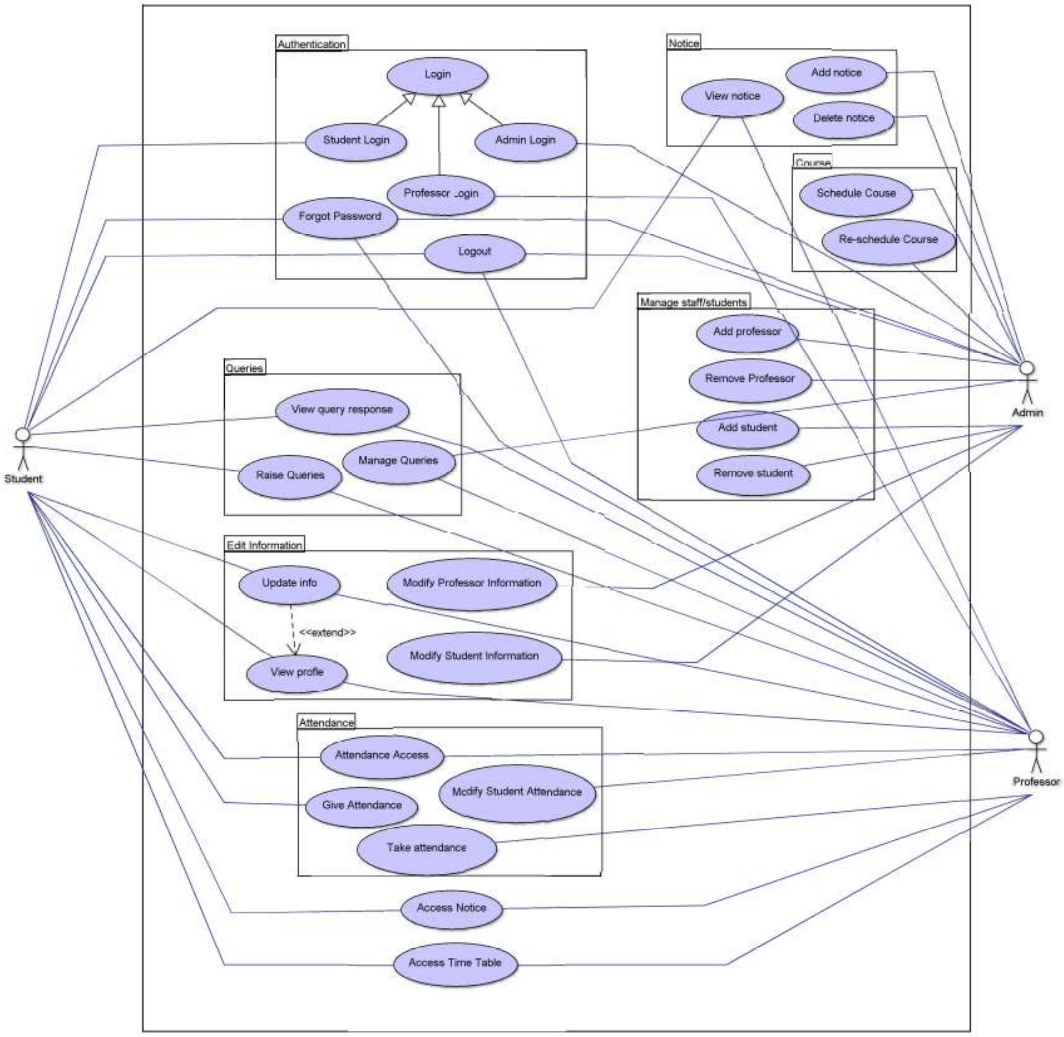
I. USE CASE MODEL



TEXT DESCRIPTION

U1: Login : Using this use case, the user can login into the website by providing the login credentials.
Scenario 1: Mainline sequence
1. User: Select ‘User login’ option.
2. System: Display prompt to enter id and password.
3. User: Enter the login details.
4 : System: Display the user’s home page.
Scenario 2: At step 4 of mainline sequence
4 : System: Displays the message that the login details are invalid.

