

Darshan University

A Project Report on

**Collage Management System**

Under the subject

**Software Engineering (2301CS405)**

B. Tech, Semester – IV

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name:Ghada Afsana R. | Enrollment No.:24010101633 |
| 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 **“Collage Management System”** submitted in partial fulfilment for the Semester-4 of **Bachelor Technology (B. Tech)** in **Computer Science and Engineering (CSE)** Department to 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.

(Ghada Afsana)

Student’s Signature

Date:

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

# CERTIFICATE

This is to certify that the SRS on **“Collage Management System” has** been satisfactorily prepared by **Ghada Afsana** (**24010101633**) 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. 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

**Ghada Afsana**

# ABSTRACT

Collage management system is a system which aims in developing a computerized system to maintain all the daily work of Collage. This system will act as a tool to transfer traditional Collage into digital Collage. This project has many features which are generally not available in general Collage system like facility of user login, Student search on one click etc. It also has a facility of admin login through which the admin can monitor the whole system. This system will be designed with the basic features such as Admin can add/view/update/delete Faculties and students' details in it.The Collage Management System is an integrated computerized system designed to manage and automate the daily administrative and academic tasks within a college. It aims to replace traditional paper-based processes with a digital solution, improving the efficiency and accessibility of various operations. The system incorporates features such as user and admin logins, book search, and easy management of student and book records.

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

**Table of Contents**

[List of Figures I](#_bookmark0)

[List of Tables II](#_bookmark1)

1. [Introduction 1](#_bookmark2)
   1. [Product perspective 1](#_bookmark3)
   2. [Product features 1](#_bookmark4)
      1. [There are five different users who will be using this product: 1](#_bookmark5)
      2. [The features that are required for the admin are 1](#_bookmark6)
      3. [The features that are required for the student are 1](#_bookmark7)
   3. [Functional Requirement 1](#_bookmark8)
      1. [Admin 1](#_bookmark9)
      2. [Faculties 2](#_bookmark10)
      3. [Student 2](#_bookmark10)
      4. [Parents 2](#_bookmark10)
      5. [Mentor 2](#_bookmark10)
   4. [Non-Functional Requirement 2](#_bookmark11)
      1. [Usability 3](#_bookmark12)
      2. [Accuracy 3](#_bookmark13)
      3. [Availability 3](#_bookmark14)
      4. [Maintainability 3](#_bookmark15)
      5. [Reliability 3](#_bookmark15)
      6. [Security 3](#_bookmark15)
      7. [Efficiency 3](#_bookmark15)
2. [Design and Implementation Constraints 4](#_bookmark16)
   1. [Use case diagram 4](#_bookmark17)
   2. [Activity diagram and Swimlane diagram for Admin Activity 5](#_bookmark19)
   3. [Activity diagram and Swimlane diagram for Student Activity 7](#_bookmark19)
   4. [Sequence diagram 8](#_bookmark22)
   5. [State diagram 10](#_bookmark24)
   6. [Class diagram 11](#_bookmark27)
   7. [Data flow diagram 12](#_bookmark29)
      1. [Context diagram (level-0) 12](#_bookmark30)
      2. [DFD Level-1 13](#_bookmark31)
      3. [DFD Level-2 14](#_bookmark34)
3. [External interface requirement (Screens) 15](#_bookmark36)
4. Screen
   1. [Screen-1: Check Student 15](#_bookmark37)
   2. [Screen-2: Add Faculty 16](#_bookmark40)

4.2 Screen-3 [Upload Material 18](#_bookmark43)

* 1. [Screen-4:Delete Material 19](#_bookmark43)
  2. [Upload Assignment 20](#_bookmark43)
  3. [Screen-6: View Result 21](#_bookmark43)

1. [Database design 23](#_bookmark46)
   1. [List of Tables 23](#_bookmark47)
2. [Stories and Scenario 24](#_bookmark52)
   1. [Story-1: Add New Student in Collage Catalogue 24](#_bookmark53)
      1. [Scenario# S1.1 24](#_bookmark54)
      2. [Scenario# S1.2 25](#_bookmark55)
   2. [Story-2: Delate Student 26](#_bookmark57)
   3. [Story-3: Search Student 27](#_bookmark58)
3. [Test cases 28](#_bookmark61)
4. [References 36](#_bookmark62)

# List of Figures

[Figure 2.1-1 Use case diagram for Collage management system 3](#_bookmark18)

[Figure 2.2-1 Activity diagram 5](#_bookmark20)

[Figure 2.2-2 Swimlane diagram 6](#_bookmark21)

[Figure 2.3-1 Sequence diagram 8](#_bookmark23)

[Figure 2.4-1 State diagram 10](#_bookmark25)

[Figure 2.4-2 State diagram 10](#_bookmark26)

[Figure 2.5-1 Class diagram for Collage management system 11](#_bookmark28)

[Figure 2.6-1 Context diagram for Collage management system 12](#_bookmark32)

[Figure 2.6-2 DFD level-1 for Collage management system 13](#_bookmark33)

[Figure 2.6-3 DFD level-2 14](#_bookmark35)

[Figure 3.1-1 Screen-1: Check Student 16](#_bookmark38)

[Figure 3.2-1 Screen-2: Add Faculty Form 17](#_bookmark41)

[Figure 3.3-1 Screen-3: Upload Material 18](#_bookmark44)

[Figure 3.4-1 Screen-4: Delate Material 19](#_bookmark44)

[Figure 3.5-1 Screen-5: Upload Assignment 20](#_bookmark44)

[Figure 3.6-1 Screen-3: View Result 21](#_bookmark44)

# List of Tables

[Table 3.1-1 Screen element of Check Student 16](#_bookmark39)

[Table 3.2-1 Screen element of Add Faculty Form 17](#_bookmark42)

