

Darshan University

A Project Report on

**“ADMISSION MANAGEMENT”**

Under the subject

**Software Engineering (2101CS503)**

B. Tech, Semester – V

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name:Nadiyapara Khushal | Enrollment No.:22010101121 |
| 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.

(Khushal Nadiyapara)

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 khushal Nadiyapara(22010101121) 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

Khushal Nadiyapara

**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.

**Table of Contents**

[List of Figures VI](#_Toc177549120)

[List of Tables II](#_Toc177549121)

[1 Introduction 1](#_Toc177549122)

[1.1 Product perspective 1](#_Toc177549123)

[1.2 Product features 1](#_Toc177549124)

[1.2.1 There are three different users who will be using this product: 1](#_Toc177549125)

[1.2.2 The features that are required for the Administrator are: 1](#_Toc177549126)

[1.2.3 The features that are required for the Applicant are: 1](#_Toc177549127)

[1.3 Functional Requirement 2](#_Toc177549128)

[1.3.1 Student 2](#_Toc177549129)

[1.3.2 Staff 2](#_Toc177549130)

[1.3.3 Admin 2](#_Toc177549131)

[1.4 Non-Functional Requirement 2](#_Toc177549132)

[1.4.1 Usability: 2](#_Toc177549133)

[1.4.2 Accuracy: 2](#_Toc177549134)

[1.4.3 Availability: 2](#_Toc177549135)

[1.4.4 Maintainability: 2](#_Toc177549136)

[2 Design and Implementation Constraints 3](#_Toc177549137)

[2.1 Use case diagram 3](#_Toc177549138)

[2.2 Activity diagram and Swimlane diagram 4](#_Toc177549139)

[2.3 Sequence diagram 6](#_Toc177549140)

[2.4 State diagram 8](#_Toc177549141)

[2.5 Class diagram 9](#_Toc177549142)

[2.6 Data flow diagram 10](#_Toc177549143)

[2.6.1 Context diagram (level-0) 10](#_Toc177549144)

[2.6.2 DFD Level-1 11](#_Toc177549145)

[3 External interface requirement (Screens) 12](#_Toc177549146)

[3.1 Screen-1: Registration Form 12](#_Toc177549147)

[3.2 Screen-2: Login Form 13](#_Toc177549148)

[3.3 Screen-3: Add Order Items 14](#_Toc177549149)

[3.4 Screen-4: Cart Page 15](#_Toc177549150)

[4 Database design 17](#_Toc177549151)

[4.1 List of Tables 17](#_Toc177549152)

[5 Stories and Scenario 19](#_Toc177549153)

[5.1 Story-1: Add New Student in Admission Management 19](#_Toc177549154)

[5.1.1 Scenario# A1.1 19](#_Toc177549155)

[5.1.2 Scenario# A1.2 19](#_Toc177549156)

[5.1.3 Scenario# A1.3 20](#_Toc177549157)

[5.1.4 Scenario# A1.4 20](#_Toc177549158)

[5.1.5 Scenario# A1.5 20](#_Toc177549159)

[6 Test cases 21](#_Toc177549160)

[7 References 24](#_Toc177549161)

# List of Figures

[Figure 2.1‑1 Use case diagram for Admission management system 3](#_Toc26066)

[Figure 2.2‑1 Activity diagram of Admission Management(Student) 4](#_Toc31524)

[Figure 2.2‑2 Activity Diagram of Admission Management 5](#_Toc24453)

[Figure 2.3‑1 Sequence diagram for student Admission 7](#_Toc28616)

[Figure 2.4‑1 State diagram of Student 8](#_Toc5755)

[Figure 2.4‑2 State diagram for Staff 8](#_Toc4678)

[Figure 2.5‑1 Class diagram for Admission management system 9](#_Toc13418)

[Figure 2.6‑1 Context diagram for Admission management system 10](#_Toc26)

[Figure 2.6‑2 DFD level-1 for Admission management system 11](#_Toc6600)

[Figure 3.1‑1 Screen-1: Registration Form 12](#_Toc11668)

[Figure 3.2‑1 Screen-2: Login Form 13](#_Toc28291)

[Figure 3.3‑1 Screen-3: Add Order Items 14](#_Toc7212)

[Figure 3.3‑1 Screen-4: Cart Page 15](#_Toc10995)

# List of Tables

[Table 3.1‑1 Screen element of Registration form 12](#_Toc28342)

[Table 3.2‑1 Screen element of Login form 13](#_Toc23797)

[Table 3.3‑1 Screen element of Add borrower book 15](#_Toc13130)

[Table 3.2‑1 Screen element of Login form 16](#_Toc15812)

[Table 4.1‑1 Table: Student 17](#_Toc22552)

[Table 4.1‑2 Table: Course 17](#_Toc998)

