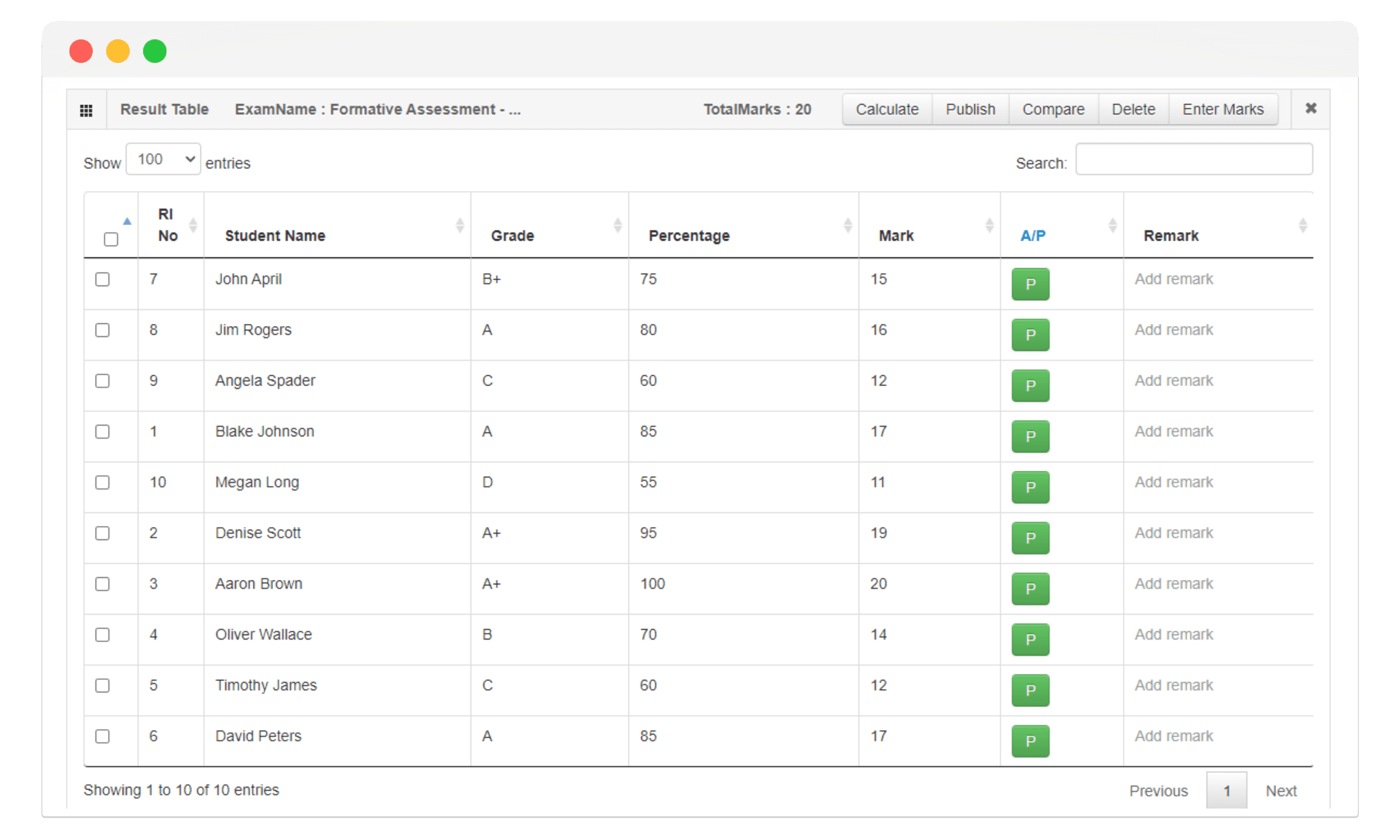
Table ‑ Screen element of Add borrower book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Monthly Collection - Collected | Label | M | 1 | Displays the collected amount for the month. |
| 2 | Monthly Collection - Expected | Label | M | 1 | Displays the expected amount for the month. |
| 3 | Online Transactions | Label | M | 1 | Displays the number of online transactions. |
| 4 | Total School Students | Label | M | 1 | Displays the total number of school students. |
| 5 | Deposit Collection - Collections | Label | M | 1 | Displays the amount collected as deposit. |
| 6 | Deposit Collection - Student | Label | M | 1 | Displays the number of students for deposit collection. |
| 7 | School Collection - Gauge | Label | M | 1 | Displays the school collection amount as a gauge. |
| 8 | Fee Structure - Template Name | Label | M | 1 | Displays the template name for fee structure. |
| 9 | Fee Structure - Count (#) | Label | M | 1 | Displays the count of the templates. |
| 10 | Fee Structure - Action | Button | M | 1 | Button to edit the template. |
| 11 | Add Button | Button | M | 1 | Button to add a new template. |
| 12 | Assign Button | Button | M | 1 | Button to assign a template. |
| 13 | All Templates Button | Button | M | 1 | Button to view all templates. |
| 14 | Selected Fee Structure - Name | Label | M | 1 | Displays the name of the selected fee structure. |
| 15 | Selected Fee Structure - Amount | Label | M | 1 | Displays the amount for the selected fee structure. |
| 16 | Selected Fee Structure - Date | Label | M | 1 | Displays the date for the selected fee structure. |
| 17 | Selected Fee Structure - Payment Type | Label | M | 1 | Displays the payment type for the selected fee structure. |
| 18 | Number of Not Paid Students | Label | M | 1 | Displays the number of students who have not paid. |
| 19 | Total Collection Amount | Label | M | 1 | Displays the total collection amount and percentage. |

