SRS Document for E- Learning Management System

SUBMITTED BY

NAME	USN
KAVYASRI R	**1MS21CS063**
JYOTHI YADAV	**1MS21CS056**

As part of the Course **Software Engineering - CS46**

SUPERVISED BY
Faculty
Darshana A Naik

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

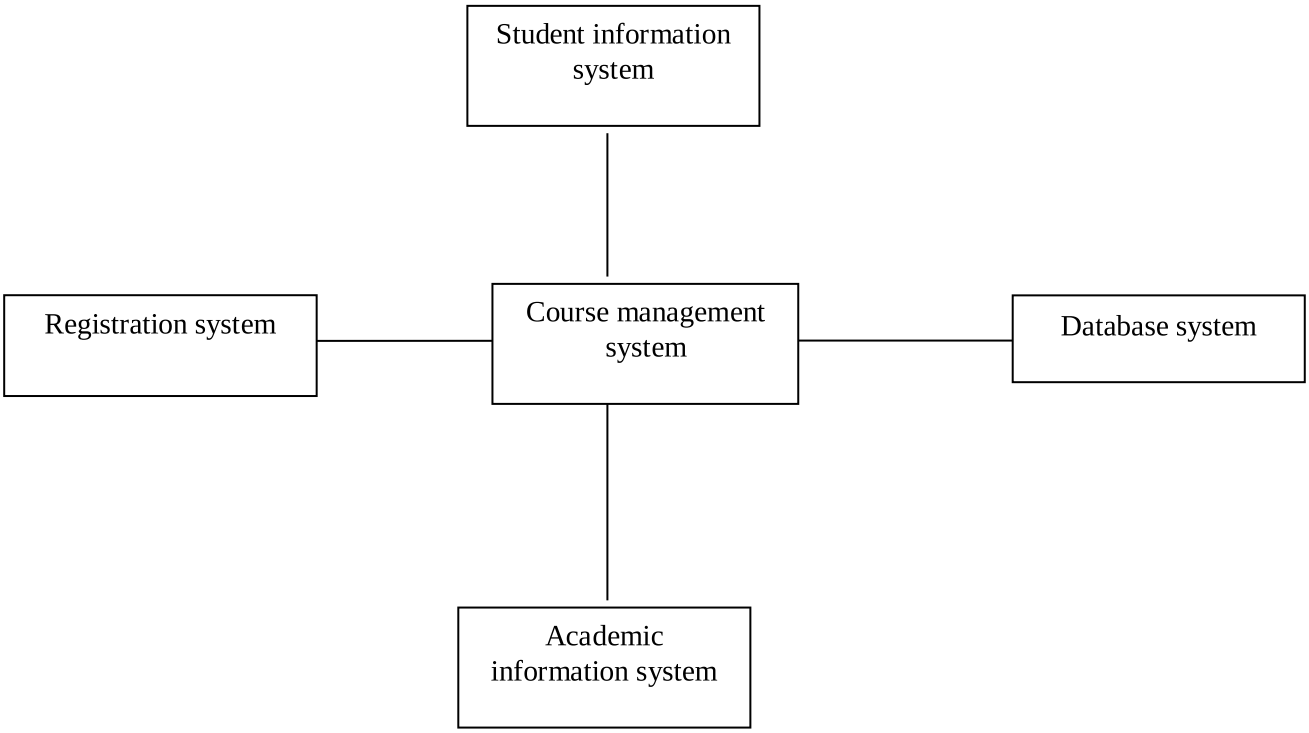
RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute, Affiliated to VTU)
Bangalore – 560054

TABLE OF CONTENTS

ABSTRACT

This document is to present a clear understanding of our E- Learning Platform—a web application designed for managing courses. It covers the purpose, features, interfaces, functionalities, operational constraints, and how the system responds to external factors. Intended for the acquirer, who acts as the owner, this document is aimed at providing a straightforward yet comprehensive insight into our envisioned E- Learning experience. The scope of the E- Learning Platform encompasses a user-friendly interface for course creation, management, and student engagement. It includes features for content delivery, assessments, and real-time collaboration. The platform aims to cater to diverse educational needs while ensuring scalability for future enhancements.

The system will operate within a university environment. This environment includes other systems that will interact with the proposed system. Therefore, interfaces between these systems are required.



