

Requirement & Design Specification

**Global Assess Management System (GAMS)**

**Version: 1.0**

– Hanoi, August 2022 –

# Record of Changes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **A\* M, D** | **In charge** | **Change Description** |
| V1.0 | 15/2 | A | KienNTHE11 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

\*A - Added M - Modified D - Deleted

Contents

[Record of Changes 2](#_gjdgxs)

[I. Overview 4](#_1fob9te)

[1. System Context 4](#_3znysh7)

[2. User Requirements 4](#_2et92p0)

[2.1 Actors 4](#_tyjcwt)

[2.2 Diagrams 5](#_3dy6vkm)

[2.3 Descriptions 6](#_1t3h5sf)

[2.4 Main Workflows 6](#_4d34og8)

[3. System Functionalities 7](#_2s8eyo1)

[3.1 Screens Flow 7](#_17dp8vu)

[3.2 Screen Authorization 8](#_3rdcrjn)

[3.3 Non-UI Functions 8](#_26in1rg)

[II. Functional Requirements 9](#_35nkun2)

[1. <<Feature Name1>> 9](#_44sinio)

[1.1 <<Screen/Function Name1>> 9](#_2jxsxqh)

[1.2 User Login 9](#_z337ya)

[1.3 … 9](#_3j2qqm3)

[2. System Administration 9](#_1y810tw)

[2.1 System Settings 9](#_4i7ojhp)

[2.2 … 10](#_1ci93xb)

[3. … 10](#_3whwml4)

[III. System Design 11](#_2bn6wsx)

[1. Database Design 11](#_qsh70q)

[1.1 Database Schema 11](#_1pxezwc)

[1.2 Table Descriptions 11](#_49x2ik5)

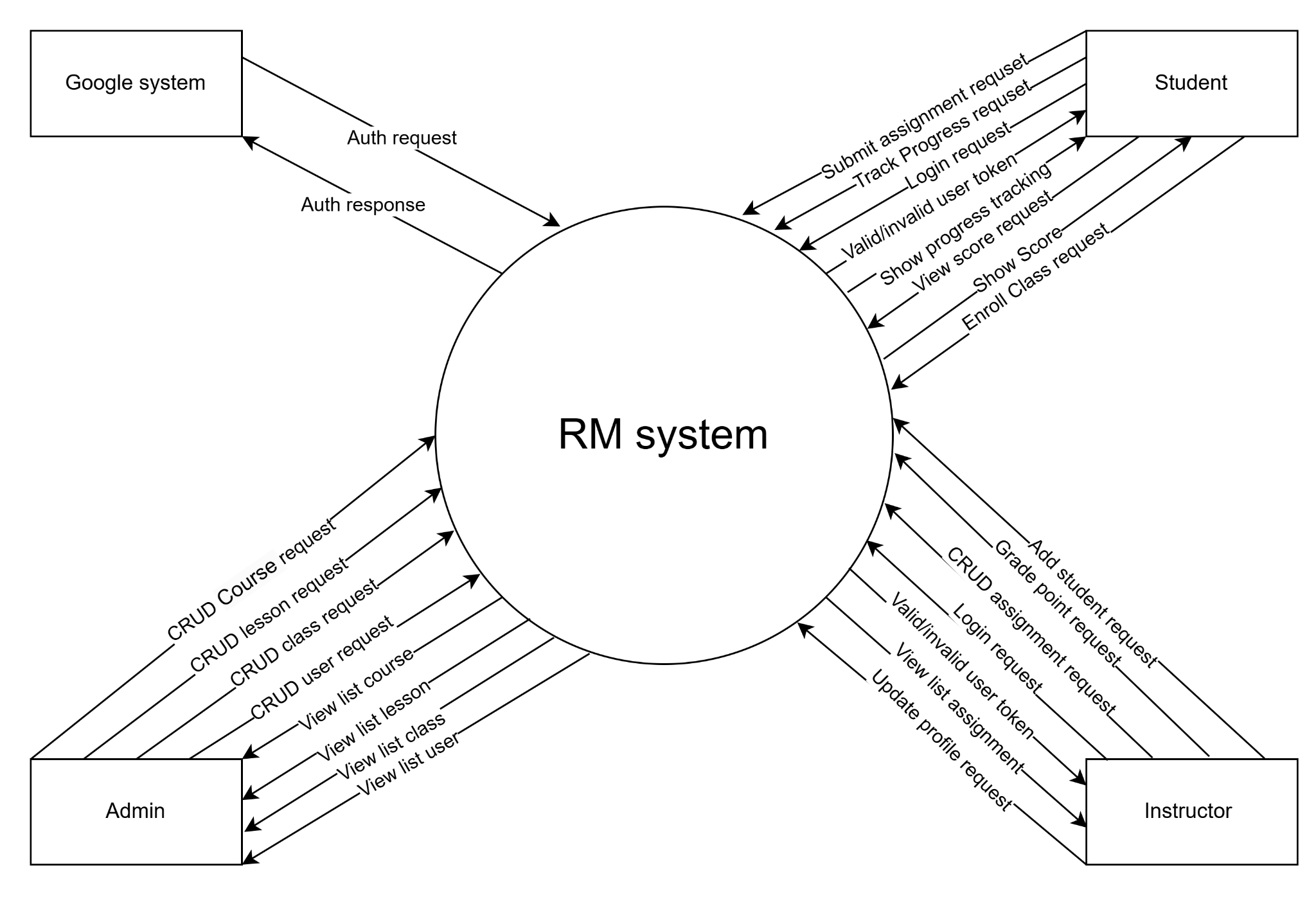
[2. Code Packages 11](#_2p2csry)

[2.1 Package Diagram 12](#_147n2zr)

[2.2 Package Descriptions 12](#_3o7alnk)

