

Darshan University

A Project Report on

**“ADMISSION MANAGEMENT”**

Under the subject

**Software Engineering (2101CS503)**

B. Tech, Semester – IV

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name:Rushi Gambhava | Enrollment No.:23010101082 |
| Academic Year  (2024-2025) | |
| Internal Guide  Prof. Raj 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 **“Admission management”** 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 **Prof, RajKumar Gondaliya** and that no part of any of report has been directly copied from any students’ reports, without providing due reference.

(Rushi Gambhav)

Student’s Signature

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

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

**CERTIFICATE**

This is to certify that the SRS on **“Admission management” has** been satisfactorily prepared by Rushi Gambhava (23010101082) 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

Rushi Gambhava

**ABSTRACT**

The Admission Management project is a digital platform designed to simplify and streamline the admissions process for educational institutions. It offers a user-friendly student portal for application submission, document uploads, and status tracking, along with an admin dashboard for reviewing applications, communicating with applicants, and making decisions. The system includes automatic email and SMS notifications to keep applicants updated, tools for generating reports and analyzing data, and integration capabilities with other systems like payment gateways and student databases. This project aims to save time, reduce errors, and improve the overall admissions experience for both students and administrators.

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

# Introduction

## Product perspective

The Admission Management project is a web-based platform designed to simplify and improve the admissions process for schools and colleges. It offers a user-friendly portal for students to submit applications, upload documents, and track their status, and a secure dashboard for staff to review applications and communicate with applicants. The system integrates with existing student databases and payment services, provides automatic notifications, and generates reports to help schools analyze and enhance their admissions process. This project aims to save time, reduce errors, and make admissions easier and more efficient for everyone involved.

## Product features

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

* **Administrator** who will be managing admissions and student data.
* **Applicant** who will be applying for admission.
* **Guest** who will request information or submit an application.

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

* Manage the movement of student applications and avoid losing important documents.
* Search for a specific application based on applicant name, admission number, program, etc.
* Print admission-related documents such as offer letters or student IDs.
* Track which applicants have completed admission requirements and submitted necessary documents. Offers modules for Application Management, Student Records, and Queries.
* Approve or reject an application.
* View the list of applicants for each program or department.
* Accept admission confirmation or withdrawal from applicants.
* Add new programs or courses to the system.
* Edit existing applicant information or program details.
* View reports on the number of applications received, admissions approved, etc.
* Access all applicant accounts and update their application status.

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

* View different programs and courses available.
* View the status of their application.
* Create and manage an account in the admission system.
* View the list of documents they have submitted or need to submit.
* Request to change program or defer admission.
* Submit applications and upload required documents online.

## Functional Requirement

### **Student**

1. **Registration**: Students register using name, email ID, and mobile number.
2. **Login**: Students log in using email ID and password.
3. **View Profile**: Students can view their profile.
4. **Edit Profile**: Students can update their profile as needed.
5. **Fill Form**: Students fill out forms for any degree as per requirement.
6. **View Program Details**: Students view details of programs offered by the university.
7. **View Faculty Details**: Students view details about faculty members.
8. **Upload Documents**: Students upload documents like living certificates, proof of ID, and 12th mark sheets.
9. **View Fee Structure**: Students view fee structures for various programs.
10. **Fees Payment**: Students pay fees using cash, check, DD, or UPI.
11. **View Scholarship Details**: Students view scholarships provided by the government or university.
12. **Check Email**: Students check emails for updates.
13. **Collect Fee Receipt**: After payment, students collect a fee receipt.
14. **Apply for Scholarship**: Students apply for scholarships.
15. **View Education Loan Details**: Students view details about education loans.
16. **Apply for Transportation Service**: Students apply for transportation services, select routes and cities, and pay fees.
17. **Track Application Status**: Students can track the status of their admission applications.
18. **View Important Dates**: Students view critical dates related to admissions, payments, and other processes.
19. **Download Admission Form**: Students can download filled admission forms for their records.
20. **Provide Feedback**: Students submit feedback about the admission process.

### **Staff**

1. **Register**: Staff register using name, email ID, and password.
2. **Login**: Staff log in using email ID and password.
3. **View Student Details**: After students fill out forms, staff can view all details.
4. **Send Email**: Staff send emails to students about admission status.
5. **Collect Fees**: Staff collect student fees.
6. **Provide Fee Receipt**: Staff issue fee receipts after payment.
7. **Application Review and Approval**: Staff review and approve applications.
8. **Verify Documents**: Staff verify the uploaded documents submitted by students.
9. **Track Pending Applications**: Staff monitor and manage pending or incomplete applications.

### **Admin**

1. **Add Student**: This feature allows the admin to define and manage the requirements for student admissions
2. **Generate Reports**: Admin generates various reports (e.g., yearly admissions, scholarship applications, fee collection).
3. **Manage Staff Accounts**: Admin creates, updates, and deletes staff accounts.
4. **Audit Logs**: Admin monitors activity logs for transparency and security.
5. **Set Application Deadlines**: Admin defines and updates deadlines for applications, fee payments, and other key events.
6. **Monitor Program Popularity**: Admin analyzes the popularity of different programs based on student applications.
7. **Update Program Details**: Admin manages details of programs offered by the university.
8. **Handle Complaints**: Admin addresses complaints and issues raised by students or staff.
9. **Approve Scholarships**: Admin reviews and approves scholarship applications.
10. **Oversee Transportation Services**: Admin monitors transportation service usage and payments.

