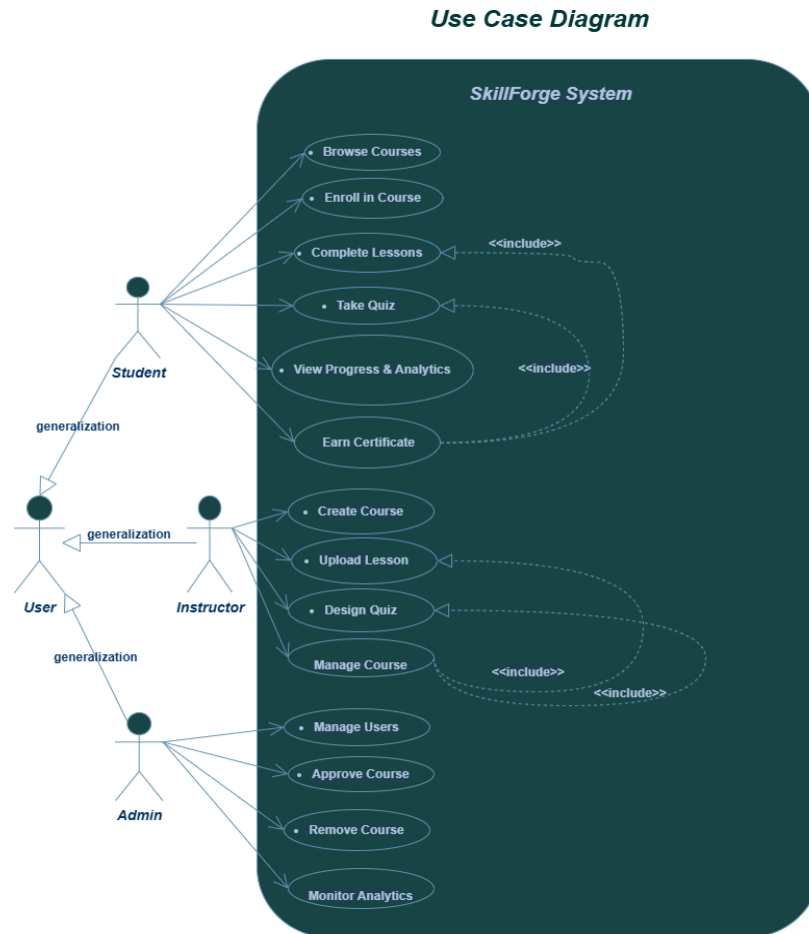# Lab6

Lab 6: SkillForge UML Design

CC272 – Programming II – Fall 2025/2026

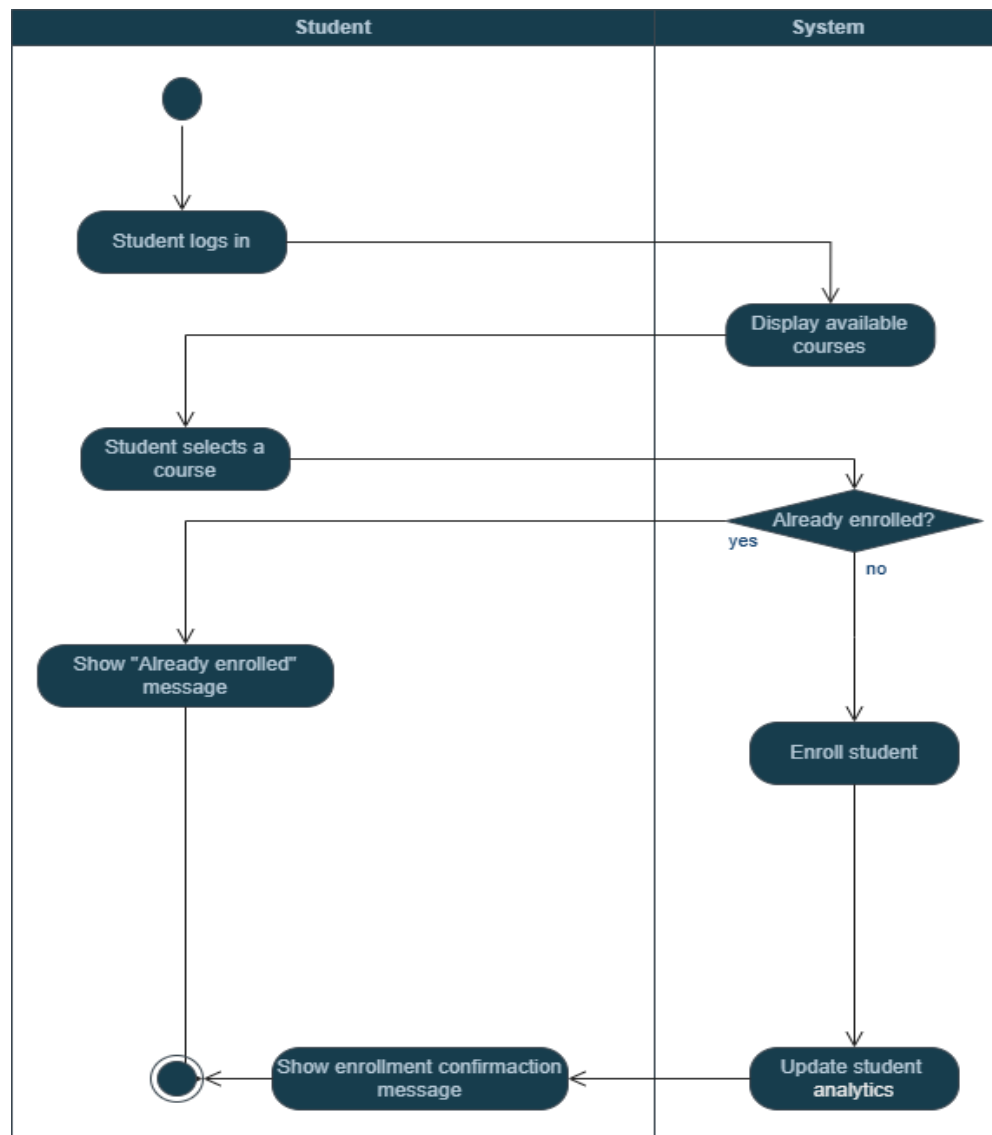---

1. Use Case Diagram

Use Case Diagram

Explanation for Use Case Diagram:

The Use Case Diagram shows how Students, Instructors, and Admins interact with SkillForge. Students browse courses, enroll, complete