[Table 3.3-1 Screen element of Upload Materia 18](#_bookmark45)

[Table 3.3-1 Screen element of Delate Material 19](#_bookmark45)

[Table 3.3-1 Screen element of Upload Assignment 20](#_bookmark45)

[Table 3.3-1 Screen element of View Result 21](#_bookmark45)

[Table 5.1-1 Table: University 23](#_bookmark48)

[Table 5.1-2 Table: Admin 23](#_bookmark49)

[Table 5.1-3 Table: Staff 23](#_bookmark50)

[Table 5.1-4 Table: Student 23](#_bookmark51)

[Table 5.1-5 Table: Parents 23](#_bookmark51)

# Introduction

## Product perspective

The **Collage Management System** (CMS) provides an efficient way to manage the day-to-day activities of a college, including student records, book inventories, and administrative tasks. It offers distinct access rights for different users such as students, librarians, and admins to streamline their respective activities.

## Product features

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

* Admin who will be acting as the administrator.
* Student who will be accessing the Application.
* .Faculties who will Upload

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

 Login to the admin panel with username and password.

 Generate various reports (student, book, teacher, issue report).

 Monitor and manage user accounts (students and librarians)Can issue a book to the student.

* Can view the data of student.
* Can update data for old student
* Can Information about Department & library
* Can Add Faculties
* Can Remove Faculties
* Can Generate a Fee Receipts
* Can generate Mark sheet for students
* Can access all the accounts of the students.

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

* Can view the different categories of books available in the library.
* Can view the List of books available in each category.
* Can own an account in the library.
* Can view the books issued to him.
* Can put a request for a new book.

## Functional Requirement

### Admin

* Store New Student Data: can store data of new student.
* Update Data For Old Student : can update data of old student.
* Check The Student Remaining On It .
* Add Faculties : can add new faculties.
* Remove Faculties : can remove faculties.
* Add Branch : Add new branch in university.
* Remove Branch :admin can some reasons remove branch .
* Manage Registration : manage a full registration .
* Information about Department & library : information about department and library .
* Generate a Fee Receipts : generate a fee receipt for paid fee by student.
* Generate Mark sheet : generate mark sheet for exams.
* Pay Fee For Staff : pay fee to staff member.
* Verify Fee pay by Student

.

### Faculties

* Login Application : Login to the Faculties panel with username and password.
* Fill Attendances : Fill Attendances to who are present todays.
* Upload Material : Upload Material in application for student.
* Delete Materiel : Faculties can delete material.
* View Notices : view notices for new event or etc.
* Show Timetable : show timetable as per day.
* Update Timetable: update timetable for some cancel lectures.
* Information About Labs : information about student lab performance.
* Evaluate Student : evaluation of student.
* Check Student Data : check student data
  + 1. Student
* Sign in : student can sign in application.
* Download Materiel : download material for faculties can forward.
* Upload A Assignment : upload assignment as under time limit
* View Result : view result for given exams.
* View Attendance : view attendance.
* Edit Personal Info : edit personal information like photo.
* Give Feedback: given feedback for faculties.
* Update Profile:update profile
* View Notices : view notice’s for some event ,scholarship.
* Update Password : student can change password.
  + 1. Parents
* Show Attendance : show attendance for student absent or present.
* Show Mark Of Student : show mark sheet of student.
* Notices for Regarding Fee : show Notices about Regarding Fee.
* Pay Fee : pay fee for new semester.
* Account Section : account detail
  + 1. Mentor
* See Mark of Branch Wise All Student
* Search Student of Any Department by Brach Wise And Roll Number Wise
* Update Data

## 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.

### Reliability:

### Backup and recovery mechanisms should ensure that no more than 5 minutes of data is lost in the event of a failure.

### **Security**

### Access control mechanisms should ensure that only authorized users can access sensitive information.

* All sensitive data (e.g., student grades, personal information) should be encrypted both at rest and in transit.

### 1.4.7 **Portability**

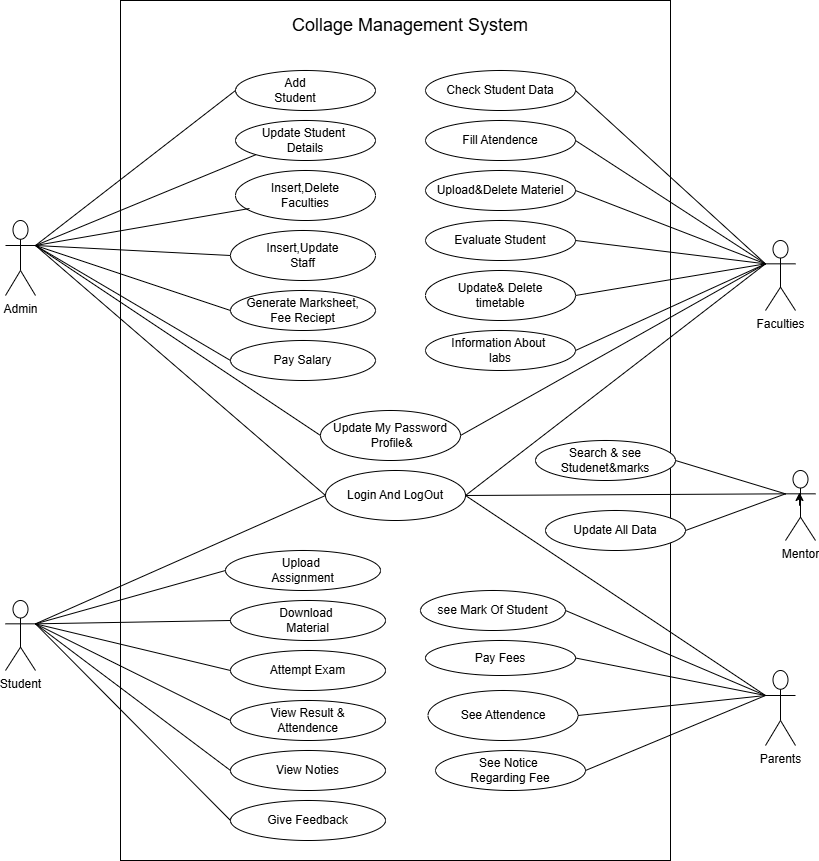
* The system should work on multiple platforms, including Windows, macOS, and popular Linux distributions.
* It should also be accessible through major web browsers (e.g., Chrome, Firefox, Safari).