[Table 4.1‑3 Table: Staff 17](#_Toc31273)

[Table 4.1‑4 Table: Faculty 17](#_Toc22851)

# 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

* Registration: They have register using name , email id , mobile no .
* Login: They have login using email id and password.
* View Profile: Student is view the profile.
* Edit Profile: Student is update the profile as per requirement.
* Fill Form: Student is fill form of any degree as per requirement.
* View program Detail: student is view the program are offered by university.
* View program Detail: student is view the all detail of offered program.
* View Faculty Detail: they are also view the faculty detail.
* Upload Document:Student have upload document like living certificate , proof id , 12th mark sheet.
* View Fees Structure : Student are view the fees structure like B.tech c.e. fees is 90000 .
* Fees payment : student is also selected the fees type like cash , check , DD , UPI etc.. .
* View Scholarship Detail: Student also view the scholarship provide by government or university and view all detail.
* Check Email : After all process student are check the mail.
* Collect the fees receipt : After fees payment student are collect fees receipt.
* Apply For Scholarship : student have apply for scholarship.
* View Education Loan Detail : students are view the education loan detail.
* Apply for transportation service : student is also apply for transportation service the select the root and city and pay fees.

### Staff

* Register: Staff is register with name ,email id ,password .
* Login : Staff is register with email id ,password.
* View Student Detail: After student fill form then staff are show all detail of the student.
* Send Email: After all process staff send email to student you are selected or not.
* Collect fess: Staff are collect the fees of student.
* Provide fees receipt : After fees payment staff are provide the receipt.
* Application review and Approval : Staff is approve the application.

### Admin

* View Yearly Record: Admin are view yearly record.

