FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGIES

FALL 2023

FINAL EXAMINATION

COURSE TITLE:	Software Design and Modelling
COURSE CODE:	SE3140
INSTRUCTOR:	Engr. Mangong Clement
NAME	NDE HURICH DILAN
MATRICULE:	ICTU20223351
EMAIL	Nde.dilan@ictuniversity.edu.cm

1. Name the four stakeholders you meet with in order to clarify your needs. 4 x 0.75 mks
   - **Instructors**: They play a key role in the course allocation process, providing feedback and confirming their schedules.
   - **Students**: They are directly impacted by the timetable and course allocation, and their feedback is essential for ensuring an effective schedule.
   - **Timetable Officer/Administrator**: Responsible for coordinating and finalizing the timetable and course allocation process.
   - **Administration**: They oversee the entire process, considering institutional constraints and ensuring fairness in course allocation and timetable scheduling.
2. List the user' requirement for the system that has to be developed, preferably in ten or more. 10 x 1/2 mks
   - ✓ Ability for instructors to receive course assignments and provide feedback.
   - ✓ Timely distribution of course schedules to instructors and students.

- ✓ Flexible scheduling options for part-time instructors.
- ✓ Limitations on the number of courses instructors can teach.
- ✓ Accessible timetable for students.
- ✓ Procedures for addressing conflicts or objections regarding the schedule.
- ✓ Authentication system for instructors, administrators, and students.
- ✓ Secure account sign-up and sign-in process.
- ✓ Support for undergraduate student levels one to three.
- ✓ Compliance with institutional constraints and regulations.

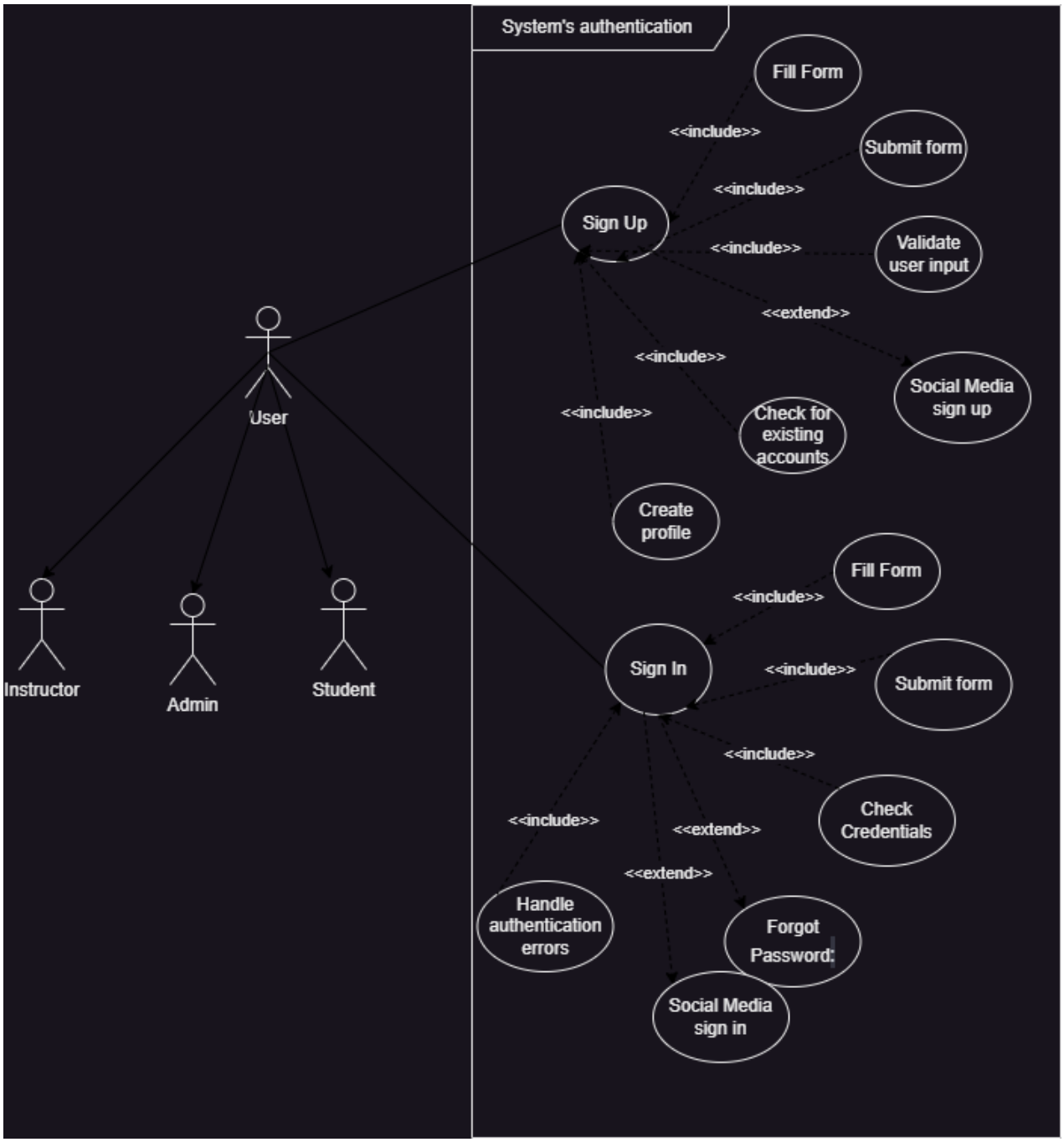
3. List four primary actors in the user's requirement. 4 x 0.75 mks