# I. Overview

## 1. System Context



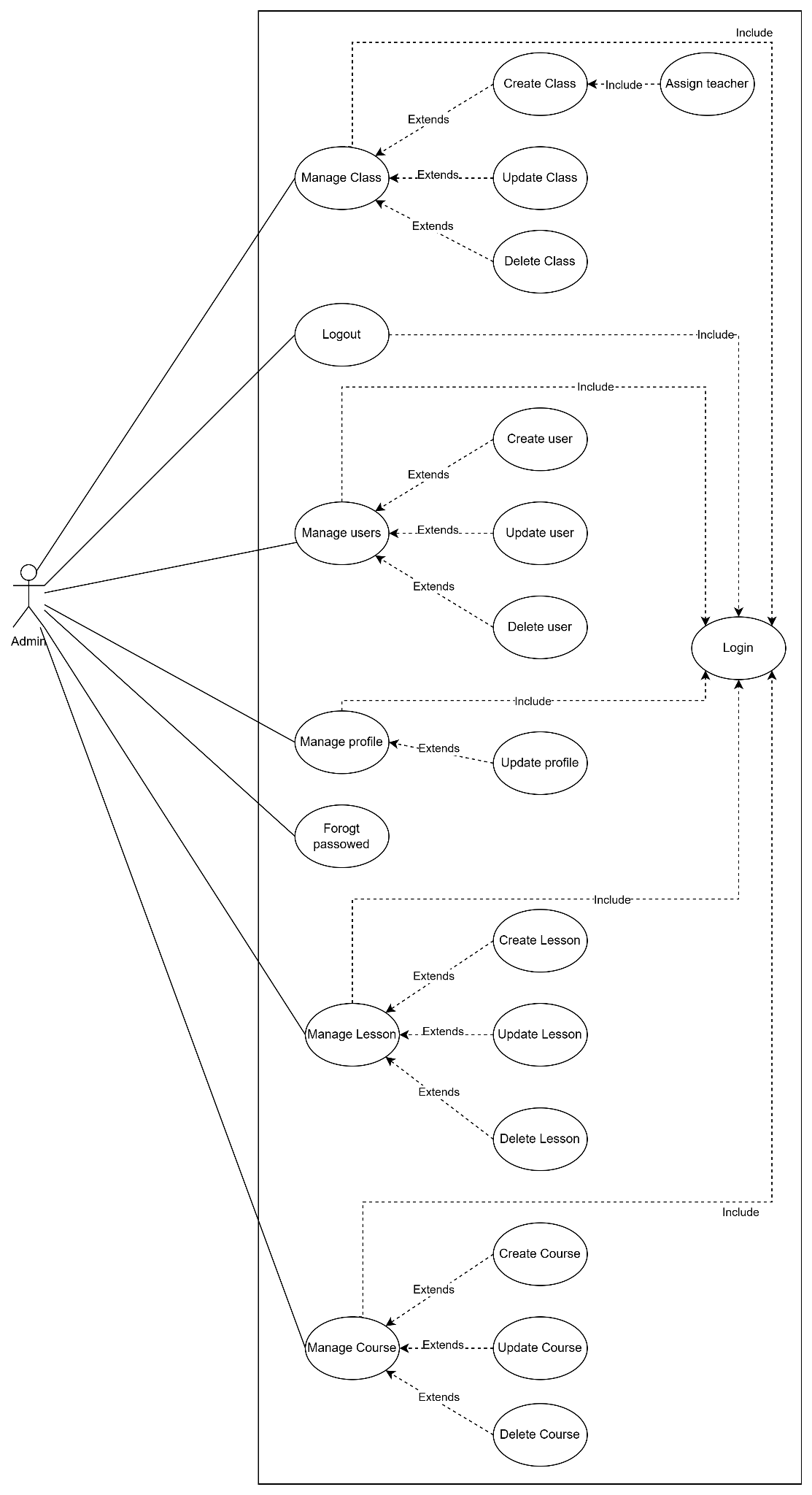
## 2. User Requirements

### 2.1 Actors

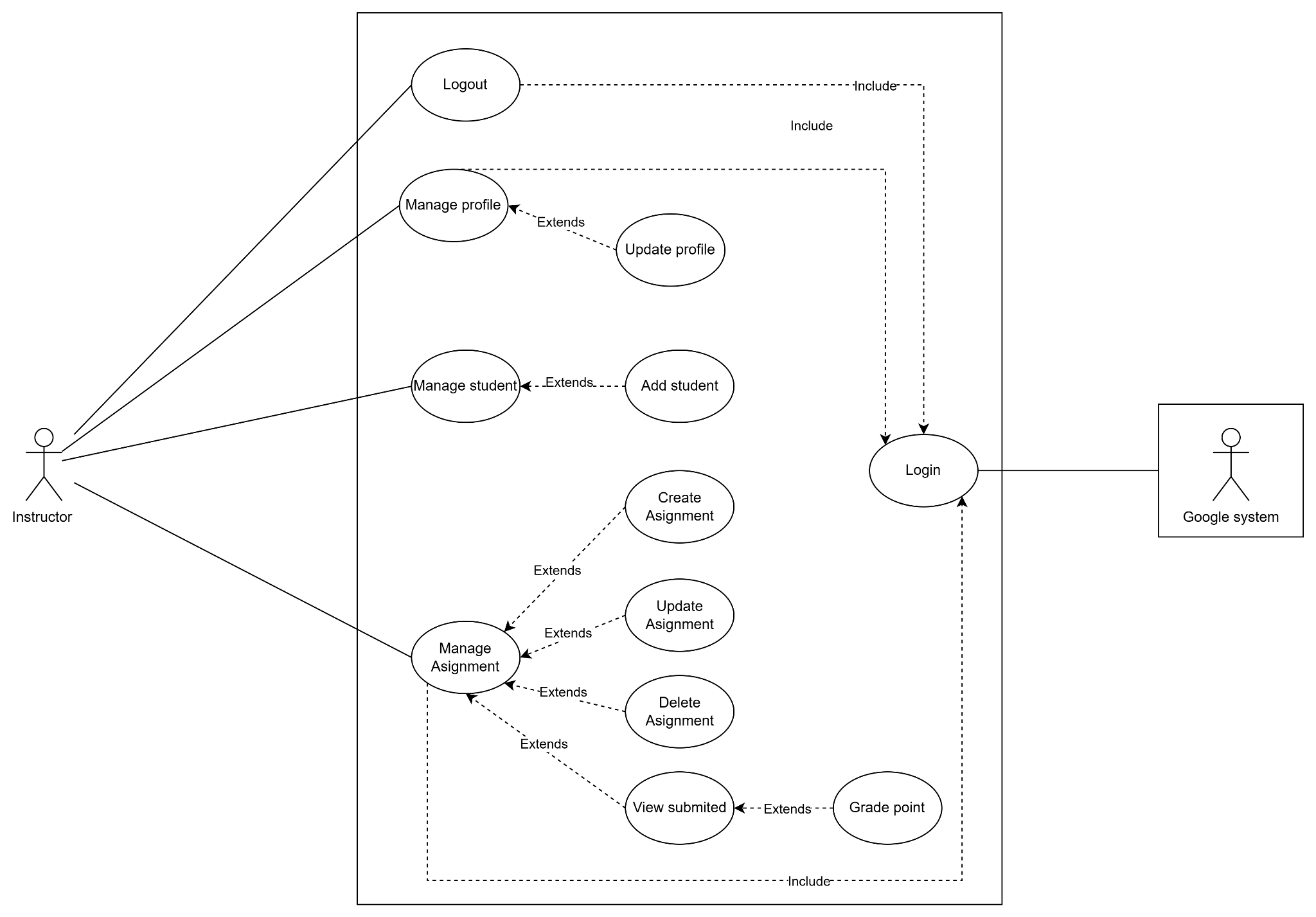
|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Description** |
| 01 | Administrator | The administrator manages the overall system. They have permissions to perform CRUD (Create, Read, Update, Delete) operations on courses, lessons, classes, and user accounts. Additionally, they can view lists of courses, lessons, classes, and users to oversee the system's operations and maintain the database. |
| 02 | Instructor | The instructor is responsible for managing classes and assignments. They can perform actions such as adding students, grading assignments, managing assignment content (CRUD operations), and tracking student progress. Instructors can also view and update their profile information. |
| 03 | Student | The student interacts with the system to participate in classes, view assignments, and submit their work. They can track their progress, view scores, enroll in classes, and perform assignment submissions. Additionally, they can log in to the system and view their progress and grades. |
| 04 | Google System | This external system is used for authentication purposes. It provides login and authentication services by validating user credentials via OAuth or similar authentication protocols. |