## 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

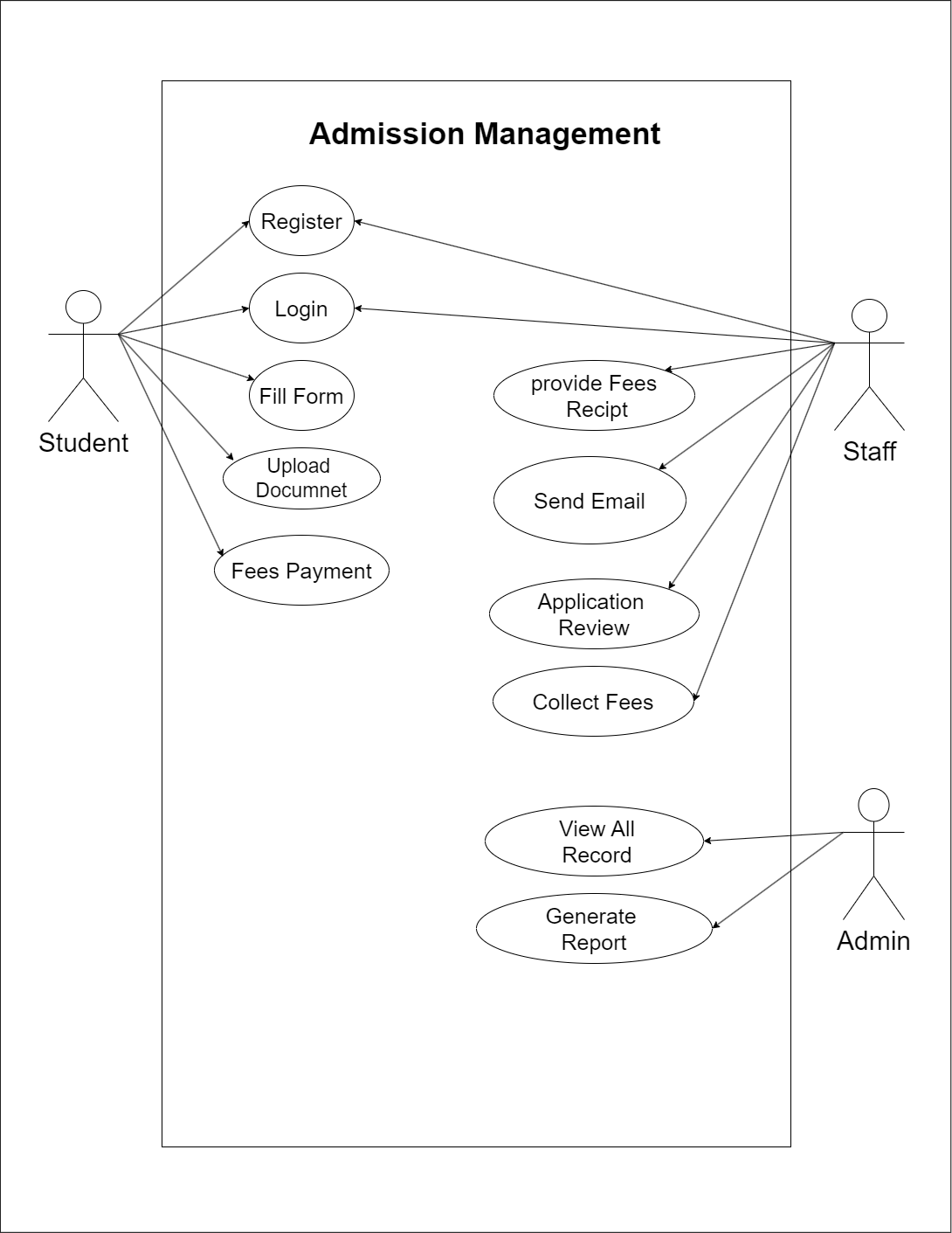


Figure 2.1‑1 Use case diagram for Admission management system

## Activity diagram and Swimlane diagram

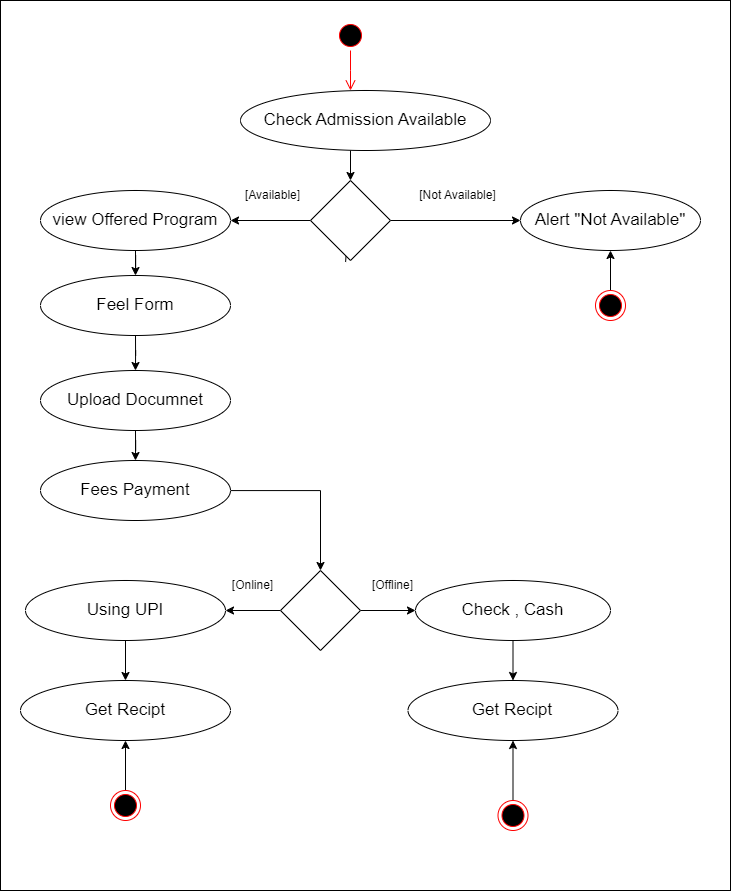


Figure 2.2‑1 Activity diagram of Admission Management(Student)

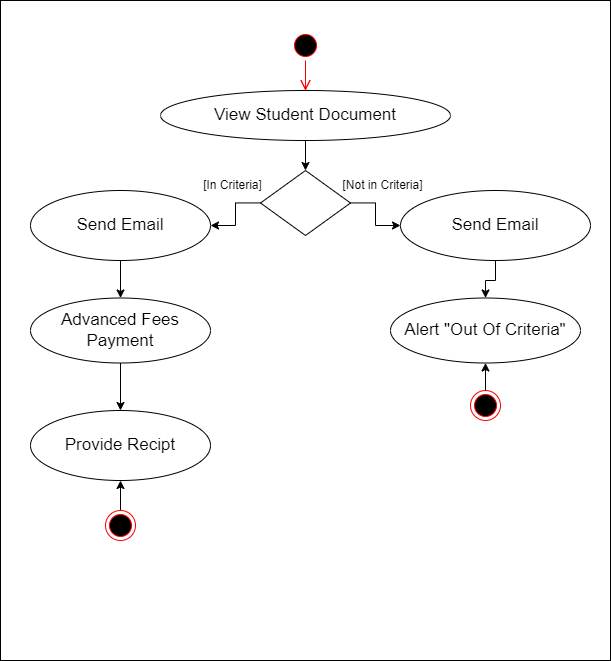


Figure 2.2‑2 Activity Diagram of Admission Management(Staff)

