–



Darshan University

A Project Report on

**“Admission Management System”**

Under the subject

**Software Engineering (2301CS405)**

B. Tech, Semester – IV

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name: Bhoomi Tulsiyani | Enrolment No.: 23010101275 |
| Academic Year  (2024-2025) | |
| Internal Guide  Prof. R. B. Gondaliya  Darshan University | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

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

**DECLARATION**

We hereby declare that the SRS, submitted along with the **Software Engineering** **(2301CS405)** for entitled **“Admission Management System”** submitted in partial fulfilment for the Semester-4 of **Bachelor Technology (B. Tech)** in **Computer Science and Engineering (CSE)** Departmentto Darshan University, Rajkot, is a record of the work carried out at **Darshan University, Rajkot** under the supervision of R. B. Gondaliya and that no part of any of report has been directly copied from any students’ reports, without providing due reference.

(Bhoomi Tulsiyani)

Student’s Signature

Date: 27/12/2024

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

**CERTIFICATE**

This is to certify that the SRS on **“Admission Management System” has** been satisfactorily prepared by **Bhoomi Tulsiyani** (**23010101275**) under my guidance in the fulfillment of the course **Software Engineering (2301CS405)** work during the academic year 2024-2025.

|  |  |  |
| --- | --- | --- |
| Internal Guide  Prof. R. B. Gondaliya  Darshan University |  | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

**Acknowledgement**

I wish to express my sincere gratitude to my project guide Prof. R. B. Gondaliyaand all the faculty members for helping me through my project by giving me the necessary suggestions and advices along with their valuable co- ordination in completing this work.

I also thank my parents, friends and all the members of the family for their precious support and encouragement which they had provided in completion of my work. In addition to that, I would also like to mention the Darshan University personals who gave me the permission to use and experience the valuable resources required for the project from the University premises.

Thus, in conclusion to the above said, I once again thank the faculties and members of **Darshan University** for their valuable support in completion of the project.

Thanking You

**Bhoomi Tulsiyani**

**ABSTRACT**

The Admission Management System is a detailed solution designed to digitalize the admission process in educational institutions. This system aims to replace traditional, paper-based methods with an efficient, computerized platform that simplifies the management of admission-related tasks. This system provides features such as user login for students, faculty and admin, enabling secure access to information. Students can fill out admission forms online, upload required documents, and track the status of their application. The system allows admin to manage the entire admission process. Admins can add, view, update, and delete student records and faculty and monitor application progress. It also includes a facility for faculties to conduct tests, communicate with applicants, and address issues.

The primary goal of this system is to reduce manual workload, enhance accuracy, and improve the overall efficiency of the admission process, making it a seamless experience for both students and administrators.

**Table of Contents**

[List of Figures I](#_Toc188285759)

[List of Tables II](#_Toc188285760)

[1 Introduction 1](#_Toc188285761)

[1.1 Product perspective 1](#_Toc188285762)

[1.2 Product features 1](#_Toc188285763)

[1.2.1 There are three different users who will be using this product: 1](#_Toc188285764)

[1.2.2 The features that are required for the Admin are: 1](#_Toc188285765)

[1.2.3 The features that are required for the Faculty are: 1](#_Toc188285772)

[1.2.4 The features that are required for the Student are: 1](#_Toc188285773)

[1.3 Functional Requirement 1](#_Toc188285774)

[1.3.1 Admin 1](#_Toc188285775)

[1.3.2 Faculty 2](#_Toc188285776)

[1.3.3 Student 2](#_Toc188285783)

[1.4 Non-Functional Requirement 2](#_Toc188285784)

[1.4.1 Usability: 2](#_Toc188285785)

[1.4.2 Accuracy: 2](#_Toc188285786)

[1.4.3 Availability: 2](#_Toc188285787)

[1.4.4 Maintainability: 2](#_Toc188285788)

[2 Design and Implementation Constraints 3](#_Toc188285789)

[2.1 Use case diagram 3](#_Toc188285790)

[2.2 Activity diagram and Swimlane diagram 4](#_Toc188285791)

[2.3 Sequence diagram 8](#_Toc188285792)

[2.4 State diagram 10](#_Toc188285793)

[2.5 Class diagram 11](#_Toc188285794)

[2.6 Data flow diagram 12](#_Toc188285795)

[2.6.1 Context diagram (level-0) 12](#_Toc188285796)

[2.6.2 DFD Level-1 13](#_Toc188285797)

[2.6.3 DFD Level-2 14](#_Toc188285798)

[3 External interface requirement (Screens) 15](#_Toc188285799)

[3.1 Screen-1: Prepare Merit List 15](#_Toc188285800)

[3.2 Screen-2: Cancel Admission 16](#_Toc188285801)

[3.3 Screen-3: Fees Payment 17](#_Toc188285802)

[3.4 Screen-4: Track Application Status 18](#_Toc188285803)

[3.5 Screen-5: Check Eligibility of Application 19](#_Toc188285804)

[4 Database design 20](#_Toc188285805)

[4.1 List of Tables 20](#_Toc188285806)

[5 Stories and Scenario 22](#_Toc188285807)

[5.1 Story-1: Add New Book in Library Catalogue 22](#_Toc188285808)

[5.1.1 Scenario# S1.1 22](#_Toc188285809)

[5.1.2 Scenario# S1.2 22](#_Toc188285810)

[5.1.3 Scenario# S1.3 22](#_Toc188285811)

[5.2 Story-2: Search Book 23](#_Toc188285812)

[5.3 Story-3: Manage due date for borrowed book 23](#_Toc188285813)

[5.4 Story-4: Renew book 23](#_Toc188285814)

[5.5 Story-5: Generate a report on book usage and availability 23](#_Toc188285815)

[6 Test cases 24](#_Toc188285816)

[7 References 28](#_Toc188285817)

# List of Figures

[Figure 2.1‑1 Use case diagram for Admission management system 3](#_Toc146706848)

[Figure 2.2‑1 Activity diagram for Apply for Admission 4](#_Toc146706849)

[Figure 2.2‑2 Activity diagram for Add Course 5](#_Toc146706849)

[Figure 2.2‑3 Swimlane diagram for Apply for Admission 6](#_Toc146706850)

[Figure 2.2‑4 Swimlane diagram for Add Course 7](#_Toc146706849)

[Figure 2.3‑1 Sequence diagram for Apply for Admission 8](#_Toc146706851)

[Figure 2.3‑2 Sequence diagram for Add Course 9](#_Toc146706851)

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

[Figure 2.4‑2 State diagram for Admission form 10](#_Toc146706853)

[Figure 2.5‑1 Class diagram for Admisison management system 11](#_Toc146706854)

[Figure 2.6‑1 Context diagram for Admission management system 12](#_Toc146706855)

[Figure 2.6‑2 DFD level-1 for Admission management system 1](#_Toc146706856)3

[Figure 2.6‑3 DFD level-2 for Admission management system 1](#_Toc146706857)4

[Figure 3.1‑1 Screen-1: Prepare Merit List 1](#_Toc146706858)5

[Figure 3.2‑1 Screen-2: Cancel Admission 1](#_Toc146706860)6

[Figure 3.3‑1 Screen-3: Fees Payment 1](#_Toc146706860)7

[Figure 3.4‑1 Screen-4: Track Application Status 1](#_Toc146706860)8

[Figure 3.5‑1 Screen-5: Check Eligibility of Application 1](#_Toc146706860)9

# List of Tables

[Table 3.1‑1 Screen element of Prepare Merit List 15](#_Toc146706861)

[Table 3.2‑1 Screen element of Cancel Admission 16](#_Toc146706862)

[Table 3.3‑1 Screen element of Fees Payment 17](#_Toc146706863)

[Table 3.4‑1 Screen element of Track Application Status 18](#_Toc146706863)

[Table 3.5‑1 Screen element of Check Eligibility of Application 1](#_Toc146706863)9

[Table 4.1‑1 Table: Course](#_Toc146706864) 20

[Table 4.1‑2 Table: Application](#_Toc146706865) 20

[Table 4.1‑3 Table: Fees 20](#_Toc146706866)

[Table 4.1‑4 Table: Student 20](#_Toc146706866)

[Table 4.1‑5 Table: Department](#_Toc146706867) 20

# Introduction

## Product perspective

This project focuses on transforming the traditional, manual admission process into an internet-based application to enhance efficiency and accessibility for users. The system allows applicants to manage their admission-related activities, such as submitting applications and tracking their status. It is a multi-user platform that supports both students and faculties. It efficiently handles fundamental functions such as student registration, document verification, application tracking, and fee management. Designed to meet the needs of educational institutions, this system provides a reliable solution to manage the entire admission process.

