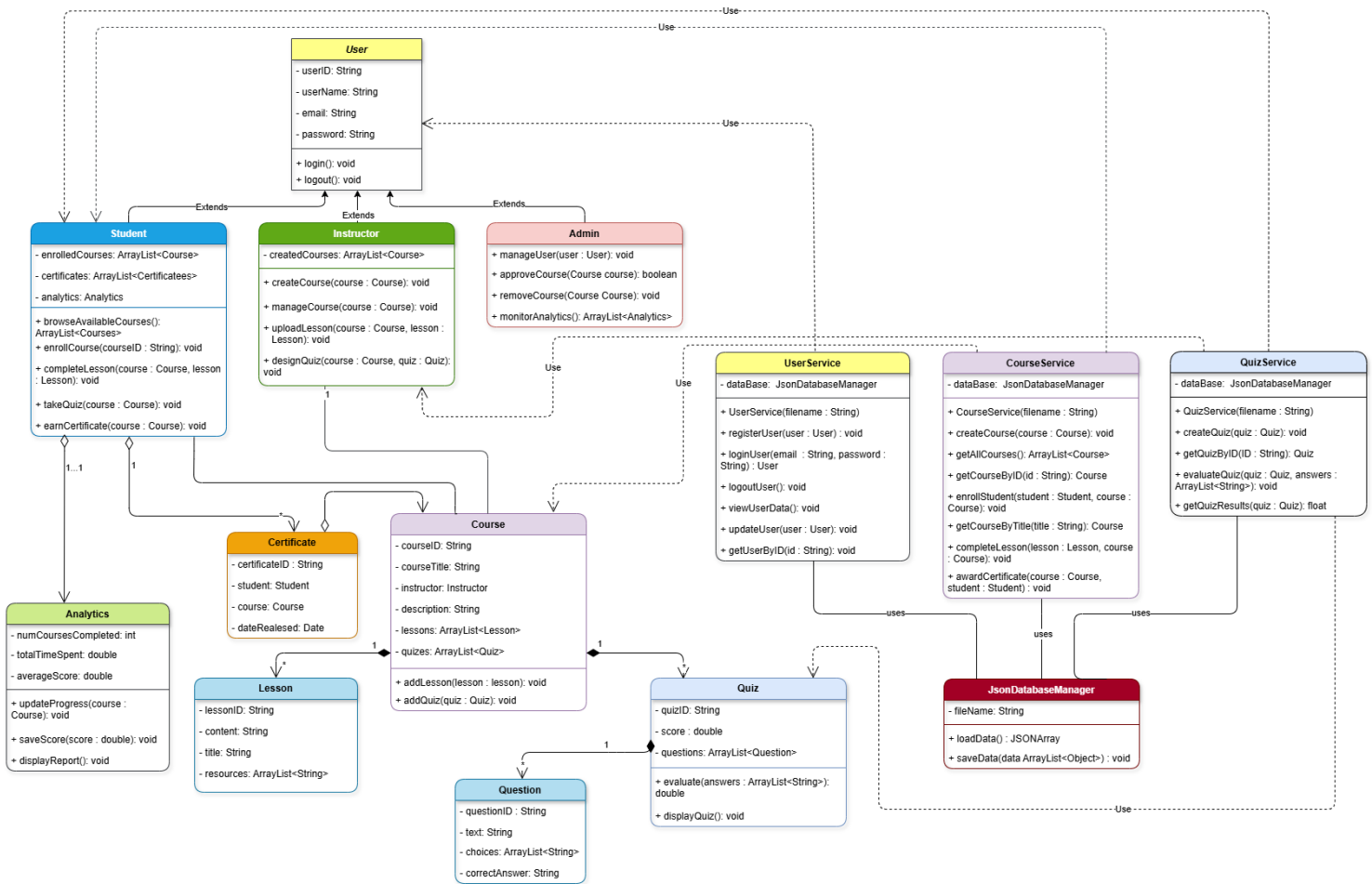