U2: Add Professor: Using this use case, admin adds a new professor to the system.
Scenario 1: Mainline sequence
1. Admin: Select ‘Add professor’ option.
2. System: Display prompt to enter all the necessary details of the new professor.
3. Admin: Enter all the necessary details of the new staff member.
4: System: Display the message that a new professor is added.
Scenario 2: At step 4 of mainline sequence
4: System: Displays message that some input information has not been entered. The system displays a prompt to enter the missing information.
Scenario 3: At step 4 of mainline sequence
4: System: Displays the message that the professor is already present.

U3: Add Student: Using this use case, admin adds a new student to the system.
Scenario 1: Mainline sequence
1. Admin: Select ‘Add student’ option.
2. System: Display prompt to enter all the necessary details of the new student.
3. Admin: Enter all the necessary details of the new student.
4: System: Display the message that a new student is added.
Scenario 2: At step 4 of mainline sequence
4: System: Displays message that some input information has not been entered. The system displays a prompt to enter the missing information.
Scenario 3: At step 4 of mainline sequence
4: System: Displays the message that the student is already present.

U4: Take Attendance: Using this use case, the professor takes the attendance.

Scenario 1: Mainline sequence

1. Professor: Select ‘Take Attendance’ option.
2. System: Display option to start taking attendance.
3. Professor: Select the ‘start taking attendance’ option.
4. System: Display option to stop taking attendance.
5. Professor: Select the ‘stop taking attendance’ option.
6. System: Attendance has been successfully taken.

U5: Give Attendance: Using this use case, the student gives his/her attendance.

Scenario 1: Mainline sequence

1. Student: Select ‘Give Attendance’ option.
2. System: Displays a message to enter the code.
3. Student: Enter the code sent to their mail.
4: System: Display message that the attendance has been taken.

Scenario 2: At step 4 of mainline sequence

4: System: Display a prompt to enter correct code.

U6: Raise query: Using this use case, the user raises the queries to the professor or admin.

Scenario 1: Mainline sequence

1. User: Select ‘Raise query’ option.
2. System: Display prompt to select for whom the query is to be sent.
3: User: Select admin.
4. System: Display prompt to enter the query.
5. User: Enter the query.
6. System: Displays a message that the query is sent successfully.

Scenario 2: At step 3 of mainline sequence

3: User: Select professor.
4. System: Display prompt to enter the query.
5. User: Enter the query.
6. System: Displays a message that the query is sent successfully.

U7: View query response: Using this use case, the user views the response for the query given by the professor/admin.

Scenario 1: Mainline sequence

1. User: Select ‘view query response’ option.

2. System: Displays the response for the query raised.

U8: Attendance access: Using this use case, the user can view the attendance statistics of the student.
Scenario 1: Mainline sequence
1: Professor: Select the ‘Attendance access’ option.
2. System: Display prompt to enter the ID of the student.
3: Professor: Enter the student ID.
4. System: Display the details of the student attendance.
Scenario 2: At step 1 of mainline sequence
1: Student: Select the ‘Attendance access’ option.
2. System: Display the details of the student attendance.

U9:Add notice: Using this use case, the admin adds the notices to the website.
Scenario 1: Mainline sequence
1. Admin: Select ‘Add notice’ option.
2. System: Display prompt to enter the text format of the notice.
3: Admin: Enter the notice and click the add option.
4. System: Displays a message that notice has been added successfully.

U10: Access Time table: Using this use case, the user can view the time table.
Scenario 1: Mainline sequence
1. User: Select ‘Show Time table’ option.
2. System: Display the time table.

U11: View notice: Using this use case, the user can view the notices.
Scenario 1: Mainline sequence
1. User: Select ‘View notice’ option.
2. System: Display the notices.