.

# Design and Implementation Constraints

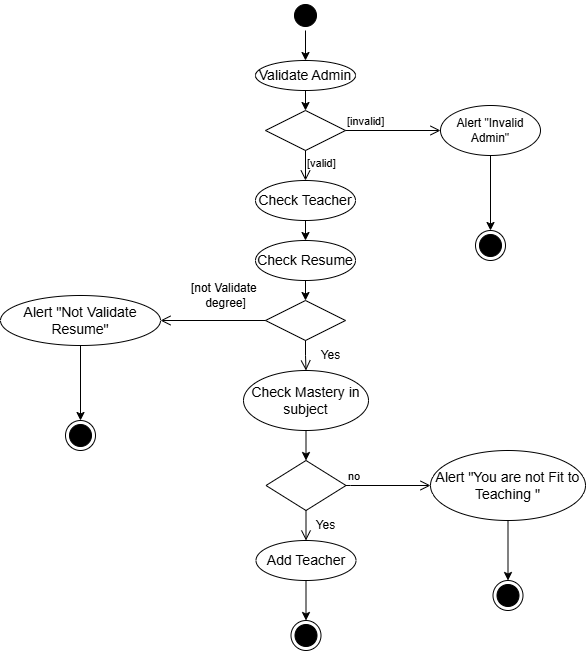
## Use case diagram

## 

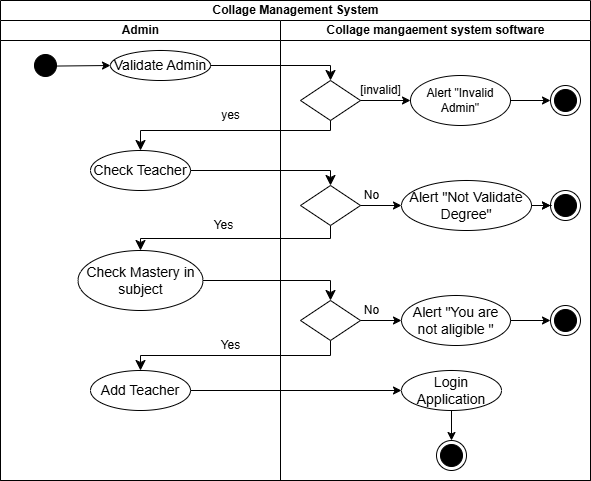


## Figure 2.1-1 Use case diagram for Collage management system

## Activity diagram and Swimlane diagram for Admin Activity

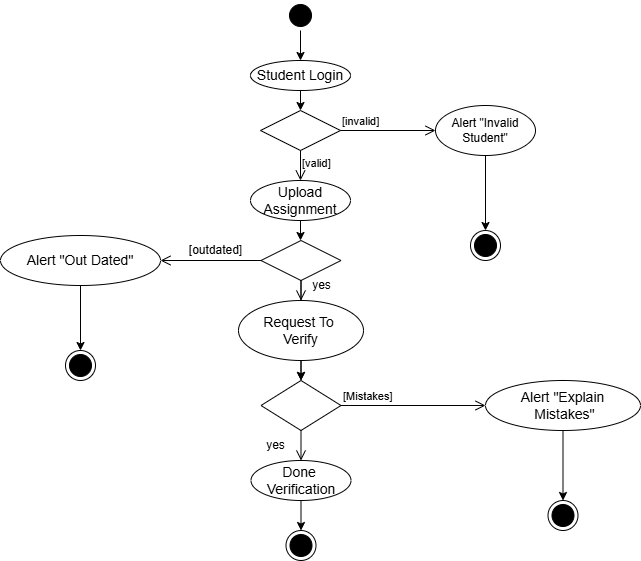
**

*Figure 2.2-1 Activity diagram for Admin Activity*

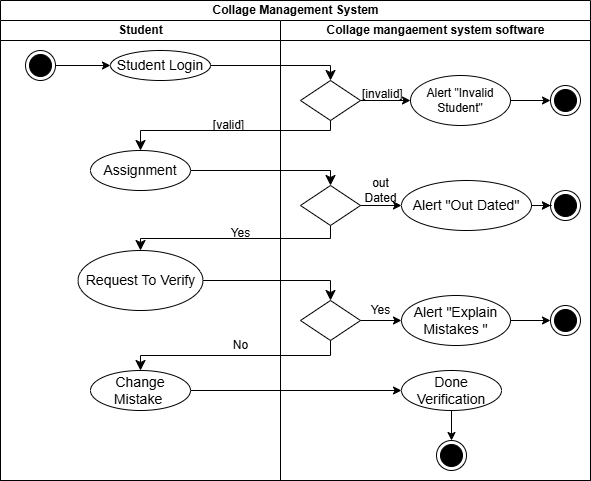
**

*Figure 2.2-2 Swimlane diagram for Admin Activity*

## Activity diagram and Swimlane diagram for Student Activity

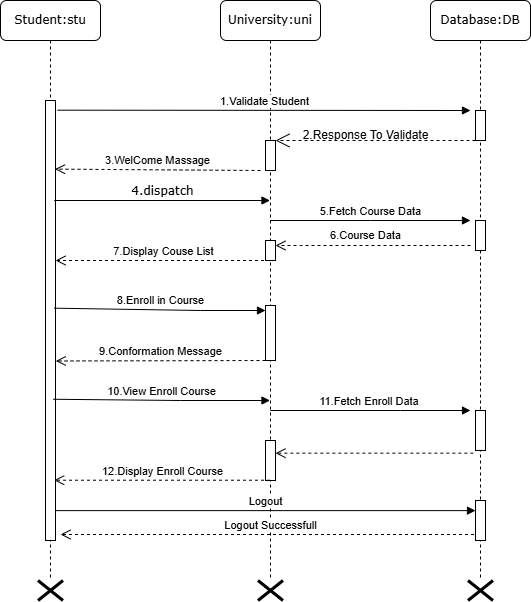


*Figure 2.2-3 Activity diagram for Student Activity*

**

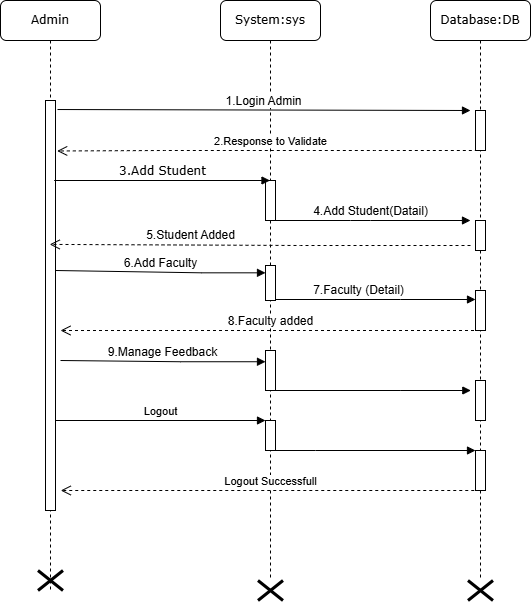
*Figure 2.2-4 Swimlane diagram for Student Activity*