- Instructor
- Student
- Timetable Officer/Administrator
- Administration

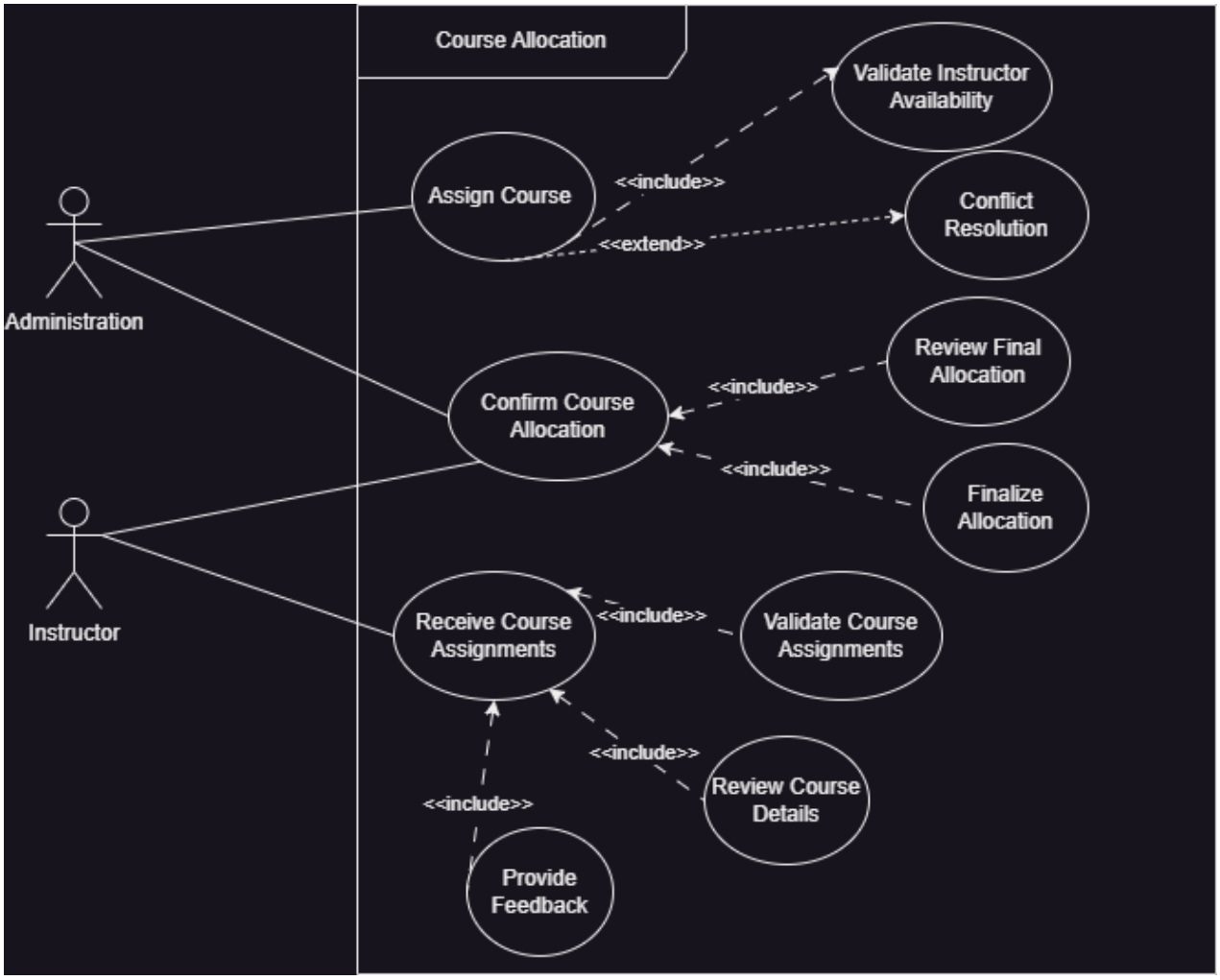
4. List four secondary actors in the user's requirements. 2 x ½ mks