U12:View Profile: Using this use case, the user can view their profile and can edit info if there is any mistake.
Scenario 1: Mainline sequence
1. User: Select ‘View Profile’ option.
2. System: Display the details of the user.
3: User: Modify the info if there is any mistake and select ‘modify’ option.
4. System: Display the message that the details are updated successfully.

U13:Modify Professor information: Using this use case, the admin can modify the details of the professor.

Scenario 1: Mainline sequence

1. Admin: Select ‘Modify Professor information’ option.
2. System: Display prompt to enter the ID of the professor.
3. Admin: Enter the ID of the professor whose details need to be changed.
4: System: Display the professor's details.
5. Admin: Modifies the details of the professor.
6. System: Display the message that the details are updated successfully.

Scenario 2: At step 4 of mainline sequence

4: System: Display the message that the ID is invalid.

U14:Modify Student information: Using this use case, the admin can modify the details of the student.

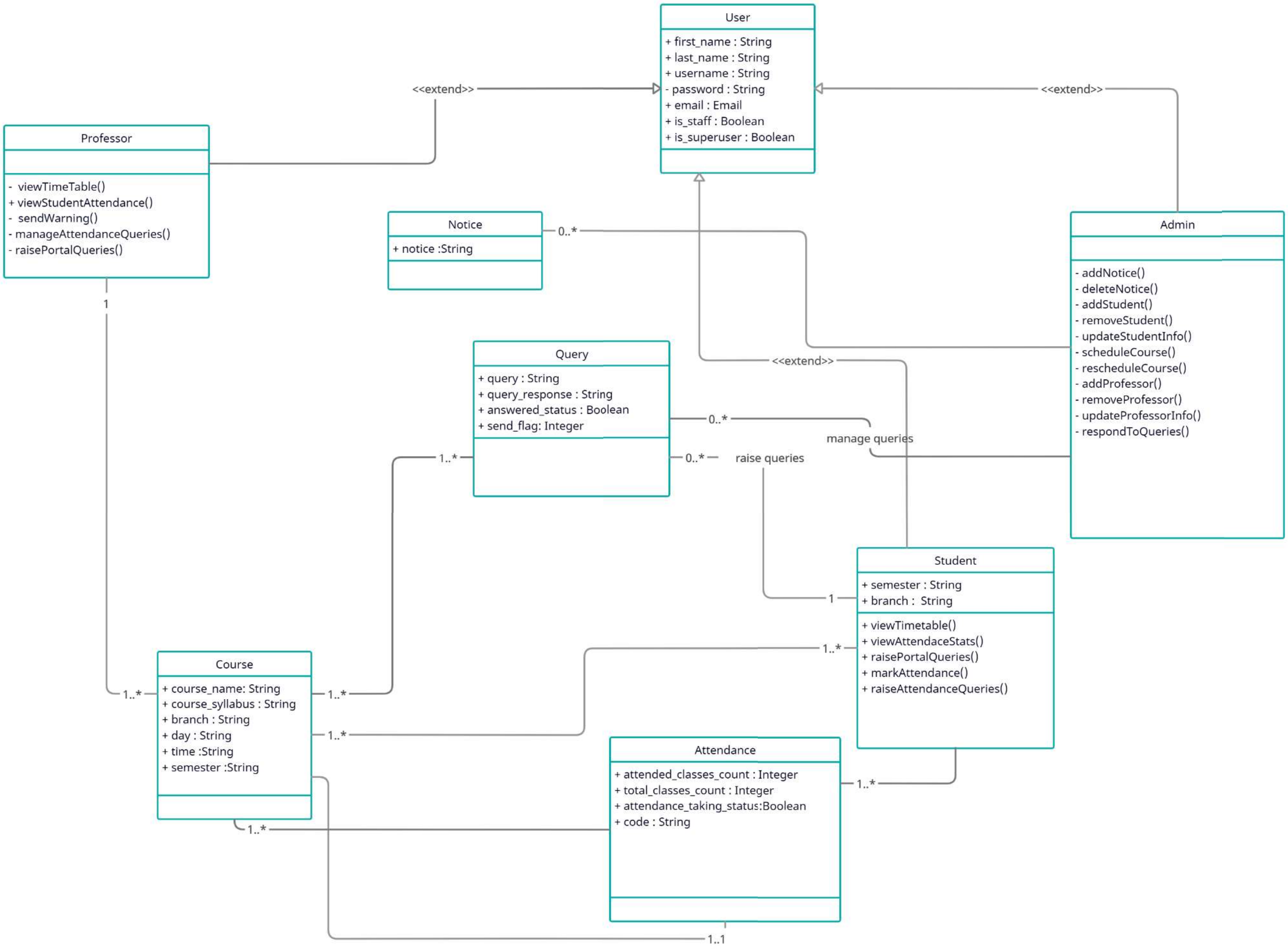
Scenario 1: Mainline sequence

1. Admin: Select ‘Modify Student information’ option.
2. System: Display prompt to enter the ID of the student.
3. Admin: Enter the ID of the student whose details need to be changed.