### 2.2 Diagrams

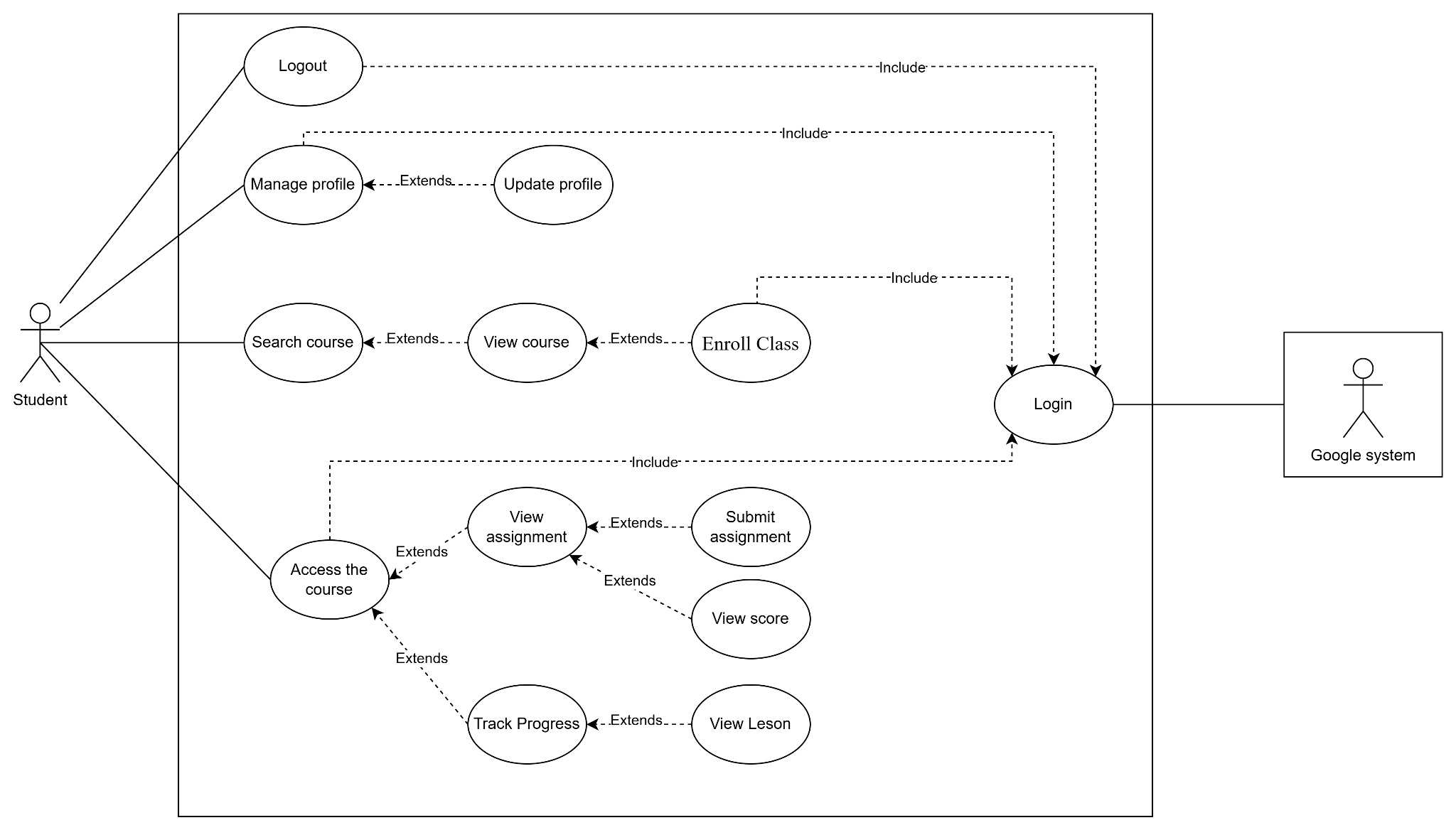
#### 2.2.1 UCs for Admin

**

#### 2.2.2 UCs for Instructor

**

#### 2.2.3 UCs for Students

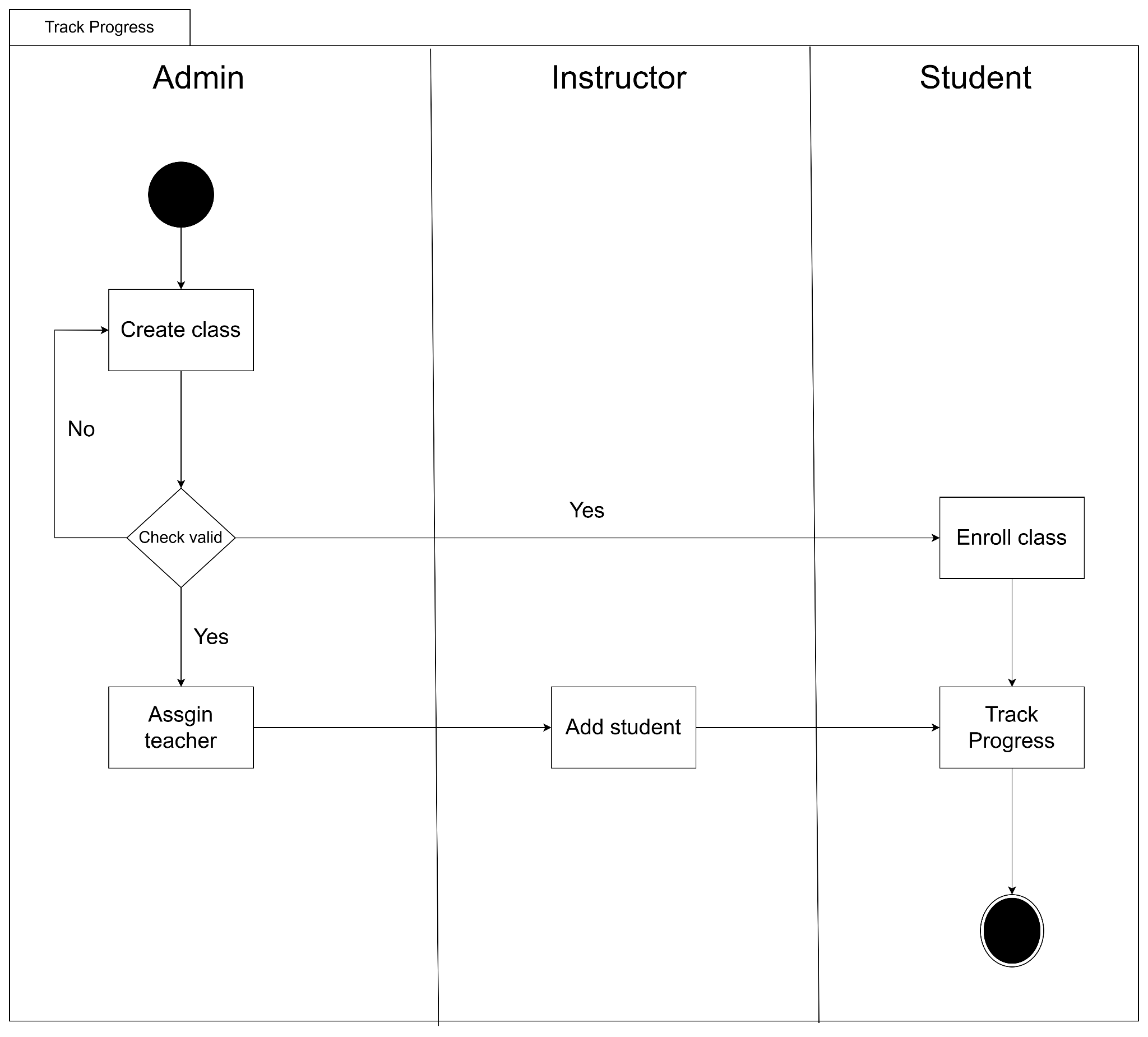


### 2.3 Descriptions

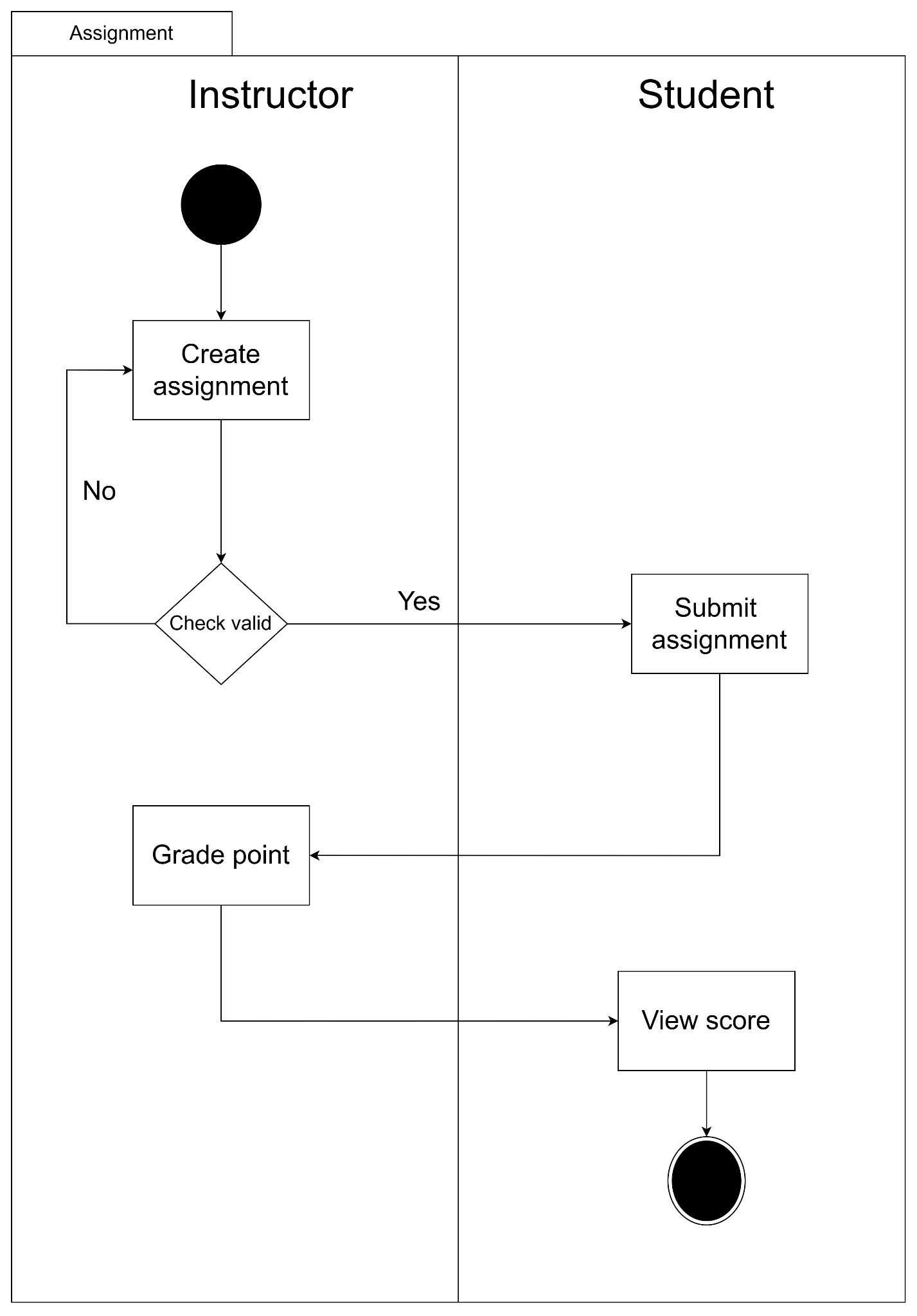
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Use Case** | **Actors** | **Use Case Description** |
| 01 | Manage Class | Admin | The admin can perform actions related to class management, including creating, updating, deleting, and assigning teachers to classes. |
| 02 | Create Class | Admin | Admin can create a new class. This use case includes assigning a teacher to the class as a subtask. |
| 03 | Update Class | Admin | Allows admin to modify the details of an existing class, such as changing the name, schedule, or assigned teacher. |
| 04 | Delete Class | Admin | Admin can remove a class from the system. |
| 05 | Manage Users | Admin | Admin has the ability to create, update, or delete user accounts. This includes managing both student and instructor accounts. |
| 06 | Create User | Admin | Admin can add a new user to the system. |
| 07 | Update User | Admin | Allows admin to update user details such as name, email, role, and status. |
| 08 | Delete User | Admin | Admin can remove a user from the system. |
| 09 | Manage Profile | Admin, Instructor, Student | All actors can update their own profile information. |
| 10 | Update Profile | Admin, Instructor, Student | Allows users to update their personal details, such as name, email, and contact information. |
| 11 | Forgot Password | Admin | Users can reset their password if they forget it. |
| 12 | Manage Lesson | Admin | Admin can create, update, or delete lessons for their courses. |
| 13 | Create Lesson | Admin | Allows instructors to create a new lesson for a course. |
| 14 | Update Lesson | Admin | Instructors can modify the content or details of an existing lesson. |
| 15 | Delete Lesson | Admin | Instructors can remove a lesson from the course. |
| 16 | Manage Course | Admin | Allows for the creation, updating, and deletion of courses. |
| 17 | Create Course | Admin | Allows for adding a new course to the system. |
| 18 | Update Course | Admin | Allows for modifying course details. |
| 19 | Delete Course | Admin | Allows for removing a course from the system. |
| 20 | Manage Assignment | Instructor | Instructors can create, update, delete, and view assignments submitted by students. |
| 21 | Create Assignment | Instructor | Instructors can add a new assignment to a class. |
| 22 | Update Assignment | Instructor | Instructors can change the details or content of an assignment. |
| 23 | Delete Assignment | Instructor | Instructors can remove an assignment from the class. |
| 24 | View Submitted Assignments | Instructor | Instructors can view assignments submitted by students for grading. |
| 25 | Grade Assignment | Instructor | Instructors can assign grades to submitted assignments. |
| 26 | Login | Admin, Instructor, Student | All users can log in to the system using their credentials. |
| 27 | Logout | Admin, Instructor, Student | All users can log out of the system. |
| 28 | Search Course | Student | Students can search for available courses. |
| 29 | Enroll in Class | Student | Students can enroll in classes they are interested in. |
| 30 | Access Course Material | Student | Students can access course materials after enrolling. |
| 31 | View Assignment | Student | Students can view assignments related to their courses. |
| 32 | Submit Assignment | Student | Students can submit completed assignments for grading. |
| 33 | View Score | Student | Students can view their grades for assignments. |
| 34 | Track Progress | Student | Students can monitor their progress in enrolled courses. |