- Student President: Facilitates student objections or feedback regarding the schedule.
- System Administrator: Manages system configurations and maintenance.

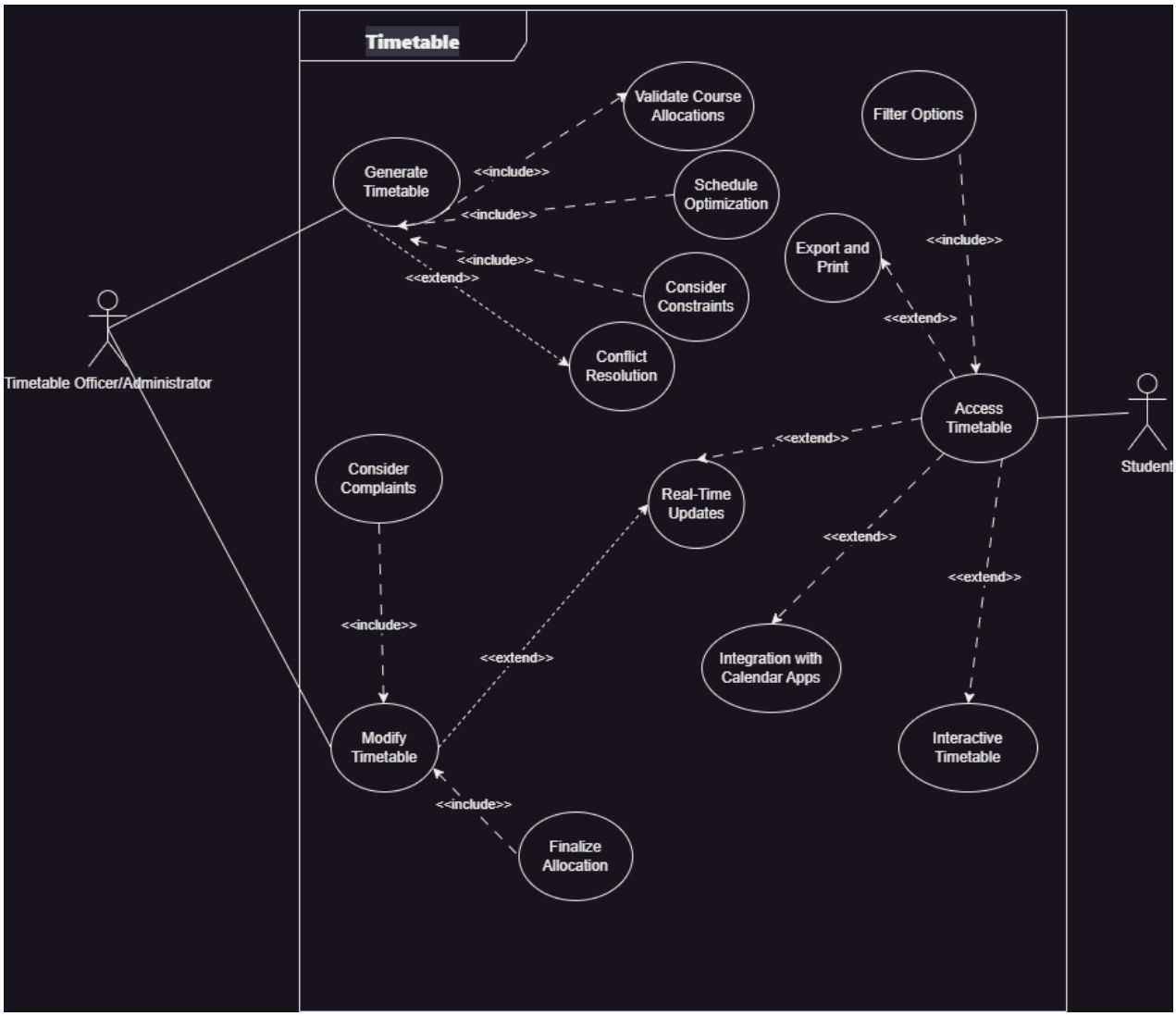
5. Create the system's authentication use case diagram, which should include use cases for account sign-up and sign-in. 4mks