## Sequence diagram

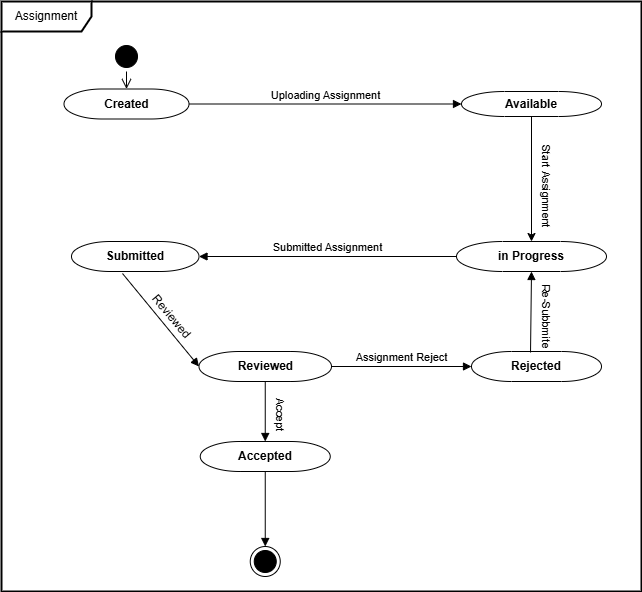
****

*Figure 2.3-1 Sequence diagram for Student course*

*Figure 2.3-2 Sequence diagram for Admin Task*

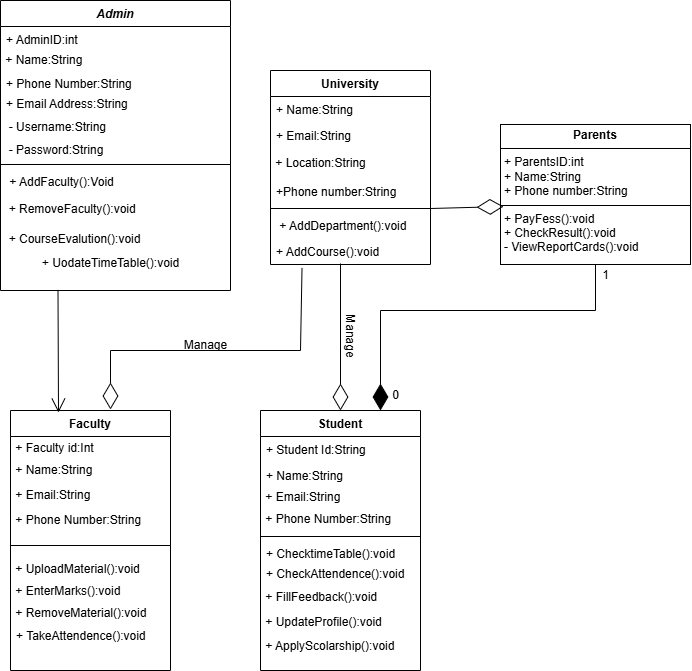


## State diagram



*Figure 2.4-1 State diagram of Assignment*

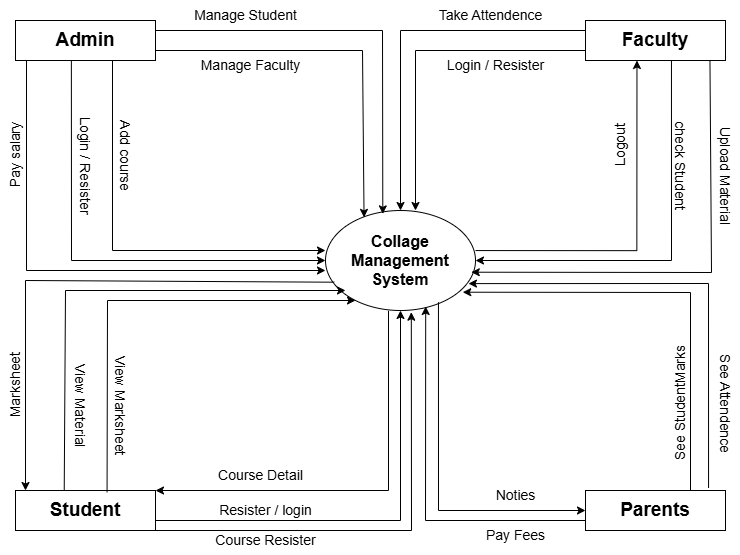
## Class diagram

****

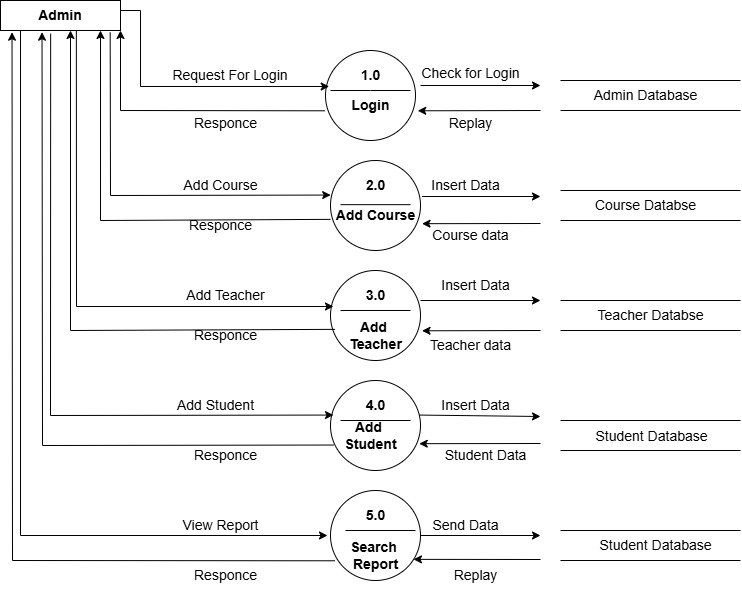
*Figure 2.5-1 Class diagram for collage management system*