## Product features

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

* Admin who will be acting as the administrator.
* Faculty who will be reviewing student applications, managing student details, or providing recommendations.
* Students who will apply for admission

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

### Manage the admission process, including application deadlines and schedules.

### Add, edit, and delete student records in the system.

### View and process submitted admission applications.

### Manage user accounts for students and faculty.

### Handle queries or requests submitted by students.

### Add, edit or delete course offered for admission.

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

* Review and evaluate student applications assigned to them.
* Provide personalized recommendations for students.
* Access student details.
* Communicate with students regarding admission requirements or feedback.

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

* Register and create an account in the system.
* Fill out and submit admission application online.
* Upload required documents.
* View the status of their application and receive notifications or updates.
* Submit queries or requests related to admissions.

## Functional Requirement

### Admin

* Add Course: Admin can add new courses for admission.
* Delete Course: Admin can remove courses that are no longer offered.
* Get Details: Admin can view specific details about courses or users.
* Update Cut-off: Admin can update the admission cut-off criteria for courses.
* Get Fees Online: Admin manages the collection of fees online.
* Add Faculty: Admin can add new faculty members.
* Remove Faculty: Admin can remove faculty members.
* Respond to Requests: Admin handles and responds to queries or requests from students or faculty.
* Cancel Admission: Admin can cancel a student’s admission if needed.
* Prepare Merit List: Admin and Faculty generate a merit list based on eligibility and criteria.

### Faculty

## Give Personalized Suggestions: Faculty can provide personalized advice to students regarding admission.

## Contact Student: Faculty can communicate directly with students.

## Conduct Tests: Faculty can organize and manage admission tests.

## Report Issue: Faculty can report issues to the admin.

## Review Application: Faculty evaluates student applications.

## Check Eligibility of Applications: Faculty verifies if applications meet eligibility criteria.

### Student

* Login: Students can log in to access the system.
* Apply for Courses: Students can apply for any course.
* View Courses: Students can browse available courses.
* Update Profile: Students can edit and update their profile details.
* Fill Admission Form: Students submit the admission form for enrolment.
* Search for Eligibility: Students can check their eligibility for specific courses.
* Upload Documents: Students can upload required documents for verification.
* Pay Fees Online: Students can make fee payments online.
* Track Application Status: Students and faculty both can check their application status.