## screen-4: Exam Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Formative Assessment - ENG | Label | M | 1 | Formative Assessment - English on 27th September 2021 |
| 2 | Unit Test - Algebra | Label | M | 1 | Unit Test - Algebra on 29th September 2021 |
| 3 | Unit Test - Geo | Label | M | 1 | Unit Test - Geography on 1st October 2021 |
| 4 | Project-5A-Craft | Label | M | 1 | Project for Class 5A - Craft on 4th October 2021 |
| 5 | Practical - Comp | Label | M | 1 | Practical Exam - Computer on 5th October 2021 |
| 6 | TERM END EXAM - MATHS | Label | M | 1 | Term End Exam - Mathematics on 6th October 2021 |
| 7 | Internal MCQ/Prac/Oral | Label | M | 1 | Internal MCQ/Practical/Oral for Classes 9th-10th on 7th October 2021 |
| 8 | Formative Assessment - FR | Label | M | 1 | Formative Assessment - French on 11th October 2021 |
| 9 | Formative Assessment - Maths | Label | M | 1 | Formative Assessment - Mathematics on 12th October 2021 |
| 10 | Unit Test - Maths | Label | M | 1 | Unit Test - Mathematics on 18th October 2021 |
| 11 | Formative Assessment - COMP | Label | M | 1 | Formative Assessment - Computer on 21st October 2021 |
| 12 | Formative Assessment - ENG | Label | M | 1 | Formative Assessment - English on 22nd October 2021 |
| 13 | Nursery | Checkbox | M | 1 | Nursery class selected |
| 14 | Junior KG | Checkbox | M | 1 | Junior KG class selected |

**3.5 Screen-5 Attendance Screen**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Exam Name: Formative Assessment | Label | M | 1 | Displays the name of the exam. |
| 2 | TotalMarks | Label | M | 1 | Displays the total marks for the exam. |
| 3 | Calculate | Button | M | 1 | Button to calculate marks. |
| 4 | Publish | Button | M | 1 | Button to publish the results. |
| 5 | Compare | Button | M | 1 | Button to compare results. |
| 6 | Delete | Button | M | 1 | Button to delete records. |
| 7 | Enter Marks | Button | M | 1 | Button to enter marks. |
| 8 | Search | TextBox | M | 1 | Input field to search students. |
| 9 | RI No | Label | M | 1 | Displays the roll number of the student. |
| 10 | Student Name | Label | M | 1 | Displays the name of the student. |
| 11 | Grade | Label | M | 1 | Displays the grade of the student. |
| 12 | Percentage | Label | M | 1 | Displays the percentage score of the student. |
| 13 | Mark | Label | M | 1 | Displays the mark of the student. |
| 14 | A/P | Label | M | 1 | Displays if the student is present or absent. |
| 15 | Remark | TextBox | M | 1 | Input field to add a remark for the student. |
| 16 | Entries Dropdown | Dropdown | M | 1 | Dropdown to select the number of entries to show. |
| 17 | Checkbox | Checkbox | M | 1 | Checkbox to select multiple entries. |