lessons and quizzes, and earn certificates upon full completion; Instructors create and manage courses by uploading lessons and designing quizzes; Admins manage users, approve or remove courses, and monitor analytics. The diagram uses generalization from a base User and «include» relationships to reflect core system behavior.
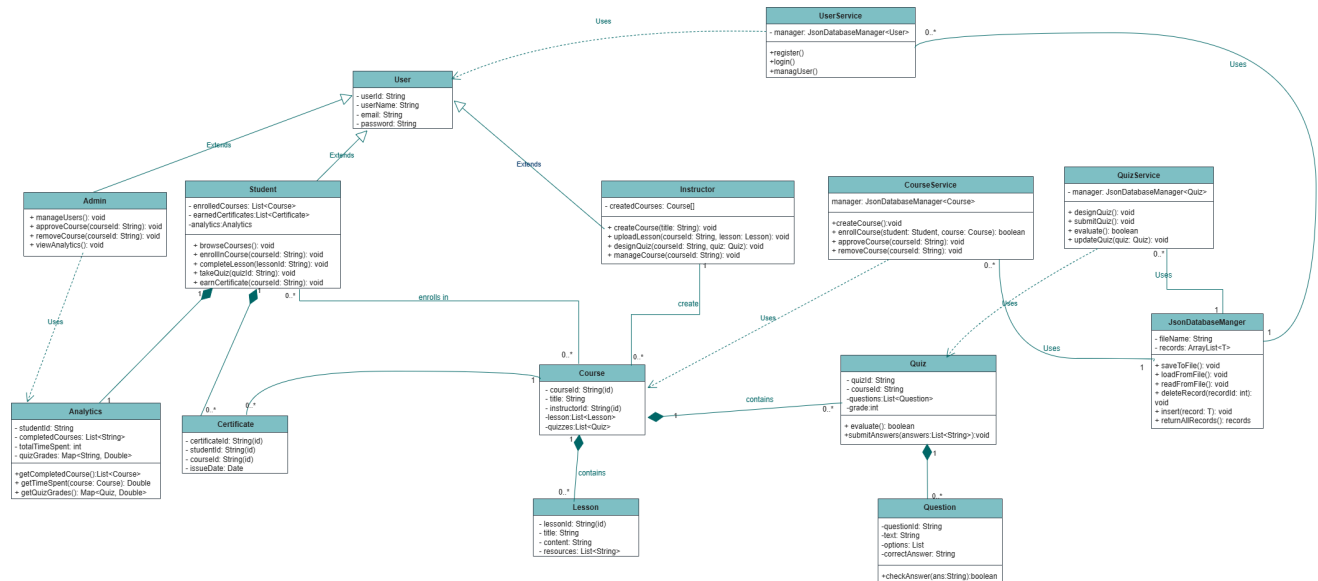
---