## 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 ‑ Use case diagram for admission management system

## Activity diagram and Swimlane diagram

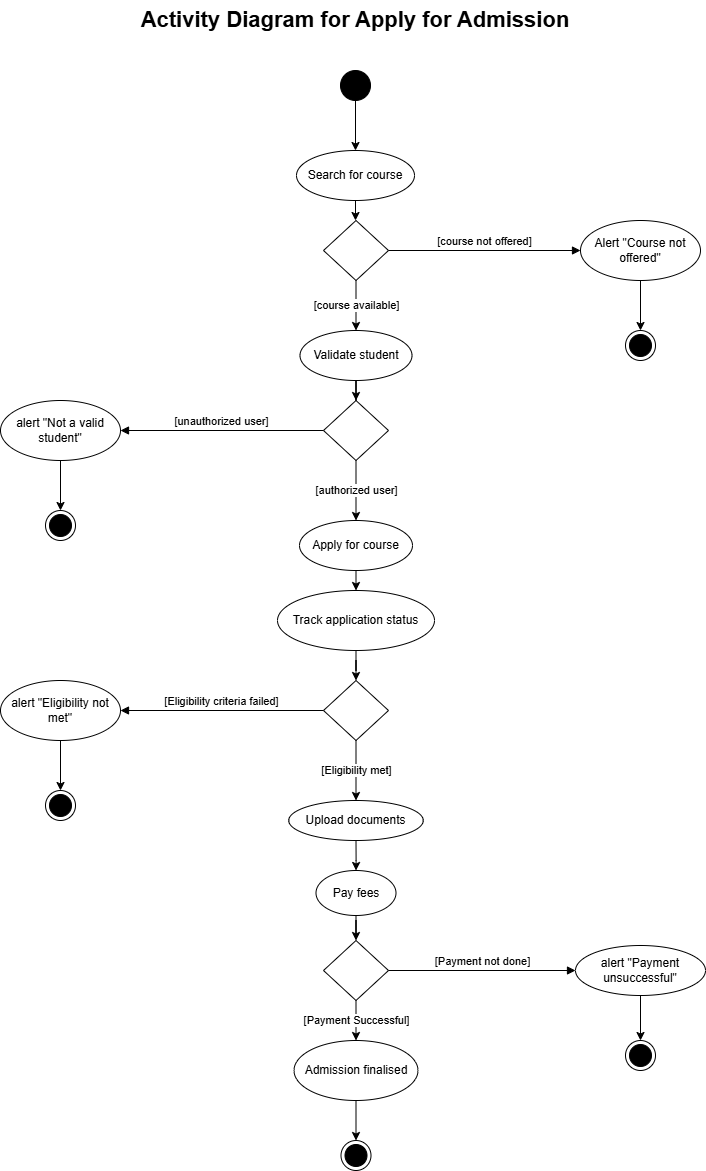


Figure ‑ Activity diagram for Apply for Admission

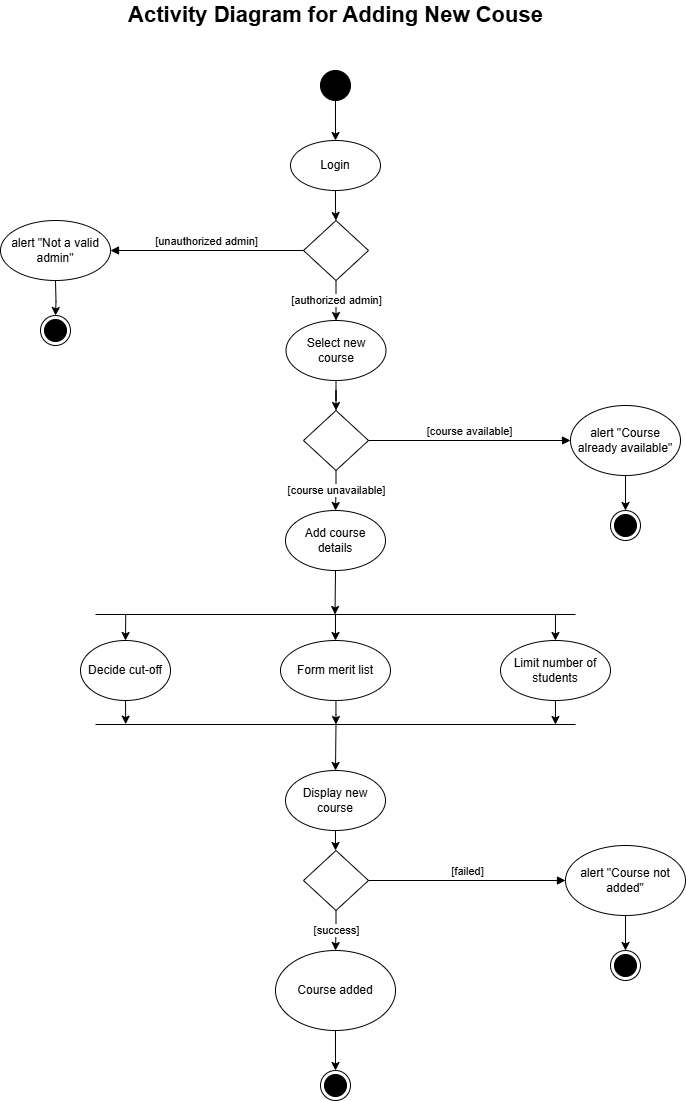


Figure ‑2 Activity diagram for Add Course

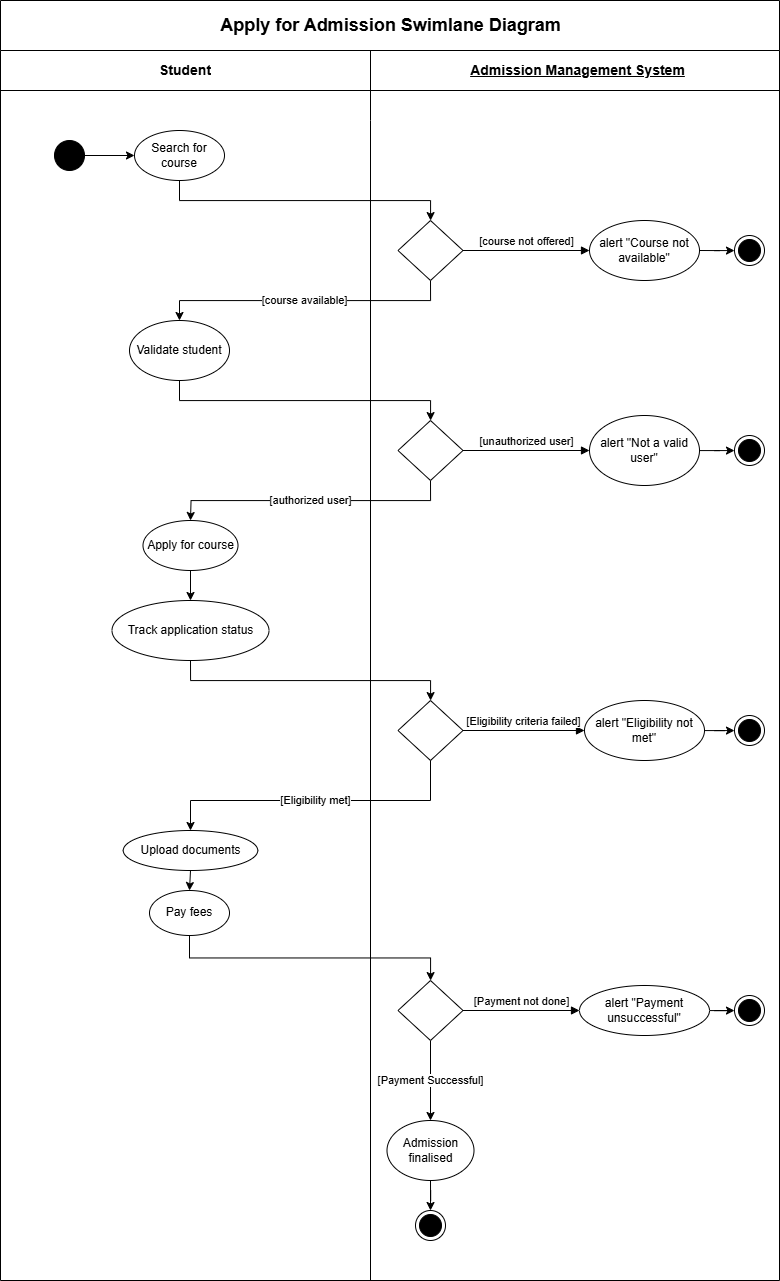


Figure ‑3 Swimlane diagram for Apply for Admission

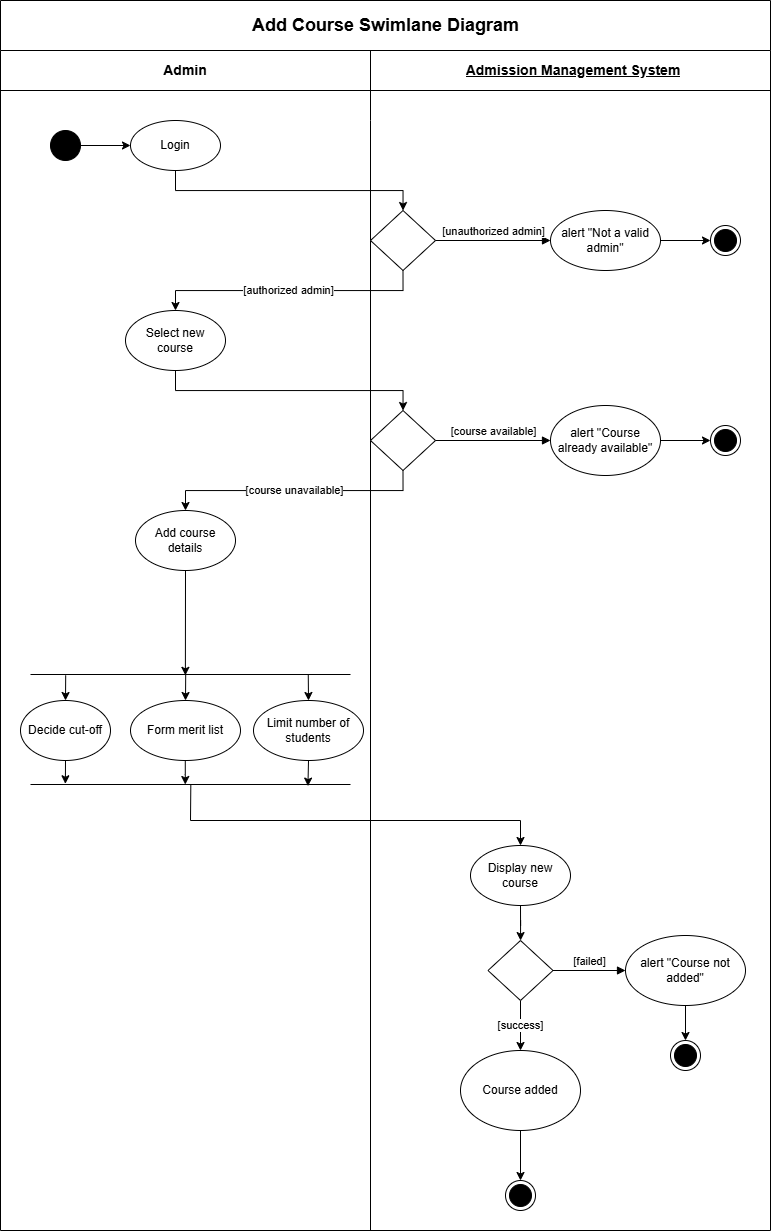


Figure ‑4 Swimlane diagram for Add Course

## Sequence diagram

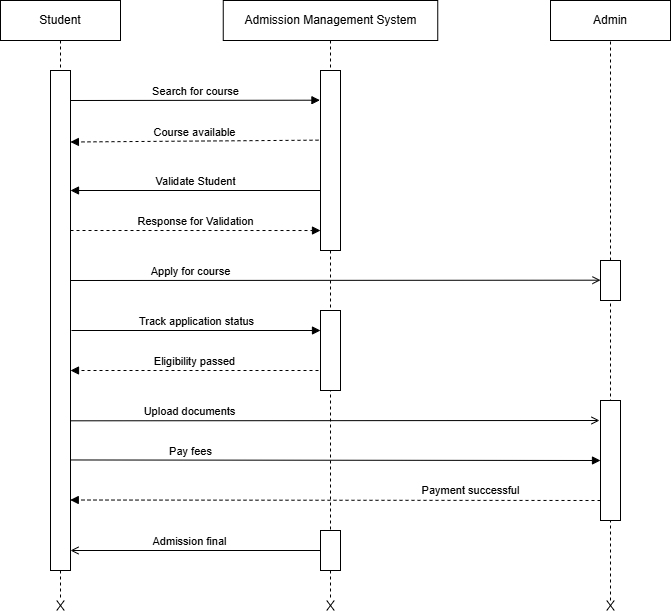


Figure ‑ Sequence diagram for Apply for Admission

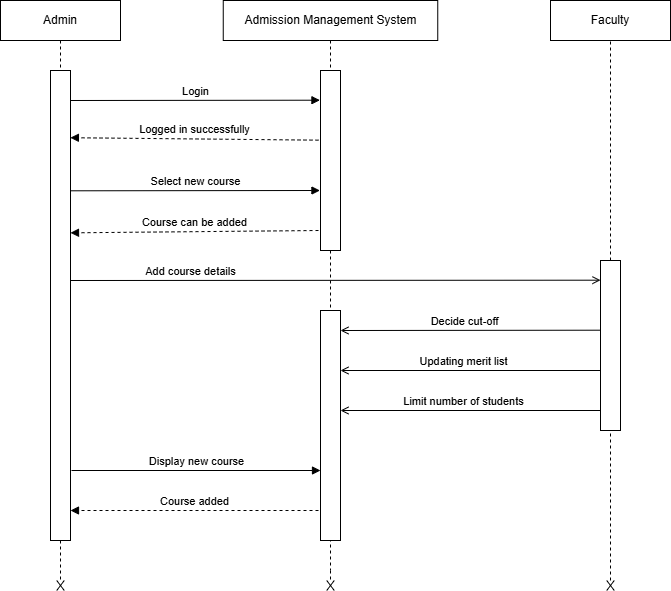


Figure ‑2 Sequence diagram for Add Course