### 2.4 Main Workflows

#### 1.2.1 Students Track Progress



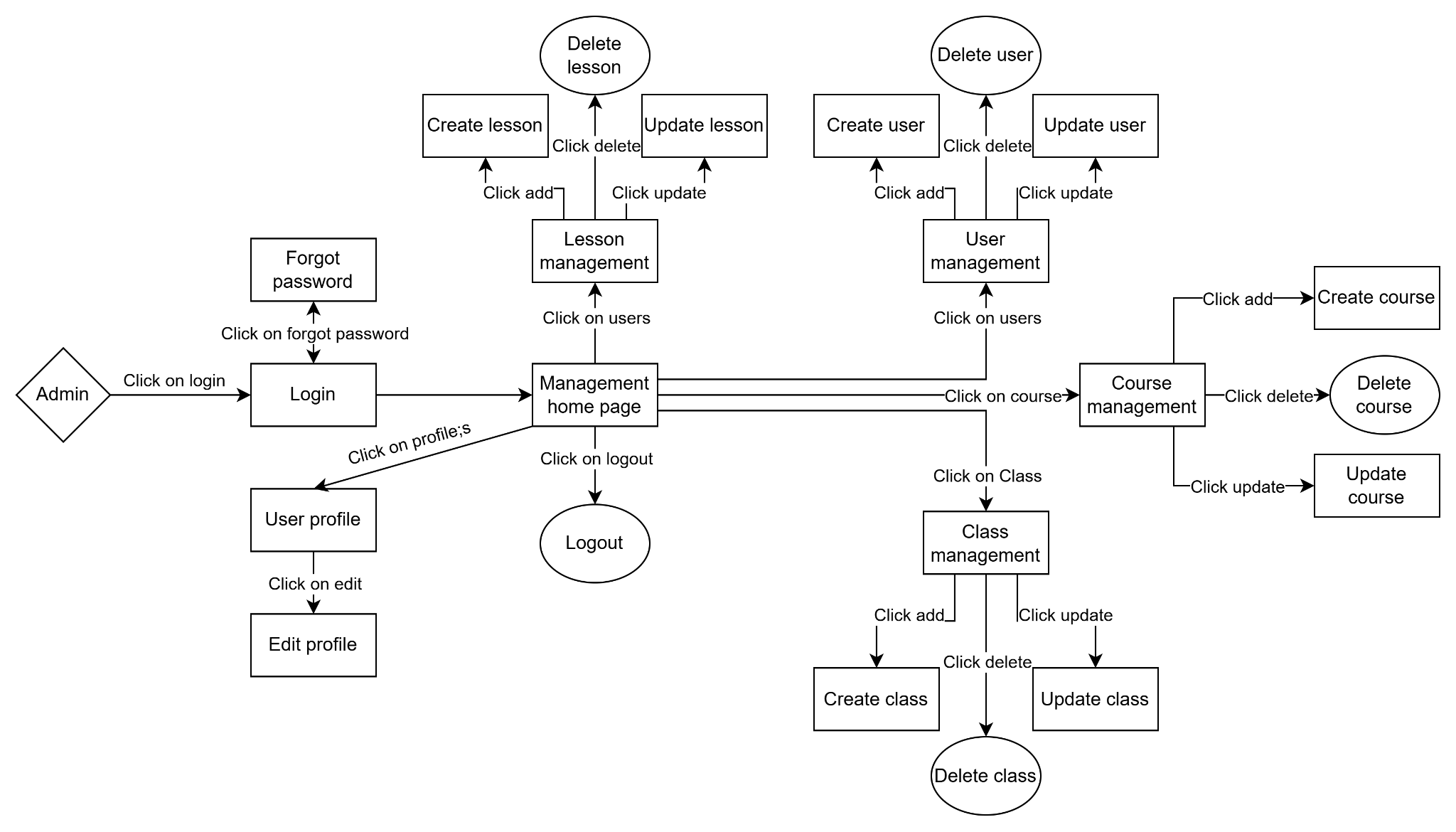
#### 1.2.2 Students do assignment



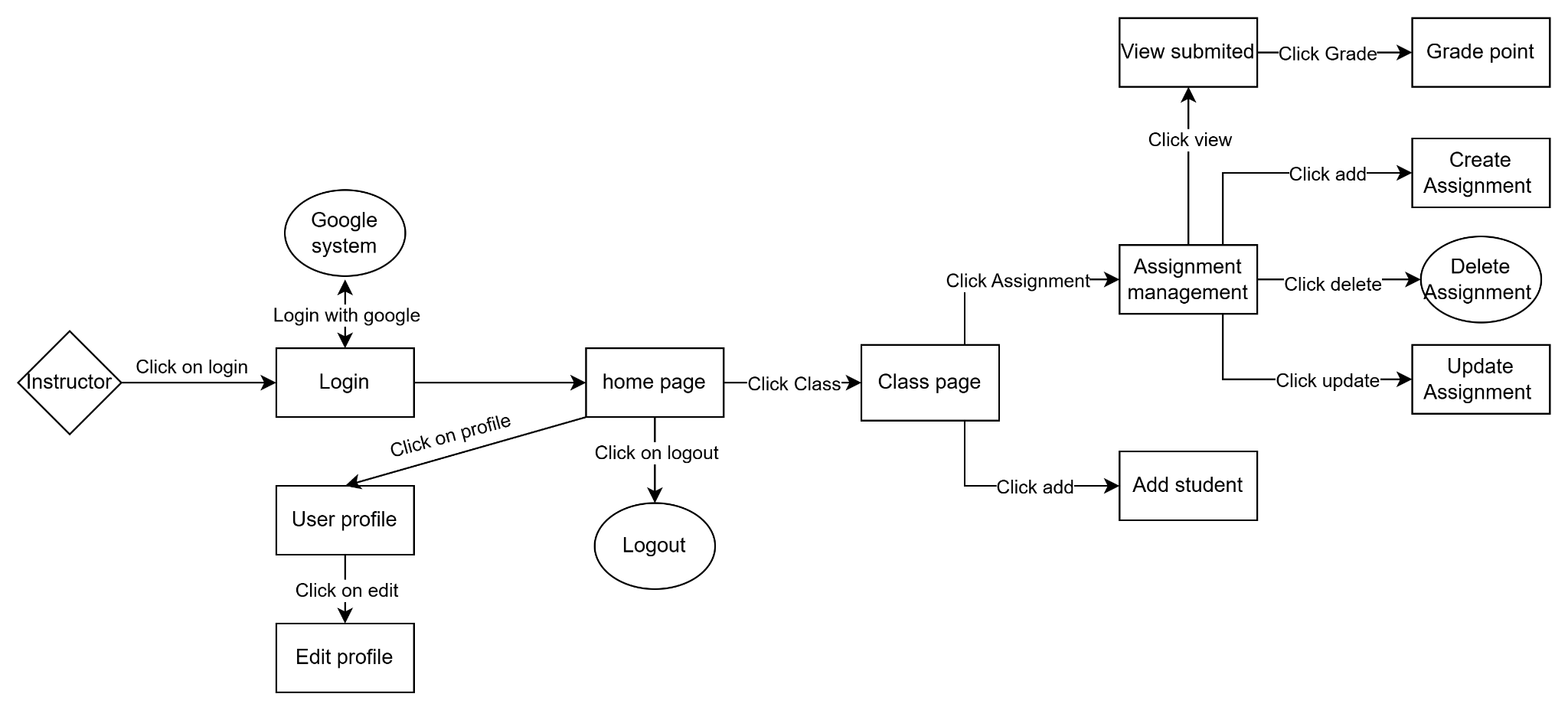
## 3. System Functionalities

### 3.1 Screens Flow

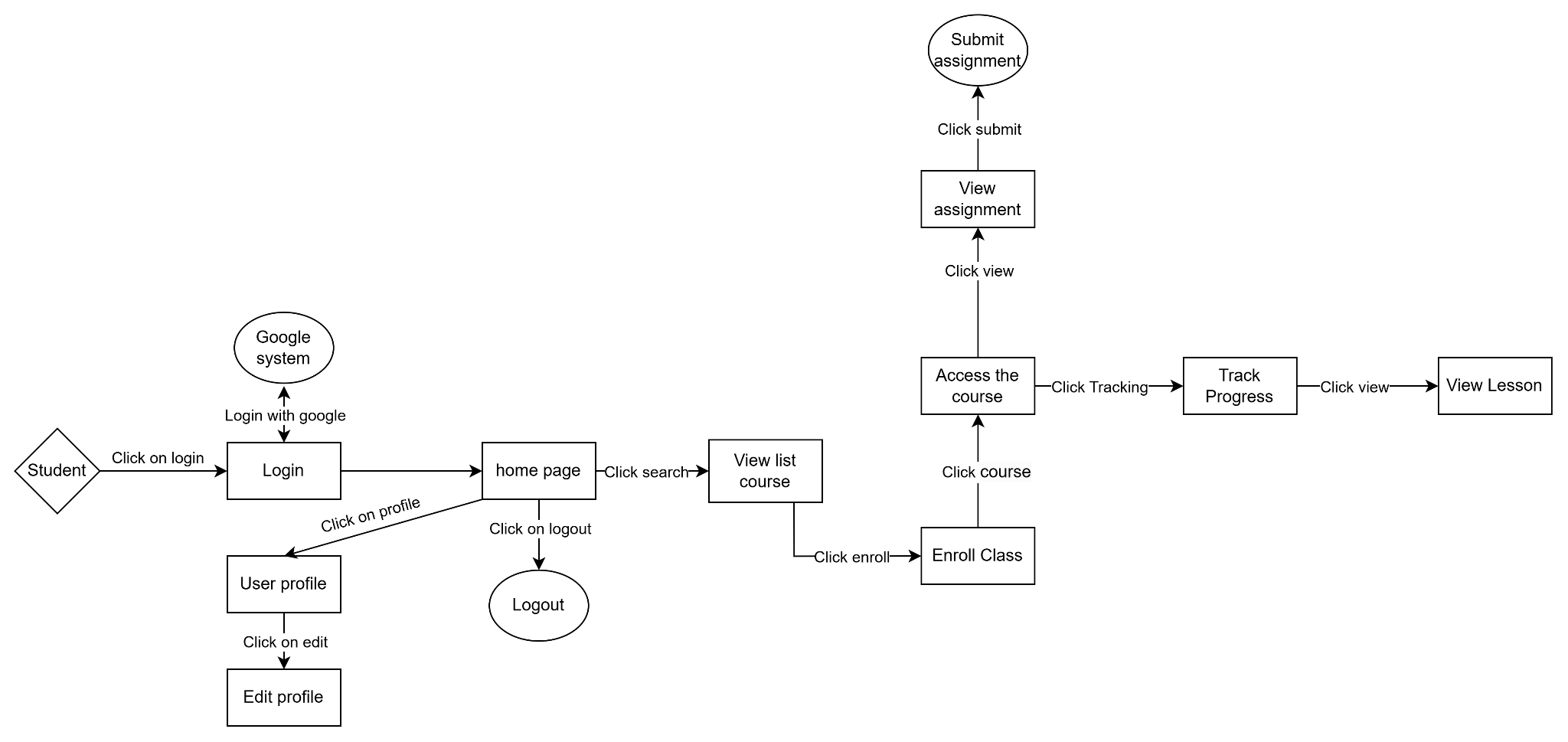
### 3.1.1 Screens Flow for admin



### 3.1.2 Screens Flow for instructor



### 3.1.3 Screens Flow for student



### 3.2 Screen Authorization

|  |  |  |  |
| --- | --- | --- | --- |
| **Screen** | **Admin** | **Instructor** | **Student** |
| Login | x | x | x |
| Home Page | x | x | x |
| User Profile | x | x | x |
| Logout | x | x | x |
| Course Management | x |  |  |