## Data flow diagram

### Context diagram (level-0)



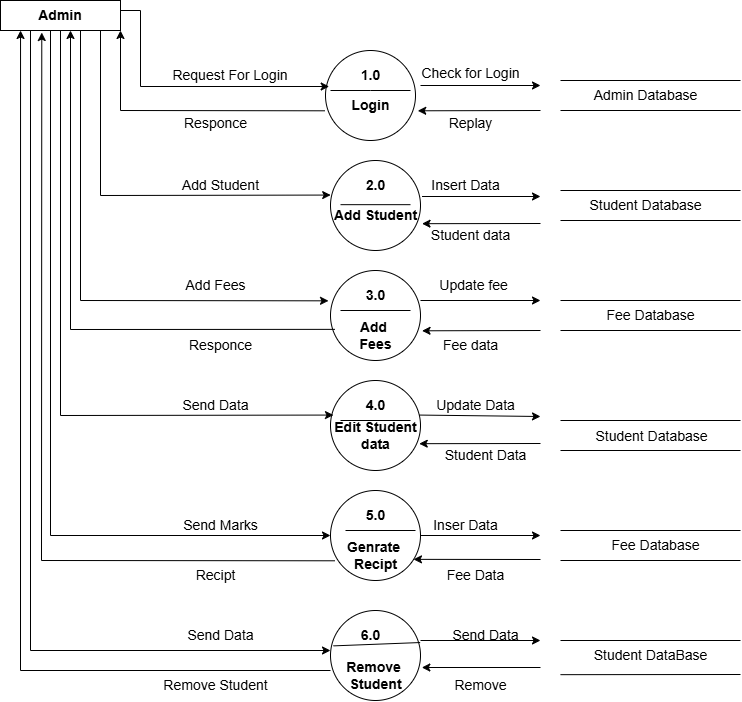
*Figure 2.6-1 Context diagram for Collage management system*



*Figure 2.6-2 DFD level-1 for collage management system*

Admin

### DFD Level-2



*Figure 2.6-3 DFD level-2 for Admin*

# External interface requirement (Screens)

## Screen-1: Check Student

## 

## 

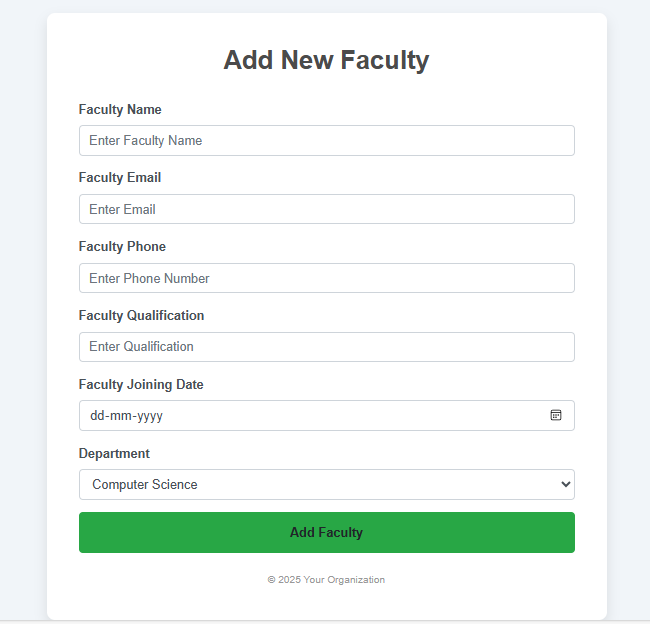
*Figure 3.1-1 Screen-1: Check student Remaining On it*