## State diagram

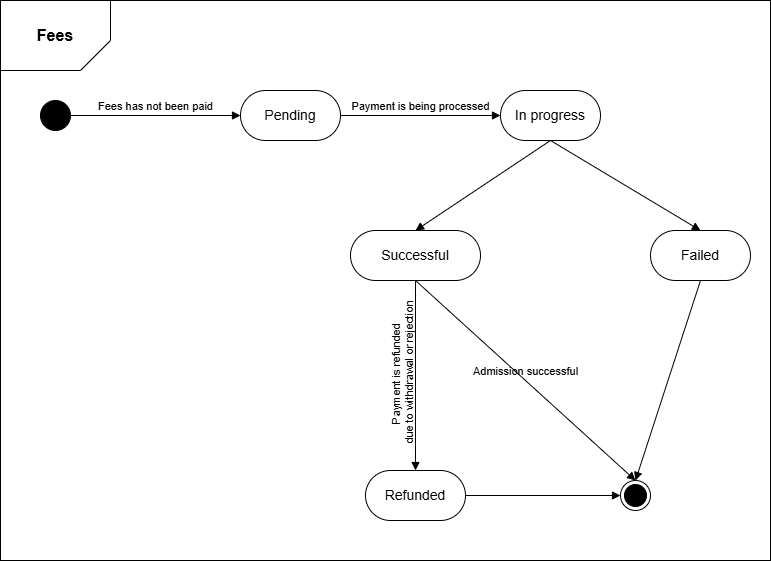


Figure ‑ State diagram of Fees

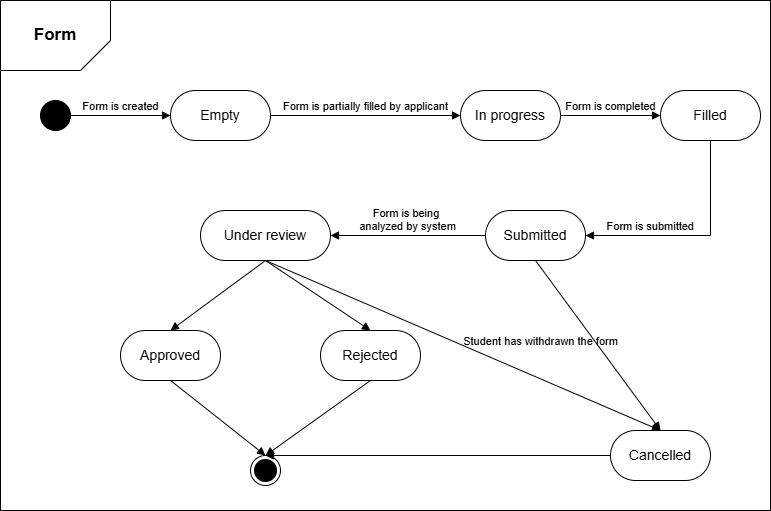


Figure ‑ State diagram for Admission Form

## Class diagram

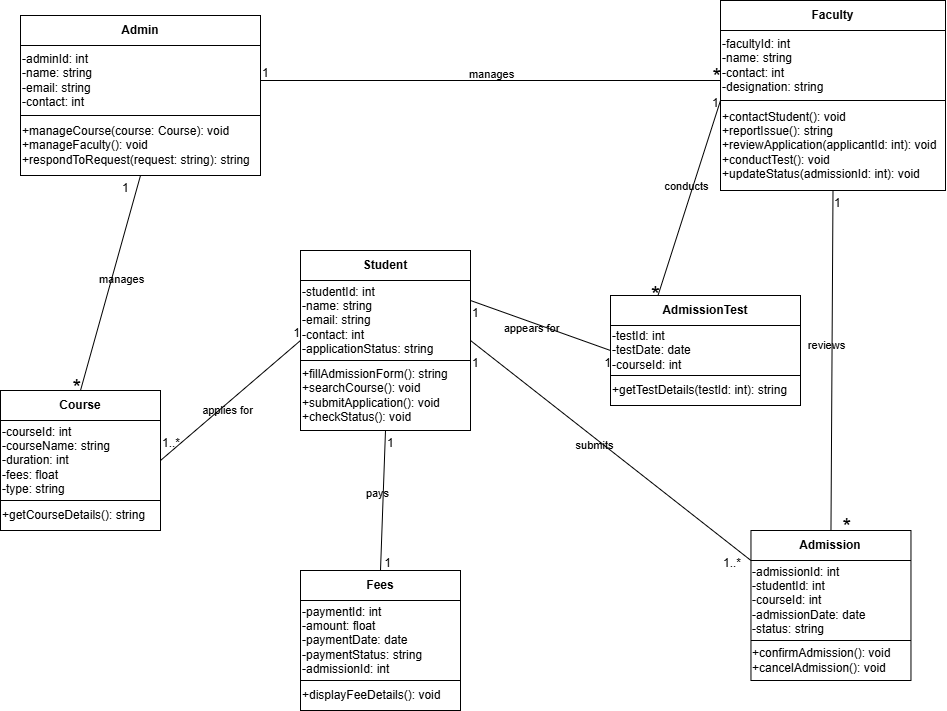


Figure ‑ Class diagram for Admission management system

## Data flow diagram

### Context diagram (level-0)

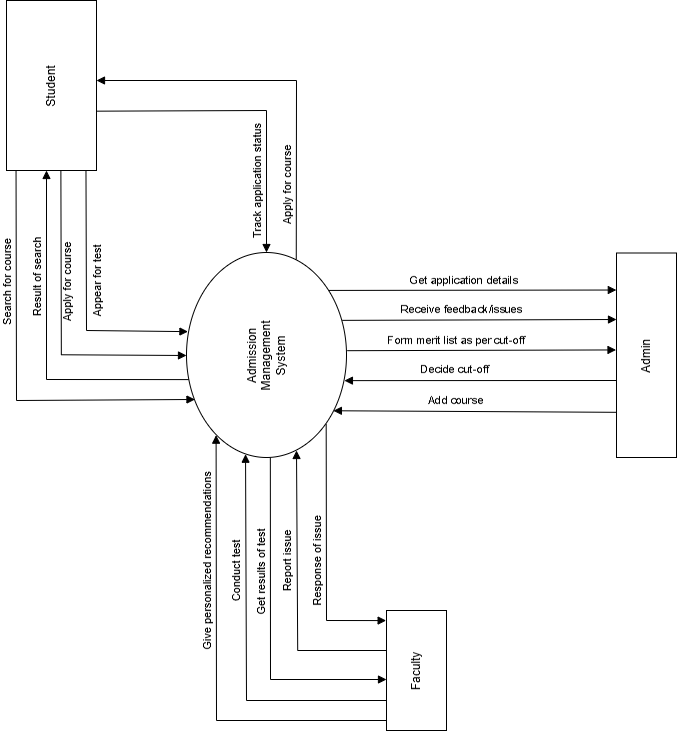


Figure ‑ Context diagram for Admission management system

### DFD Level-1

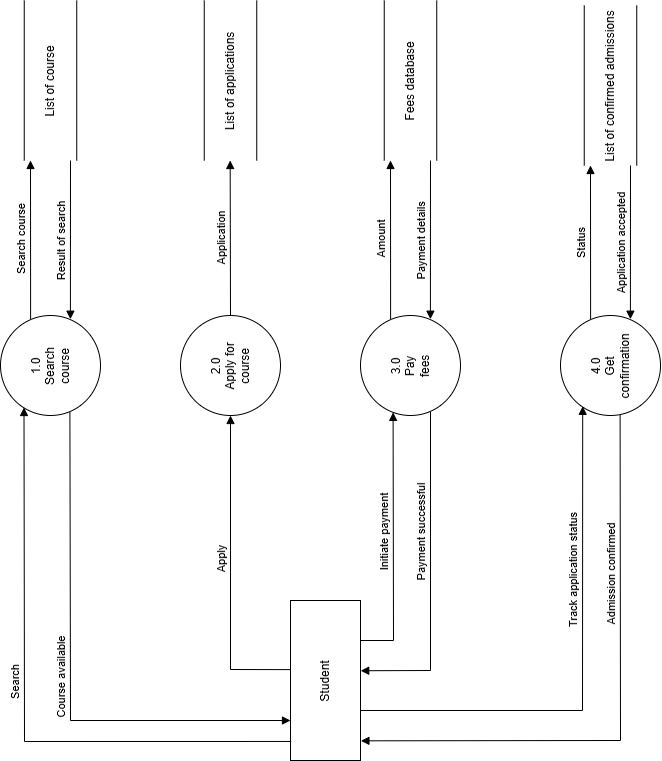


Figure ‑ DFD level-1 for Admission management system

### DFD Level-2

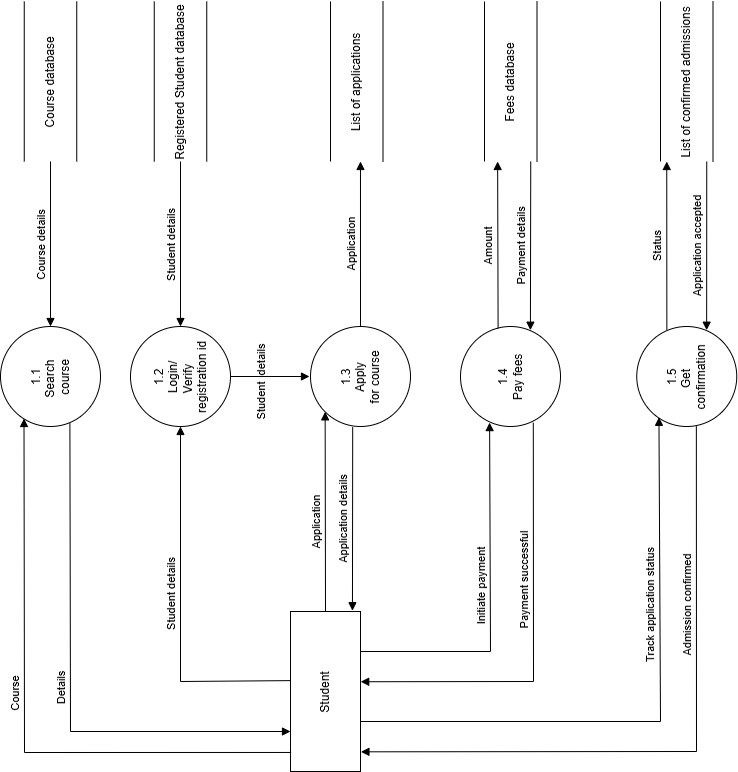


Figure ‑ DFD level-2 for Admission management system