6. Create the use case diagram for the system that will be designed for course allocation. 4mks



7. Create the use case diagram for the time table. 4 mks



8. Write the use case descriptions for the course allocation, timetable, and authentication of two chosen use cases. 2 x 2mks

Course Allocation System:

1. Receive Course Assignments:

- **Use Case Name:** Receive Course Assignments
- **Actors:** Instructor, Administration
- **Description:**
    - **Instructor Perspective:**
        - Instructors receive preliminary course assignments from the administration approximately three weeks before the academic year commences.

        - They review the assigned courses, considering their expertise, availability, and workload capacity.
        - Instructors provide feedback to the administration regarding the assigned courses, expressing preferences, rejecting assignments if necessary, or requesting adjustments.
    - **Administration Perspective:**
        - The administration prepares and distributes preliminary course assignments to instructors based on institutional needs and instructor qualifications.
        - They await feedback from instructors regarding their course assignments, considering their input for further adjustments.

2. Allocate Courses:

- **Use Case Name:** Allocate Courses
- **Actors:** Administration
- **Description:**
    - The administration collects feedback from instructors regarding their course assignments and incorporates their preferences and constraints into the allocation process.
    - They consider various factors such as instructor availability, course demand, classroom availability, and institutional policies during the allocation process.
    - The administration allocates courses to instructors, ensuring a balanced workload distribution and adherence to institutional guidelines.
    - Once the allocation is finalized, instructors are provided with their updated course schedules for confirmation.

Timetable Management System:

1. Generate Timetable:

- **Use Case Name:** Generate Timetable
- **Actors:** Timetable Officer/Administrator
- **Description:**
    - The timetable officer initiates the timetable generation process at the beginning of each academic term.
    - They collect course allocation data, including course schedules, instructor availability, and classroom assignments, from the course allocation system.

- Constraints such as room capacity, instructor preferences, and student enrollment levels are considered during the timetable generation process.
- The timetable officer applies scheduling algorithms to optimize the timetable, minimizing conflicts and maximizing resource utilization.
- Once the timetable is generated, it is reviewed and finalized by the timetable officer before distribution to students and instructors.

2. Access Timetable:

- **Use Case Name:** Access Timetable
- **Actors:** Student
- **Description:**
  - Students access the timetable system to view their individual schedules for the academic term.
  - Upon logging into the system, students are presented with options to view their timetable.
  - The system retrieves and displays the student's schedule, including course names, instructors, class timings, and classroom locations.
  - Students have the ability to customize their timetable view by filtering courses based on criteria such as course code, instructor name, or day of the week.
  - The timetable can be exported to various formats or printed for offline reference.
  - Any changes or updates to the timetable are reflected in real-time, and students are notified accordingly.