## Non-Functional Requirement

### Usability:

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

### Accuracy:

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

### Availability:

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

### Maintainability:

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

# Design and Implementation Constraints

## Use case diagram

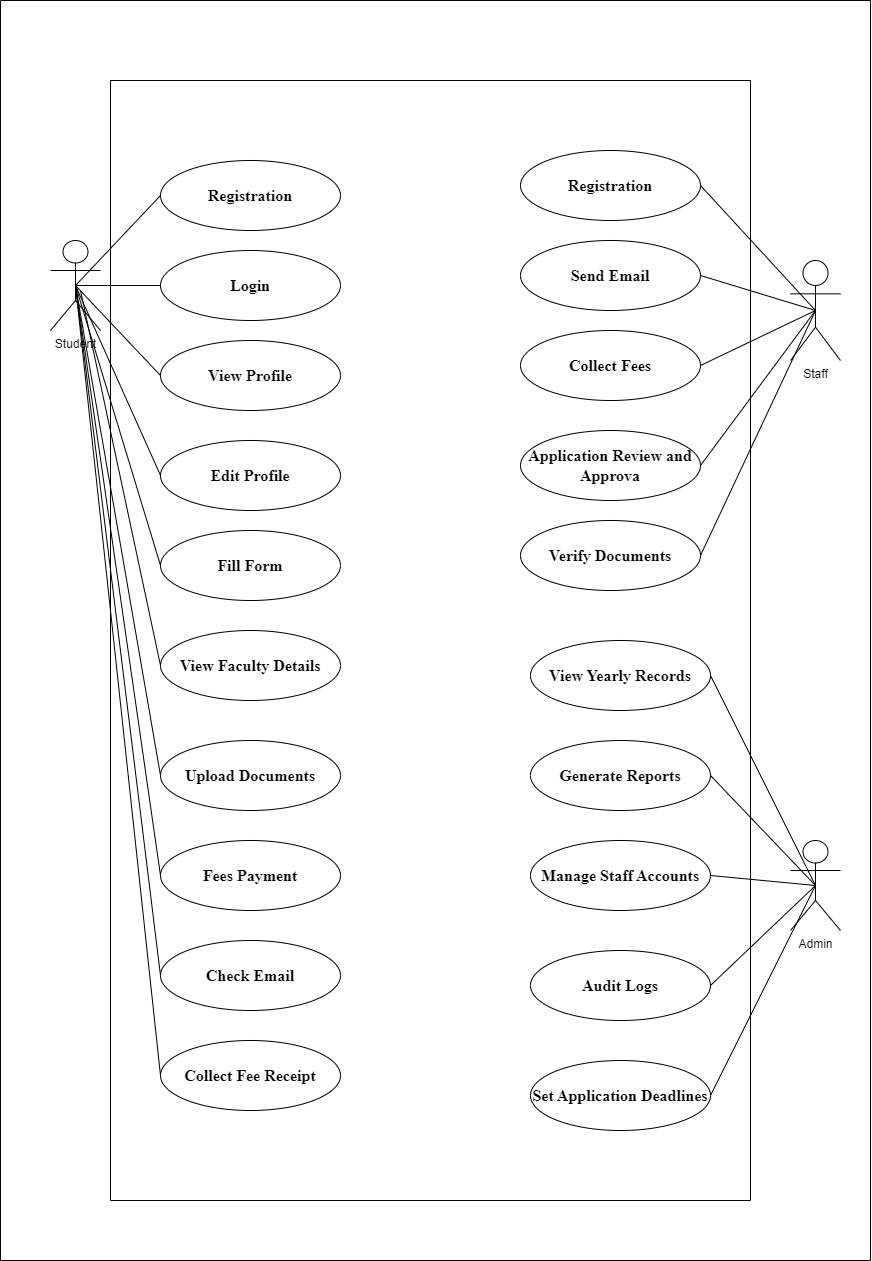


Figure 2.1‑1 Use case diagram for Admission management system

## Activity diagram and Swimlane diagram

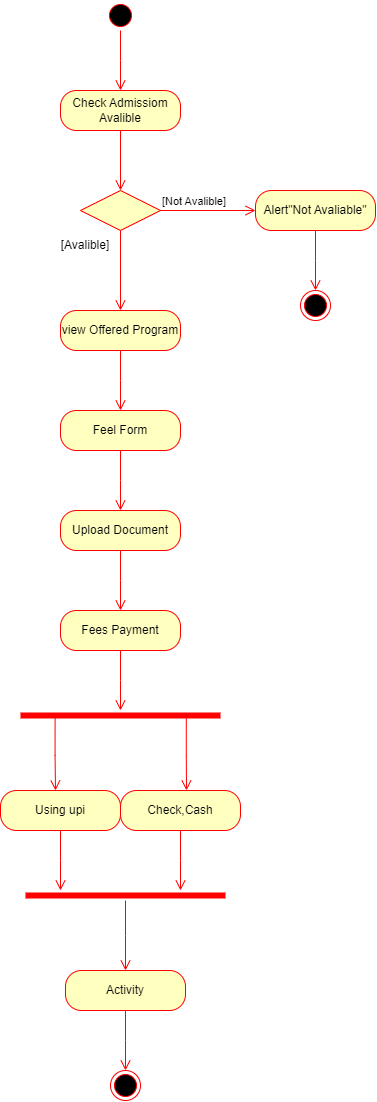


Figure 2.2‑1 Activity diagram of Admission Management(Student)