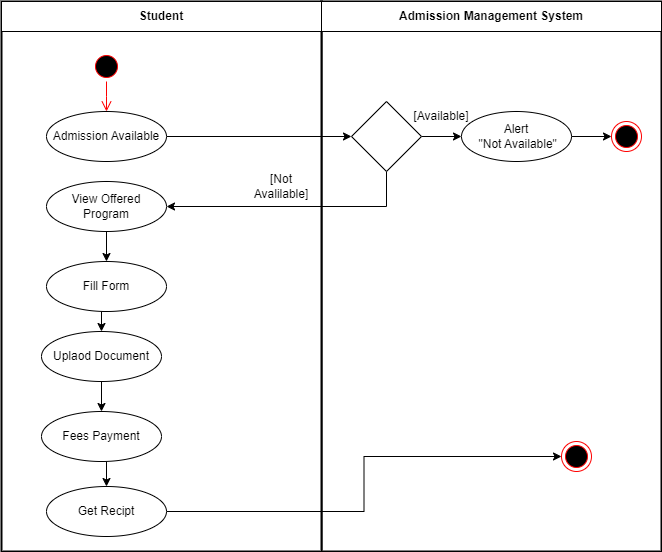


Figure 2.3‑Swimlane Diagram of Admission Management

## Sequence diagram

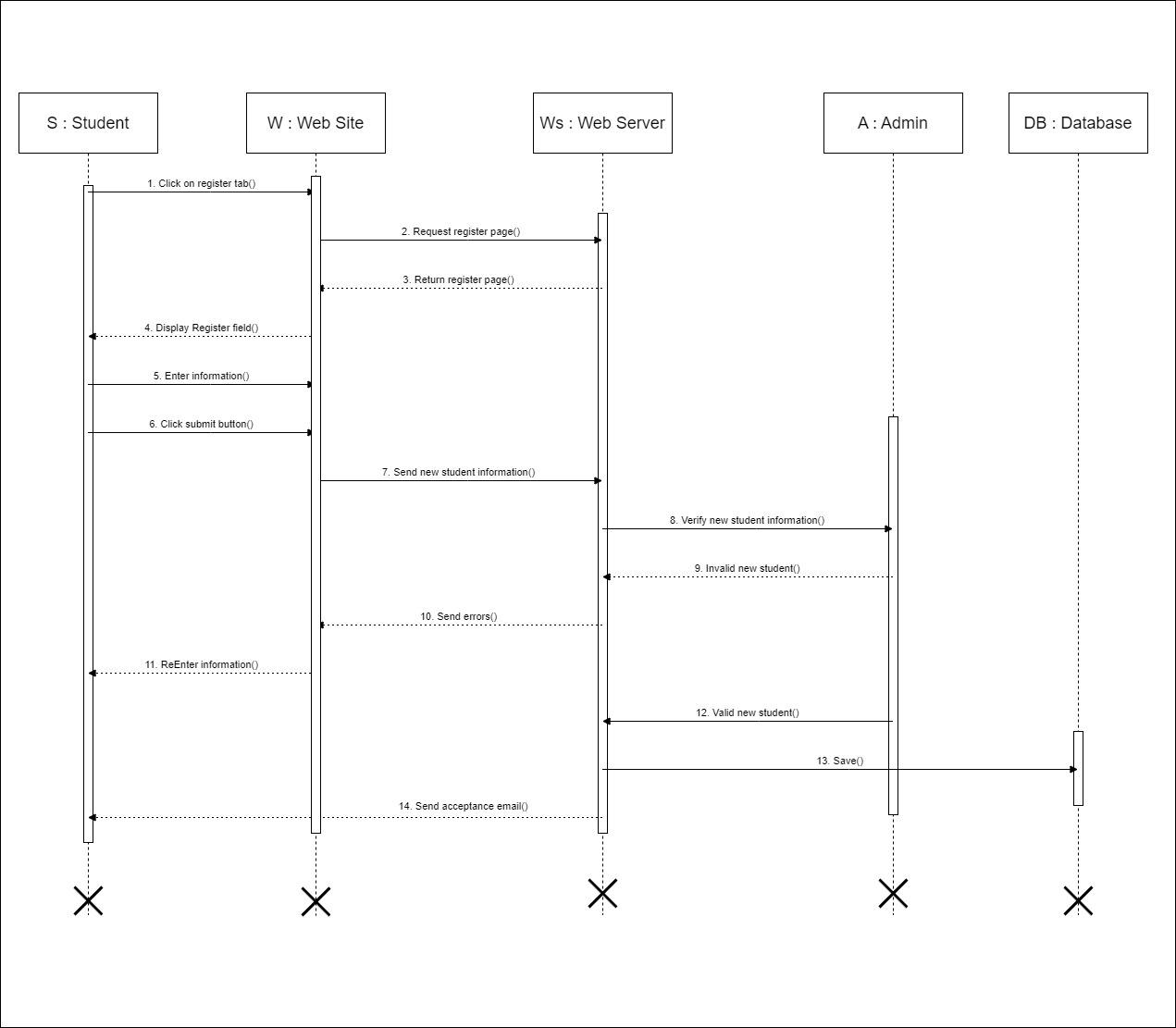


Figure 2.3‑1 Sequence diagram for student Admission

## State diagram

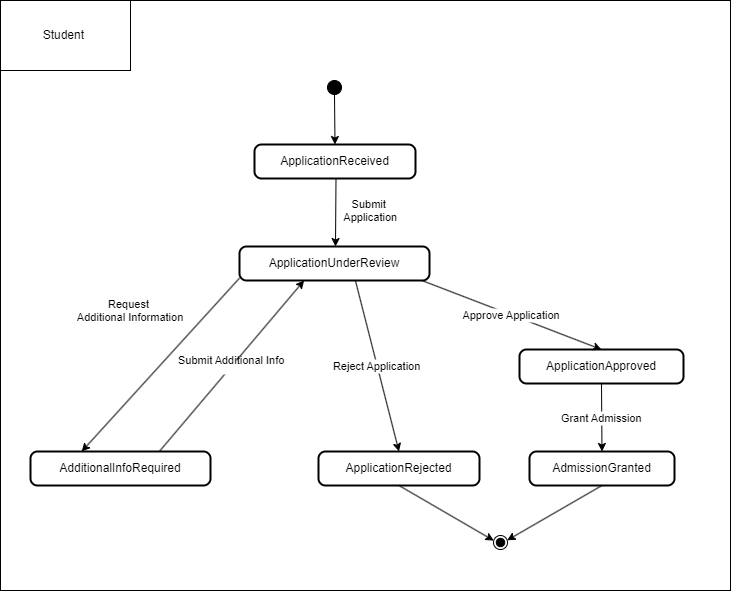


Figure 2.4‑1 State diagram of Student

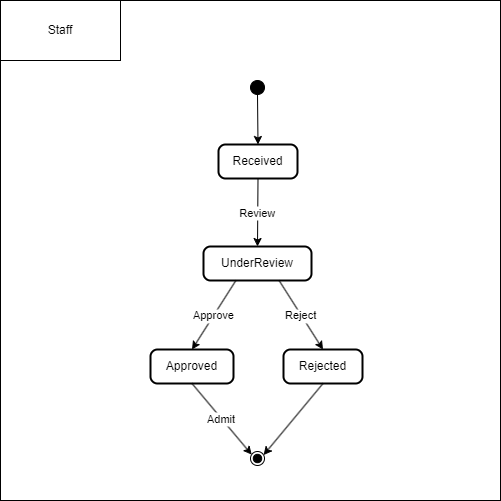


Figure 2.4‑2 State diagram for Staff

## Class diagram

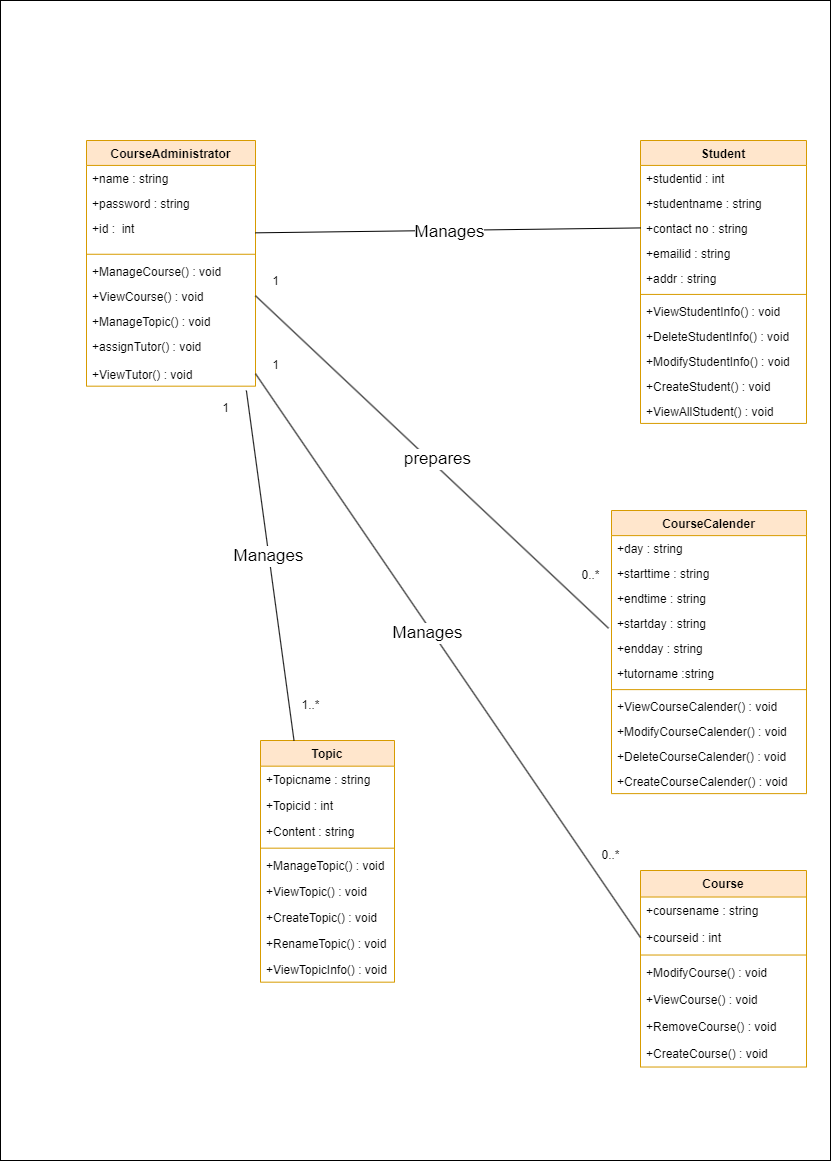


Figure 2.5‑1 Class diagram for Admission management system

## Data flow diagram

### Context diagram (level-0)

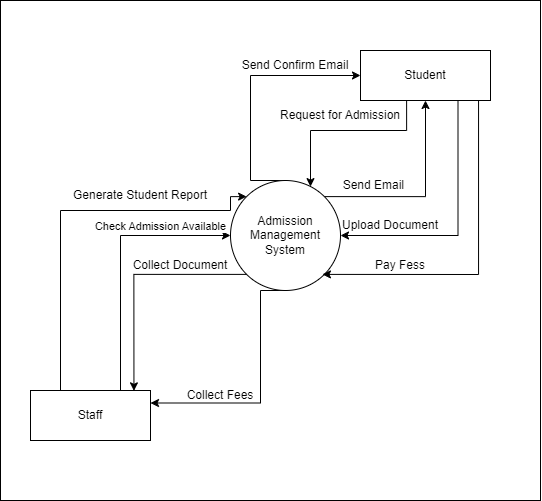


Figure 2.6‑1 Context diagram for Admission management system

### DFD Level-1

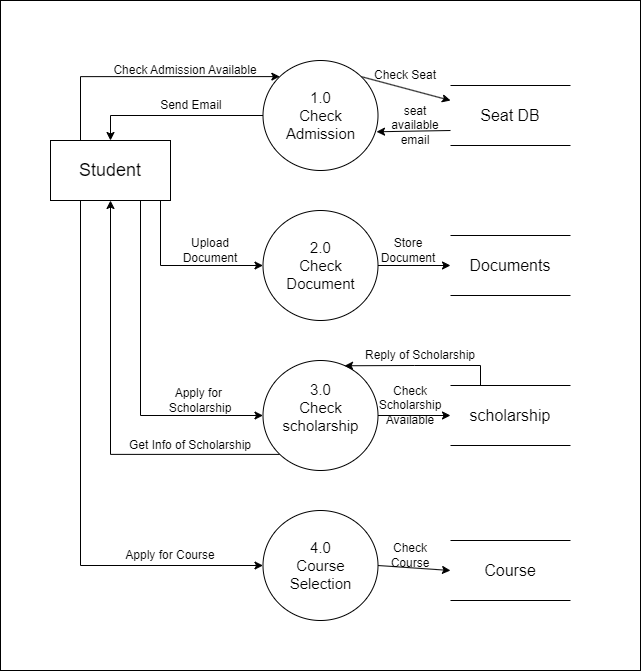


Figure 2.6‑2 DFD level-1 for Admission management system

# External interface requirement (Screens)

## Screen-1: Registration Form

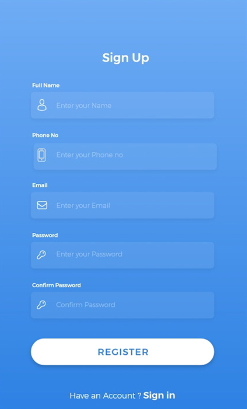


Figure 3.1‑1 Screen-1: Registration Form

**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-2: Login Form

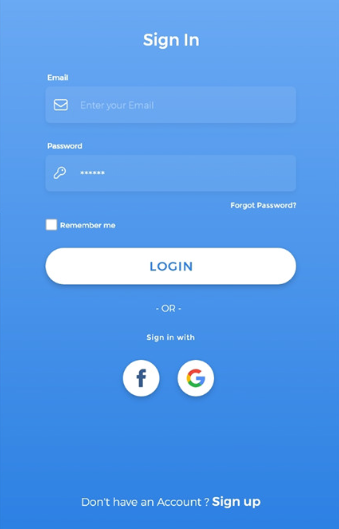


Figure 3.2‑1 Screen-2: Login Form