2. Activity Diagram

This activity diagram models the detailed workflow of the "Enroll in a Course" use case. It includes a decision point to prevent duplicate enrollment, updates student analytics upon successful registration, and ensures a consistent and user-friendly experience throughproper validation and feedback
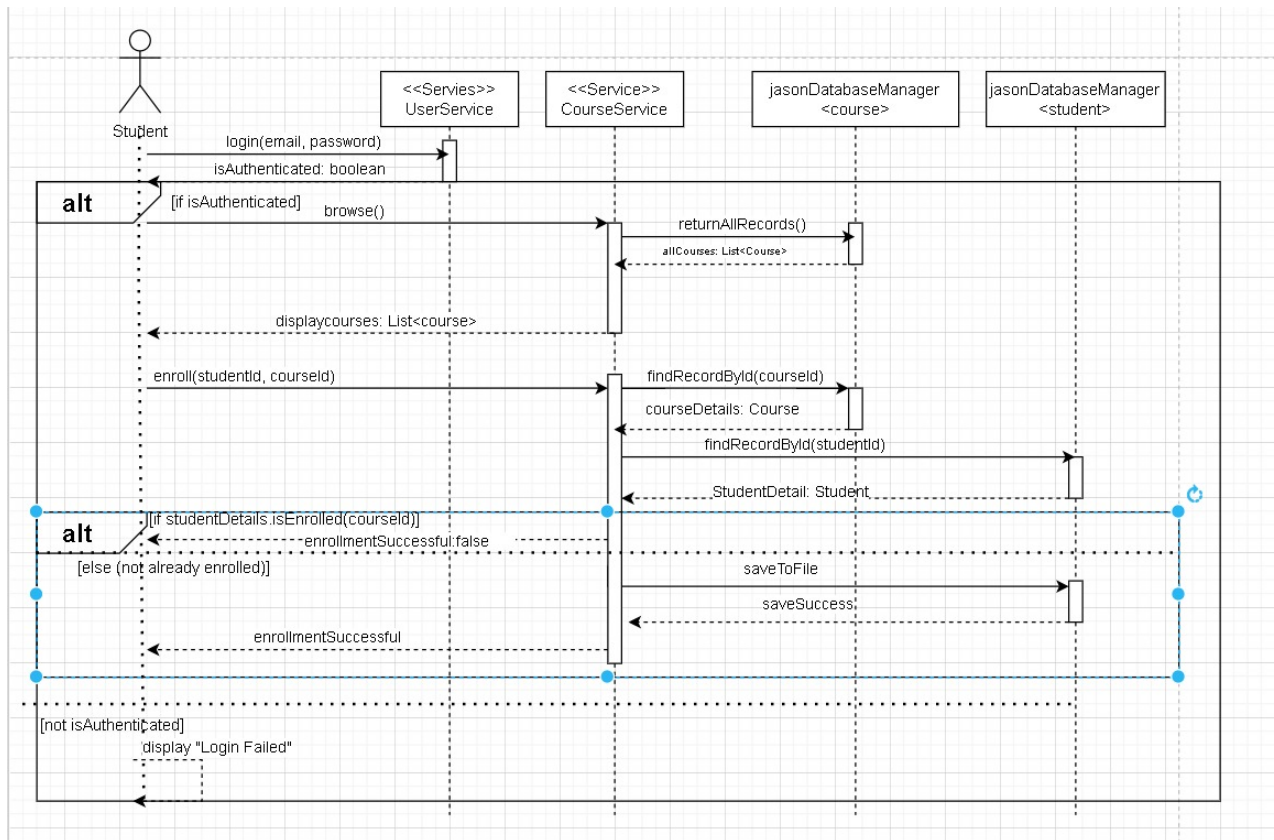
3. Class Diagram

