11/2/2025

Lab 6 Report

UML design for SkillForge system

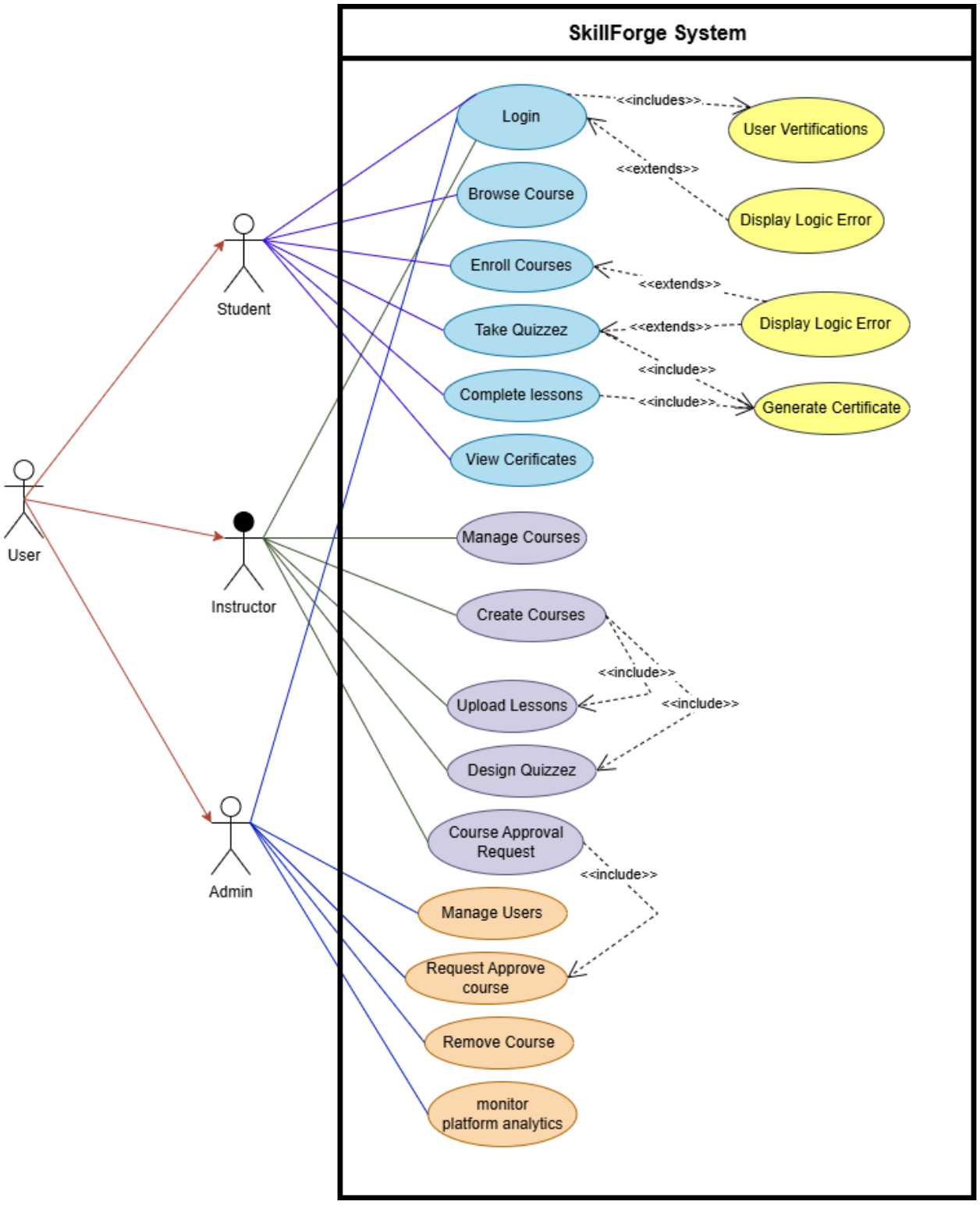
Nour Khaled 9468

Mariam Yamen 9559

Ali Tarek 9676

Moaz Refaat 9649

1)Use Case Diagram:



Description:

The Use Case Diagram shows the functions of the SkillForge System by demonstrating the interactions between the actors (users) where User role is generalized into Student, Admin, and Instructor. It also shows relationships like extend and include which help organize the behavior of the system.