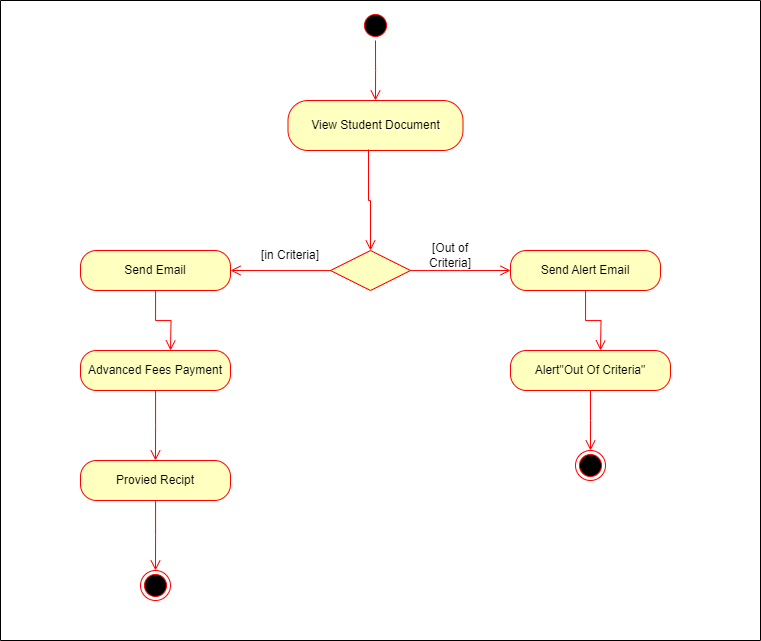


Figure 2.2‑2 Activity Diagram of Admission Management(Staff)