**Purpose:** This form will be check Student Available in this application . return a particular Student detail.

*Table 3.1-1 Screen element of Registration form*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | Enroll no | Textbox | M | 1 | A unique identifier for each student. |
| **2** | Name | Textbox | M | 1 | The full name of the student. |
| **3** | Age | Number | M | 1 | The age of the student. |
| **4** | Course | Dropdown | M | 1 | The course or program the student is enrolled in (e.g., Mathematics, Physics). |
| **5** | Student Id | Textbox | M | 1 | A unique ID assigned to the student by the institution. |
| **6** | Email | Link | ------ | ------ | The email address of the student in a valid email format (e.g., name@domain.com). |
| **7** | Check Remaining on it | Button | ------ | ------ | Check Botton Chek Student Available Or Not. |

## Screen-2: Add Faculty Form



*Figure 3.2-1 Screen-2: Add Faculty form*

**Purpose:** The purpose of the **"Add New Faculty"** form is to allow administrators to efficiently register and manage faculty details in the system, ensuring organized faculty data storage, quick retrieval, and seamless integration into institutional operations.

*Table 3.2-1 Screen element of Add Faculty form*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | |  | | --- | | Faculty Name |  |  | | --- | |  | | Textbox | M | 1 | Full name of the faculty member being added. |
| **2** | |  | | --- | | FacultyEmail |  |  | | --- | |  | | Email | M | 1 | The email address of the faculty member in a valid email format. |
| **3** | Faculty Phone | Number | M | 1 | The contact phone number of the faculty member. |
| **4** | Faculty Qualification | Textbox | M | 1 | Educational qualifications of the faculty member. |
| **5** | Faculty Joining Date | Date Picker | M | 1 | The date when the faculty member joined the institution. |
| **6** | Department | Dropdown | M | 1 | The department to which the faculty member belongs (e.g., Computer Science). |
| **7** | Add Faculty | Button | ------- | ------ | Button to submit the form and store faculty data in the database. |

## Screen-3: Upload Material

## 

*Figure 3.3-1 Screen-3: Upload Material*

**Purpose:** The purpose of the **"Upload Material"** form is to allow Faculty to upload academic materials such as lecture notes, presentations any other resources into the system for student access.

*Table 3.3-1 Screen element of Upload Material*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | Material Title | Textbox | M | 1 | Title of the academic material to identify its content clearly. |
| **2** | Description | Textbox | O | 1 | A brief description providing details about the material being uploaded. |
| **3** | Material Type | Dropdown | M | 1 | Type of material (e.g., Lecture Notes, Assignments, or Tutorials). |
| **4** | Upload File | File Input | M | 1 | Upload field for selecting the material file to be added to the system. |
| **5** | Upload Material | Button | ------ | ------ | Last Name field should be editable and accept  only string as a borrower last name. |

## Screen-4: Delete Material

## 

*Figure 3.4-1 Screen-4: Delete Material*

**Purpose:** The purpose of the above interface is to manage academic resources by listing materials with options to delete them. It ensures efficient organization and removal of outdated or unnecessary content.

*Table 3.4-1 Screen element of Delete Material*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | |  | | --- | | Material Title |  |  | | --- | |  | | Textbox | M | N | |  | | --- | | Displays the title of the material to identify it clearly. |  |  | | --- | |  | |
| **2** | |  | | --- | | Material Type |  |  | | --- | |  | | Textbox | M | N | |  | | --- | | Specifies the type of material, such as Lecture Notes, Research Papers, etc. |  |  | | --- | |  | |
| **3** | |  | | --- | | Uploaded By |  |  | | --- | |  | | Textbox | M | N | |  | | --- | | Displays the name of the uploader to track the contributor. |  |  | | --- | |  | |
| **4** | |  | | --- | | Actions |  |  | | --- | |  | | Button | M | N | "Delete" button to remove a specific material from the system. |

## 4.5 Screen-5:Upload Assignment

## 

## 

*Figure 3.5-1 Screen 5:Upload Assignment*

**Purpose**: The purpose of the "Upload Assignment" interface is to allow users to submit assignments by providing details like title, description, due date, and attaching relevant files. It organized submission for academic tasks.

*Table 3.5-1 Screen element of upload Assignment*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | |  | | --- | | Assignment Title |  |  | | --- | |  | | Textbox | M | 1 | Input field for entering the title of the assignment. |
| **2** | Description | Textbox | O | 1 | |  | | --- | | Optional field for providing a brief  description of the assignment. |  |  | | --- | |  | |
| **3** | |  | | --- | | Due Date |  |  | | --- | |  | | Date | M | 1 | |  | | --- | | Input field for selecting the due date of the  assignment. |  |  | | --- | |  | |
| **4** | Upload Assignment File | File Input | M | 1 | |  | | --- | | Field for attaching the file to be uploaded,  such as the assignment document. |  |  | | --- | |  | |
| **5** | Upload Assignment | Button | ------ | ------ | Button to confirm and upload the entered assignment details along with the selected file. |

## 4.6 Screen-6:View Result

## 

*Figure 3.6-1 Screen 5:View Result*

**Purpose:** The purpose of the above interface is to manage academic resources by listing materials with options to delete them. It ensures efficient organization and removal of outdated or unnecessary content