**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 3.2‑1 Screen element of Login form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | Email | Textbox | M | 1 | Email field should be editable and accept the email. |
| **2** | Password | Password | M | 1 | Password field should be editable and accept the password and display as star or dot. |
| **3** | Forgot Password | Link | ------ | ------ | Link for navigate to Forgot password page for allows users to recover password. |
| **4** | Remember Me | Checkbox | M | 1 | Saving login credentials through remember me checkbox |
| **6** | Login | Button | ------ | ------ | Login button navigates to another page even if valid login credentials. |
| **7** | Sign in with Facebook or Google | Button | ----- | ----- | Button for navigate to sign in with facebook page or sign in with google page |
| **8** | Sign up | Link | ----- | ----- | If you are new user then click on this link. |

## Screen-3: Add Order Items



Figure 3.3‑1 Screen-3: Add borrower book

**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 3.3‑1 Screen element of Add borrower book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | Cart | Button | ------ | ------- | Cart button is show all product in store in to your cart. |
| **2** | Back | Button | ------ | ------ | Back button is navigate to the previous slide. |
| **3** | Product Detail | Box | ------ | ------ | Show the product name and price. |
| **4** | Add to cart(+) | Button | ------ | ------ | Add to cart button is add your item in cart screen. |

## Screen-4: Cart Page

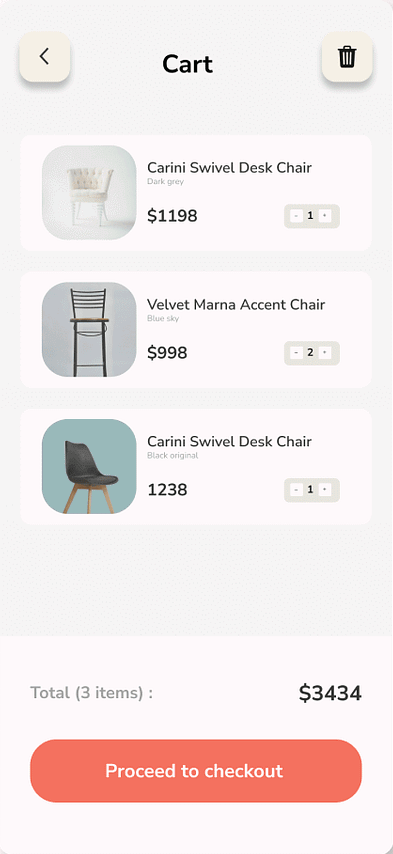


Figure 3.3‑1 Screen-3: Add borrower book

**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 3.2‑1 Screen element of Login form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | Delete | Button | ------ | ------- | Delete button is delete your product. |
| **2** | Back | Button | ------ | ------ | Back button is navigate to the previous slide. |
| **3** | Product Detail | Box | ------ | ------ | Show the product image, product name and price. |
| **4** | Item Increase(+) | Button | ------ | ----- | Increase the item. |
| **5** | Item Decrease  (-) | Button | ------ | ------ | Decrease the item. |
| **4** | Proceed to checkout | Button | ------ | ------ | This button is navigate to the payment screen. |

# Database design

## List of Tables

* Student
* Course
* Staff
* Faculty
* Document

Table 4.1‑1 Table: Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **StudentID** | int | NN | PK (Auto Increment) |  |