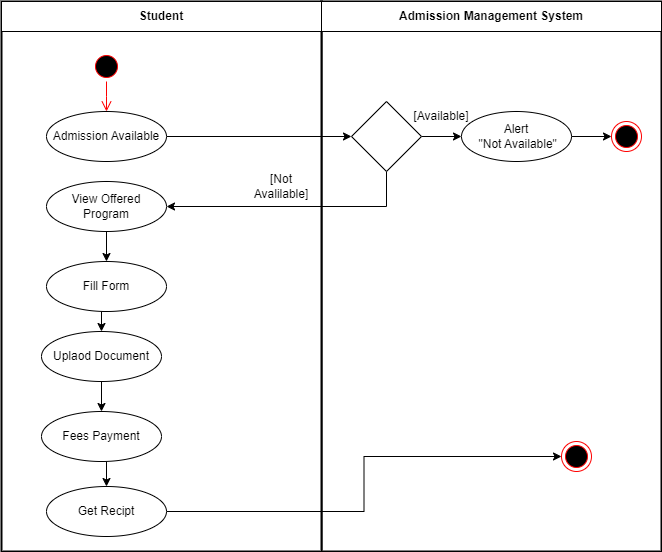


Figure 2.3‑Swimlane Diagram of Admission Management

## Sequence diagram

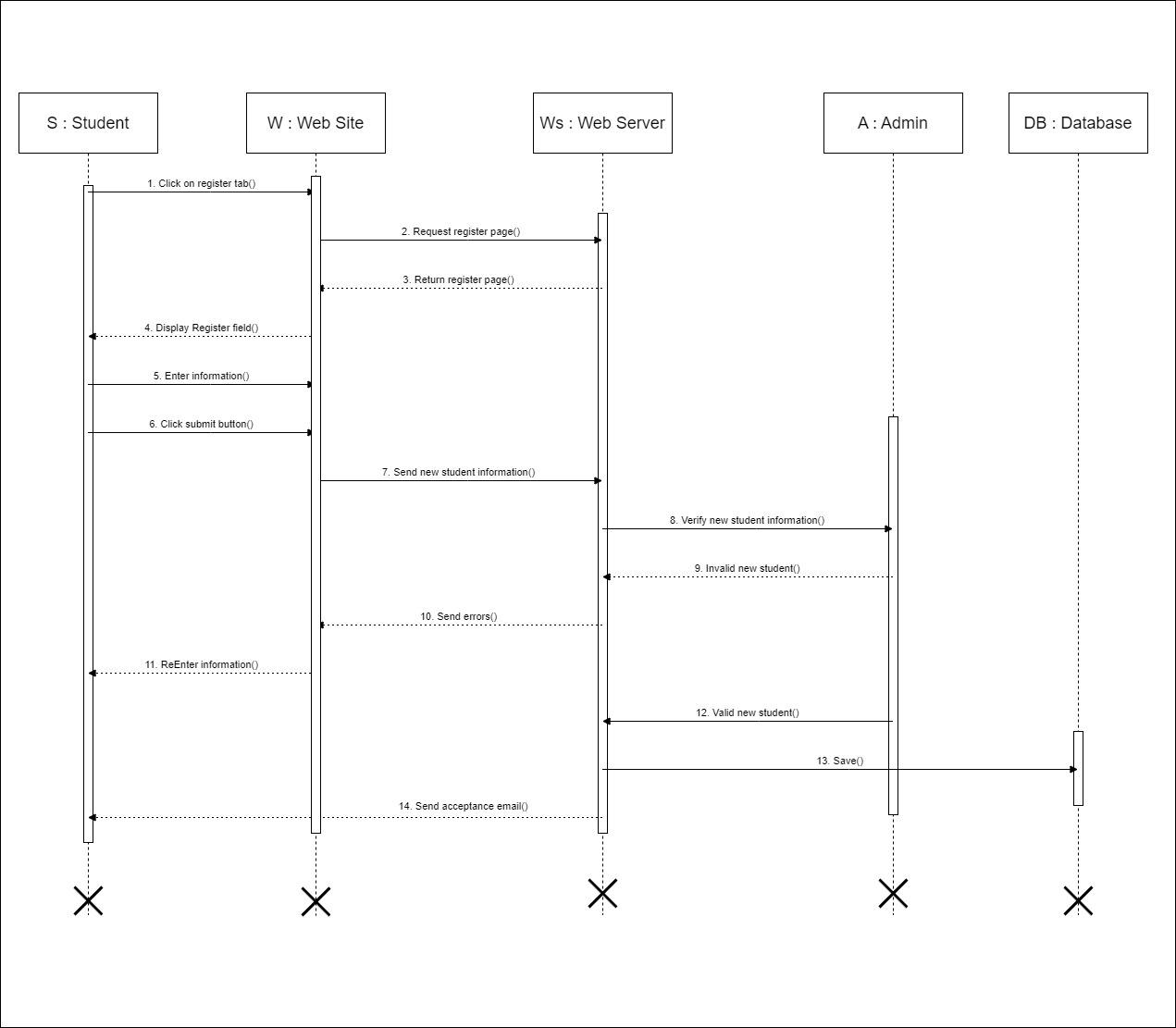
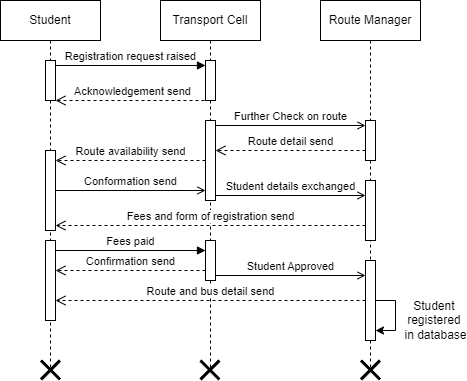