# External interface requirement (Screens)

## Screen-1: Prepare Merit List

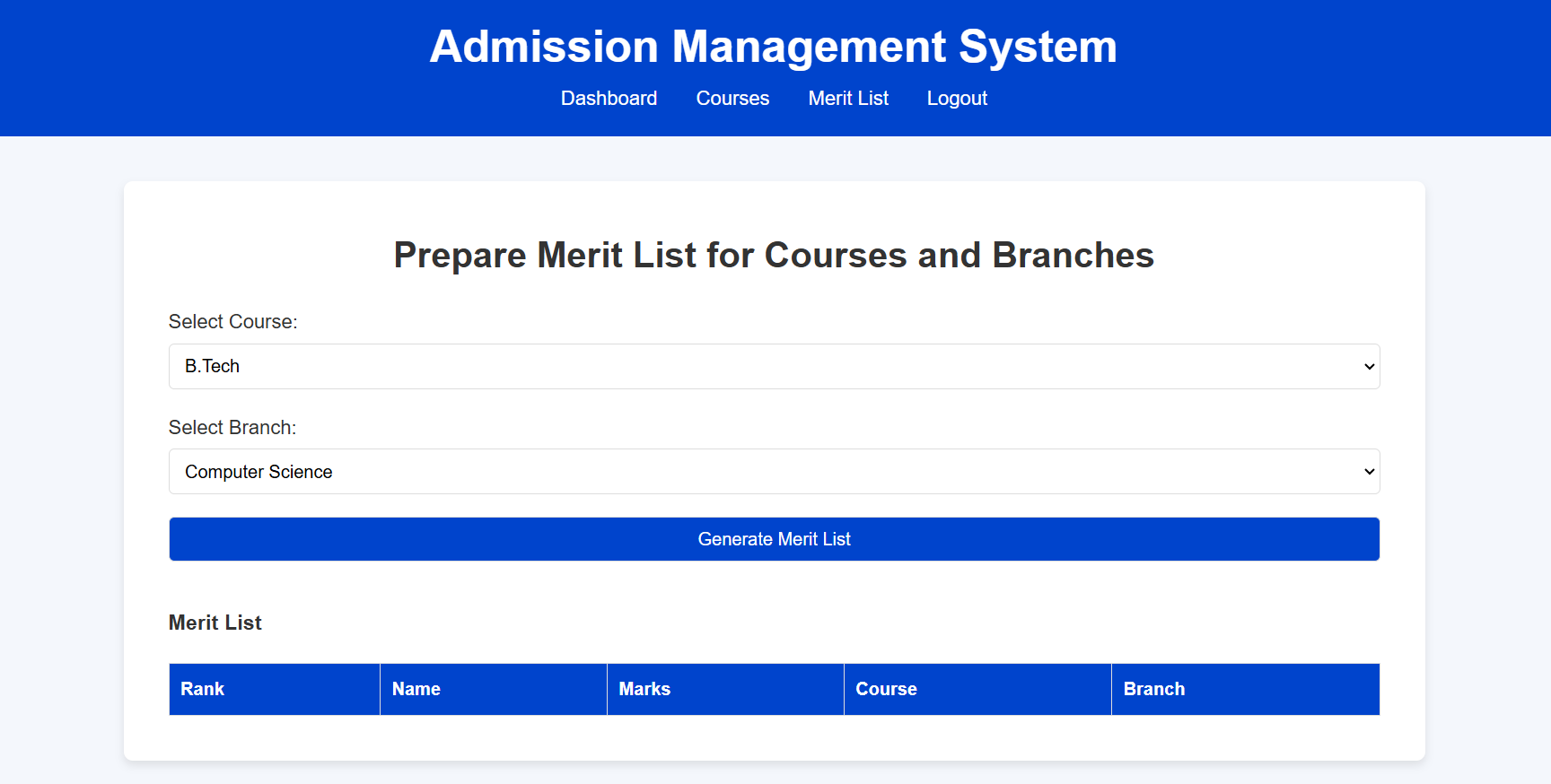


Figure ‑ Screen-1: Prepare Merit List

**Purpose:** This form will allow the admin to prepare merit list in the system for different courses and branches. To apply, the following information will be encoded in the system.

Table ‑ Screen element of Prepare Merit List

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Select Course | Dropdown | M | 1 | Course field should be selected from the dropdown. |
| 2 | Select Branch | Dropdown | M | 1 | Branch field should be selected from the dropdown. |
| 3 | Generate Merit list | Button | ------ | ------ | Generate Merit List is a button to store the criteria for merit list. |