| **StudentName** | varchar(100) | NN |  |  |
| **Email** | varchar(100) | AN |  |  |
| **Age** | int | AN |  |  |
| **Address** | varchar(100) | AN |  |  |
| **ScholarshipID** | int | NN | FK | Reference of scholarship detail |
| **CourseID** | int | NN | Fk | Reference of Course. |

Table 4.1‑2 Table: Course

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **CourseID** | int | NN | PK (Auto Increment) |  |
| **CourseName** | varchar(100) | NN |  |  |
| **CourseDetail** | Varchar(100) | AN |  |  |
| **FacultyID** | int | AN | FK | Reference of Faculty |

Table 4.1‑3 Table: Staff

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **StaffID** | int | NN | PK (Auto Increment) |  |
| **StaffName** | varchar(100) | AN |  |  |
| **IsAdmin** | Boolean | AN |  |  |
| **Designation** | varchar(100) | AN |  |  |
| **StudentID** | int | NN | FK | Reference of student |

Table 4.1‑4 Table: Faculty

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **FacultyID** | int | NN | PK (Auto Increment) |  |
| **FacultyName** | varchar(100) | AN |  |  |
| **Email** | varchar(100) | AN |  |  |
| **Department** | varchar(100) | AN |  |  |
| **Contact** | number(10,0) | AN |  |  |