4: System: Display the student’s details.
5. Admin: Modifies the details of the student.
6. System: Display the message that the details are updated successfully.

Scenario 2: At step 4 of mainline sequence

4: System: Display the message that the ID is invalid.

II. UML CLASS DIAGRAM



Classes, Member variables and Member functions

1. **User**

   Variables:
   - **first_name**
   - **last_name**
   - **username**
   - **email**
   - **password**
   - **is_staff**
   - **is_superuser**

2. **Professor**

   Functions:
   - **viewTimeTable()**
   - **viewStudentAttendance()**
   - **sendWarning()**
   - **manageAttendanceQueries()**
   - **raisePortalQueries()**

3. **Student**

   Variables:
   - **semester**
   - **branch**

   Functions:
   - **viewTimeTable()**
   - **raisePortalQueries()**
   - **markAttendance()**
   - **raiseAttendanceQueries()**
   - **viewAttendanceStats()**

4. **Admin**

   Functions:
   - **addNotice()**
   - **deleteNotice()**
   - **addStudent()**
   - **removeStudent()**

- **updateStudentInfo()**
- **updateProfessorInfo()**
- **addProfessor()**
- **removeProfessor()**
- **scheduleCourses()**
- **rescheduleCourses()**

5. **Query**

   Variables:
   - **query**
   - **query_response**
   - **send_flag**
   - **response_status**