Authentication System:

1. Sign Up:

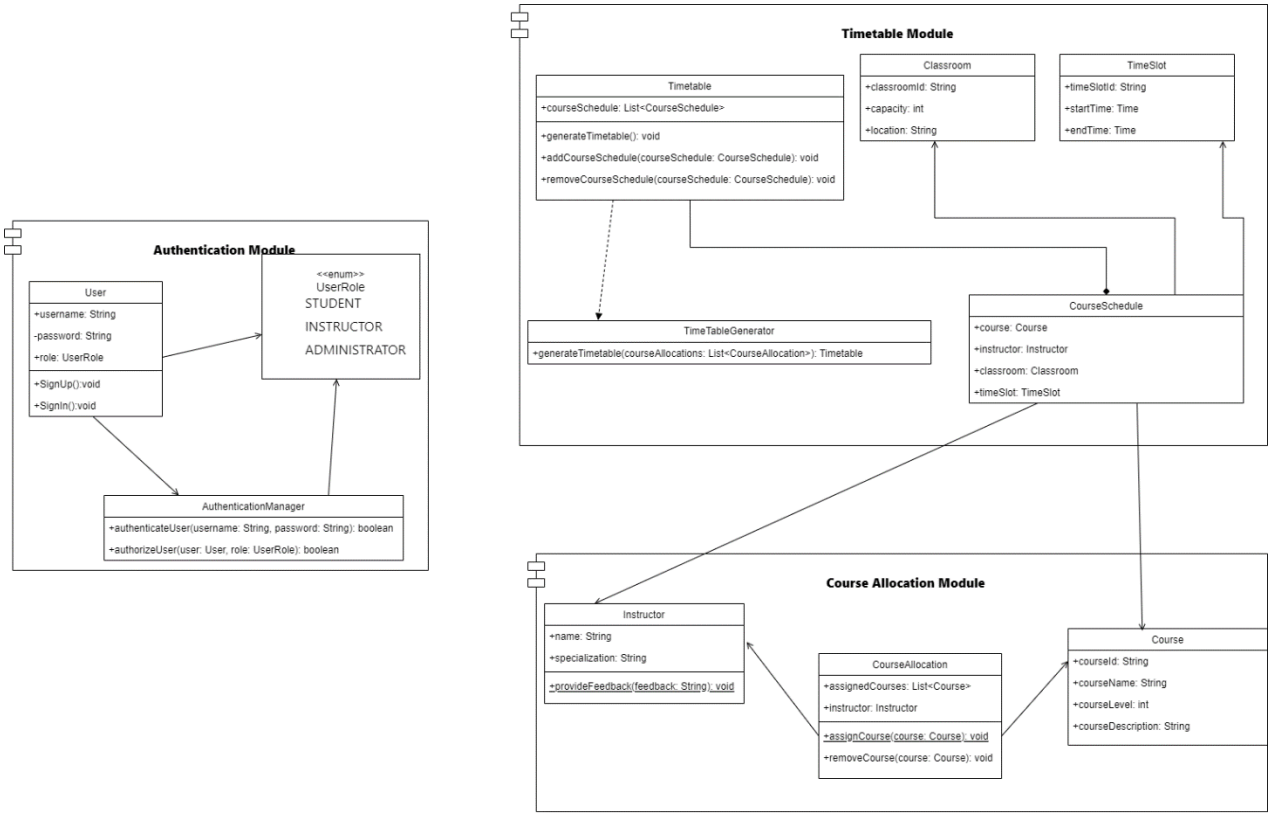
- **Use Case Name:** Sign Up
- **Actors:** User
- **Description:**
  - Users initiate the sign-up process by providing required information such as email address, password, and personal details.
  - The system validates the provided information, ensuring that it meets the required criteria and is not associated with an existing account.

- Upon successful validation, the system creates a new user account and stores the provided information securely.
- Users may be prompted to verify their email address through a confirmation link sent to their registered email.
- Upon confirmation, the sign-up process is completed, and users gain access to the system's functionalities.

2. Sign In:

- **Use Case Name:** Sign In
- **Actors:** User
- **Description:**
    - Users initiate the sign-in process by providing their registered email address and password.
    - The system validates the provided credentials, ensuring that they match the information stored in the system's database.
    - Upon successful validation, users are granted access to their accounts and directed to the system's dashboard or main interface.
    - In case of invalid credentials, users are notified and prompted to re-enter their information or reset their password if necessary.
    - The sign-in session remains active until the user logs out or the session expires due to inactivity.

