IT314 - Software Engineering

Group 26

Online course management system

Team Member :

Student ID	Name
202001242	Chintan Suthar
202001277	Nikunj Adhiya
202001448	Priyanshu Parmar
202001244	Parth Masalawala
202001269	Kunj Gupta
202001460	Raj Shah
202001246	Milind Patel
202001263	Pranav Thumar
202001440	Jay Shukla
202001404	Aditya Patel

Table of Contents

Needs

1. An online course management system is needed to deliver, manage, and track online courses. It reduces the requirement for the workforce and eliminates the need for paperwork. It is a valuable tool for instructors, students, and institutes that provide online education.

2. The system facilitates communication between students and instructors, as well as among students, by providing a platform for discussions. The system allows students to collaborate on projects and assignments, improving the learning experience and fostering teamwork.

3. The system will enable students to view and enroll/drop in online courses and communicate with their instructors from anywhere.

4. It helps students who cannot attend offline classes(Distance Learning) due to location, work, or other personal obligations with an internet connection, making education more accessible and convenient.

5. The system provides students with access to various educational resources, such as video lectures, notes, readings, interactive simulations, and timetables, which can enhance their learning experience. It helps students to stay regularly updated with the

course files and other notices for their respective courses. They don’t need to go to the library and fetch the course material.

6. Students can give their teachers feedback, which helps to improve the learning process. This active interaction between students and teachers aids in the progression of learning.

7. It creates a user-friendly, efficient, and effective system for managing and delivering online courses that can improve the institution's overall operation and enhance the student, teacher, and academic institution's experience.

8. The system allows instructors and students to track student progress, monitor their engagement, and assess their performance, providing valuable data for both.

9. Students register courses by themselves, reducing administrators' manual workload of registering each student’s data manually.

10. Online result declaration and generating reports can be beneficial to academic administrators.

Features

1. **User Management:** The system should have the capability to manage multiple users with different roles such as students, instructors, and administrators.

2. **Enroll**: If the student is interested in a particular course then he/she can enroll.

3. **Add course**: During the registration, the student needs to add the course in his/her list and they will be enrolled after the deadline.

4. **Drop course:** During the registration, the student needs to drop the course in his/her list and they will be Disenrolled after the deadline.

5. **Track progress:** When the student submits the online assignment or quizzes the Progress data is updated.

6. **Generate report:** At the end of the course the result of each student needs to be generated and features are included.

7. **Course materials**(Content, videos, notes), **variety of documents** (.zip,.pdf,.doc): with the continuity of the course the course material is given to the student for the revision and exam and assignments purpose.

8. **Online assignments, quizzes:** submission of the student is recorded.

9. **Communication and Collaboration:** The system should provide tools for students and instructors to communicate, discuss and collaborate on course content.

10. **Reporting and Analytics:** Reports and analytics dashboards to provide insights into student performance, course completion rates, and other key metrics.

11. **Content Delivery:** Supports various types of multimedia content such as videos, audio, images, and text.

12. **Course Management:** Ability to create and manage courses, add course content, and assignments.

13. **Feedback Management:** Providing instructors with a platform to give feedback on student assignments, including written comments.

Functional Requirements

1. **User Management:** The system should allow for the creation of student and instructor accounts, and enable administrators to manage these accounts, including adding, editing, and deleting users.

2. **Course creation and management (For instructors):**
    - **a.** Users should be able to create new courses and upload course content, such as text, video, audio, and images.
    - **b.** Users should be able to organize course content into modules and lessons.
    - **c.** Users should be able to set up quizzes and assignments to test student knowledge.
    - **d.** The system should track student progress and provide real-time feedback to students.

3. **Student enrollment and management:**
    - **a.** The system should allow administrators or instructors to manage student enrollments, including adding and removing students from courses.
    - **b.** The system should track student progress and provide real-time feedback on assignments and tests.
    - **c.** The system should provide students with the ability to provide feedback on course materials and courses as a whole.

d. Students can search courses and make requests to join particular courses.

4. **Content delivery:**
   a. The system should support the delivery of course content in various formats, such as text, video, audio, and images.
   b. The system should support multimedia content, such as videos, quizzes, and assignments. The system shall accept submissions in multiple formats, including .zip, .cpp, .txt, .doc, etc.
   c. The system should allow students to access course content at any time and from any location.