# Database design

## List of Tables

* Student
* Course
* Transport
* Faculty
* Exam

Table ‑ Student Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constraints** | **Default Value & Description** |
| **StudentID** | int | NN | PK (Auto Increment) |  |
| **FirstName** | varchar(50) | NN |  |  |
| **LastName** | varchar(50) | NN |  |  |
| **DateOfBirth** | date | NN |  |  |
| **Gender** | varchar(10) | AN | CHECK (Gender IN ('Male', 'Female', 'Other')) |  |
| **Address** | text | AN |  |  |
| **PhoneNumber** | varchar(15) | AN | UNIQUE |  |
| **Email** | varchar(100) | AN | UNIQUE |  |
| **EnrollmentDate** | date | NN |  |  |

Table ‑ Course Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constraints** | **Default Value & Description** |
| **CourseID** | int | NN | PK (Auto Increment) |  |
| **CourseName** | varchar(100) | NN | UNIQUE |  |
| **Description** | text | AN |  |  |
| **Credits** | int | NN | CHECK (Credits > 0) |  |
| **Department** | varchar(50) | NN |  |  |

Table ‑ Transport Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constraints** | **Default Value & Description** |
| **TransportID** | int | NN | PK (Auto Increment) |  |
| **VehicleNumber** | varchar(20) | NN | UNIQUE |  |
| **DriverName** | varchar(50) | NN |  |  |
| **DriverContact** | varchar(15) | NN |  |  |
| **Route** | text | NN |  |  |
| **Capacity** | int | NN | CHECK (Capacity > 0) |  |
| **Schedule** | text | NN |  |  |

Table ‑ Faculty Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constraints** | **Default Value & Description** |
| **FacultyID** | int | NN | PK (Auto Increment) |  |
| **FirstName** | varchar(50) | NN |  |  |
| **LastName** | varchar(50) | NN |  |  |
| **DateOfBirth** | date | NN |  |  |
| **Gender** | varchar(10) | AN | CHECK (Gender IN ('Male', 'Female', 'Other')) |  |
| **Address** | text | AN |  |  |
| **PhoneNumber** | varchar(15) | AN | UNIQUE |  |
| **Email** | varchar(100) | AN | UNIQUE |  |
| **HireDate** | date | NN |  |  |
| **Department** | varchar(50) | NN |  |  |

Table ‑5 Exam Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constraints** | **Default Value & Description** |
| **ExamID** | int | NN | PK (Auto Increment) |  |
| **CourseID** | int | NN | FK (references Course(CourseID)) |  |
| **ExamDate** | date | NN |  |  |
| **StartTime** | time | NN |  |  |
| **EndTime** | time | NN |  |  |
| **Venue** | varchar(100) | NN |  |  |

# Stories and Scenario

## Story-1: Add New Course

|  |  |  |
| --- | --- | --- |
| *Story # S1* | : | As an Administrator,  I want to add a new course to the university system  So that students can enrol in the new course and faculty can manage it. |
| Priority | **:** | High |
| Estimate | **:** | XL |
| Reason | **:** | Adding new courses is crucial for expanding the academic offerings of the university and meeting the educational needs of students. |

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

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.1* | : | Adding a New Course with Valid Information |
| Prerequisite | **:** | Administrator is logged in to the College Management System. |
| Acceptance Criteria | **:** | **Given:**  The Administrator is on the course management page. Valid course information, including course title, code, description, and credits, is entered.  **When:**  The administrator selects the "Add New Course" option and enters valid course details. The administrator clicks the "Save" button to add the course to the system.  **Then:** The system successfully adds the course, and the administrator receives a confirmation message with the course details. |

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

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.2* | : | Adding a New Course with Invalid Information |
| Prerequisite | **:** | Administrator is logged into the College Management System. |
| Acceptance Criteria | **:** | **Given:** The administrator is on the course management page. Incomplete or incorrect course details are entered.  **When:** The administrator selects the "Add New Course" option and clicks the "Save" button.  **Then:** The system displays error messages for the incorrect or missing information, and the course is not added to the system. |