Figure 2.3‑1 Sequence diagram for student Admission



*Figure 2.3-2 Sequence diagram for Trasport Registration*

## State diagram

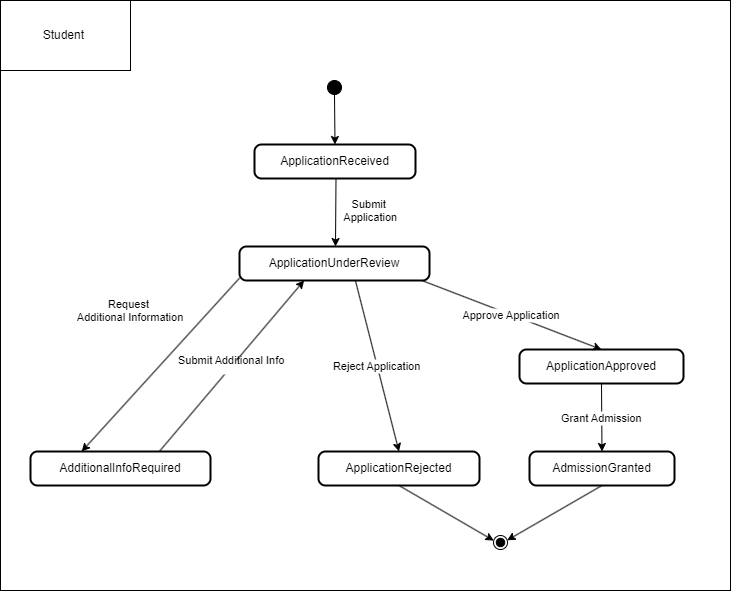


Figure 2.4‑1 State diagram of Student

## Class diagram

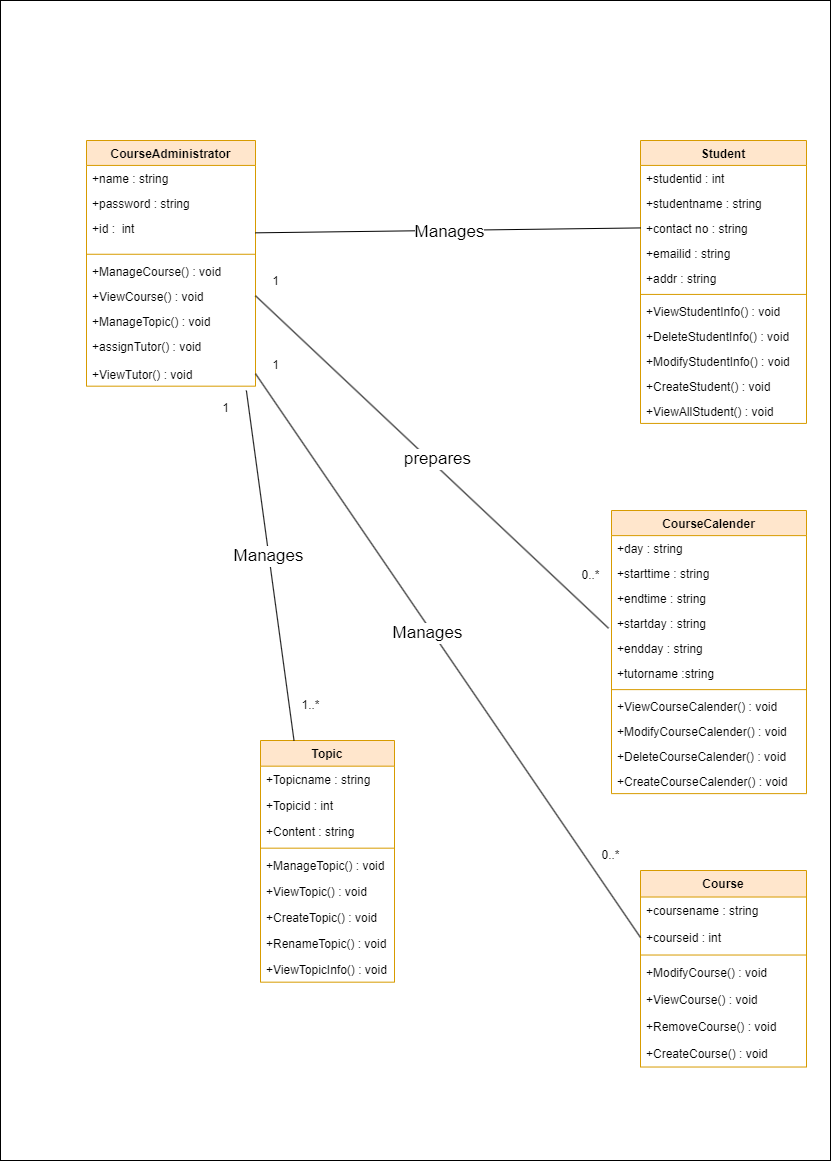


Figure 2.5‑1 Class diagram for Admission management system

Data flow diagram

### Context diagram (level-0)

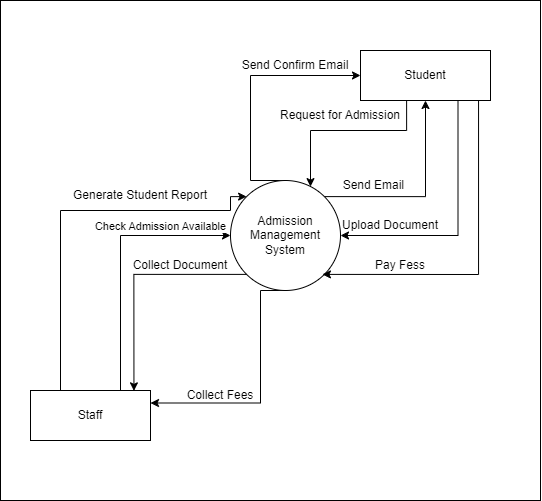


Figure 2.5‑2 Context diagram for Admission management system

### DFD Level-1

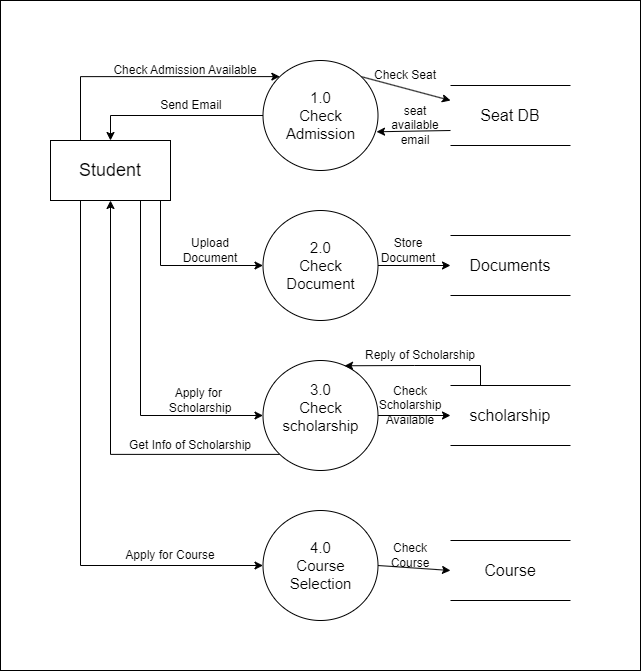


Figure 2.5‑3 DFD level-1 for Admission management system

# External interface requirement (Screens)

## Screen-1: Registration Form

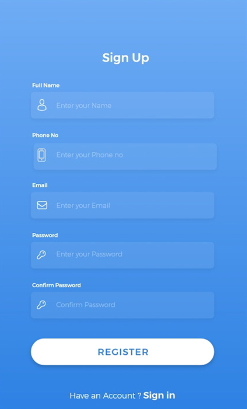


Figure 3.1‑1 Screen-1: Registration Form

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

Table 3.1‑1 Screen element of Registration form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Full Name | Textbox | M | 1 | Full name field should be editable and accept the student full name. |
| 2 | Phone No | Textbox | M | 1 | Phone no field should be editable and accept student phone no. |
| 3 | Email | Textbox | M | 1 | Email field should be editable and accept the email with proper format. |
| 4 | Password | Password | M | 1 | Password field should be editable and accept the password and display as star or dot. |
| 5 | Confirm password | Password | M | 1 | Confirm Password field should be editable and accept the password and display as star or dot. |
| 6 | Register | Button | ------ | ------ | Register is a button for store the entered data into database. |
| 7 | Have a sign in | Link | ----- | ------ | If you are Successfully register then click in this link to redirect into the sign in or login page. |

## Screen-3: View Profile Items

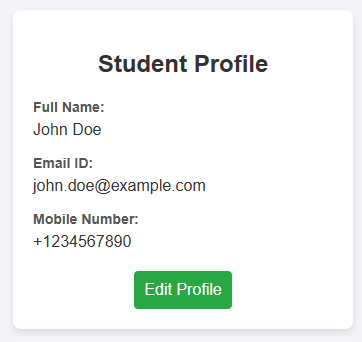


Figure 3.2-1 Screen-2: Add borrower book