| Class Management | x |  |  |
| Lesson Management | x |  |  |
| Assignment Management |  | x |  |
| Add Student to Class |  | x |  |
| Submit Assignment |  |  | x |
| View Assignment |  |  | x |
| Track Progress |  |  | x |
| Enroll Class |  |  | x |

### 3.3 Non-UI Functions

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **System Function** | **Description** |
| 1 | Login with google | Login | Students and instructors can login with google. |

# II. Functional Requirements

## 1. Login

### 1.1 User Login

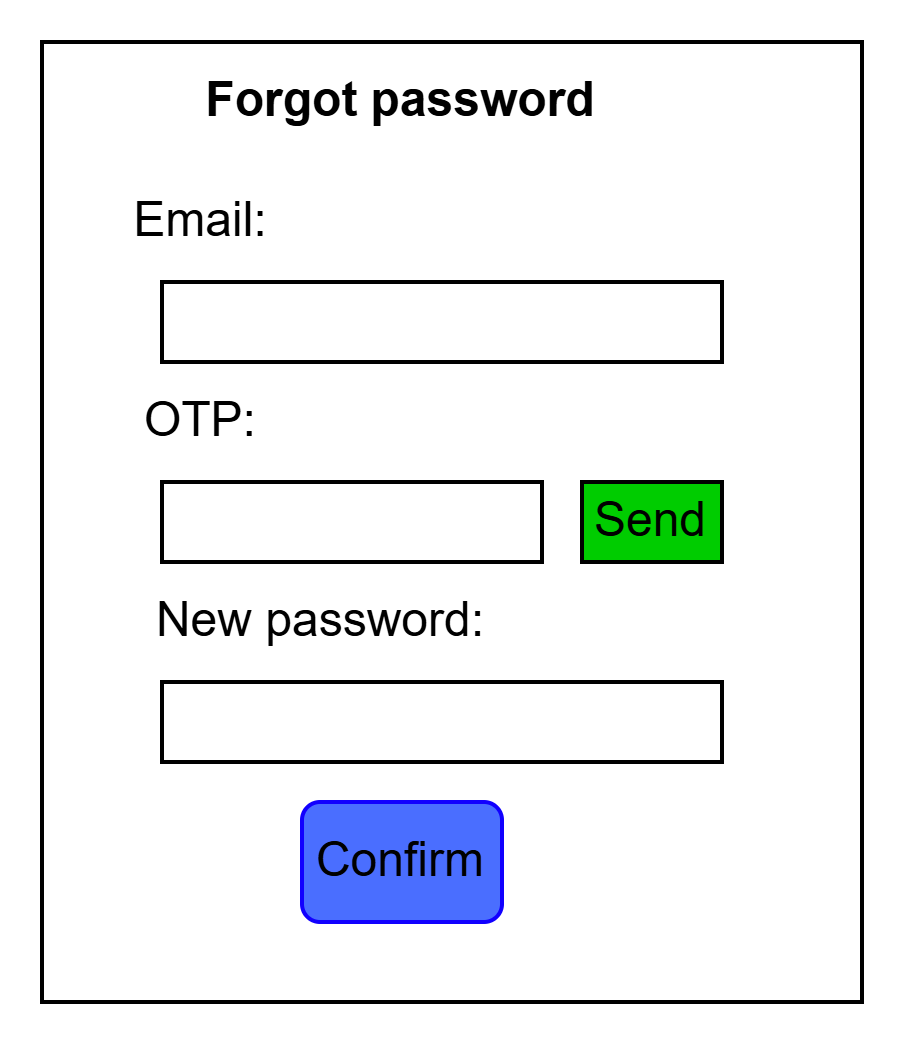
This screen allows users to be authenticated to the system screens/functionalities.