*Table 3.5-1 Screen element of View Result*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sr.** | **Screen Element** | **Input Type** | **O/M** | **1/N** | **Description** |
| **1** | |  | | --- | | Student ID |  |  | | --- | |  | | Textbox | M | 1 | |  | | --- | | Input field for entering the student’s  unique ID. |  |  | | --- | |  | |
| **2** | Password | Password | M | 1 | |  | | --- | | Field for securely entering the student’s  account password. |  |  | | --- | |  | |
| **3** | |  | | --- | | Select Exam |  |  | | --- | |  | | Dropdown | M | 1 | |  | | --- | | Dropdown menu for selecting the exam for  which the result is to be viewed. |  |  | | --- | |  | |
| **4** | |  | | --- | | Select Year |  |  | | --- | |  | | Dropdown | M | 1 | |  | | --- | | Dropdown menu for selecting the year of  the exam. |  |  | | --- | |  | |
| **5** | |  | | --- | | Captcha |  |  | | --- | |  | | Textbox | M | 1 | |  | | --- | | Field to enter the displayed CAPTCHA code  to verify user authenticity. |  |  | | --- | |  | |
| **6** | Refresh Captcha | Button | ------ | ------ | |  | | --- | | Button to generate a new CAPTCHA code if  the current one is unclear. |  |  | | --- | |  | |
| **7** | |  | | --- | | View Result |  |  | | --- | |  | | Button | ------ | ------ | Button to submit all details and view the result. |

# Database design

## List of Tables

* University
* Admin
* Staff
* Student
* Parents

*Table 4.1-1 Table: Admin*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **AdminID** | int | NN | PK (Auto Increment) |  |
| **AdminName** | varchar(100) | NN |  |  |
| **PhoneNum** | Number(10,0) | AN |  |  |
| **Email** | Varchar(50) | AN |  |  |
| **UserName** | varchar(50) | NN |  |  |
| **Password** | Varchar(50) | NN |  |  |

*Table 4.1-2 Table: University*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **UNIID** | int | NN | PK (Auto Increment) |  |
| **Name** | varchar(100) | NN |  |  |
| **Address** | Varchar(100) | NN |  |  |
| **ContactNum** | Number(10,0) | AN |  |  |
| **Email** | Varchar(50) | AN |  |  |

*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 | FK | REFERENCE THE ADMIN TABLE |
| **Designation** | varchar(100) | NN |  |  |
| **Joining Date** | DateTime | NN |  |  |
| **Salary** | int | AN |  |  |

*Table 4.1-4 Table: Student*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **StudentID** | int | NN | PK (Auto Increment) |  |
| **StudentName** | varchar(100) | AN |  |  |
| **Gender** | varchar(100) | AN |  |  |
| **DOB** | DateTime | AN |  |  |
| **Department** | varchar(100) | AN |  |  |
| **Contact** | number(10,0) | AN |  |  |

*Table 4.1-4 Table: Parents*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Null** | **Keys & Constrains** | **Default Value & Description** |
| **ParentID** | int | NN | PK (Auto Increment) |  |
| **ParentName** | varchar(100) | AN |  |  |
| **Contact** | number(10,0) | AN |  |  |
| **Student Id** | int | NN | FK | REFERENCE THE STUDENT TABLE |

# Stories and Scenario

## Story-1: Add New Student in Collage Catalogue

|  |  |  |
| --- | --- | --- |
| *Story #* ***S1*** | **:** | **As a** Admin,  **I want to** add a new student to the college management system  **So that** their academic records can be managed efficiently. |
| **Priority** | **:** | High |
| **Estimate** | **:** | XL |
| **Reason** | **:** | The addition of a new student to the college management system is crucial for ensuring that student records are accurately maintained and accessible for academic and administrative purposes. |

### Scenario# S1.1

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***S1.1*** | **:** | Adding a New Student with Valid Information |
| **Prerequisite** | **:** | The administrator is logged in to the college management system. |
| **Acceptance Criteria** | **:** | **Given:** The administrator is navigated to the student management page. Valid student information, including name, date of birth, student ID, department, and other relevant details, is added.  **When:**  The administrator selects the "Add New Student" option, enters valid student details, and clicks the "Save" button to add the student to the system.  **Then t**he system successfully adds the student to the database and  The administrator receives a confirmation message with the student's identification number. |

* + 1. Scenario# S1.2

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

## Story-2: DELETE STUDENT

|  |  |  |
| --- | --- | --- |
| *Story #* ***S2*** | : | **As a** Admin,  **I want to** delete a student from the college management system **So that** their records are removed when they are no longer part of the institution. |
| **Priority** | **:** | High |
| **Estimate** | **:** | L |
| **Reason** | **:** | The removal of a student from the college management system is crucial for ensuring that records remain accurate and up-to-date, preventing outdated or irrelevant data from affecting administrative processes. |

### 6.2.1Scenario# S2.1

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***S2.1*** | **:** | Deleting a Student with Valid Information |
| **Prerequisite** | **:** | The administrator is logged in to the college management system. |
| **Acceptance Criteria** | **:** | **Given:** The administrator is navigated to the student management page. The student to be deleted exists in the system.  **When:**  administrator selects the "Delete Student" option, searches for the student using valid details (e.g., Student ID or name), and confirms the deletion. The system successfully removes the student from the database and provides a confirmation message.  **Then t**he system successfully removes the student from the database and administrator receives a confirmation message stating that the student has been deleted. |

### 6.2.2Scenario# S2.2

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***S2.2*** | **:** | Attempting to Delete a Non-Existing Student |
| **Prerequisite** | **:** | The administrator is logged into the college management system. |
| **Acceptance Criteria** | **:** | **Given:** The administrator is on the student management page.  **When:**  administrator selects the "Delete Student" option, searches for a student who does not exist in the system, and attempts to delete the student, the system displays an error message indicating that the student record does not exist, and no changes are made to the database.  **Then t**he system successfully adds the student to the database and  The administrator receives a confirmation message with the student's identification number. |

* 1. **Story-3: SEARCH STUDENT**

|  |  |  |
| --- | --- | --- |
| *Story #* ***S3*** | **:** | **As a** Admin,  **I want** Search a student from the college management system  **So that** I can quickly access their academic and personal details when needed. |
| **Priority** | **:** | High |
| **Estimate** | **:** | M |
| **Reason** | **:** | The ability to search for a student is essential for efficient record management, enabling quick retrieval of student information for administrative and academic purposes. |