**Purpose**: This module will allow users to view and manage their profile information. Users can edit their full name, email address, and mobile number by clicking the "Edit Profile" button. It ensures that the profile information remains up-to-date and accurate.

Table 3.2‑1 Screen element of Add borrower book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Full Name | Textbox | M | 1 | Full name field should be editable and accept the student full name. |
| 2 | |  |  | | --- | --- | | |  | | --- | | Email ID | | | Textbox | M | 1 | Email field should be editable and accept a valid email in proper format |
| 3 | |  |  | | --- | --- | | |  | | --- | | Mobile Number | | | Textbox | M | 1 | Mobile number field should be editable and accept a valid phone number. |
| 4 | |  | | --- | |  |  |  | | --- | | Edit Profile | | Button | ------ | ------ | Edit Profile button should enable editing of the fields in the profile. |

## Screen-4: Cart Page

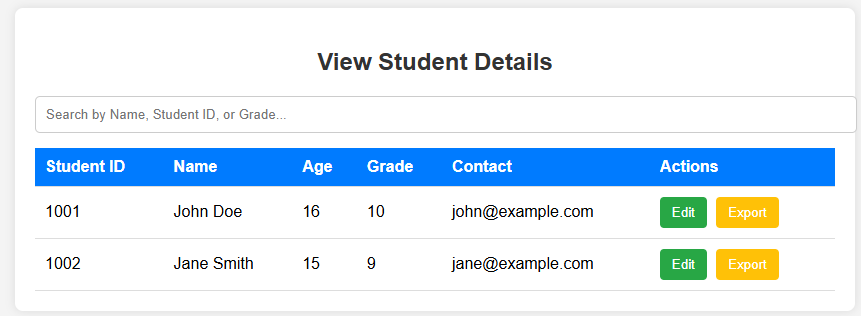


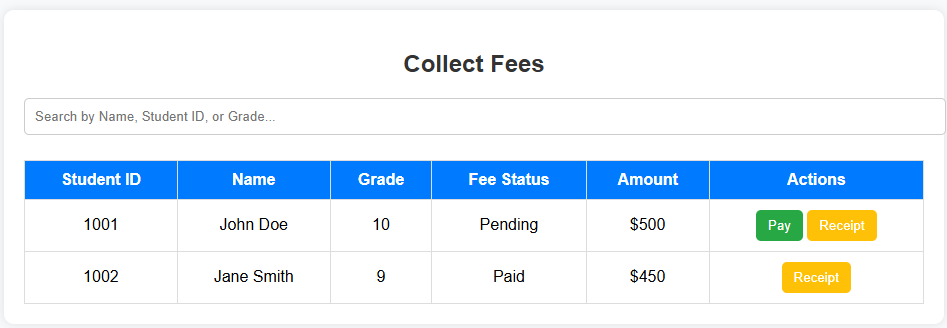
Figure 3.3-1 Screen-3: Add borrower book

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

Table 3.3‑1 Screen element of Login form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | **Search Box** | Textbox | O | N | Allows staff to search for students using Name, Student ID, or Grade. |
| 2 | **Student ID** | |  | | --- | |  |  |  | | --- | | Auto-generated | | M | 1 | Unique identifier for each student. |
| 3 | **Name** | Text | M | 1 | Displays the student's full name. |
| 4 | **Age** | Number | M | 1 | Displays the student’s age. |
| 5 | **Grade** | Dropdown | M | 1 | Displays the student’s grade level. |
| 6 | **Contact** | Email | M | 1 | |  | | --- | |  |  |  | | --- | | Shows the student’s registered email ID. | |
| 7 | **Edit Button** | Button (Green) | O | 1 | |  | | --- | |  |  |  | | --- | | Allows staff to edit student details. | |
| 8 | **Export Button** | Button (Yellow) | O | 1 | Allows staff to export student details. |