## Screen-2: Cancel Admission

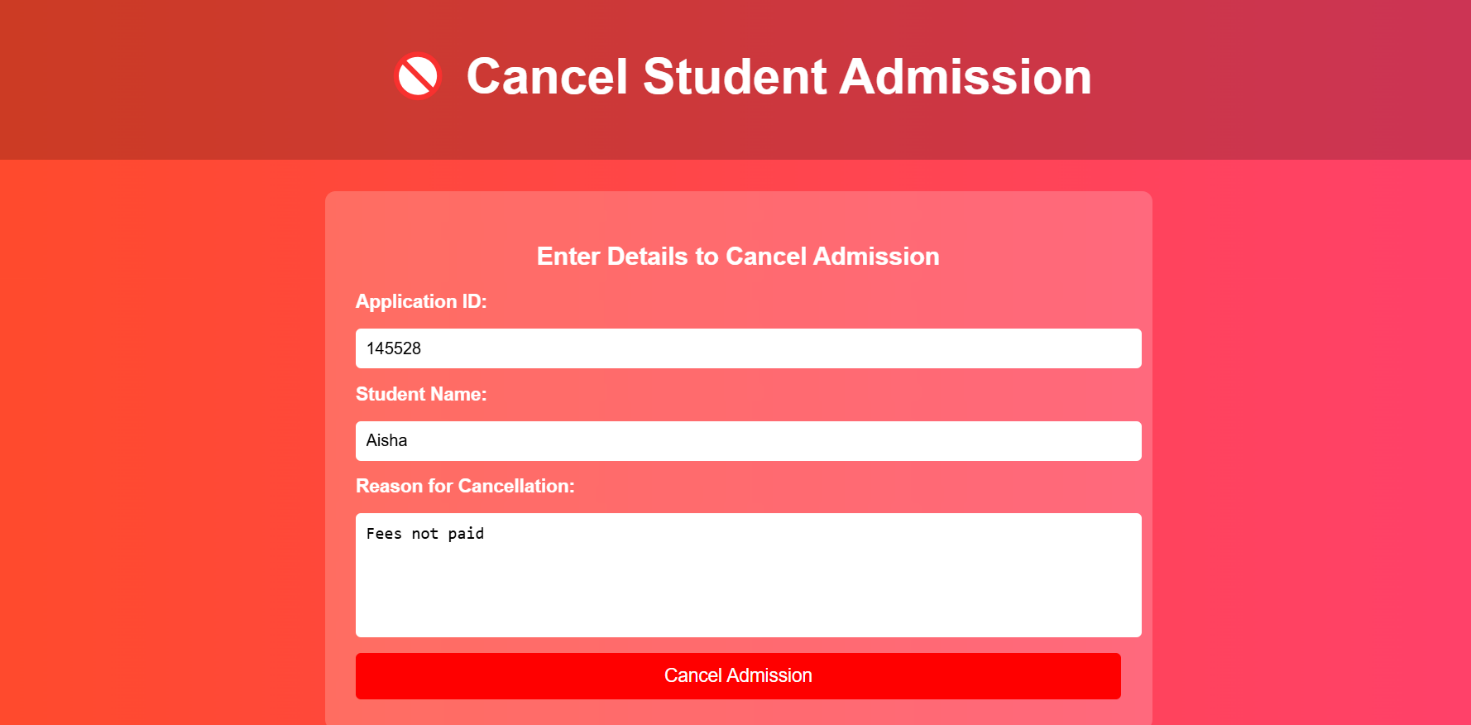


Figure ‑ Screen-2: Cancel Admission

**Purpose:** This form will be used by the admin to cancel admission of students under exceptional circumstances.

Table ‑ Screen element of Cancel Admission

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Application ID | Textbox | M | 1 | Application ID field should be editable and accept the Application ID. |
| 2 | Student Name | Textbox | M | 1 | Name field should be editable and accept the name with proper format. |
| 3 | Reason for Cancellation | Textbox | M | 1 | Reason for cancelling admission must be written. |
| 4 | Cancel Admission | Button | ------ | ------ | This button confirms the cancelling of admission by admin. |

## Screen-3: Fees Payment

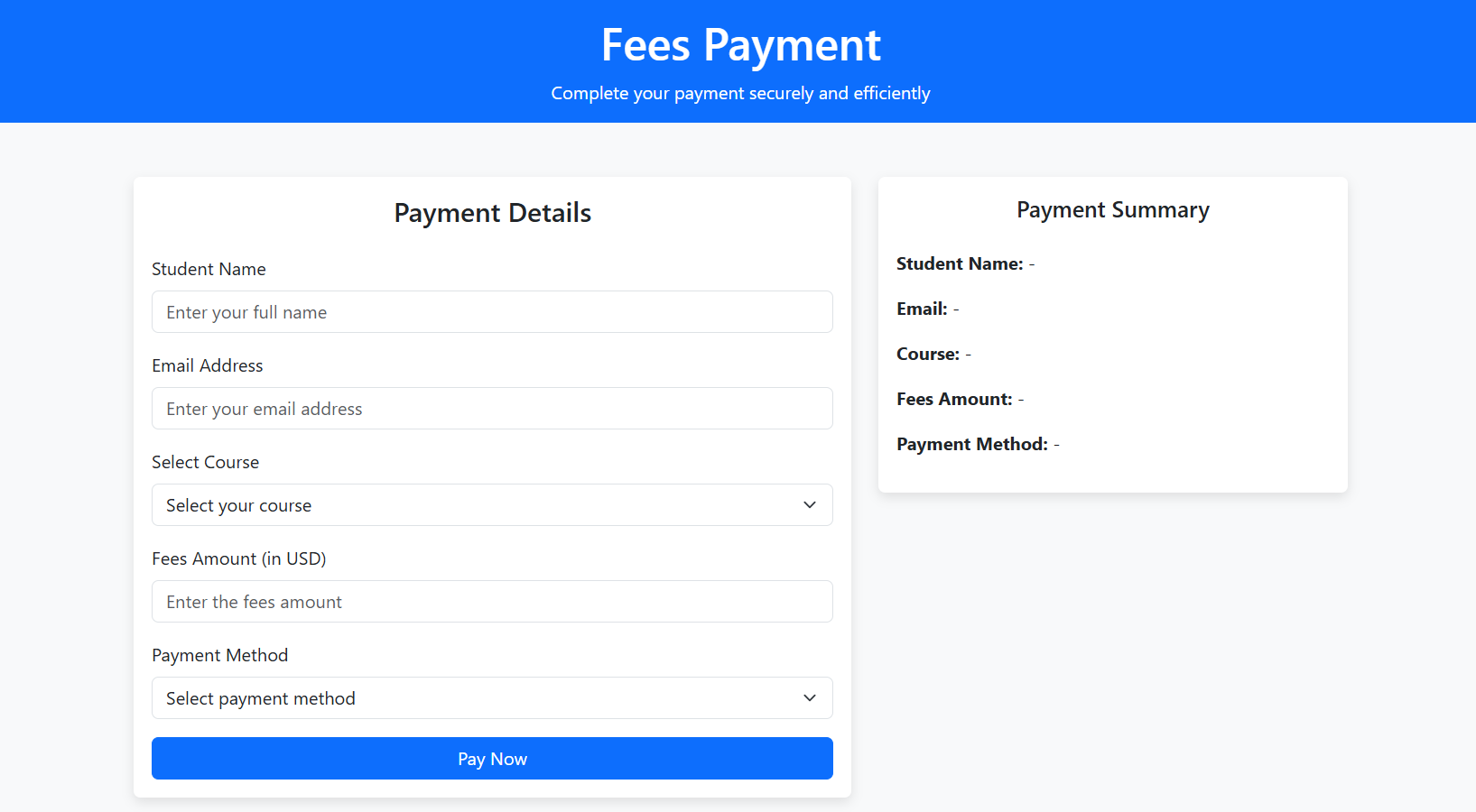


Figure ‑ Screen-2: Fees Payment

**Purpose:** This form will be used by the students to pay fees online securely.

Table ‑ Screen element of Fees Payment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Student Name | Textbox | M | 1 | Name field should be editable and accept the student name. |
| 2 | Email Address | Textbox | M | 1 | Email field should be editable and accept the email with proper format. |
| 3 | Select Course | Dropdown | M | 1 | Course can be selected from the dropdown showing all course options available. |
| 4 | Fees Amount | Textbox | M | 1 | Fees amount will automatically be filled when course is selected. |
| 5 | Payment Method | Dropdown | M | 1 | Payment method can be selected from the dropdown menu showing all options. |
| 6 | Pay Now | Button | ------ | ------ | This button confirms payment by user. |
| 7 | Payment Summary | ------ | ------ | ------ | This shows the details of payment summary by the customer. |