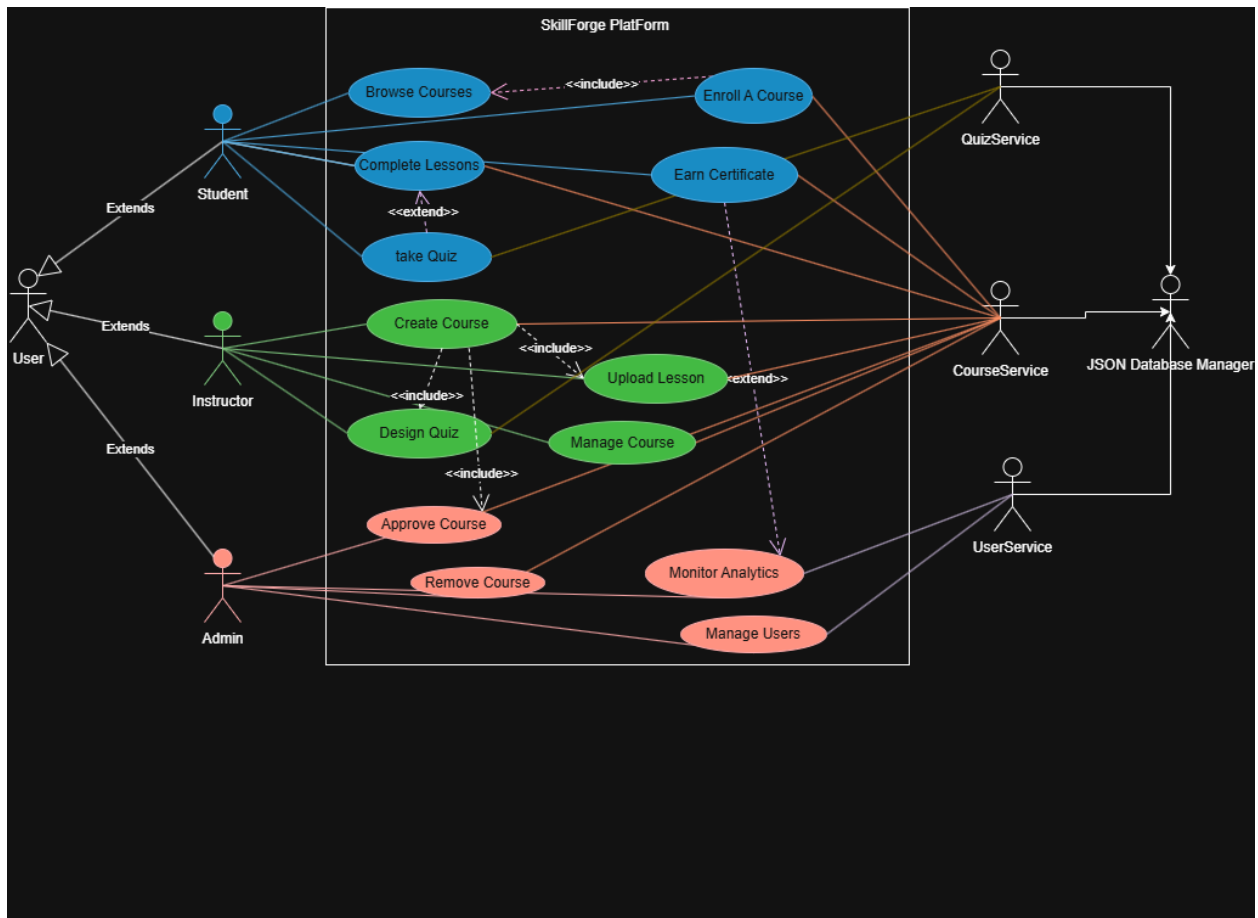# 1.Class Diagram