Main Actors

1)User :Which is the superclass actor or a generalized actor that represents any person that is interacting with the system

2)Student : is an actor which is a generalization of the User

3)instructor: is an actor which is a generalization of the User

4)Admin : is an actor which has is a generalization of the User

KEY USE CASE :

1)Student : login , browse and enroll courses, take quizzes, complete lessons , earn and view certificates.

2)Instructor : login, manage and create courses, upload lessons and send a request to approve course upload

3)Admin: login ,manage users, approve course requests , remove courses and monitor platform analytics.

Use Case Description

Student

1)login : used for authentication and allow only certain users to enter the website , when the User enters its username and password the system will verify the User (User Verification) if the User is not found in the file it will display Logical Error.

2)Browse Courses: allows for searching and viewing available courses

3)Enroll courses: allows student to enroll for available courses. If the User is already enrolled in that course or there is no available seats in that course it will display Logical Error

4)Take Quizzes: Allow student to take a quiz to check their understanding to the lesson , the user can't take a quiz if they didn't finish the lesson , it will display Logical Error

5)Complete Lessons: Allow user to see their completed lesson .

5)View Certificate : allow student to view all the certificates they earned.

Instructor

1)login : For authentication and validate instructor

2)Manage Courses : To edit , update or organize the course

3)Create courses: To create a new course and it must have lessons and quizzes in order to be created

4)Upload Lessons: to upload lessons to the course

5)Design Quizzes : to design and create quizzes related to the lesson or course

6)Courses Approval Request: Request admin to approve the course to be uploaded to the website .

Admin

1)Login : to authenticate admin

2)Manage Users: to edit, add ,and remove users.

3)Request Approve Course: to view course and decide whether to approve or decline the course to uploaded

4)Remove Course : to remove courses that doesn't follow the website policies

5)Monitor Platform Analytics: allow admin to monitor and track the system usage and performance .