6. **Notice class:**

   Variables:
   - **notice**

7. **Attendance class:**

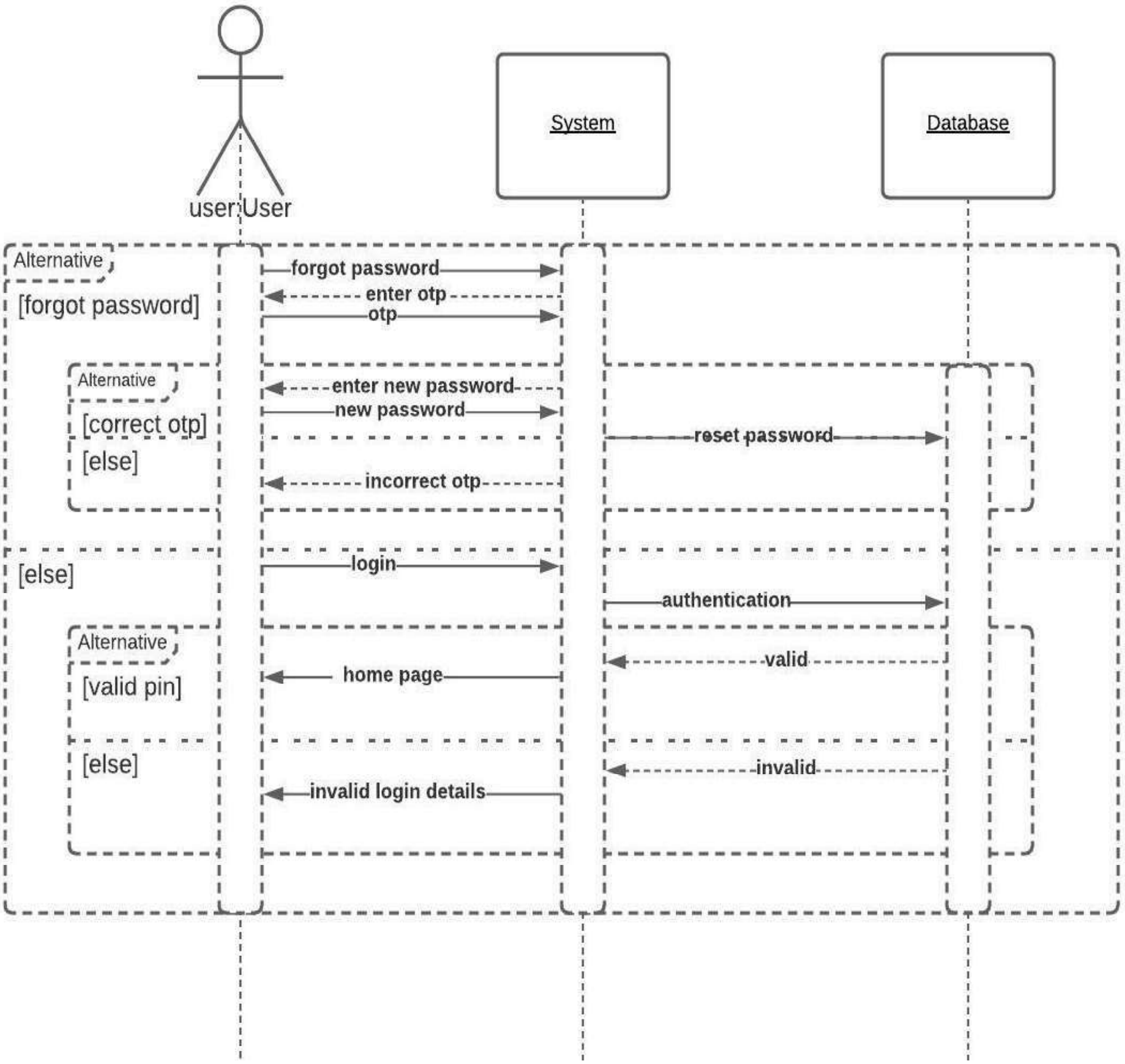
   Variables:
   - **attended_classes_count**
   - **total_classes_count**
   - **attendance_taking_status**
   - **code**