# Description of the Class Diagram:

The UML Class Diagram shows the main structure of the SkillForge platform, including the key classes, their attributes, methods, and relationships. It serves as a blueprint for the system's design, helping developers understand how different parts of the platform interact. At the center of the system is the User class, which defines common attributes such as userID, email, username, and password. It is extended by Student and Admin subclasses. The Student class handles course enrollment, progress tracking, and viewing analytics, while the Admin class manages users and courses.

The Course class represents the learning content, holding details like courseID, title, and description. Each course contains multiple Lesson and Quiz objects. Lessons store educational content, while quizzes test understanding through Question objects. Quizzes are composed of several questions, maintaining consistency between their content and purpose. Students can enroll in courses, complete activities, and earn Certificates, which link students to completed courses and include the issue date. The Analytics class tracks progress, scores, and completion rates, connecting students and their learning data.

Data management is handled by the JsonDatabaseManager, which loads and saves information about users, courses, certificates, and analytics in JSON files.

The diagram uses different relationships:

**– Inheritance:** The Student, Instructor, and Admin classes inherit from the User class. This means they share common attributes like username, email, and password, but each subclass adds its own specific functions, such as course management for instructors or user control for admins.

**– Association:** The Student class is associated with the Course class, as students can enroll in multiple courses. Similarly, the Instructor class is connected to the Course class as instructors are responsible for creating and managing the courses.

**– Composition:** The Course class is composed of multiple Lesson and Quiz objects. This indicates a strong relationship where lessons and quizzes cannot exist without their parent course

**– Aggregation:** The Certificate and Analytics classes are aggregated with the Student and Course classes. These elements depend on student and course data but can exist independently once created, such as a certificate remaining in the system even after a course is completed.

# 2.UseCase Diagram



# Description of the Use Case Diagram:

The Use Case Diagram helps in understanding Actors and the key use cases by which they interact with the system.

**Student:** can "Browse Courses" then choose to "Enroll Course", "Enroll Course" is linked to Course Service dealing with Database, once enrolled he can complete lesson and take quiz. "Earn Certificate" as Admin "Monitor Analytics" to verify earning certificate once he completes all lessons.

**Instructor:** "Create Course" requires "Upload Lessons" and "design Quizzes" which are linked to Course Service and Quiz Service. "Manage Course" so he can Upload Lessons and
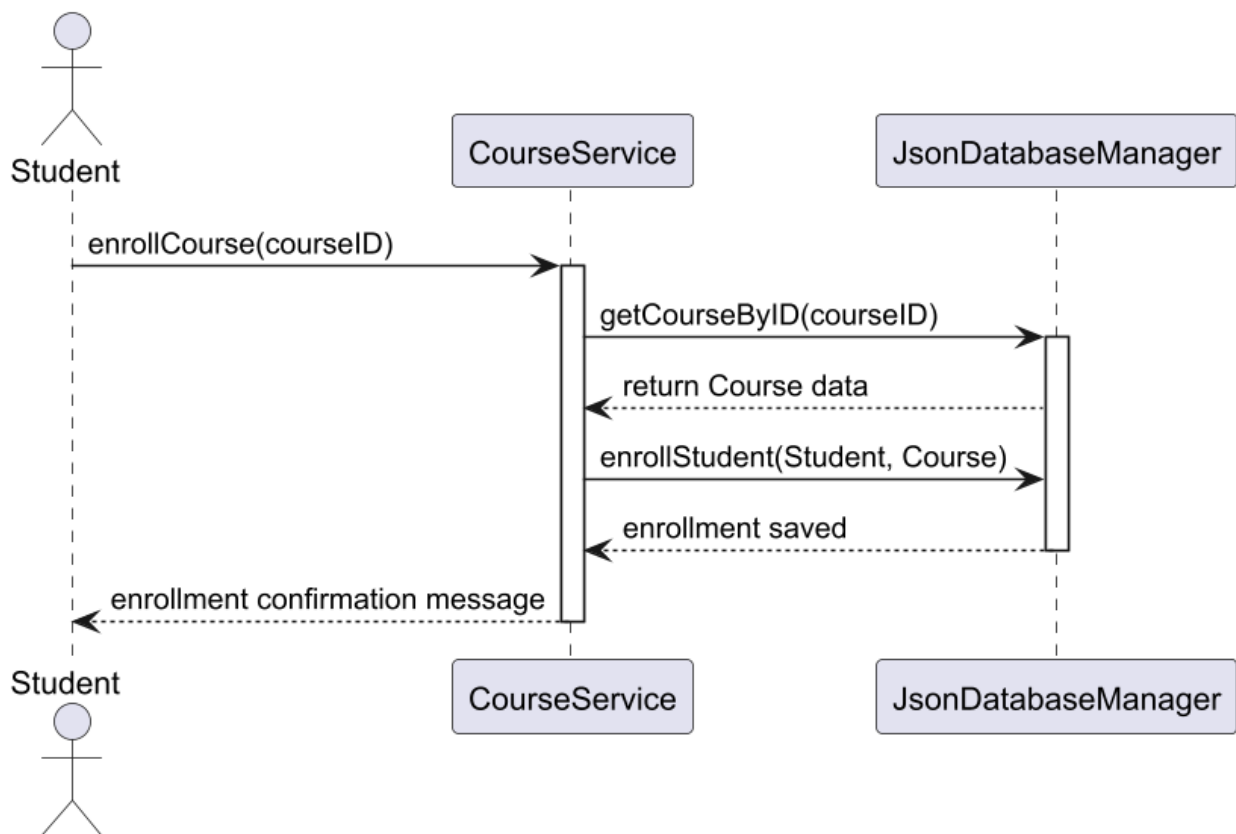
create Quizzes or/remove them. "Upload Lesson" to add lessons to a course so students can complete them. "Design Quiz" so students can take quizzes.