Table 4.1‑5 Table: Document

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **DocumentID** | int | NN | PK (Auto Increment) |  |
| **DocumentDetail** | varchar(100) | AN |  |  |
| **StudentID** | 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. |

### Scenario# A1.3

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***A1.3*** | **:** | Adding a New Student with Invalid 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 student information, such as an invalid age or incorrect contact number format, is provided.  **When**: The Admission Officer selects the "Add New Student" option  And enters invalid student details  And clicks the "Save" button to add the student to the admission database.  **Then**: The system displays error messages indicating the invalid fields, and the student is not added to the database until valid information is provided. |

### Scenario# A1.4

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***A1.4*** | **:** | Adding a New Student with Duplicate 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. The information provided matches an existing student record in the database (e.g., same name and birthdate).  **When**: The Admission Officer selects the "Add New Student" option  And enters student details that match an existing record  And clicks the "Save" button to add the student to the admission database.  **Then**: The system displays a warning about the duplicate entry and prompts the Admission Officer to review or confirm the addition. |

### Scenario# A1.5

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***A1.5*** | **:** | Adding a New Student After Session Timeout |
| **Prerequisite** | **:** | Admission Officer is logged in to the admission management system, and the session is active. |
| **Acceptance Criteria** | **:** | **Given**: The Admission Officer is navigated to the student admission management page but takes too long to enter student information, causing the session to time out.  **When**: The Admission Officer selects the "Add New Student" option  And begins entering student details  But the session times out before clicking the "Save" button.  **Then**: The system logs out the Admission Officer due to inactivity, displays a session timeout message, and requires the Admission Officer to log in again to continue adding the student. |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Name:** | **Admission Management** | **Test Designed by:** | 1. K. Nadiyapara |
| **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** | Verify accessibility of Admission Management System dashboard | Functional | Verify that the dashboard of the Admission Management System is accessible | TC\_001 |
| **TC\_002** | Verify student application form accessibility and elements | GUI | Verify the availability and accessibility of all elements in the student form | TC\_002 |
| **TC\_003** | Submit student application with valid data | Functional | Verify that the student application form can be submitted with valid data and that the system handles the submission correctly | TC\_003 |

|  |  |
| --- | --- |
| **Test Case Title** | Verify accessibility of Admission Management System dashboard |
| **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 Admission Management System URL | The site should launch properly | Site launched successfully | Pass |  | https://admission.system.com/dashboard |  |
| 2 | Navigate to dashboard | User should be able to access the dashboard without login | Dashboard accessed without login | Pass |  |  |  |
| 3 | Verify the loading time of the dashboard | Dashboard should load within 5 seconds | Dashboard loaded within 5 seconds | pass |  | Password: rbgondaliya |  |
| 4 | Verify that the dashboard is responsive on different devices (mobile, tablet, desktop) | Dashboard should be responsive and display correctly on all devices | Dashboard displayed correctly on all devices | Pass | Tested on mobile and tablet |  |  |
| 5 | Verify all dashboard elements (charts, menus, navigation links) are loaded correctly | All elements should load and be functional without errors | All elements loaded correctly | pass |  |  |  |

|  |  |
| --- | --- |
| **Test Case Title** | Verify student application form accessibility and elements |
| **Test Type** | GUI |
| **Test Priority** | HIGH |
| **Pre-condition** | The web application must be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Access the student application form from the menu | The form should open and be accessible | Form opened successfully | Pass |  |  |  |
| 2 | Verify all form fields (name, date of birth, address, etc.) are available | All fields should be displayed and editable | All fields displayed correctly | Pass |  |  |  |
| 3 | Verify "Submit" button exists | "Submit" button should be present | Submit button visible | Pass |  |  |  |
| 4 | Verify form field validation messages (if any) | Appropriate validation messages should appear for required fields if left empty | Validation messages appeared correctly | Pass | Tested by leaving fields empty |  |  |

|  |  |
| --- | --- |
| **Test Case Title** | Submit student application with valid data |
| **Test Type** | Functional |
| **Test Priority** | HIGH |
| **Pre-condition** | Web application and student form must be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Enter valid student data in all required fields (name, address, etc.) | Form should accept the valid data | Form accepted all inputs | Pass |  |  |  |
| 2 | Verify that the submitted data is saved correctly in the database | Data should be stored in the database accurately | Data saved correctly in the database | Pass | Verified by checking database entries |  |  |
| 3 | Verify that a confirmation email is sent to the applicant upon successful form submission | Applicant should receive a confirmation email | Confirmation email received successfully | Pass |  |  |  |
| 4 | Click on the "Submit" button | Data should be submitted, and confirmation message should appear | Data submitted successfully with confirmation message | Pass |  |  |  |

# References

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