### 6.3.1Scenario# S3.1

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***S3.1*** | **:** | Searching for a Student with Valid Information |
| **Prerequisite** | **:** | The administrator is logged in to the college management system. |
| **Acceptance Criteria** | **:** | **Given:** The administrator is on the student management page.  **When:**  administrator enters a valid search query (e.g., Student ID, name, or department) and clicks the "Search" button, the system retrieves  **Then t**he displays the student’s details, allowing the administrator to view the student’s academic and personal records. |

### 

### 6.3.2Scenario# S3.2

|  |  |  |
| --- | --- | --- |
| *Scenario#* ***S3.2*** | **:** | Searching for a Non-Existing Student |
| **Prerequisite** | **:** | The administrator is logged into the college management system. |
| **Acceptance Criteria** | **:** | **Given:** The administrator is on the student management page.  **When:**  administrator enters a search query for a student who does not exist in the system and clicks the "Search" button.  **Then t**he system displays a message stating that no matching student record was found, and no student details are displayed. |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Name:** | **Collage Manegment** | **Test Designed by:** | **Ghada Afsana** |
| **Module Name:** | **Add Student** | **Test Designed date:** | 22-02-2025 |
| **Release Version:** | **1.0** | **Test Executed by:** |  |
|  |  | **Test Execution date:** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pre-condition**: **Web application should be accessible** | | | | |
| **Test Case ID** | **Test Title** | **Test Type** | **Description** | **Test Case ID** |
| **TC\_001** | Verify Add Faculty page load | Functional | |  | | --- | |  |  |  | | --- | | Verify that the Add New Faculty  Page load successfully. | | TC\_001 |
| **TC\_002** | Submit Form with Valid Data | Functional | Verify that the form submits successfully with valid input data. | TC\_002 |

|  |  |
| --- | --- |
| Test Case Title | Verify Add Faculty Page Load |
| 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 Add Faculty page URL | Add New Faculty Page should launch properly. | Add New Faculty page launched successfully | Pass |  | URL: ADD faculty page URL. |  |
| 2 | Verify "Add New Faculty" heading | The page should display the heading "Add New Faculty". | Heading displayed as expected. | Pass |  |  |  |
| 3 | |  | | --- | |  |  |  | | --- | | Verify  Faculty  Name field | | The Faculty Name field should be editable and accept input. | Faculty Name field accepted input. | pass |  | Name:GhadAfsana |  |
| 4 | Verify Faculty Email field | The Faculty Email field should be editable and accept input | faculty Email field accepted input. | Pass |  |  |  |

|  |  |
| --- | --- |
| Test Case Title | Submit Form with Valid Data |
| 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 | Enter valid Faculty Name Password | Faculty Name field should accept the valid name input. | Name accepted without error. | Pass |  | Name:Afsana |  |
| 2 | Enter valid Faculty Email &Password | Faculty Email field should accept valid email input. | Email and Password  accepted successfully. | Pass |  | Email:xyz@g.com  Pass:cdbdh |  |
| 3 | Enter Faculty Qualification. | Qualification field should accept text input | Qualification accepted. | Pass |  | Qualification:  PHD |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 4 | Click on "Add Faculty" button. | Form should submit successfully with a success message. | Succes massag  Display | Pass |  |  |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Name:** | **Collage Manegment** | **Test Designed by:** | **Ghada Afsana** |
| **Module Name:** | **Upload Material** | **Test Designed date:** | 22-02-2025 |
| **Release Version:** | **1.0** | **Test Executed by:** |  |
|  |  | **Test Execution date:** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pre-condition**: **Web application should be accessible** | | | | |
| **Test Case ID** | **Test Title** | **Test Type** | **Description** | **Test Case ID** |
| **TC\_001** | Verify Upload Material Page Load | Functional | |  | | --- | |  |  |  | | --- | | Verify that the Upload Material  page loads successfully. | | TC\_001 |
| **TC\_002** | Submit Form with Valid Data | Functional | Verify that the form submits successfully with valid data. | TC\_002 |

|  |  |
| --- | --- |
| Test Case Title | Verify Upload Material Page Load |
| 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 | Verify "Upload Material" heading | Page should display the heading "Upload Material". | Heading displayed as expected. | Pass |  |  |  |
| 2 | Verify Material Title field | Material Title field should be editable and accept input. | Material Title field accepted input. | Pass |  |  |  |
| 3 | |  | | --- | |  |  |  | | --- | | Verify Descri-  ption field | | Description field should be editable and accept input. | Description field accepted input. | pass |  |  |  |
| 4 | Verify Upload Material button | Upload Material button should be clickable. | Button clicked and processed input. | Pass |  |  |  |

|  |  |
| --- | --- |
| Test Case Title | Submit Form with Valid Data |
| 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 | Enter valid Material Title | Material Title field should accept valid input. | Name accepted without error. | Pass |  |  |  |
| 2 | Enter valid Description | Description field should accept text input. | Description accepted successfully. | Pass |  |  |  |
| **3** | Select valid Material Type | Material Type dropdown should accept selection. | Material Type selected successfully. | pass |  |  |  |
| **4** | Upload a valid file | Choose File should accept valid file formats. | File uploaded successfully. | Pass |  |  |  |
| 4 | Click Upload Material button | Form should submit successfully with valid data. | Form submitted and confirmation received. | Pass |  |  |  |

**References**

* <http://www.w3schools.com/html/html_intro.asp>
* https://[www.w3schools.com/php/default.asp](http://www.w3schools.com/php/default.asp)
* https://[www.javatpoint.com/uml](http://www.javatpoint.com/uml)