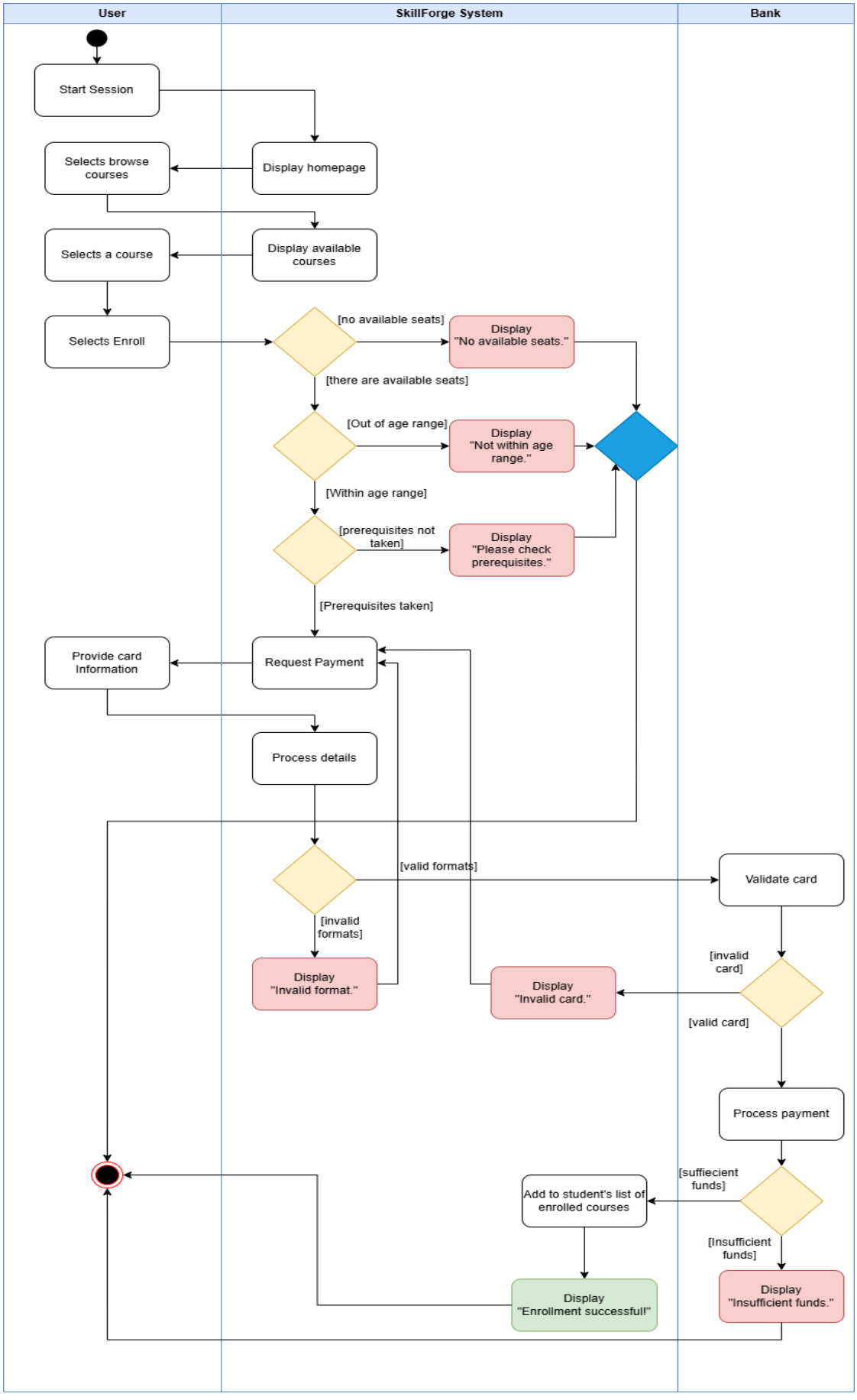
System

1)Generate Certificate: the system will generate a certificate only when the user finishes all their quizzes and completed all their lesson related to the course

2)Display Logical Error: if the user didn’t pass the user verification process or if a student trying to take a quiz and they didn’t finish the lessons or if a student was going to enroll in an already registered course or a course with unavailable seats.

3)User Verification: for authentication and check if there is a user with this username and password.

