Coursera SRS

Date: 4/4/2024

1. Introduction

The purpose of this document is to present a detailed description of the online courses platform Coursera. It serves as a blueprint for Coursera, detailing its purpose, functionalities, and interactions with users. By facilitating user authentication, seamless course browsing, enrollment, and access to course materials. This document is intended for the stakeholders and the developers.

Moreover, Coursera is designed to integrate seamlessly with existing educational infrastructure and systems, including payment organizations. Coursera can interface with these systems to ensure a cohesive learning ecosystem.

Furthermore, Coursera aligns with the overarching strategic objectives of the organization commissioning this software. By offering a scalable, reliable, and user-friendly platform, Coursera not only meets the immediate educational needs of its users but also contributes to the organization’s long-term goals of fostering continuous learning and skill development.

2. User Requirements

2.1 Student

- As a student I shall be able to create an account and login using my credentials.
- As a student I shall be able to search for courses easily and enroll in my course of choice.
- As a student I shall be able to view my course material, as well as viewing my progress.
- As a student I should be able to communicate with the course instructors and other students who are enrolled in the same courses.
- As a student I shall receive a certificate upon completion of the course.
- As a student I should be able to give my feedback on the course.

2.2 Instructor

- As an instructor I shall be able to create an account and login using my credentials.
- As an instructor I shall be able to upload my courses, which include videos, documents, assignments and quizzes.
- As an instructor I should be able to communicate with the students who are enrolled in my courses.
- As an instructor I should be able to read the feedback given by the students on my courses.

2.3 Admin

- As an admin I shall be able to approve instructors' accounts before they start uploading courses to the platform.
- As an admin I shall be able to manage courses.

3. Functional Requirements

User Authentication:

- Description / Action: Users should be able to create accounts, log in, and log out securely.
- Requirements / Inputs: User-provided information (e.g., username, email, password) for account creation and login.
- Source: User input via registration and login forms.
- Pre-condition: The user is on the registration page.
- Post-condition: User is logged in (for login), or user account is created (for registration).
- Output: Successful login message or confirmation of account creation.

Course Browsing:

- Description / Action: Users should be able to browse courses by subject, university, or topic.
- Requirements / Inputs: User-selected filters (e.g., subject, university) for course browsing.
- Source: User input via filter options provided on the course browsing interface.
- Pre-condition: User is logged in.
- Post-condition: User sees a list of courses matching the selected filters.
- Output: List of courses with relevant details (e.g., title, description, instructor).

Course Enrollment:

- Description / Action: Users should be able to enroll in courses they're interested in.
- Requirements / Inputs: User-selected course (e.g., Course ID, Course Name) for enrollment.
- Source: User input via the course enrollment interface.
- Pre-condition: User is logged in and has not already enrolled in the selected course.
- Post-condition: User is enrolled in the selected course.
- Output: Confirmation message indicating successful enrollment.

Course Materials Access:

- Description / Action: Users should be able to access course materials, including lectures, assignments, and readings.
- Requirements / Inputs: Selected course by the user.
- Source: User input via course selection or navigation within the course interface.
- Pre-condition: User is enrolled in the selected course.

- Post-condition: User gains access to course materials.
- Output: Access to lectures, assignments, readings, and other course materials.

Progress Tracking:

- Description / Action: Users should be able to track their progress within courses and see their completed and upcoming assignments.
- Requirements / Inputs: User progress data (e.g., completed assignments, viewed lectures).
- Source: User activity within the course interface.
- Pre-condition: User is enrolled in the course and has completed some activities.
- Post-condition: User can view their progress and upcoming assignments within the course.
- Output: Progress dashboard displaying completed activities and upcoming assignments.

Certification:

- Description / Action: Users should be able to receive certifications upon completing courses or programs.
- Requirements / Inputs: Completion status of course or program by the user.
- Source: Course completion data tracked by the system.
- Pre-condition: User has completed all requirements for the course or program.
- Post- condition: User is eligible to receive a certification.
- Output: Certificate of completion issued to the user.

Communication:

- Description / Action: Users should be able to communicate with instructors and other students through discussion forums or messaging.
- Requirements / Inputs: User-generated messages or posts, recipient (instructor or student)
- Source: User input via messaging interface or discussion forums.
- Pre-condition: User is enrolled in the course and has access to communication features.
- Post-condition: Messages or posts are successfully sent and visible to intended recipients.
- Output: Display of sent messages or posts in the communication interface.

Feedback:

- Description / Action: Users should be able to provide feedback on courses and instructors.

- Requirements / Inputs: User-provided feedback (e.g., ratings, comments).
- Source: User input via feedback forms or rating systems.
- Pre-condition: User has completed or interacted with the course or instructor.
- Post-condition: Feedback is submitted and stored for analysis.
- Output: Confirmation message indicating successful submission of feedback.