## Screen-4: Track Application Status

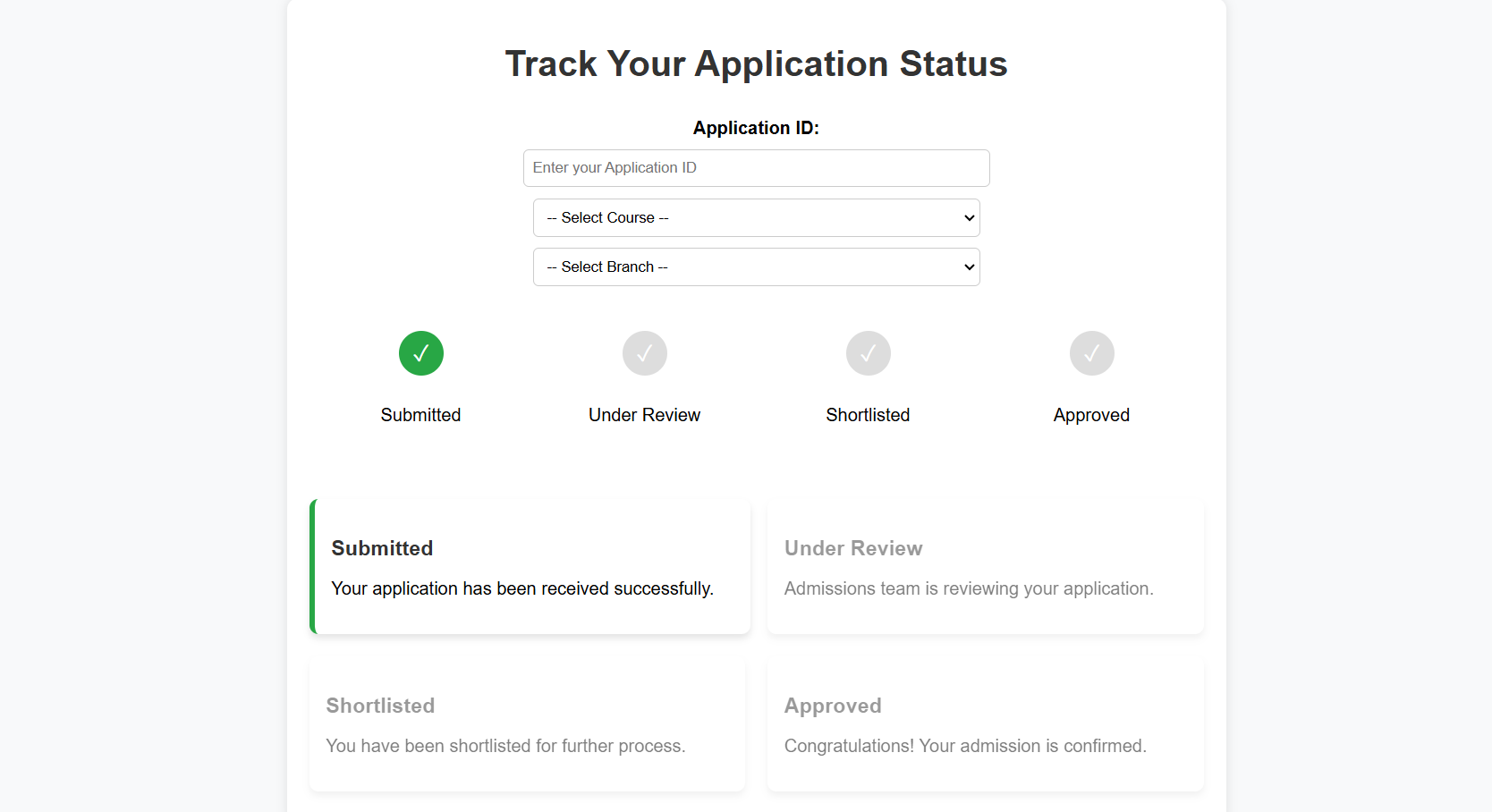


Figure ‑ Screen-3: Track Application Status

**Purpose:** This module will allow the student to track their application status.

Table ‑ Screen element of Track Application Status

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Application ID | Textbox | M | 1 | Application ID field should be editable. |
| 2 | Select Course | Dropdown | M | 1 | Course field should be selected from the dropdown. |
| 3 | Select Branch | Dropdown | M | 1 | Branch field should be selected from the dropdown. |

## Screen-5: Check Eligibility of Applications

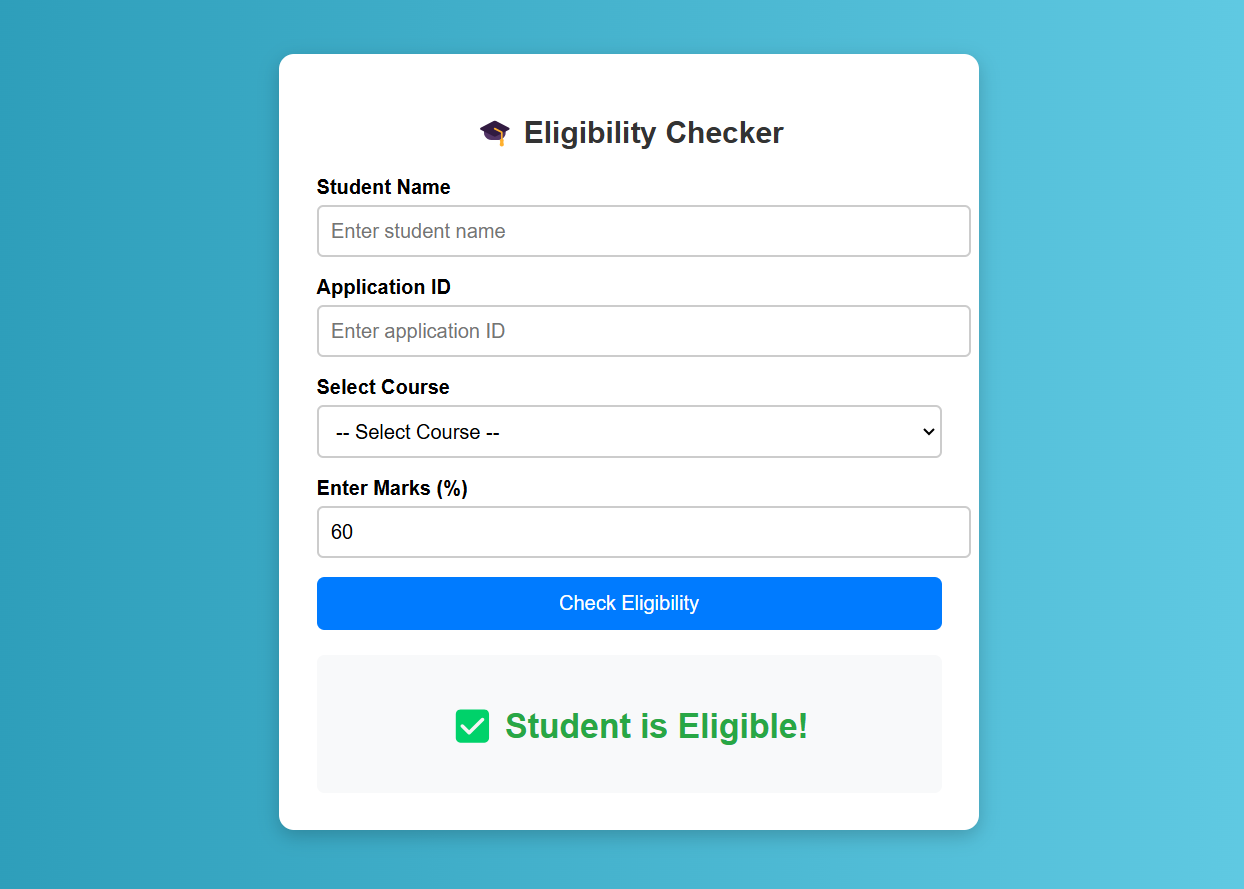


Figure ‑ Screen-3: Check Eligibility of Application

**Purpose:** This module will allow faculty to check eligibility of application of students.

Table ‑ Screen element of Check Eligibility of Application

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Student Name | Textbox | M | 1 | Name field should be editable. |
| 2 | Application ID | Textbox | M | 1 | Application ID field should be editable. |
| 3 | Select Course | Dropdown | M | 1 | Course field should be selected from the dropdown. |
| 4 | Enter Marks | Textbox | M | 1 | Marks of student should be entered here to check eligibility. |
| 5 | Check Eligibility | Button | ----- | ----- | This button checks the marks entered as per eligibility criteria of that course and displays result below. |

# Database design

## List of Tables

* Course
* Application
* Fees
* Student
* Department

Table ‑ Table: Course

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| CourseID | int | NN | PK (Auto Increment) |  |
| CourseName | varchar(100) | AN |  |  |
| DepartmentID | varchar(100) | NN | FK | Reference of Department Table |
| Credits | int | AN |  |  |
| StudentID | int | NN | FK | Reference of Student Table |

Table ‑ Table: Application

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| ApplicationID | int | NN | PK (Auto Increment) |  |
| StudentID | int | NN | FK | Reference of Student Table |
| CourseID | int | NN | FK | Reference of Course Table |
| ApplicationDate | DateTime | AN |  |  |