5. **Communication tools:**
   a. The system should provide discussion forums for students and instructors to communicate and exchange ideas.
   b. The system should provide private messaging capabilities to allow students and instructors to communicate directly.
   c. The system should allow administrators to send announcements to all students enrolled in a course.

6. **Assessment and Evaluation:** The system should allow instructors to create and administer quizzes, exams, and assignments, and provide students with instant feedback on their performance.Accept submissions in multiple formats: The system shall accept submissions in multiple formats, including .zip, .cpp, .txt, .doc, etc.

7. **Feedback Management:** Instructors should be able to provide feedback on student assignments, including written comments and annotations.

8. The system should prevent unauthorized access to the platform.

9. **Integration:**
    a. The system should be able to integrate with other tools and systems, such as student information systems.

10. **Reporting and analytics:**
    a. The system should provide reporting and analytics capabilities to provide insights into student performance, engagement, and completion rates.
    b. The system should provide overall platform usage and effectiveness statistics.

11. On-premise software in the form of web application is required.

Non-functional Requirements

1. **User Experience:** The system should be easy to use and navigate, with a user-friendly interface. The system should provide a seamless and efficient experience for students and instructors.

2. **Usability:** The system should be easy to use and it should be able to achieve the objective with efficiency and effectiveness. It should be easily understandable for students, instructors, and administrators.

3. **Scalability:** The system should be able to handle a large number of users and courses without degradation in performance. The system should be scalable and able to accommodate growth in the number of courses, students, and instructors without sacrificing performance.

4. **Performance:** The system should have a fast load with minimal latency and downtime, good efficiency, and the least response time to minimize frustration and improve the user experience. The system should be able to handle high traffic and large amounts of data without slowing down.

5. **Security:** The system should be secure, protect sensitive information such as student grades, personal information, and confidential communications and prevent unauthorized access.

Passwords should be encrypted and protect sensitive student information.

6. **Reliability:** The system should be available and functional when users need it. The system should be reliable and available 24/7, with minimal downtime for maintenance and updates. The system should be reliable, with consistent and accurate performance. The system should have appropriate backup and recovery mechanisms in place to ensure the availability of data and services.

7. **Compatibility:** The system should work with a variety of devices. The system should be mobile-friendly and accessible from different devices and platforms.

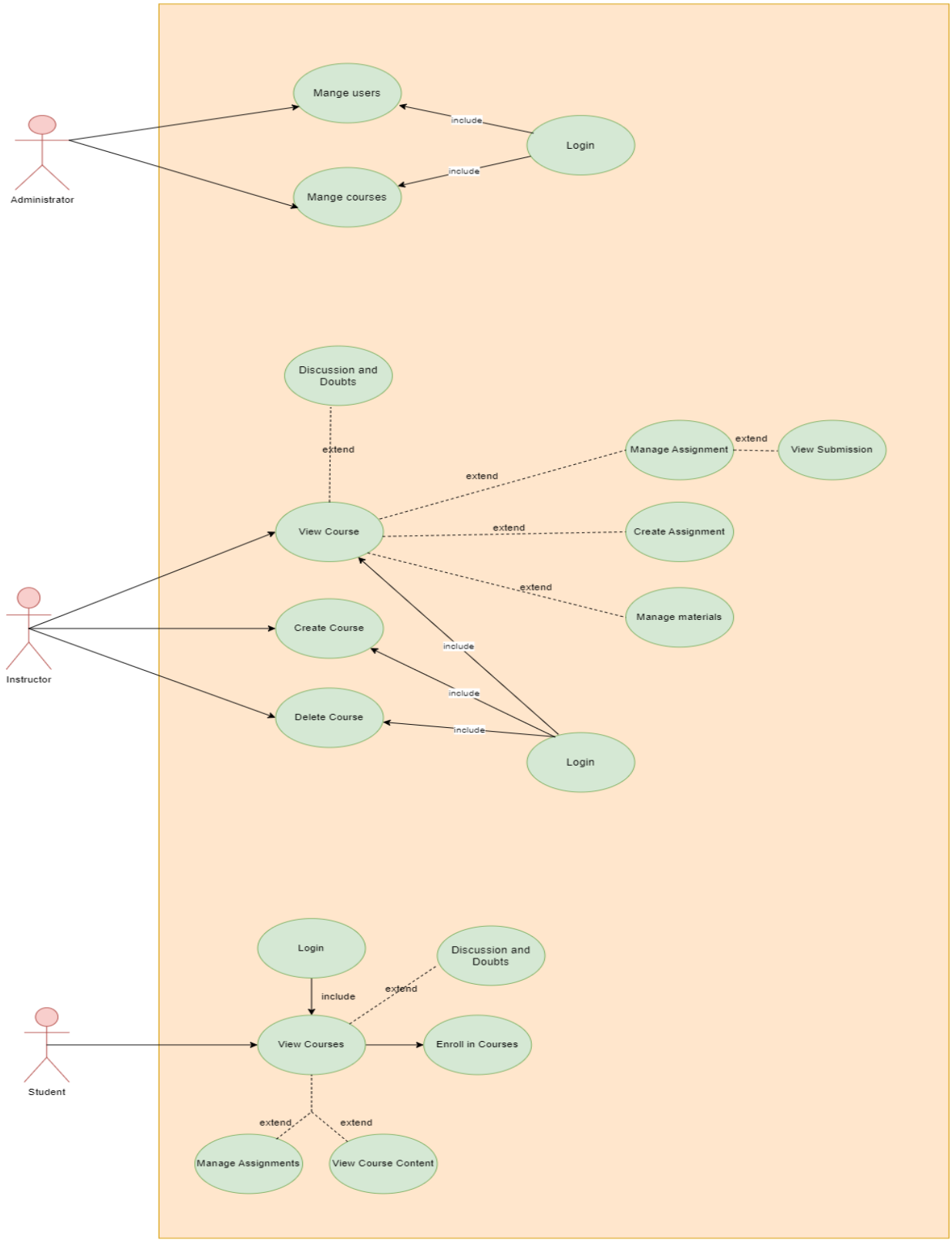
8. **Interoperability:** The system should be operable from different browsers or versions of the same browser.