Course Management:

- Description / Action:
  Instructors should be able to manage their uploaded courses, including adding new content, modifying existing content, and removing content. Admins should review and approve them before they are published on the platform.
- Requirements / Inputs:
  Instructor-provided course materials (e.g., lectures, assignments, quizzes).
  Admin approval for course publication.
- Source:
  Instructor input via course management interface.
  Admin input via course approval interface.
- Pre-condition:
  For instructors: Instructor is logged in and has permission to manage the course.
  For admins: Admin is logged in and has authorization and ownership to approve courses.
- Post-condition: Course is updated on the platform upon admin approval.
- Output:
  Confirmation message indicating successful modification of course content by instructor. Notification to the instructor upon admin approval of the course.

Account Management:

- Description / Action: Admins should be able to manage user accounts, including creating new accounts, modifying existing accounts, and deactivating accounts as necessary.
- Requirements / Inputs: Admin-provided user information (e.g., username, email).
- Source: Admin input via account management interface.
- Pre-condition: Admin is logged in and has permission to manage user accounts.
- Post-condition: User account details are updated according to admin actions.
- Output: Confirmation message indicating successful modification of user accounts.

4. Non- Functional Requirements

Performance:

- The system shall respond to user interactions within X milliseconds/seconds.
- Video streaming: Video lectures and content should stream smoothly without buffering or lag.
- System responsiveness: The system should be able to handle a high number of concurrent users without performance degradation.

Security:

- The system shall ensure user authentication and authorization mechanisms to prevent unauthorized access to course materials and user data.
- The system shall encrypt sensitive user data such as passwords and personal information.

Scalability:

- The software should be able to accommodate a growing number of courses, students and instructors.

Reliability & Availability:

- Data integrity: Course content, student data and progress should be stored securely and reliably with minimal risk of data loss.
- Error handling: The system should gracefully handle errors and provide clear feedback to users.

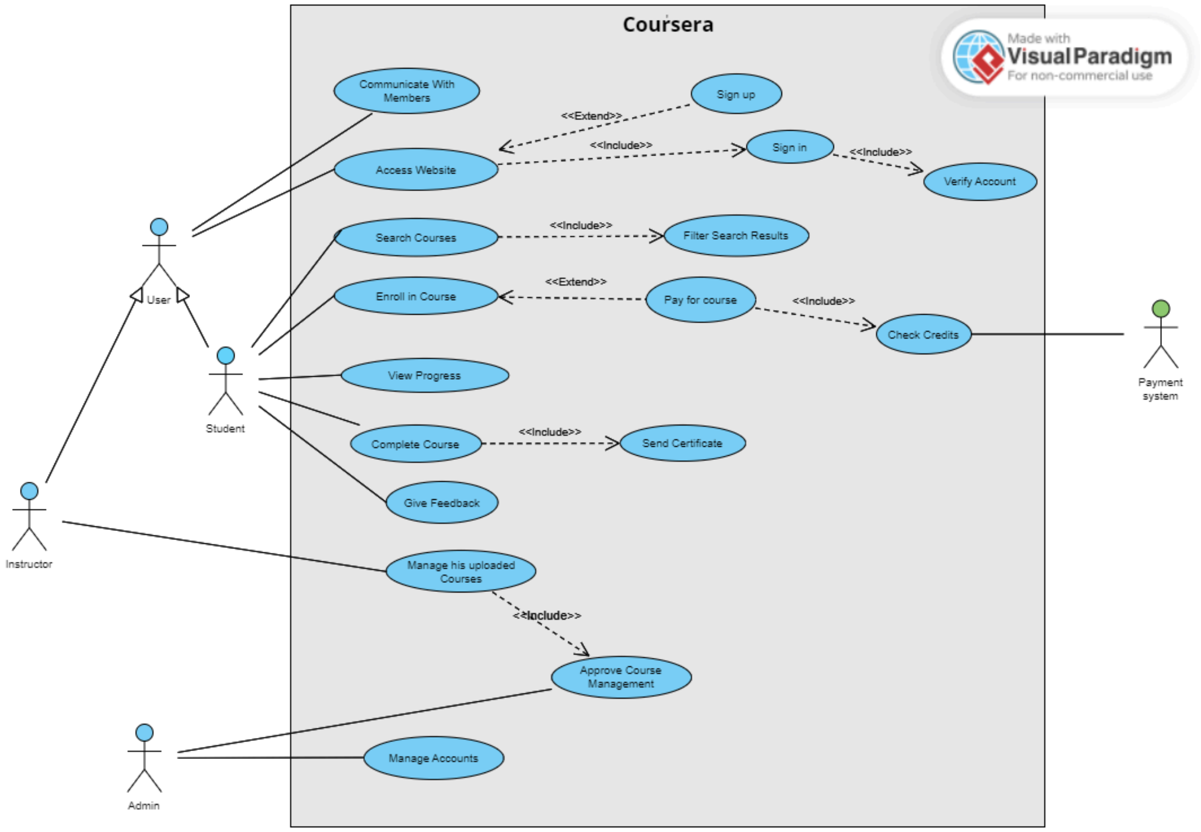
Maintainability:

- The codebase should be well-documented and modular for easy maintenance and updates.
- The system should be designed to allow for easy addition of new features and functionalities in the future.