**Admin:** "Manage Users" as for creating accoustudentsin/Verifications.

"Approve Course" each course created by the instructor must be approved by an admin. "Remove Course" Admin has authority to remove any course on the system. "Monitor Analytics" Admin monitor analytics to give certificates to students and keep track of performance
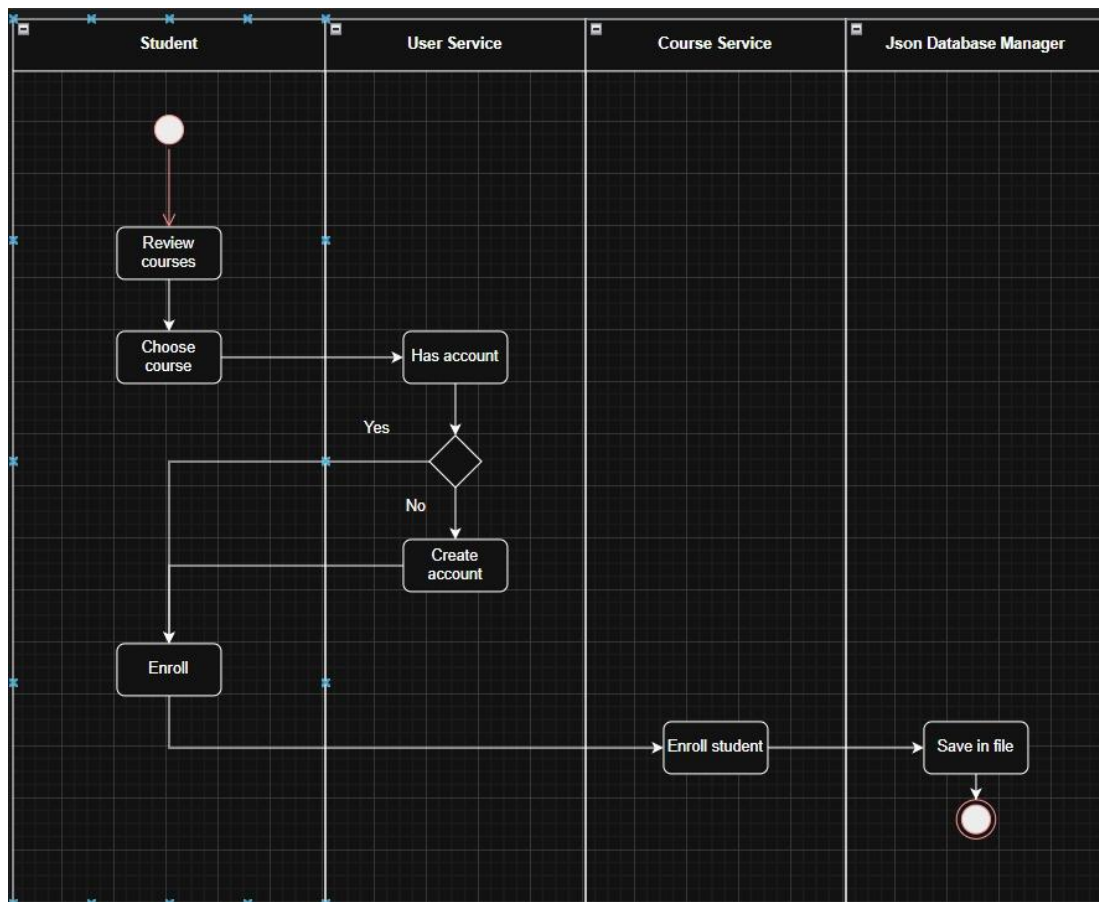
# 3.Sequence Diagram

# Description of the Sequence Diagram:

This sequence diagram models the "Enroll in a Course" process in the SkillForge platform. The student sends an enrollment request to the CourseService, which retrieves the course details from the JsonDatabaseManager, and stores the enrollment record. Finally, the CourseService sends an enrollment confirmation back to the student.

# 4.Activity Diagram



# Description of the Activity Diagram:

The activity diagram shows the steps taken for a user (student) to enroll in a course. The student browses the available courses then chooses one. If the user is already registered on the course, it won't be available. If the user doesn't have an account, they have to create one. Once the student is enrolled, the data is saved in the Json file.