9. **Accessibility:** The system should be accessible to users with disabilities in accordance with relevant regulations such as the Web Content Accessibility Guidelines (WCAG).

10. **Maintainability:** The system should be easy to maintain and update, with clear documentation and a modular design.

11. **Customizability:** The system should be customizable to meet the specific needs of different institutions and courses.

Use Case Diagram:



Process Model and Justification:

Agile Model (Scrum(Testing Methods))

Agile is a highly iterative and flexible project management approach that has proven to be effective in a variety of contexts, including online course management systems. There are several reasons why the agile model is particularly well-suited for online course management development:

1) **Iterative and flexible development process:** Agile model takes short development cycles which are called sprints, during which small portions of a project are completed. This allows for course content to be developed and tested incrementally, reducing the risk of investing a lot of time and resources into a course that may not meet the needs of students.

2) **Continuous feedback and improvement:** When frequent changes are required Agile model encourages regular feedback from stakeholders, including instructors, students,Teaching assistants, and administrators. This allows for the course to be continuously refined and improved based on the needs and preferences of the target audience.

3) **Customer is ready to be involved:** In this whole Project till completion we assume that we have got the feedback from the TA's about the Requirements that anyone would need, also we the project the TA's would be there and they will be ready to involve to have a meeting with a software team all the time.

4) **Project size is small:** Here with the Project timing of 3 months this is a small Project and developing it with Agile model would be more efficient.

5) **Collaboration and teamwork:** Agile places a strong role of collaboration and teamwork, which is particularly important in the context of our online course management development where multiple users, such as instructors, students,Teaching assistants, and administrators may be involved.

6) **Emphasis on value delivery:** Agile places a strong emphasis on delivering value to the customer, which in the case of online courses, would be the students. This means that the course content and delivery methods are continuously reviewed and optimized to ensure that students are receiving the best possible learning experience.

In conclusion, the Agile model's iterative, flexible, collaborative, and value-focused approach makes it well-suited to online course management, helping to ensure that courses are delivered effectively and efficiently.

Scrum(Agile Testing Methods) is an agile development process focused primarily on ways to manage tasks in team-based development conditions.

There are three roles in it, and their responsibilities are:

- Scrum Master
- Product owner
- Scrum Team

Here us per the definition of sprint is a short, time-boxed period when a scrum team works to complete a set amount of work. Here we have the lab every week so the sprint can be defined for one week and insurance of the work that is done can be seen.

Tools and Technologies:

- ASP.NET/ Node.js
- HTML, CSS, EJS/ Blazor or Razor
- PostgresQL