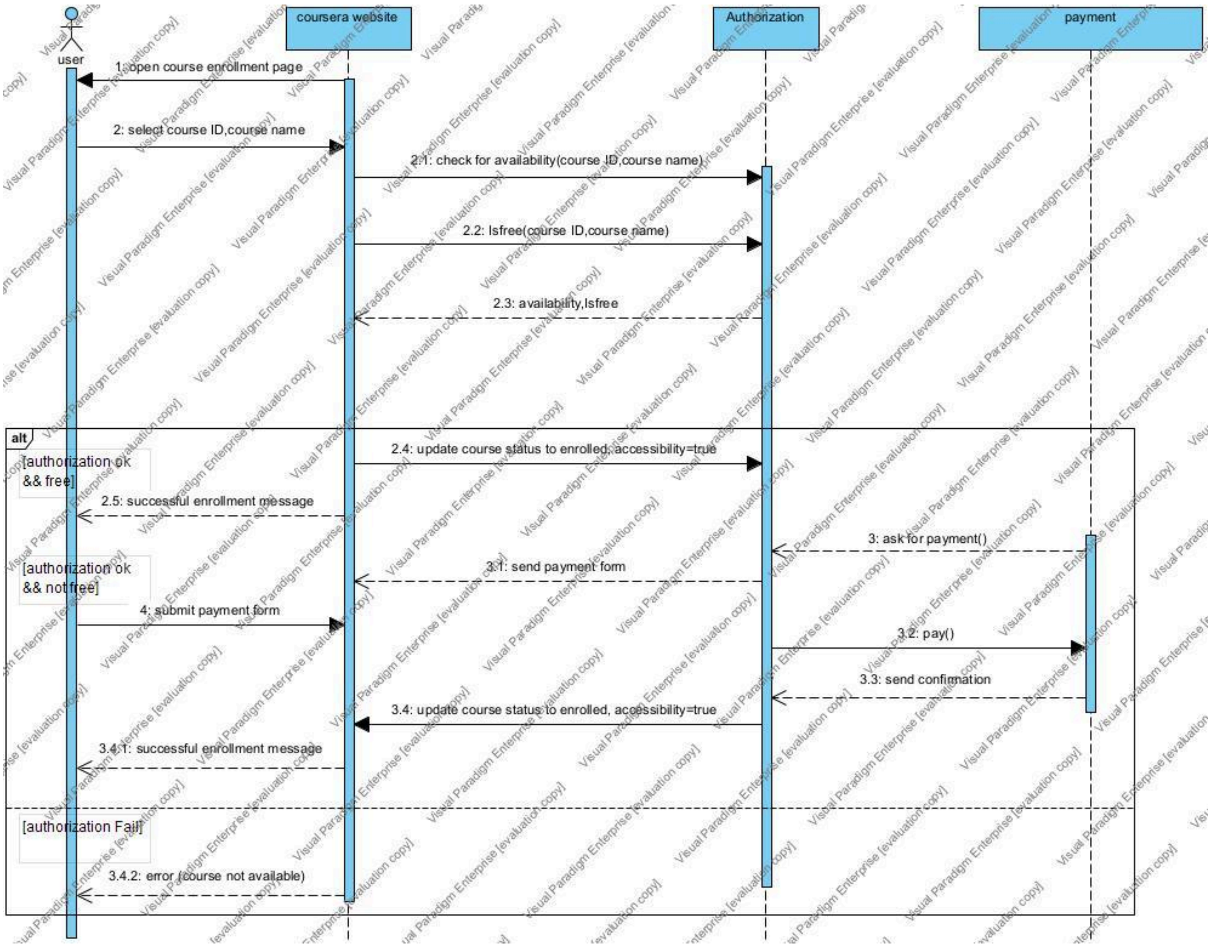
Usability:

- User Interface (UI): The user interface should be intuitive and easy to navigate for both students and instructors to ensure efficiency and satisfaction.

5. Use Case Diagram:



6. Sequence Diagram for course enrollment:



user
coursera website
Authorization
payment
1: open course enrollment page
2: select course ID,course name
2.1: check for availability(course ID,course name)
2.2: Isfree(course ID,course name)
2.3: availability,Isfree
alt
[authorization ok && free]
2.4: update course status to enrolled, accessibility=true
2.5: successful enrollment message
[authorization ok && not free]
3: ask for payment()
3.1: send payment form
4: submit payment form
3.2: pay()
3.3: send confirmation
3.4: update course status to enrolled, accessibility=true
3.4.1: successful enrollment message
[authorization Fail]
3.4.2: error (course not available)