The main login screen is shown to the user in one of below cases:

* He/she clicks the Login link in the pages’ header
* He/she is accessing one of the authenticated page while not logging-in yet

|  |  |
| --- | --- |
| ***S1\_User Login screen for admin*** | ***S2\_Select account to login (with Google) for user*** |
|  |  |

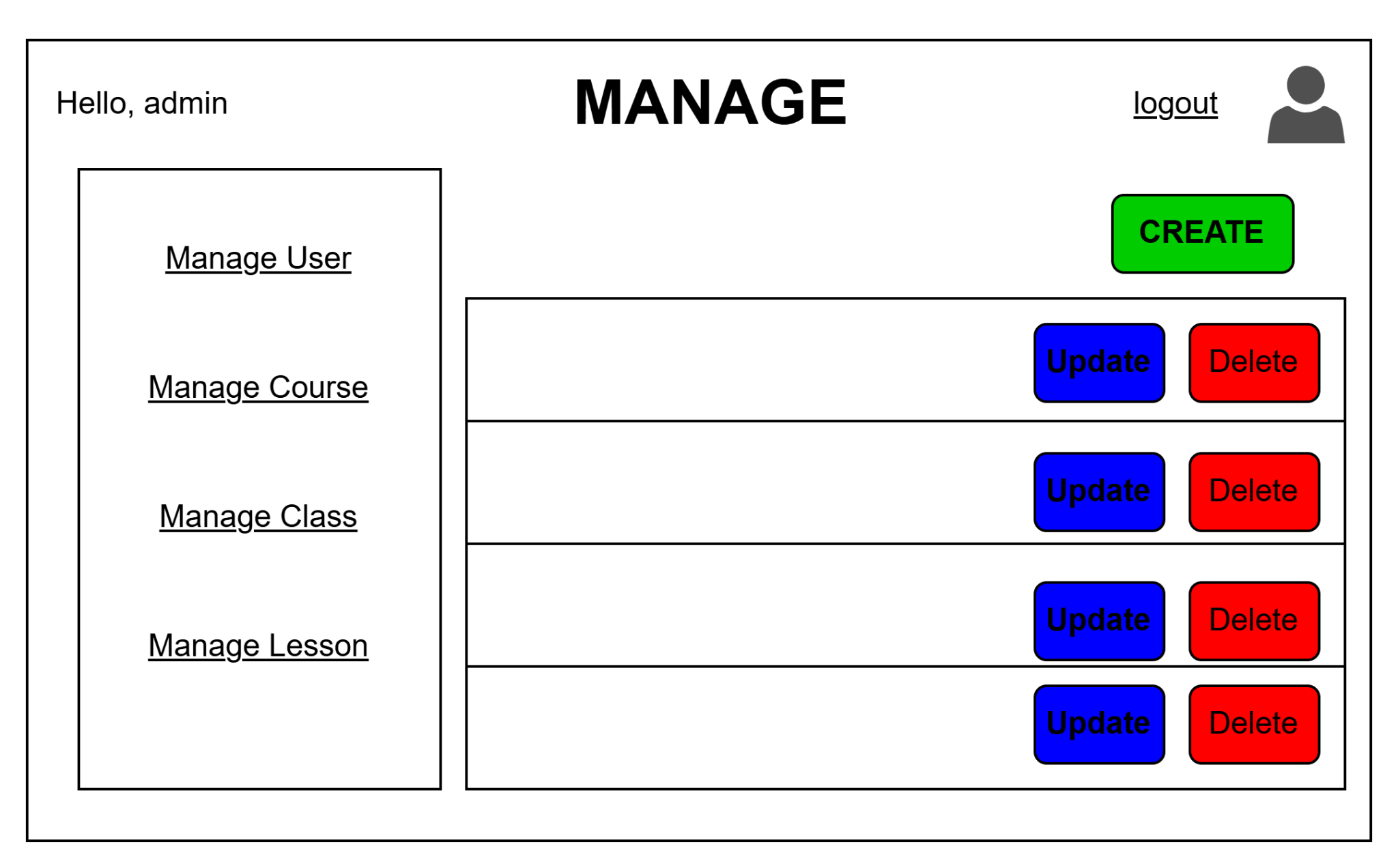
### 1.2 Forgot password



## 2. Admin management

### 2.1 Management

#### 2.1.1 Management List



**Manage User, Course, Class, and Lesson Links (Sidebar)**:

* Located on the left-hand side, this sidebar allows the administrator to navigate between different management sections, such as "Manage User," "Manage Course," "Manage Class," and "Manage Lesson."
* Clicking on each link filters the main content area to display the corresponding management actions for the selected entity.

**Create Button**:

* The "CREATE" button located on the top-right of the page allows the administrator to add a new item (user, course, class, or lesson) to the system.
* When clicked, it will redirect to a form where the admin can input details for the new entry.

**Item List Table (Main Content Area)**:

* Displays a list of the current items that can be managed (depending on the selected section).
* Each row corresponds to an item (user, course, etc.), showing details like name and status.

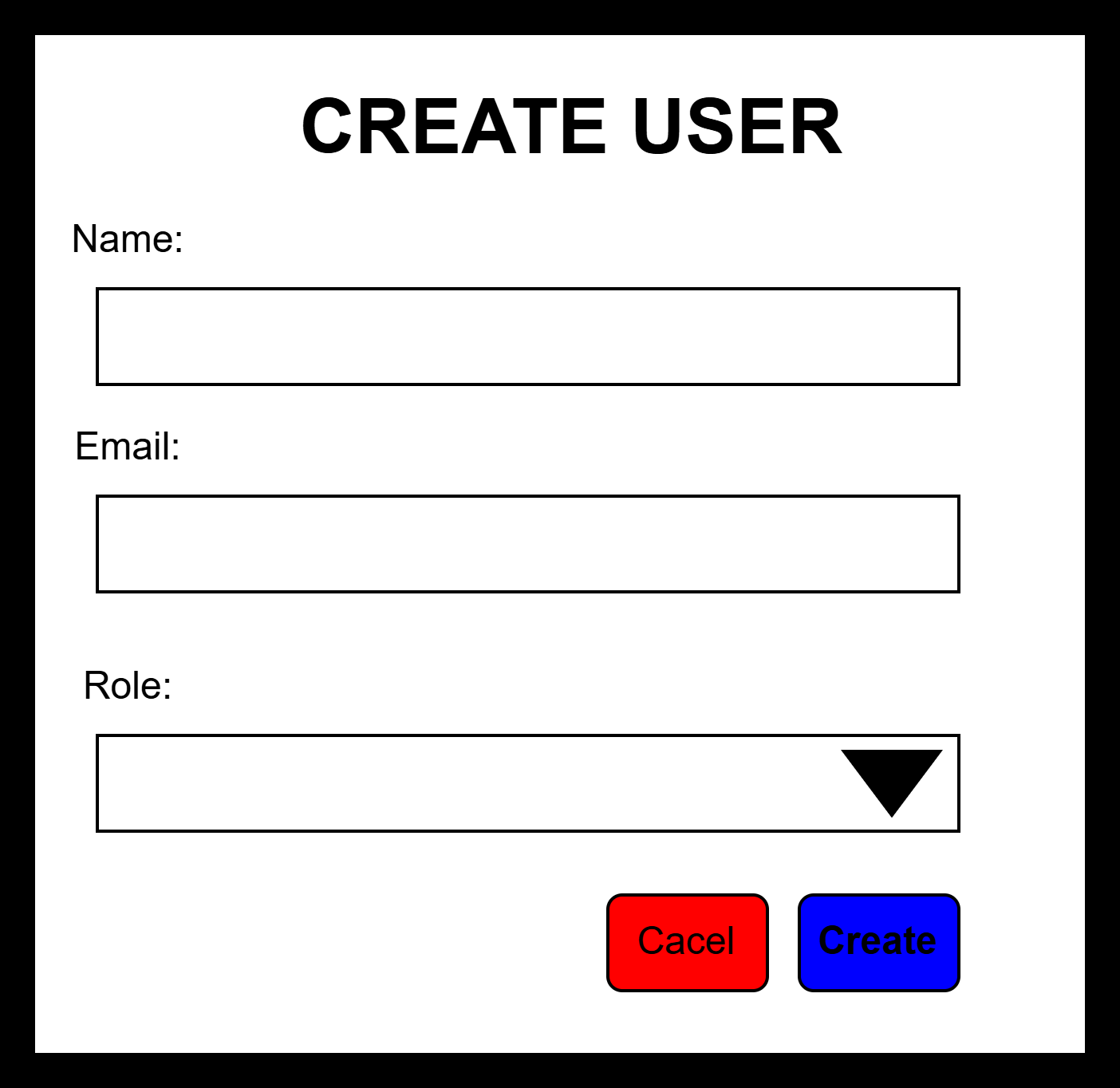
**Update and Delete Actions**:

* Each row has "Update" (blue) and "Delete" (red) buttons.
* **Update**: Allows the admin to modify details for the selected item.
* **Delete**: Enables the admin to remove the item from the system. A confirmation modal will prompt before deletion to prevent accidental removals.

**Header and User Information**:

* The top header shows a greeting (e.g., "Hello, admin") and includes a "logout" link for signing out.
* There is also a profile icon indicating that the logged-in user can access their profile settings.

#### 2.1.2 Create user



### **Form Fields:**

1. **Name:**
   * A text input field where the administrator can enter the new user's full name.
   * This field is mandatory to proceed with user creation.
2. **Email:**
   * A text input field for entering the user's email address.
   * The email serves as a unique identifier for the user and is required.
3. **Role:**
   * A dropdown menu (combobox) that allows the administrator to assign a role to the new user.
   * The dropdown contains options like "Admin," "Instructor," "Student," or other relevant roles.
   * This field is required to determine the user's permissions in the system.