8. **Course class:**

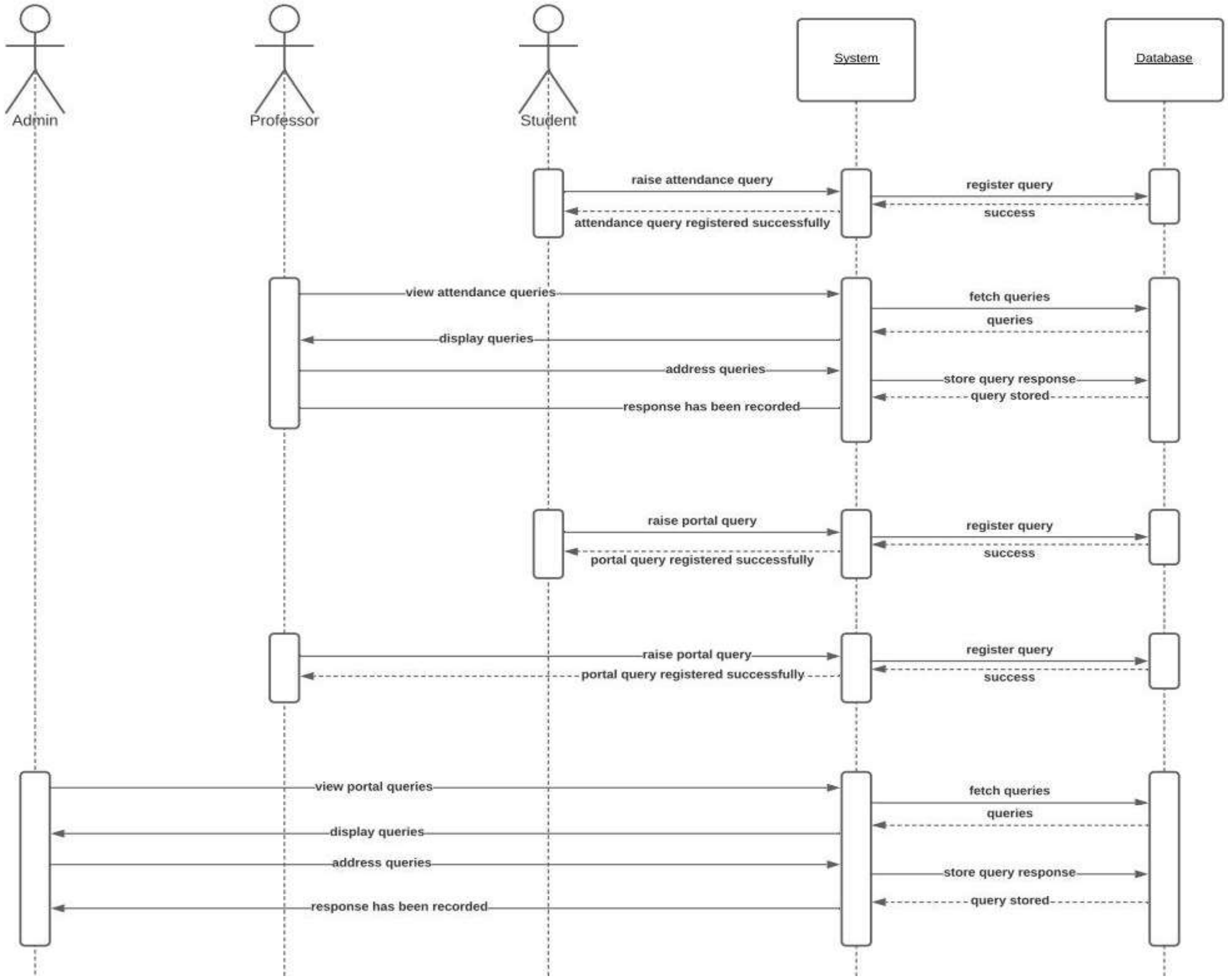
   Variables:
   - **course_name**
   - **course_syllabus**
   - **day**
   - **time**
   - **branch**
   - **semester**