Table ‑ Table: Fees

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| FeesID | int | NN | PK (Auto Increment) |  |
| StudentID | int | NN | FK | Reference of Student Table |
| CourseID | int | NN | FK | Reference of Course Table |
| FeesStatus | varchar(100) | AN |  |  |
| TransactionDate | DateTime | NN |  |  |

Table ‑ Table: Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| StudentID | int | NN | PK (Auto Increment) |  |
| StudentName | varchar(100) | AN |  |  |
| Phone | int | AN |  |  |
| DOB | DateTime | AN |  |  |
| CourseID | int | NN | FK | Reference of Course Table |

Table 4.1‑5 Table: Department

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| DepartmentID | int | NN | PK (Auto Increment) |  |
| DepartmentName | varchar(100) | AN |  |  |
| CourseList | varchar(100) | NN | FK | Reference of Course Table |
| AllotedSeats | int | AN |  |  |

# Stories and Scenario

## Story-1: Add New Book in Library Catalogue

|  |  |  |
| --- | --- | --- |
| *Story # S1* | : | As a Librarian,  I want to add a new book in library catalogue  So that everyone can easily find and borrow it. |
| Priority | **:** | High |
| Estimate | **:** | XL |
| Reason | **:** | The addition of a new book to the library catalogue is crucial for ensuring that the library's collection is up-to-date and accessible to everyone. |

### Scenario# S1.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.1* | : | Adding a New Book with Valid Information |
| Prerequisite | **:** | Librarian is logged in to the Library management system. |
| Acceptance Criteria | **:** | **Given:**  The Librarian is navigated to the library catalog management page. Valid book information, including title, author, ISBN, and other relevant details is added.  **When:**  The librarian selects the "Add New Book" option  And The librarian enters valid book details  The librarian clicks the "Save" button to add the book to the catalog.  **Then t**he system successfully adds the book to the catalog and the librarian receives a confirmation message with the book's identification number. |

### Scenario# S1.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.2* | : | Adding a New Book with Invalid Information. |
| Prerequisite | **:** | The librarian is logged into the library management system. |
| Acceptance Criteria | **:** | **Given:** The librarian is on the library catalogue management page  **When:** The librarian selects the "Add New Book" option and the librarian enters an incomplete or incorrect book details and librarian clicks the "Save" button to add the book to the catalogue.  **Then t**he system displays error messages for the incorrect or missing information and the book is not added to the catalogue. |

### Scenario# S1.3

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.3* | : | Attempting to Add a Duplicate Book |
| Prerequisite | **:** | The librarian is logged into the library management system and the librarian is on the library catalogue management page |
| Acceptance Criteria | **:** | **Given**: The book information, including title, author, ISBN, and other relevant details, is available and the book with the same ISBN is already in the catalogue.  **When**: User Clicks on “Add book” button. Enter a number of copies with the same book detail mentioned in the field.  **Then**: Generate unique book id, barcode and spine label for various book of same title. |

## Story-2: Search Book

|  |  |  |
| --- | --- | --- |
| *Story # S2* | : | As a Librarian or member,  I want to search for books by title, author, or keyword,  So that I can quickly find books that match my interests. |
| Priority | **:** | High |
| Estimate | **:** | M |
| Reason | **:** | Implementing a search functionality is essential for enhancing the user experience, as it allows librarian and member to efficiently discover and access the library's resources. |

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

|  |  |  |
| --- | --- | --- |
| *Story # S3* | : | As Librarian,  I want to manage due dates for borrowed books  So that I can ensure that books are returned on time and avoid overdue fines. |
| Priority | **:** | High |
| Estimate | **:** | M |
| Reason | **:** | Proper due date management is crucial for maintaining the library's collection and ensuring that books are available for all members. |

## Story-4: Renew book

|  |  |  |
| --- | --- | --- |
| *Story # S3* | : | As Librarian,  I want to renew a book that I have borrowed  So that I can extend my borrowing period if needed. |
| Priority | **:** | Medium |
| Estimate | **:** | M |
| Reason | **:** | Book renewal functionality is a convenience feature for librarian, allowing them to keep a book for an extended period if no one else has requested it. |

## Story-5: Generate a report on book usage and availability

|  |  |  |
| --- | --- | --- |
| *Story # S3* | : | As Librarian,  I want to generate reports on library usage and book availability  So that I can make informed decisions about library’s books. |
| Priority | **:** | Medium |
| Estimate | **:** | L |
| Reason | **:** | Reporting functionality helps librarians track the usage of library resources and make data-driven decisions to improve services and collections. |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | EMI Calculator | Test Designed by: | P. U. Jadeja |
| Module Name: | **Login** | **Test Designed date:** | 01-10-2023 |
| Release Version: | **1.0** | **Test Executed by:** | **R. B. Gondaliya** |
|  |  | **Test Execution date:** | 15-01-2023 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pre-condition: Web application should be accessible | | | | |
| Test Case ID | **Test Title** | **Test Type** | **Description** | **Test Case ID** |
| TC\_001 | Login to web application with valid credential | Functional | Login to Library management system web application through valid credential | TC\_001 |
| TC\_002 | Login to web application with invalid credential | Functional | Login to Library management system web application through invalid credential | TC\_002 |
| TC\_003 | Varify login page elements | GUI | varify that all elements are availabe on login page | TC\_003 |

|  |  |
| --- | --- |
| **Test Case Title** | Login to web application with valid credential |
| **Test Type** | Functional |
| **Test Priority** | High |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **BUG ID** |
| 1 | Access Web application URL | The site launched properly | Site launched successfully | Pass |  | <https://accounts.google.com/ServiceLogin> |  |
| 2 | Enter valid Username in username field | Username field should be editable and accept the Username | Username input accepted | Pass |  | Username:  Rbgondaliya@gmail.com |  |
| 3 | Enter valid Password in Password field | Password field should be editable and accept the password and display as star or dot | Password input displayed in dot and accepted | pass |  | Password: rbgondaliya |  |
| 4 | Enter valid captcha code in captch field | Captch field should editable and accept captcha and captcha is case sensitive | Captcha input accepted | Pass | Step required when human action validation perform | get captcha from image which is near by captcha field |  |
| 5 | Click on login button | User should login into site and navigated to dashboard | User navigated to dashboard and username should br display in top of the right side. | pass |  |  |  |

|  |  |
| --- | --- |
| **Test Case Title** | Login to web application with invalid credential |
| **Test Type** | Functional |
| **Test Priority** | Medium |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Verify that User is not able to Login with invalid Username and invalid Password | Should be display an error message enter wrong username or password | Display an error of wrong username and password | Pass |  |  |  |
| 2 | Verify that User is not able to Login with Valid Username and invalid Password | Should be display an error message enter wrong password | Display an error of wrong password | Pass |  |  |  |
| 3 | Verify that User is not able to Login with invalid Username and Valid Password | Should be display an error message User not found | Display an error Username not found | Pass |  |  |  |
| 4 | Verify that User is not able to Login with blank Username or Password | Set required field validation message for Username and Password | Display an error of wrong username and password | Fail | Not performa a validation function fix it |  | Bug\_002 |

|  |  |
| --- | --- |
| **Test Case Title** | Varify login page elements |
| **Test Type** | GUI |
| **Test Priority** | Medium |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Launch application with the given url | The site launched properly | Site launched successfully | Pass |  | <https://accounts.google.com/ServiceLogin> |  |
| 2 | Verify that the login screen contains elements such as Username, Password, Sign in button, Remember password check box, Forgot password link, and Create an account link. | All listed control displayed properly on the page | Login page loaded successfully | Pass |  |  |  |
| 3 | Verify that cursor is focused on “Username” text box on the page load | Cursor is focused in Username textbox | Cursor focus in Username textbox | Pass |  |  |  |
| 4 | Verify that tab functionality is working properly or not | When tab pressed cursor move in next control | Cursor moving in next control | Pass |  |  |  |
| 5 | Verify that all the fields such as Username, Password has a valid placeholder | All text fields have proper placeholder | All text fields have proper placeholder | Pass |  |  |  |
| 6 | Verify that the labels float upward when the text field is in focus or filled (In case of floating label) | When field is focused or filled, label display on top of the filled | When field is focus or filled, label display on top of the filled | Pass | step required when fields with floating label |  |  |
| 7 | verify that forgot password link working properly | when click on forgot password load forgot passworg page | forgot password link not working | Fail |  |  |  |

# References

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