9. Create the class diagram that includes modules for timetables, course allocation, and authentication. 3x3 mks



10. Use a creational, structural, and behavioural pattern to improve your class diagrams. 3 x 2 mks

Creational Pattern: Singleton Pattern

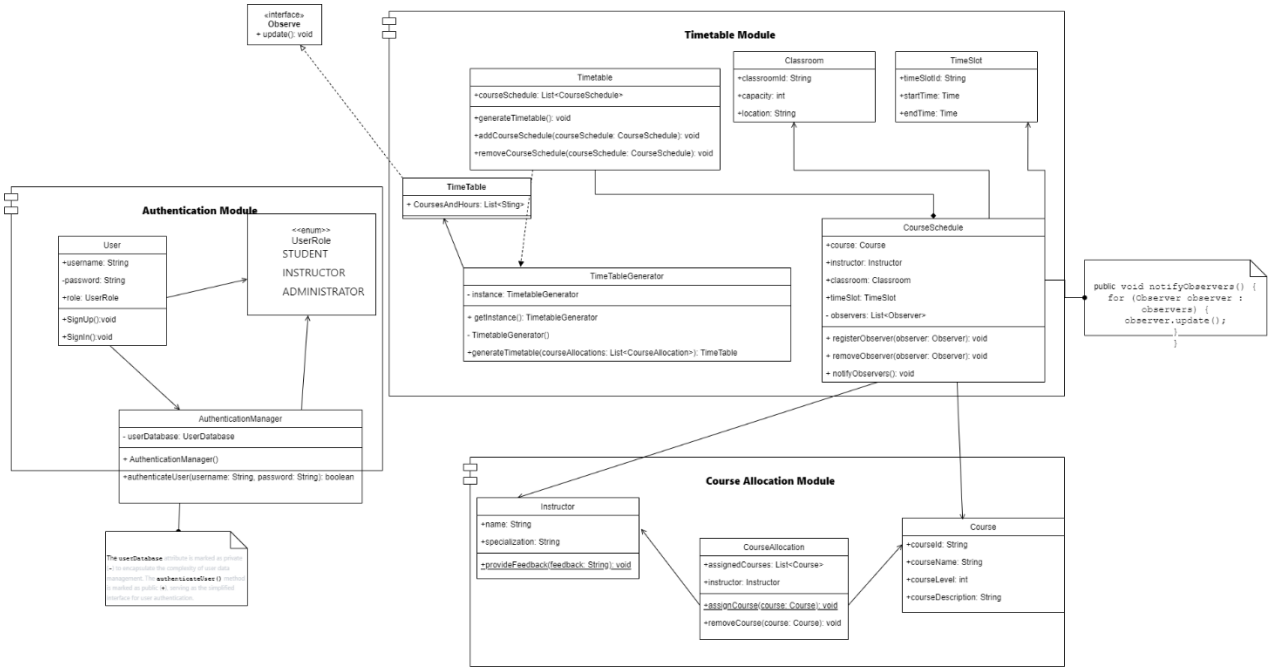
We can apply the Singleton pattern to the TimetableGenerator class to ensure that there is only one instance responsible for generating timetables throughout the system.

Structural Pattern: Facade Pattern

We can apply the Facade pattern to the AuthenticationManager class to simplify the authentication process for users.

Behavioral Pattern: Observer Pattern

We can apply the Observer pattern to the CourseSchedule class to notify the timetable whenever there is a change in the course schedule.



11. To improve your class diagram and create the final class diagrams, apply the sound design concepts – the SOLID design principles. 3 x 3mks

Without Applying SOLID Principles:

Timetable Class:

Attributes:

courseSchedules: List<CourseSchedule>

Methods:

generateTimetable(): void

addCourseSchedule(courseSchedule: CourseSchedule): void

removeCourseSchedule(courseSchedule: CourseSchedule): void

With Applying SOLID Principles:

Timetable Class:

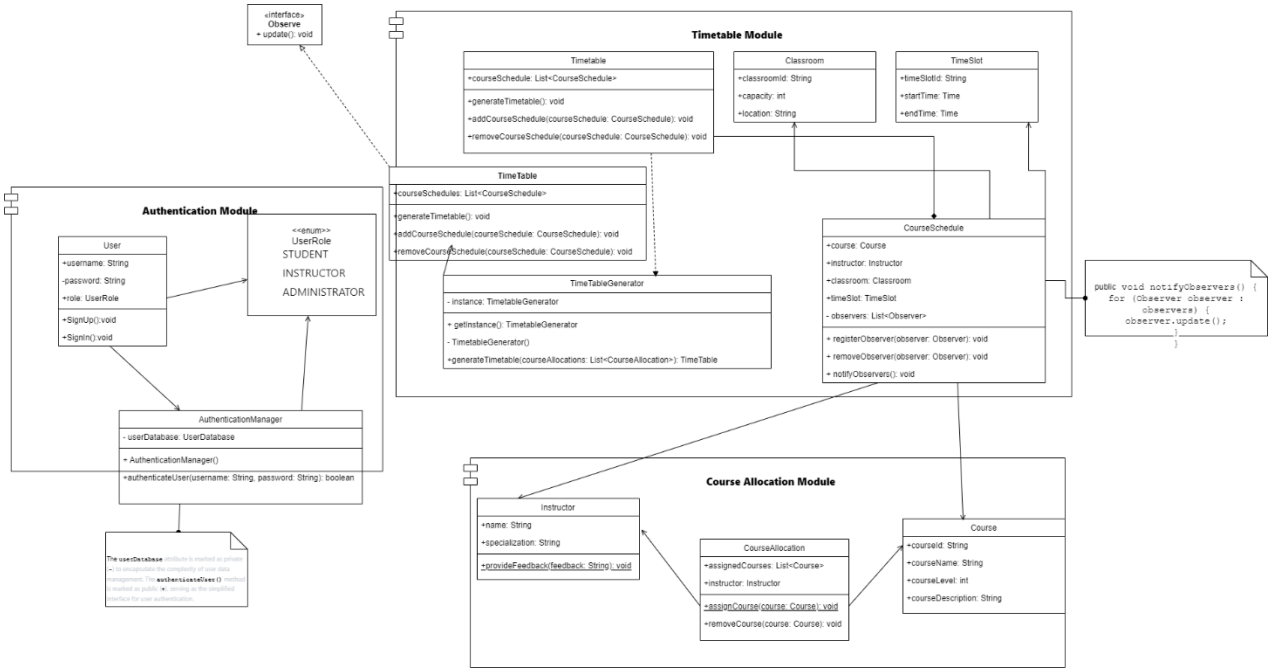
Attributes:

courseSchedules: List<CourseSchedule>

Improved Responsibilities:

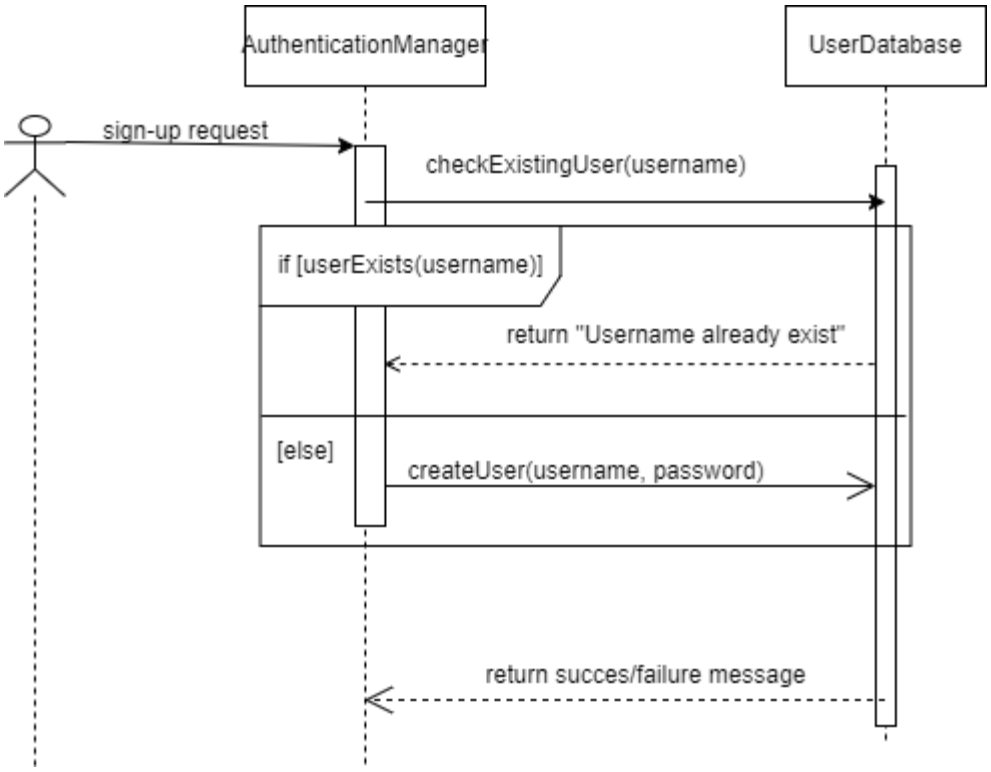
The Timetable class will focus solely on managing the timetable data, adhering to the Single Responsibility Principle.