III. UML SEQUENCE DIAGRAMS

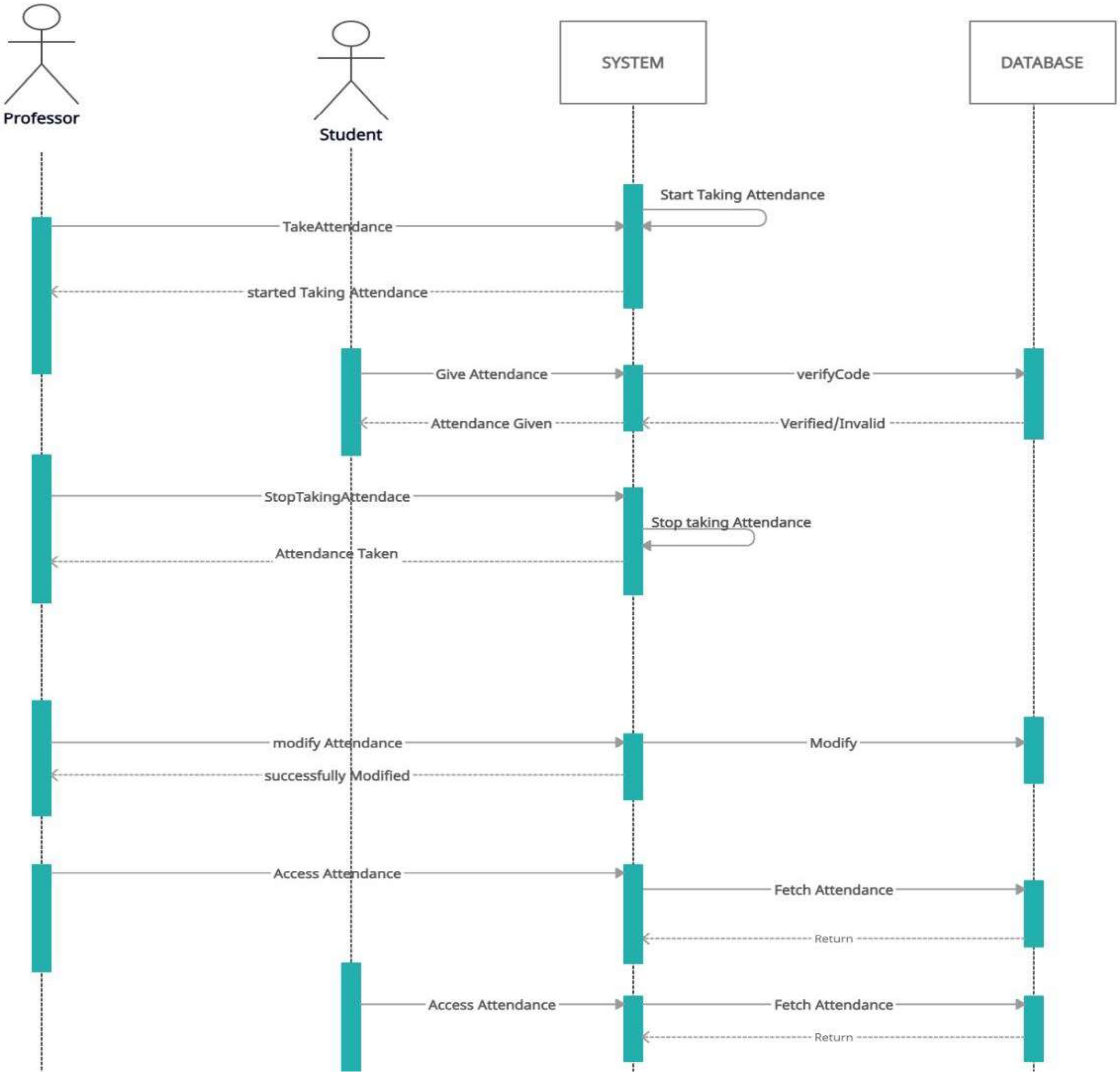
- **Login**



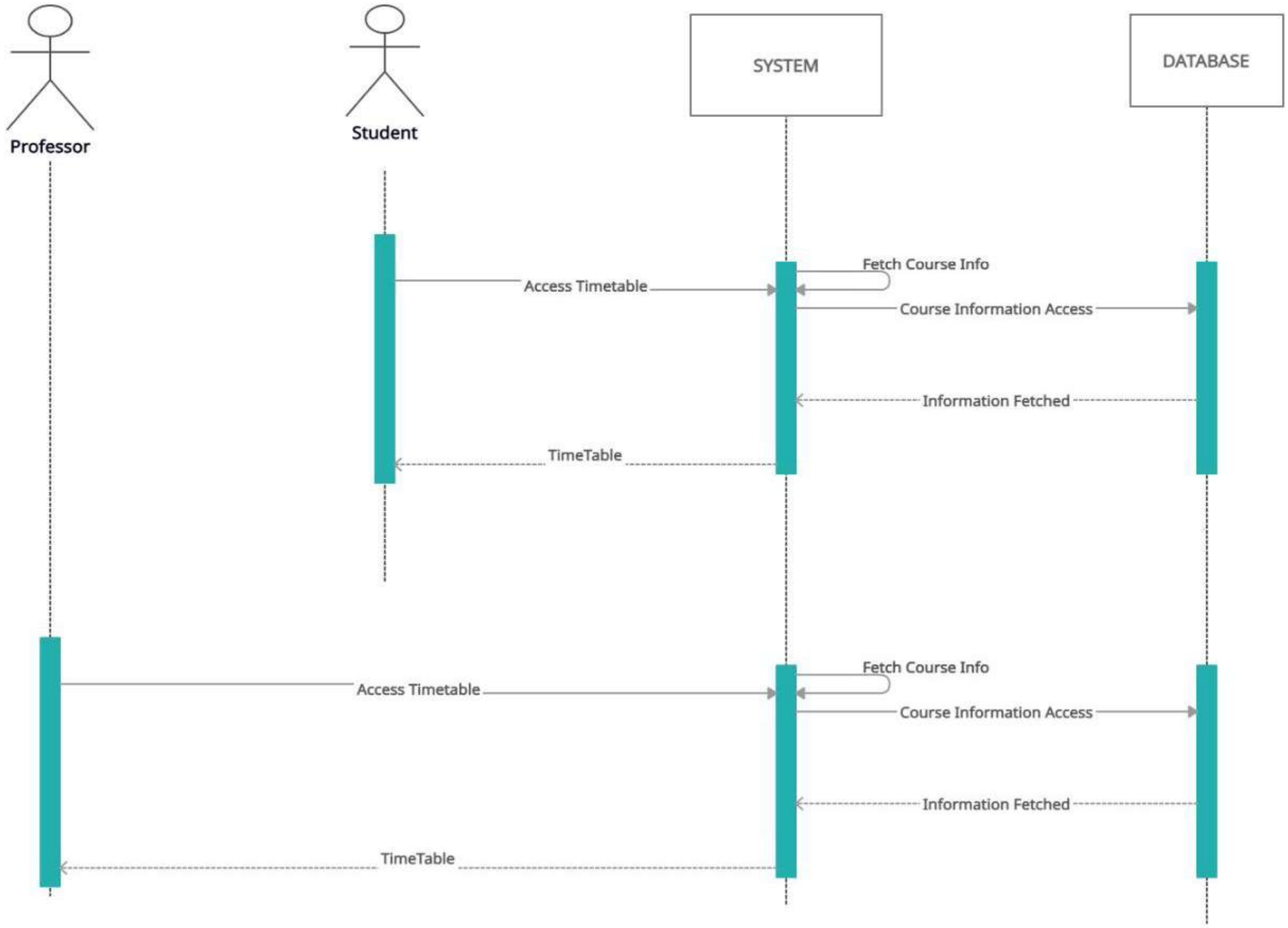
- **Queries**


Admin
Professor
Student
System
Database
raise attendance query
register query
success
attendance query registered successfully
view attendance queries
fetch queries
queries
display queries
address queries
store query response
query stored
response has been recorded
raise portal query
register query
success
portal query registered successfully
raise portal query
register query
portal query registered successfully
success
view portal queries
fetch queries
queries
display queries
address queries
store query response
query stored
response has been recorded

- **Attendance**


Professor
Student
SYSTEM
DATABASE
Start Taking Attendance
TakeAttendance
started Taking Attendance
Give Attendance
verifyCode
Attendance Given
Verified/Invalid
StopTakingAttendace
Stop taking Attendance
Attendance Taken
modify Attendance
Modify
successfully Modified
Access Attendance
Fetch Attendance
Return
Access Attendance
Fetch Attendance
Return

- **Timetable**


Professor
Student
SYSTEM
DATABASE
Access Timetable
Fetch Course Info
Course Information Access
Information Fetched
TimeTable
Access Timetable
Fetch Course Info
Course Information Access
Information Fetched
TimeTable