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

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.3* | : | Attempting to Add a Duplicate Course |
| Prerequisite | **:** | Administrator is logged into the College Management System and is on the course |
| Acceptance Criteria | **:** | **Given**: The course information, including title and code, already exists in the system.  **When**: The administrator tries to add a course with the same title and code.  **Then**: The system generates a unique identifier for the new course if needed, and provides a message indicating that a course with the same title and code already exists. |

## Story-2: Enrol Student in a Course

|  |  |  |
| --- | --- | --- |
| *Story # S2* | : | As a student,  I want to enroll in a course,  So that I can attend classes and earn credits. |
| Priority | **:** | High |
| Estimate | **:** | M |
| Reason | **:** | Course enrollment is a fundamental aspect of academic life, allowing students to participate in their chosen courses and progress through their studies. |

### **Scenario # S2.1:**

|  |  |  |
| --- | --- | --- |
| *Scenario# S2.1* | : | Enrolling in a Course with Available Seats |
| Prerequisite | **:** | Student is logged in to the College Management System. |
| Acceptance Criteria | **:** | **Given**: The student is on the course enrolment page, and the course has available seats.  **When**: The student selects a course and clicks the "Enrol" button.  **Then**: The system successfully enrols the student in the course and provides a confirmation message with the enrolment details. |

### **Scenario # S2.2:**

|  |  |  |
| --- | --- | --- |
| *Scenario# S2.2* | : | Enrolling in a Course with No Available Seats |
| Prerequisite | **:** | Student is logged in to the College Management System. |
| Acceptance Criteria | **:** | **Given**: The student is on the course enrollment page, and the course is fully booked.  **When**: The student selects the course and clicks the "Enroll" button.  **Then**: The system displays a message indicating that the course is full and enrollment cannot be completed. |

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

|  |  |  |
| --- | --- | --- |
| *Story # S3* | : | As Faculty Member,  I want to manage and update student grades  So that So that I can accurately reflect students' academic performance. |
| Priority | **:** | High |
| Estimate | **:** | M |
| Reason | **:** | Accurate management of student grades is essential for tracking academic progress and ensuring that students receive fair evaluations. |

### **Scenario # S3.1:**

|  |  |  |
| --- | --- | --- |
| *Scenario# S2.1* | : | Updating Grades for a Student |
| Prerequisite | **:** | Faculty member is logged in to the College Management System and has access to grading. |
| Acceptance Criteria | **:** | **Given**: The faculty member is on the grade management page, and valid grades are available.  **When**: The faculty member selects a student and enters the new grade. The faculty member clicks the "Save" button to update the grade.  **Then**: The system successfully updates the student's grade and provides a confirmation message. |

### **Scenario # S3.2:**

|  |  |  |
| --- | --- | --- |
| *Scenario# S3.2* | : | Updating Grades with Invalid Information |
| Prerequisite | **:** | Faculty member is logged in to the College Management System. |
| Acceptance Criteria | **:** | **Given**: The faculty member is on the grade management page and enters invalid grade information.  **When**: The faculty member clicks the "Save" button.  **Then**: The system displays an error message for the invalid grade and does not update the grade. |

## Story-4: Generate Student Transcript

|  |  |  |
| --- | --- | --- |
| *Story # S4* | : | As Student,  I want to generate a transcript of my academic records  So that I can provide proof of my academic achievements to potential employers or institutions. |
| Priority | **:** | Medium |
| Estimate | **:** | M |
| Reason | **:** | Generating transcripts is important for students who need to provide evidence of their academic history for job applications or further education. |

### **Scenario # S4.1:**

|  |  |  |
| --- | --- | --- |
| *Scenario# S4.1* | : | Generating a Transcript with Complete Records |
| Prerequisite | **:** | Student is logged in to the College Management System and has completed all required courses. |
| Acceptance Criteria | **:** | **Given**: The student is on the transcript generation page.  **When**: The student requests to generate a transcript.  **Then**: The system generates the transcript with complete academic records and provides a downloadable PDF with all course grades and credits. |