## Screen-4: Student Result Page



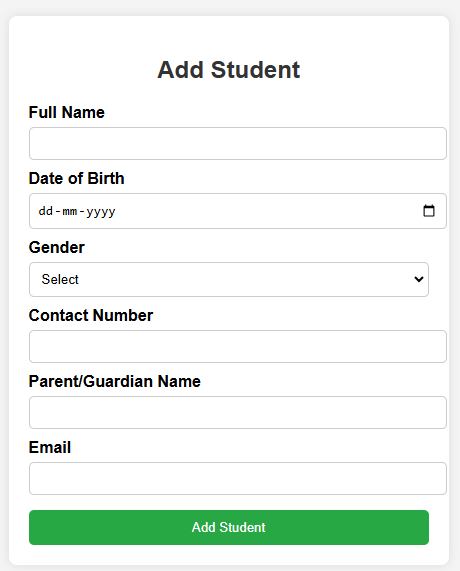
*Figure 3.4-1 Screen-4: collage fees page*

**Purpose:** This Page is Result of Student It Display student’s result of all exam they gave.

Table 3.3‑1 Screen element of Login form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | |  | | --- | |  |  |  | | --- | | Student ID | | Text | M | 1 | |  | | --- | |  |  |  | | --- | | Displays the unique student ID for identification. | |
| 2 | Name | |  | | --- | |  |  |  | | --- | | Text | | M | 1 | Displays the student's full name. |
| 3 | Fee Status | Label | M | 1 | Shows the fee payment status (e.g., "Pending" or "Paid"). |
| 4 | Amount | Label | M | 1 | Displays the fee amount to be paid. |
| 5 | Search Bar | Textbox | O | N | |  | | --- | |  |  |  | | --- | | Allows staff to search for students by name, student ID, or grade. | |
| 6 | |  | | --- | |  |  |  | | --- | | Pay Button | | Button | O | N | Enables payment processing for students with pending fees. |
| 7 | |  | | --- | |  |  |  | | --- | | Receipt Button | | Button | O | N | Generates a payment receipt for the transaction. |

## Screen-5: Certificate Page



*Figure 3.5-1 Screen-5: student Result Page*

**Purpose:** This is Certificate request from that accept the which kind of certificate student and generate that certificate for student.

Table 3.3‑1 Screen element of Login form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | |  | | --- | |  |  |  | | --- | | Full Name | | |  | | --- | | Textbox |  |  | | --- | |  | | M | 1 | |  | | --- | |  |  |  | | --- | | Full name field should be editable and accept the student's full name. | |
| 2 | Date of Birth | |  | | --- | |  |  |  |  | | --- | --- | | |  | | --- | | Date Picker | | | M | 1 | Allows selecting the student's date of birth in dd-mm-yyyy format. |
| 3 | Gender | Dropdown | M | 1 | Dropdown to select the student's gender (Male, Female, Other). |
| 4 | |  | | --- | |  |  |  | | --- | | Contact Number | | Textbox | M | 1 | Field should be editable and accept the student's contact number. |
| 5 | Parent/Guardian Name | Textbox | M | 1 | |  | | --- | |  |  |  | | --- | | Allows staff to search for students by name, student ID, or grade. | |
| 6 | Email | Textbox | M | 1 | Enables payment processing for students with pending fees. |
| 7 | |  | | --- | |  |  |  | | --- | | Add Student Button | | Button | ----- | ----- | Generates a payment receipt for the transaction. |

# Database design

## List of Tables

* Student
* Course
* Staff
* Fees
* Document

Table 4.1‑1 Table: Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| Student ID | int | NN | PK (Auto Increment) |  |
| Student Name | varchar(100) | NN |  |  |
| Email | varchar(100) | AN |  |  |
| Age | int | AN |  |  |
| Address | varchar(100) | AN |  |  |
| Scholarship ID | int | NN | FK | Reference of scholarship detail |
| Course ID | int | NN | Fk | Reference of Course. |

Table 4.1‑2 Table: Course

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| Course ID | int | NN | PK (Auto Increment) |  |
| Course Name | varchar(100) | NN |  |  |
| Course Detail | Varchar(100) | AN |  |  |
| Faculty ID | int | AN | FK | Reference of Faculty |

Table 4.1‑3 Table: Staff

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| Staff ID | int | NN | PK (Auto Increment) |  |
| Staff Name | varchar(100) | AN |  |  |
| Is Admin | Boolean | AN |  |  |
| Designation | varchar(100) | AN |  |  |
| Student ID | int | NN | FK | Reference of student |

Table 4.1‑4 Table: Faculty

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| fee\_id | int | NN | PK (Auto Increment) |  |
| student\_id | int | AN | FK | Reference of student |
| fee date | date | AN |  |  |
| fee type | varchar(100) | AN |  |  |
| payment status | int | AN |  |  |

Table 4.1‑5 Table: Document

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| DocumentID | int | NN | PK (Auto Increment) |  |
| Document Detail | varchar(100) | AN |  |  |
| Student ID | varchar(100) | NN | FK | Reference of student |