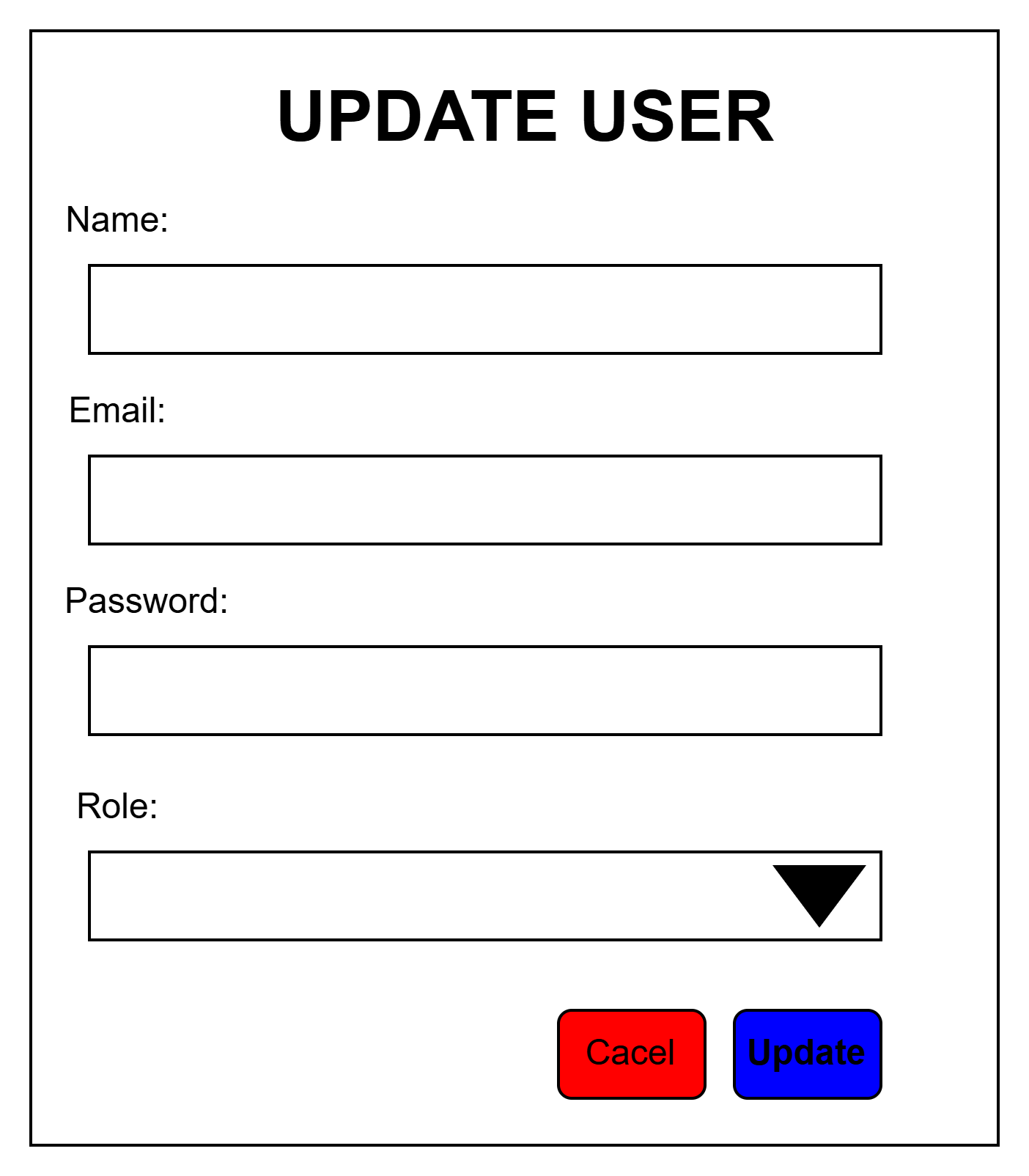
### **Action Buttons:**

1. **Cancel Button (Red):**
   * Allows the administrator to exit the form without saving any changes.
   * Clicking this button will return the admin to the previous page.
2. **Create Button (Blue):**
   * Initiates the process of creating a new user with the provided details.
   * This button is only enabled if all required fields (Name, Email, and Role) are filled in.
   * On successful submission, the new user will be added to the system, and the admin will be redirected to the user management page.

### **Form Validation:**

* The form should include validation to ensure all required fields are filled out.
* Email validation should be performed to confirm a valid email format.
* If the form is incomplete or contains invalid data, appropriate error messages should be displayed to guide the admin in correcting the input.

#### 2.1.3 Update user



### **Form Fields:**

1. **Name:**
   * A text input field where the administrator can update the user's full name.
   * This field is pre-populated with the current name of the user for editing.
2. **Email:**
   * A text input field for modifying the user's email address.
   * The existing email is shown by default, allowing for updates if necessary.
3. **Password:**
   * A text input field where the administrator can set a new password for the user.
   * If left blank, the user's existing password will remain unchanged.
4. **Role:**
   * A dropdown menu (combobox) that allows the administrator to update the user's role.
   * The dropdown contains options like "Admin," "Instructor," "Student," or other relevant roles.
   * The current role of the user is pre-selected by default.

### **Action Buttons:**

1. **Cancel Button (Red):**
   * Allows the administrator to exit the form without saving any changes.
   * Clicking this button will return the admin to the previous page.
2. **Update Button (Blue):**
   * Saves the changes made to the user's details.
   * This button will only be enabled if the form fields meet the validation requirements.
   * Upon successful submission, the updated information will be saved, and the admin will be redirected back to the user management page.

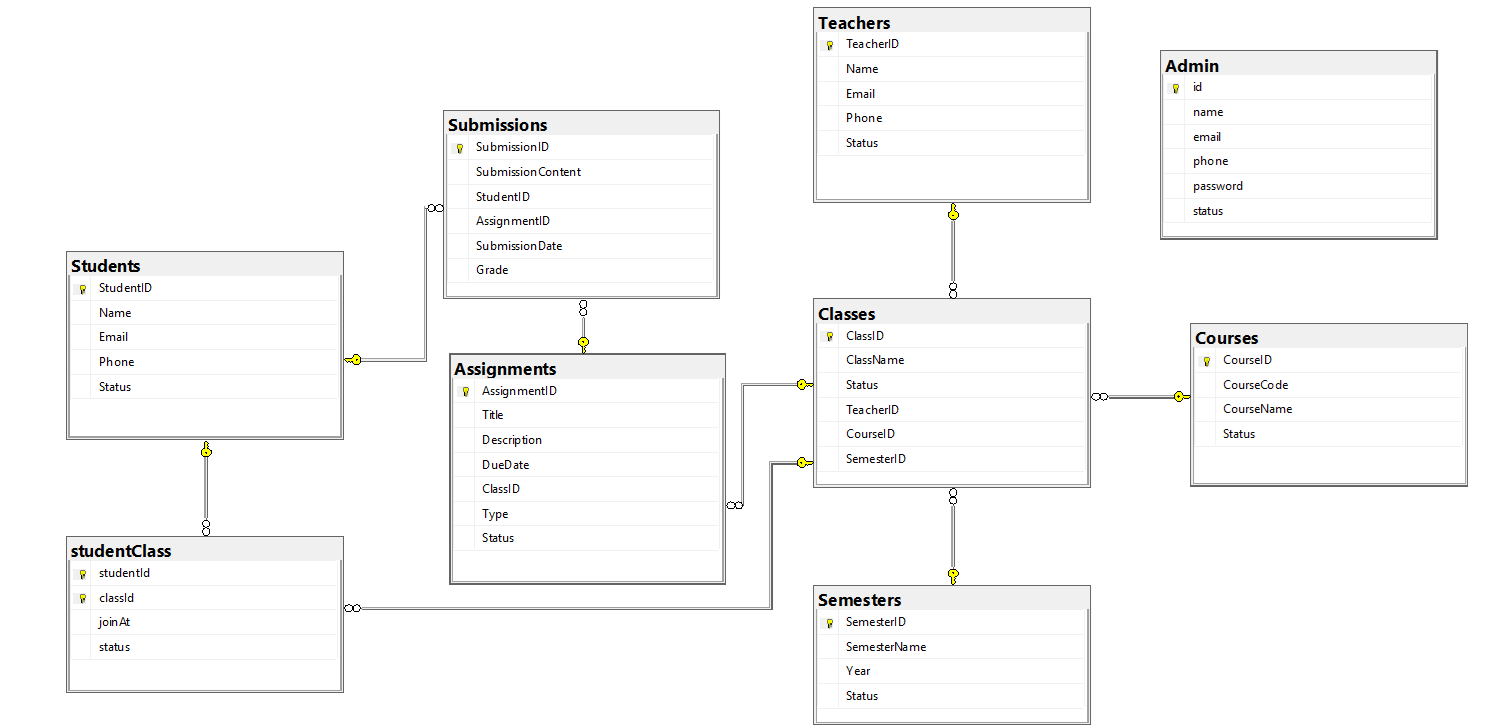
### **Form Validation:**

* The form should include validation to ensure the "Name" and "Email" fields are not left empty.
* Email format validation should be implemented to ensure the updated email is valid.
* Password validation should only be required if a new password is entered.
* If the form contains any invalid data, error messages should be displayed to guide the administrator.