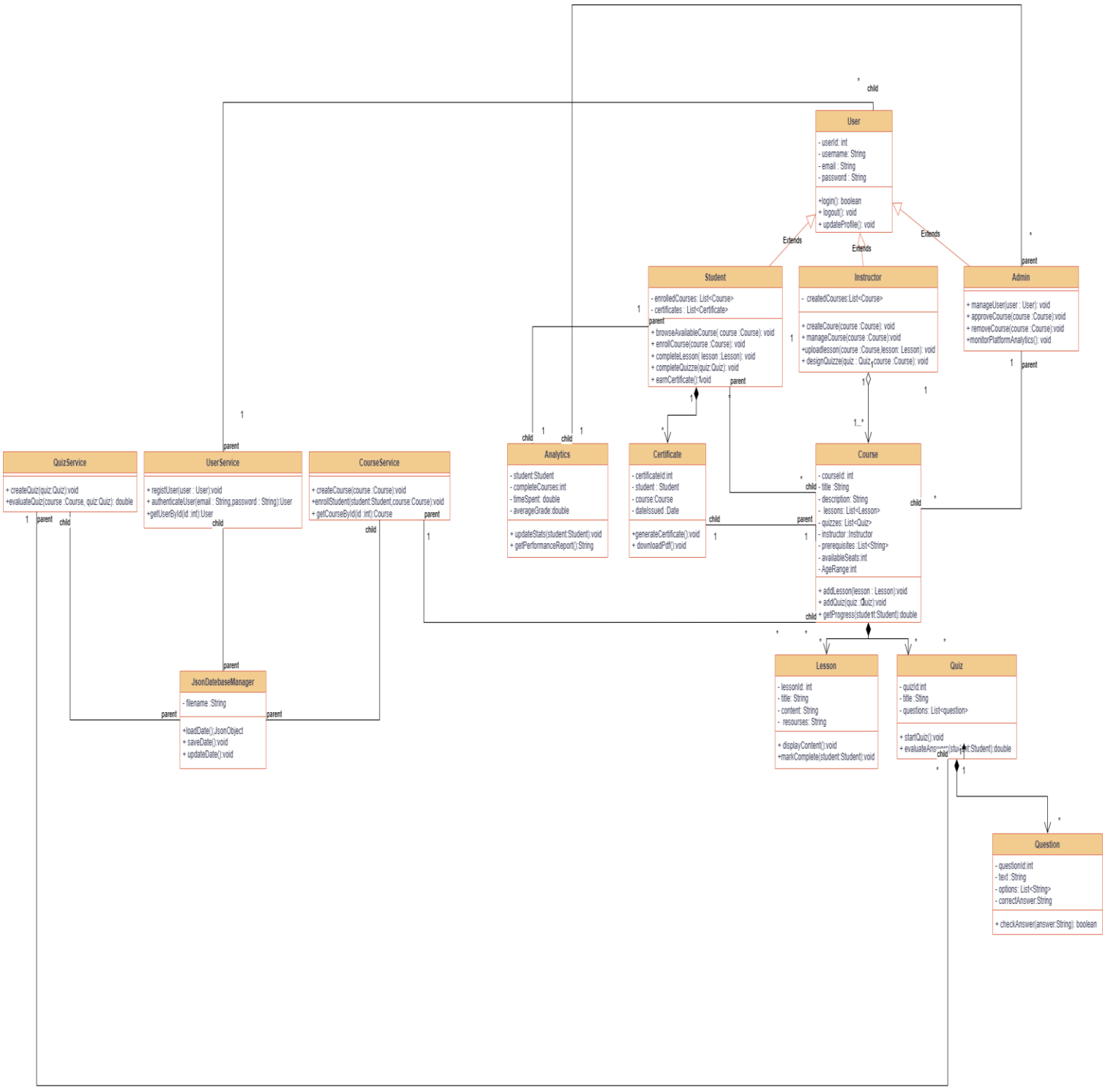
2) Activity Diagram:



Description:

The use case selected here is “student enrollment in a course”, the user is assumed to be registered as a student.
The student will choose to browse available courses and select a course to enroll in.
The system then checks if there are available seats, and if student is eligible for this course by checking the age range, and if there are any prerequisites for the selected course.
In case of failure in any of the above cases, an error message is displayed and will end the enrollment session.
Otherwise, the system prompts the user to enter payment details and will check for any syntax errors. If there are any errors an error message is displayed and will prompt the user to enter payment details again, but if all is correct the system will communicate with the bank for card validation.
If the card is invalid, an error message is displayed and will prompt the user to enter payment details again.
Otherwise, the bank will check the funds, if they are sufficient, enrollment will be successful, a message is displayed to the student and the session will end, but if they are insufficient, an error message is displayed, and the session will end.

3)Class Diagram:



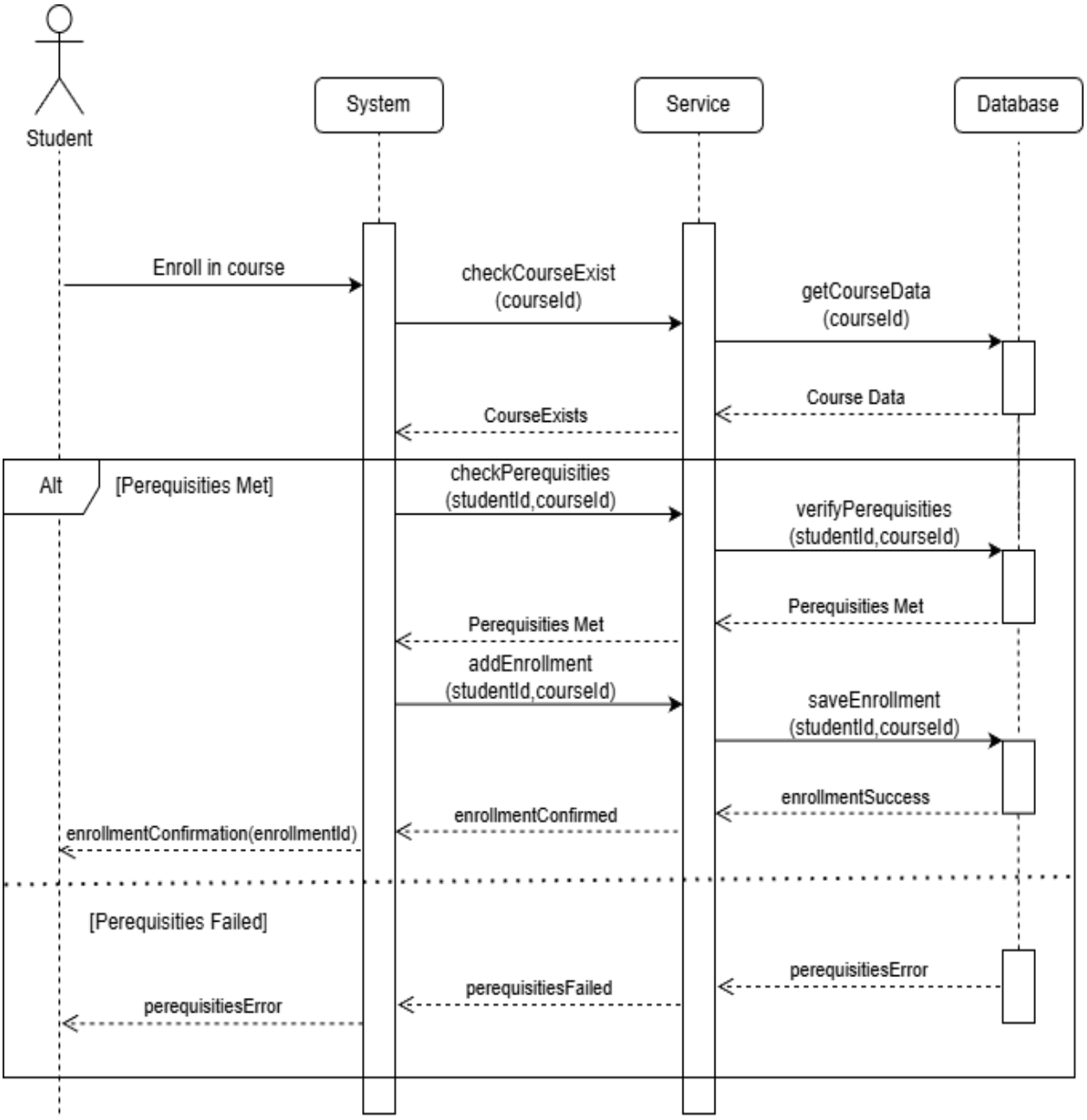
Description:

The Class Diagram shows the main structure of the SkillForge platform.

It illustrates the key classes such as User, Course, Lesson, Quiz, and Certificate, along with their attributes, methods, and relationships.

This diagram is important because it helps understand how different parts of the system are connected and supports building the system in an organized and efficient way.

4) Sequence diagram:



Description:

This sequence diagram models the "Enroll in a Course" use case, illustrating the chronological interactions between the student actor and system components during the enrollment process. It shows how the student initiates enrollment, how the system validates course availability and prerequisites, and how the database persists the enrollment data upon successful validation.

Importance in System Design:

This diagram is crucial for understanding the real-time object interactions and message flow between system layers. It clearly defines the responsibilities of each component (System, Service, Database) and demonstrates proper error handling for prerequisite validation, ensuring robust enrollment functionality in the SkillForge platform.