For the Open/Closed principle we just need to ensure that our classes are open for extension and close for modification, and on our class diagram it's well apply

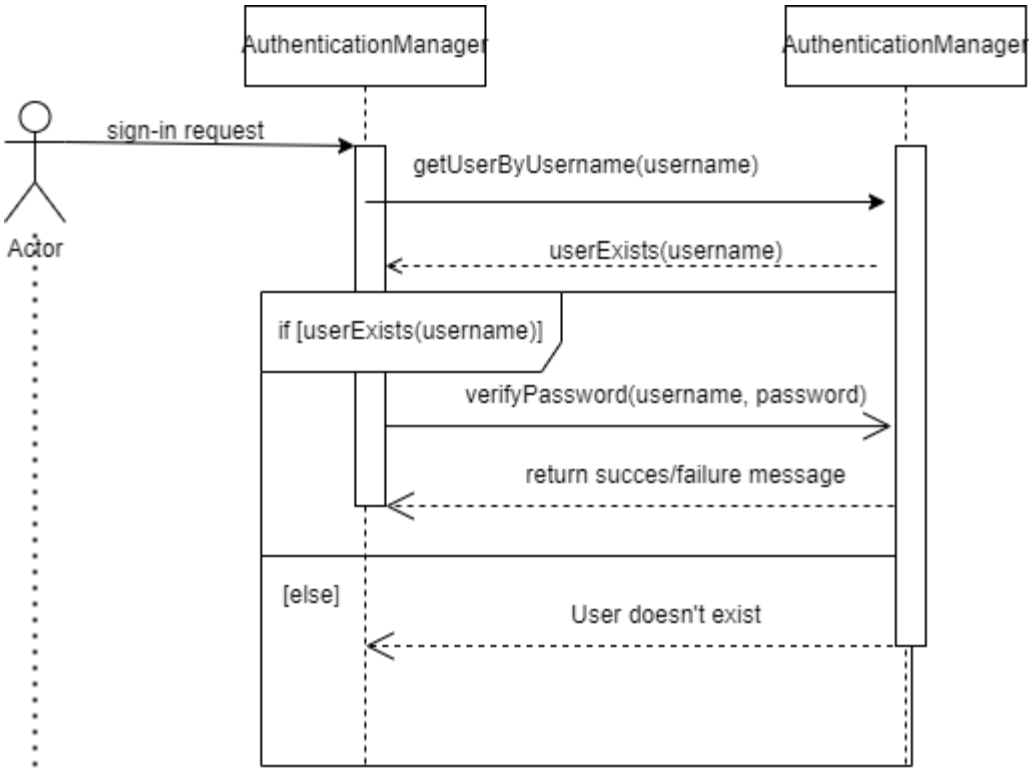


12. Create the sequence diagram using the two use cases that you have chosen from the authentication class diagram. 2 x 4mks

Sequence Diagram for "Sign Up" Use Case:



Sequence Diagram for "Sign In" Use Case



13. Create the sequence diagram's collaboration diagram. 2 x 3mks

14. Create the state machine diagram using two selected actors. 3mks

ALL THE BEST