# III. System Design

## 1. Database Design

### 1.1 Database Schema

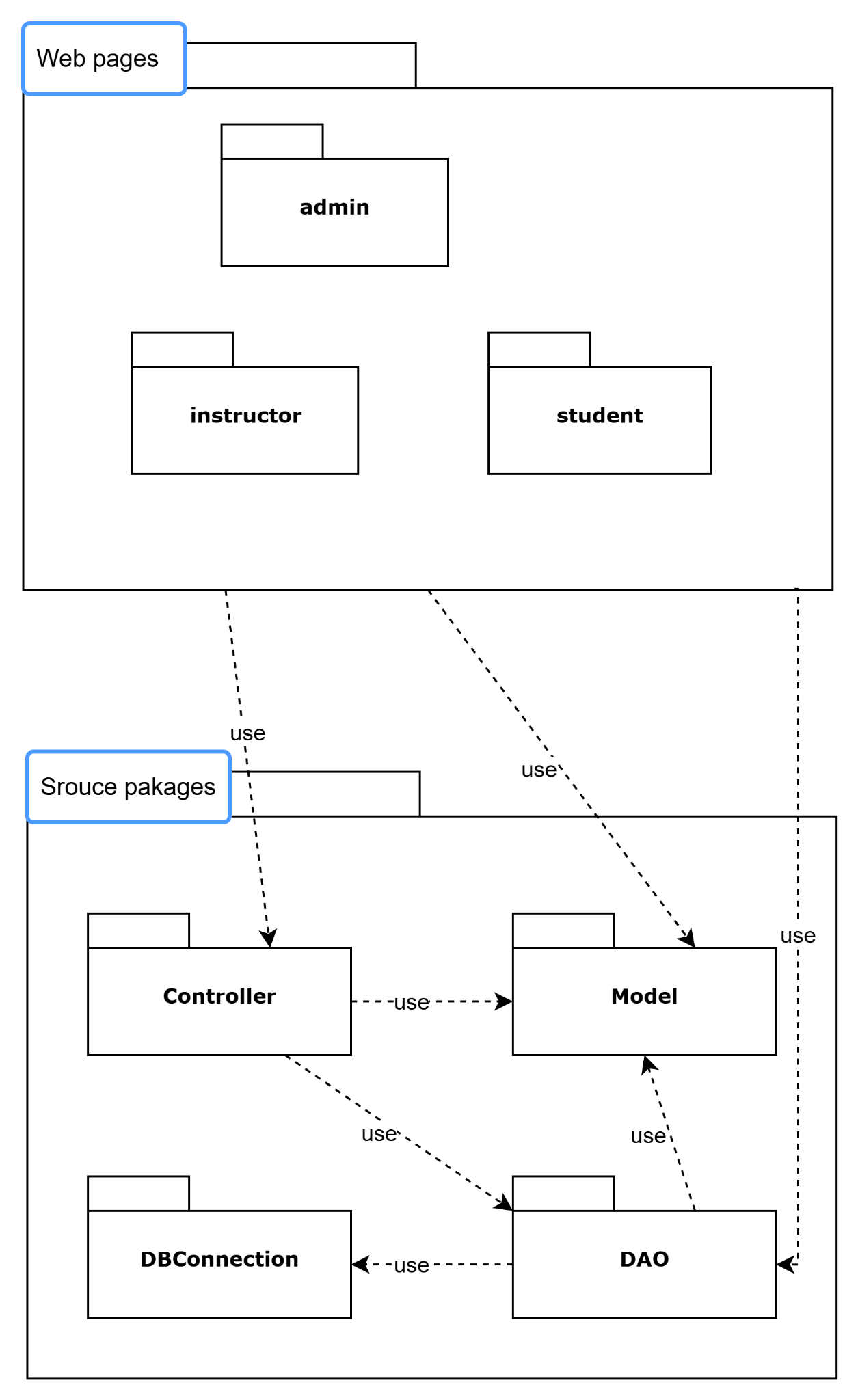


### 1.2 Table Descriptions

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| 01 | Students | Description: Contains information about students.  Primary Key: StudentID  Fields:  StudentID (Primary Key): Unique identifier for each student.  Name: Name of the student.  Email: Student's email address.  Phone: Student's contact number.  Status: Status indicating whether the student is active or inactive. |
| 02 | Teachers | Description: Contains information about teachers.  Primary Key: TeacherID  Fields:  TeacherID (Primary Key): Unique identifier for each teacher.  Name: Name of the teacher.  Email: Teacher's email address.  Phone: Teacher's contact number.  Status: Status indicating whether the teacher is active or inactive. |
| 03 | Admin | Description: Stores administrative user information.  Primary Key: id  Fields:  id (Primary Key): Unique identifier for each admin.  name: Admin's name.  email: Admin's email address.  phone: Admin's contact number.  password: Admin's login password.  status: Status indicating whether the admin account is active or inactive. |
| 04 | Classes | Description: Represents different classes.  Primary Key: ClassID  Foreign Keys: TeacherID, CourseID, SemesterID  Fields:  ClassID (Primary Key): Unique identifier for each class.  ClassName: Name of the class.  Status: Status indicating whether the class is active or inactive.  TeacherID (Foreign Key): References Teachers.  CourseID (Foreign Key): References Courses.  SemesterID (Foreign Key): References Semesters. |
| 05 | Courses | Description: Contains information about different courses.  Primary Key: CourseID  Fields:  CourseID (Primary Key): Unique identifier for each course.  CourseCode: Code for the course.  CourseName: Name of the course.  Status: Status indicating whether the course is active or inactive. |
| 06 | Semesters | Description: Represents academic semesters.  Primary Key: SemesterID  Fields:  SemesterID (Primary Key): Unique identifier for each semester.  SemesterName: Name of the semester.  Year: Academic year of the semester.  Status: Status indicating whether the semester is active or inactive. |
| 07 | Assignments | Description: Contains information about assignments.  Primary Key: AssignmentID  Foreign Keys: ClassID  Fields:  AssignmentID (Primary Key): Unique identifier for each assignment.  Title: Title of the assignment.  Description: Description of the assignment.  DueDate: Due date for the assignment submission.  ClassID (Foreign Key): References Classes.  Type: Type of assignment.  Status: Status indicating whether the assignment is active or inactive. |
| 08 | Submissions | Description: Stores submission details for assignments.  Primary Key: SubmissionID  Foreign Keys: StudentID, AssignmentID  Fields:  SubmissionID (Primary Key): Unique identifier for each submission.  SubmissionContent: Content of the student's submission.  StudentID (Foreign Key): References Students.  AssignmentID (Foreign Key): References Assignments.  SubmissionDate: Date of submission.  Grade: Grade received for the submission. |
| 09 | studentClass | Description: Represents the relationship between students and classes.  Primary Key: Composite Key (studentId, classId)  Foreign Keys: studentId (References Students), classId (References Classes)  Fields:  studentId (Primary Key, Foreign Key): References Students.  classId (Primary Key, Foreign Key): References Classes.  joinAt: Date when the student joined the class.  status: Status indicating the student's participation in the class. |

## 2. Code Packages

### 2.1 Package Diagram



### 2.2 Package Descriptions

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| 01 | Admin | Description: The admin package contains web pages and functionality specific to administrators. It is used for managing the entire system, including user accounts, course offerings, assignments, and classes.  Contents:  Web pages for managing system settings.  Administrative tools for overseeing courses, users, and reports.  Controllers and services for admin-related operations. |
| 02 | Instructor | Description: This package includes functionalities and web pages for instructors. It allows teachers to manage their classes, assignments, and track student progress.  Contents:  Web pages for instructors to create and manage assignments.  Controllers for handling instructor-related actions.  Services for grade submissions and tracking student performance. |
| 03 | student | Description: The student package is used to manage the student's interface and actions within the application. It provides access to course materials, assignments, and submission functionality.  Contents:  Web pages for students to view courses, assignments, and grades.  Controllers for handling student-related requests.  Services for managing course enrollment and assignment submission. |
| 04 | Controller | Description: Contains controller classes that act as intermediaries between the views (web pages) and the models. It handles the user requests, updates the models, and returns the appropriate responses.  Contents:  Controllers for managing user requests (e.g., login, course enrollment).  Logic to connect the web pages to the appropriate data processing services.  Routing functionalities to direct requests to the correct services. |
| 05 | Model | Description: This package includes the data models that represent the core entities within the application (e.g., users, courses, assignments). These models map to the database tables.  Contents:  Entity classes representing different objects in the system, such as Student, Course, Assignment.  Data validation logic for model properties.  Relationships and associations between different models. |
| 06 | DBConnection | Description: The DBConnection package manages the connection to the database. It is responsible for setting up and closing connections, and managing database configurations.  Contents:  Database connection configurations.  Utilities for opening and closing connections.  Exception handling for database-related operations. |
| 07 | DAO | Description: This package is used for data access operations. It contains methods to interact with the database, such as retrieving, inserting, updating, and deleting records.  Contents:  Classes for performing CRUD (Create, Read, Update, Delete) operations.  Database queries for managing the entities.  Data mapping logic to convert database rows into model objects. |