# Stories and Scenario

## Story-1: Add New Student in Admission Management

|  |  |  |
| --- | --- | --- |
| *Story # A1* | : | As a Admission Officer,  I want to add a new student to the admission database  So that their information is recorded and they can proceed with the admission process. |
| Priority | **:** | High |
| Estimate | **:** | XL |
| Reason | **:** | The addition of new students to the admission database is crucial for managing the admission process and ensuring accurate records are maintained. |

### Scenario# A1.1

|  |  |  |
| --- | --- | --- |
| *Scenario#* A1.1 | : | Adding a New Student with Valid Information |
| Prerequisite | **:** | Admission Officer is logged in to the admission management system. |
| Acceptance Criteria | **:** | **Given:**  The Admission Officer is navigated to the student admission management page. Valid student information, including name, age, grade, and contact details, is added.  **When:**  The Admission Officer selects the "Add New Student" option  And The Admission Officer enters valid student details  The Admission Officer clicks the "Save" button to add the student to the admission database.  **Then** The system successfully adds the student to the admission database, and the Admission Officer receives a confirmation message with the student's identification number. |

### Scenario# A1.2

|  |  |  |
| --- | --- | --- |
| *Scenario# A1.2* | : | Adding a New Student with Missing Required Information. |
| Prerequisite | **:** | Admission Officer is logged in to the admission management system. |
| Acceptance Criteria | **:** | **Given:** The Admission Officer is navigated to the student admission management page. Some required student information fields, such as name or age, are left blank.  **When:** The Admission Officer selects the "Add New Student" option And enters incomplete student details And clicks the "Save" button to add the student to the admission database. **Then**: The system prompts the Admission Officer to fill in the missing required fields, and the student is not added to the database until all required information is provided. |

## Story-2: View Profile Student in Admission Management

|  |  |  |
| --- | --- | --- |
| *Story # A2* | : | **As an** Admission Officer, **I want to** view a student's profile in the admission management system **So that** I can access their details and verify or update information as needed |
| Priority | **:** | High |
| Estimate | **:** | L |
| Reason | **:** | Accessing a student's profile is crucial for reviewing admission details, updating records, and ensuring accurate information is maintained. |

### Scenario# A2.1

|  |  |  |
| --- | --- | --- |
| *Scenario#* A1.1 | : | Viewing a Student Profile with Valid Information |
| Prerequisite | **:** | The Admission Officer is logged into the admission management system. |
| Acceptance Criteria | **:** | **Given**: The Admission Officer navigates to the student profile management page and searches for a student using a valid identifier (e.g., name, student ID).  **When**:  The Admission Officer enters the student's details in the search bar.  The system retrieves and displays the student's profile with complete information (e.g., name, age, grade, contact details).  **Then**: The Admission Officer successfully views the student's profile with all details displayed correctly. |

### Scenario# A2.2

|  |  |  |
| --- | --- | --- |
| *Scenario# A1.2* | : | Attempting to View a Non-Existent Student Profile |
| Prerequisite | **:** | The Admission Officer is logged into the admission management system. |
| Acceptance Criteria | **:** | **Given**: The Admission Officer navigates to the student profile management page and searches for a student using an incorrect or non-existent identifier.  **When:**  The Admission Officer enters invalid student details in the search bar.  The system attempts to find the student but does not retrieve any records.  **Then**: The system displays an error message stating that the student profile does not exist, prompting the Admission Officer to check the entered details or add a new student |

## Story-3: Fees Payment in Admission Management

|  |  |  |
| --- | --- | --- |
| *Story # A3* | : | As an Admission Officer,  I want to process and manage student fee payment So that student payment records are updated, and they can proceed with the admission process. |
| Priority | **:** | High |
| Estimate | **:** | XL |
| Reason | **:** | Ensuring that student fees are correctly recorded is crucial for managing the admission process and preventing discrepancies in financial records. |

### Scenario# A3.1

|  |  |  |
| --- | --- | --- |
| *Scenario#* A1.1 | : | Processing Fee Payment Successfully |
| Prerequisite | **:** | Admission Officer is logged into the admission management system. |
| Acceptance Criteria | **:** | **Given:** The Admission Officer is on the fee payment page of the admission management system.  **When:**   * The Admission Officer selects the student and navigates to the fee payment section. * The Admission Officer enters the correct fee amount and payment details. * The Admission Officer clicks the "Submit Payment" button.   **Then:**   * The system successfully records the payment. * The Admission Officer receives a confirmation message with the transaction ID. * The student’s fee status is updated to "Paid." |

### Scenario# A3.2

|  |  |  |
| --- | --- | --- |
| *Scenario# A1.2* | : | Handling Payment Failure or Incomplete Payment |
| Prerequisite | **:** | Admission Officer is logged into the admission management system. |
| Acceptance Criteria | **:** | **Given:** The Admission Officer is on the fee payment page of the admission management system.  **When:**   * The Admission Officer attempts to process a payment but enters incomplete or incorrect payment details. * The Admission Officer clicks the "Submit Payment" button.   **Then:**   * The system displays an error message prompting the officer to check the details. * The payment is not processed, and the student’s fee status remains "Unpaid." * The Admission Officer can retry the payment after correcting the errors.   does not exist, prompting the Admission Officer to check the entered details or add a new student |