UserService
- manager: JsonDatabaseManager<User>

+register()
+login()
+managUser()

User
- userId: String
- userName: String
- email: String
- password: String

Admin
+ manageUsers(): void
+ approveCourse(courseId: String): void
+ removeCourse(courseId: String): void
+ viewAnalytics(): void

Student
- enrolledCourses: List<Course>
- earnedCertificates:List<Certificate>
-analytics:Analytics

+ browseCourses(): void
+ enrollInCourse(courseId: String): void
+ completeLesson(lessonId: String): void
+ takeQuiz(quizId: String): void
+ earnCertificate(courseId: String): void

Instructor
- createdCourses: Course[]

+ createCourse(title: String): void
+ uploadLesson(courseId: String, lesson: Lesson): void
+ designQuiz(courseId: String, quiz: Quiz): void
+ manageCourse(courseId: String): void

CourseService
- manager: JsonDatabaseManager<Course>

+createCourse():void
+ enrollCourse(student: Student, course: Course): boolean
+ approveCourse(courseId: String): void
+ removeCourse(courseId: String): void

QuizService
- manager: JsonDatabaseManager<Quiz>

+ designQuiz(): void
+ submitQuiz(): void
+ evaluate(): boolean
+ updateQuiz(quiz: Quiz): void

Analytics
- studentId: String
- completedCourses: List<String>
- totalTimeSpent: int
- quizGrades: Map<String, Double>

+getCompletedCourse():List<Course>
+ getTimeSpent(course: Course): Double
+ getQuizGrades(): Map<Quiz, Double>

Certificate
- certificateId: String(id)
- studentId: String(id)
- courseId: String(id)
- issueDate: Date

Course
- courseId: String(id)
- title: String
- instructorId: String(id)
-lesson:List<Lesson>
-quizzes:List<Quiz>

Quiz
- quizId: String
- courseId: String
-questions:List<Question>
-grade:int

+ evaluate(): boolean
+submitAnswers(answers:List<String>):void

JsonDatabaseManger
- fileName: String
- records: ArrayList<T>

+ saveToFile(): void
+ loadFromFile(): void
+ readFromFile(): void
+ deleteRecord(recordId: int): void
+ insert(record: T): void
+ returnAllRecords(): records

Lesson
- lessonId: String(id)
- title: String
- content: String
- resources: List<String>

Question
-questionId: String
-text: String
-options: List
-correctAnswer: String

+checkAnswer(ans:String):boolean

Relationships (labels): Uses, Extends, enrolls in, create, contains, 0.., 1*

Explanation for Class Diagram:

This diagram captures the static structure of SkillForge, showing core classes like User, Course, and Quiz with their attributes, methods, and relationships. It uses inheritance, composition, and dependency to model real-world interactions accurately. The generic JsonDatabaseManager ensures clean data persistence and system modularity

4. Sequence Diagram

This sequence diagram illustrates the dynamic interaction during course enrollment. It shows how the GUI delegates requests to CourseService, which validates enrollment status, updates student data and analytics, and persists changes via JsonDatabaseManager—ensuring loose coupling and clear message flow between components.