### **Scenario # S4.2:**

|  |  |  |
| --- | --- | --- |
| *Scenario# S4.2* | : | Generating a Transcript with Missing Records |
| Prerequisite | **:** | Student is logged in to the College Management System. |
| Acceptance Criteria | **:** | **Given**: The student has incomplete course records or pending grades.  **When**: The student requests to generate a transcript.  **Then**: The system generates a transcript with a note about the missing or pending records and provides a downloadable PDF. |

## Story-5: Schedule University Events

|  |  |  |
| --- | --- | --- |
| *Story # S3* | : | As an Event Coordinator,  I want I want to schedule university events  So that students, faculty, and staff can participate and stay informed about upcoming activities. |
| Priority | **:** | Medium |
| Estimate | **:** | L |
| Reason | **:** | Scheduling and managing university events are essential for fostering community engagement and keeping everyone informed about important activities. |

### **Scenario # S5.1:**

|  |  |  |
| --- | --- | --- |
| *Scenario# S4.2* | : | Scheduling a New Event |
| Prerequisite | **:** | Event Coordinator is logged in to the College Management System. |
| Acceptance Criteria | **:** | **Given**: The event coordinator is on the event scheduling page and has valid event details.  **When**: The event coordinator enters the event details, including date, time, location, and description. The coordinator clicks the "Save" button to schedule the event.  **Then**: The system successfully schedules the event and sends notifications to relevant participants. |

### **Scenario # S5.2:**

|  |  |  |
| --- | --- | --- |
| *Scenario# S5.2* | : | Scheduling an Event with Conflicting Dates |
| Prerequisite | **:** | Student is logged in to the College Management System. |
| Acceptance Criteria | **:** | **Given**: The event coordinator is on the event scheduling page, and the proposed date/time conflicts with another scheduled event.  **When**: The event coordinator enters conflicting event details and clicks the "Save" button.  **Then**: The system displays a conflict warning and prompts the coordinator to choose a different date/time or resolve the conflict. |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | College Management System | Test Designed by: | Keval Dhandhukiya |
| Module Name: | **Course Enrollment** | **Test Designed date:** | 15-02-2025 |
| Release Version: | **1.0** | **Test Executed by:** | **R. B. Gondaliya** |
|  |  | **Test Execution date:** | 04-03-2025 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 College Management System web application through valid credential | TC\_001 |
| TC\_002 | Login to web application with invalid credential | Functional | Login to College Management System web application through invalid credential | TC\_002 |
| TC\_003 | Verify login page elements | GUI | verify that all elements are available on login page | TC\_003 |
| TC\_004 | Reset Password with Valid Email | Functional | Helps to reset the email with valid credentials | TC\_004 |

|  |  |
| --- | --- |
| **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:  kd@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: kd@1234 |  |
| 4 | Enter valid captcha code in captcha field | Captcha 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 performing  validation function fixes it |  | Bug\_002 |

|  |  |
| --- | --- |
| **Test Case Title** | Verify 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 on Username textbox | Cursors focus in Username textbox | Pass |  |  |  |
| 4 | Verify that tab functionality is working properly or not | When tab pressed cursor moves 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 password page | forgot password link not working | Fail |  |  |  |

|  |  |
| --- | --- |
| **Test Case Title** | Reset Password with Valid Email |
| **Test Type** | Functional |
| **Test Priority** | Medium |
| **Pre-condition** | Web application should be accessible, and the user should have a valid account. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Step | Test Case Description | Expected Result | Actual Result | Status | Comment | Data | BUG ID |
| 1 | Click on Forgot Password link | Forgot password page should load properly | Pass |  |  |  |  |
| 2 | Enter valid registered email and click "Reset" | The system should send a password reset link to email | Pass |  | Email: kd@gmail.com |  |  |
| 3 | Verify reset link received in email | A password reset email should be received | Pass |  |  |  |  |
| 4 | Click on reset link and set new password | The system should allow setting a new password | Pass |  |  |  |  |
| 5 | Login using new password | Login should be successful with the newly set password | Pass |  |  |  |  |

# References

* http://www.w3schools.com/html/html\_intro.asp
* https://www.w3schools.com/php/default.asp
* <https://www.javatpoint.com/uml>
* <https://chatgpt.com>
* https://app.diagrams.net/