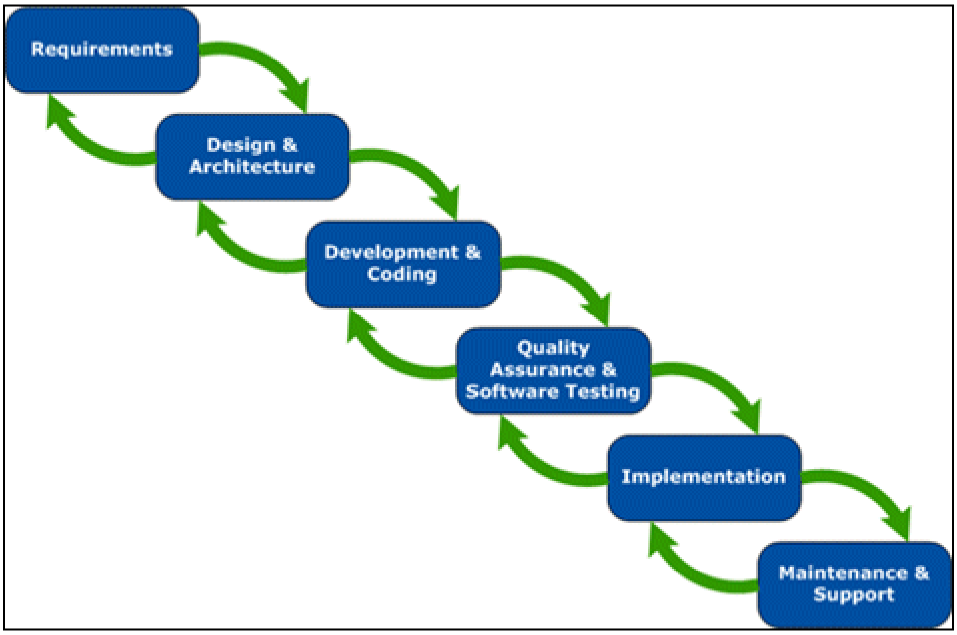
2. INTRODUCTION

PROBLEM STATEMENT

The challenge addressed by our E- Learning Platform lies in creating an accessible and efficient digital environment that seamlessly connects educators and learners. In response to the growing demand for remote education, our aim is to establish a user-friendly web application facilitating course creation, management, and interactive learning experiences. We strive to bridge the gap between educators and students in a dynamic online educational landscape.

SOFTWARE PROCESS MODEL

The choice of a software process model depends on various factors such as the project's size, complexity, requirements clarity, and the team's experience. For an E-Learning Platform (course management system), considering the need for adaptability and iterative development, an Agile process model could be suitable. Agile methodologies, like Scrum, allow for flexibility, regular feedback, and incremental improvements, which align well with the evolving nature of educational technology.



3. Software Requirement Specification

3.1 Functional Requirements

1. GUI: The E- Learning Platform's GUI aims for user-friendly interactions across various roles. The registration process features a clear form for users to select roles, input personal details, and verify via email. Academic staff find an intuitive interface for CV management, course addition, and notifications, while students easily enroll in courses, browse offerings, and engage in video conferences and chat functionalities. Subscription selection is streamlined for students and universities. Admin login, tailored for administrators, ensures a secure and separate interface for comprehensive user management. The GUI prioritizes simplicity, responsiveness, and a cohesive design for an enhanced user experience

2. Registration

Role(s): Student, Academic staff, University

Feature description/steps:

- The user will be given required and optional fields to fill out.
- The user will be given option to register as Student, Academic staff, or University.
- The user enters first and last name/University name, email address, password, repeat password.
- The user will need to verify registration via confirmation link that will be sent to the provided email address.

Dependencies/constraints: All required fields must be filled out with valid information. Registration must be confirmed in order to continue.

3. Login

Role(s): Student, Academic staff

Feature description/steps:

- The user will be given email and password.
- If user’s credentials are not valid display error message and ask the user to try again.
- After entering correct information user can access the system.

Dependencies/constraints: Users must be registered in order to login

4. Upload CV

Role(s): Academic staff

Feature description/steps:

- In the staff's profile page there will be an option to upload a CV file.
- The file can be changed and removed.
- The CV must be reviewed by our own staff in order for the employee to become verified.

Dependencies/constraints: Users must be registered as an Employee. File must be a pdf file.

5. Add course

Role(s): Academic staff

Feature description/steps:

- In the Academic staff's profile page there will be an option to add a course.
- The Academic staff will upload a course details file providing details about the course.
- The Academic staff will upload course materials for students.
- The Academic staff will upload the lecture schedule.

Dependencies/constraints: The Academic staff must be a verified staff member by the University (if enrolled via university) in order to add courses to the system. The course details file must be a pdf file.

6. Enroll course

Role(s): Student

Feature description/steps:

- Student will be able to enroll a course from the browse page by using course enrollment cred its.
- Enrolling in a course will give the student access to learning material provided by the academic staff, course lectures and feedback from the staff.

Dependencies/constraints: Student must have available course enrollment credits.

7. Browse courses

Role(s): Any user, registered or not

Feature description/steps:

Users will be able to browse courses on the browse page. Here they can find a variety of recommended courses as well as search for a course. Each course has a overview explaining what the course is about (in text or video form).

Dependencies/constraints: User must be registered to see recommended courses.

8. Video conferencing

Role(s): Student, Academic Staff

Feature description/steps:

- Academic staff should be able to create video chat for meetings or lessons. video chat can be temporary or permanent.
- Video chat for exams will be automatically created and deleted after the exam.

Dependencies/constraints: The video chat can have a maximum of 200 participants per call.

9. Chat

Role(s): Student, Academic Staff

Feature description/steps:

- Students can message each other in course group chat
- Students can message other students from the same university.
- Students can message other students by adding them using their student tag.
- Chat for lectures will be automatically created and deleted after the lecture.
- Students can message academic Staff.

10. Subscription selection

Role(s): Student, University

Feature description/steps:

- Students can select from various subscription packages giving them access to a certain number of course enrollment credits (prices vary) which they can later use for course enrollment. Universities can also select a subscription plan for the university, giving access to the creation of a certain number of courses depending on the package as well as additional options in the form of the number of registered students

Dependencies/constraints: Users must have valid payment info entered in the system to subscribe.

11. Admin login

Role(s): Administrator

Feature description/steps:

- Admin of LSM have a separate login form and from that position he can manage all users and their information easily.
- If certain problems occur, the administrator can easily delete or immediately disable the work of a particular university, professor or student.

Dependencies/constraints: The admin data is entered directly into the database by the project owner, so no register is required

3.2 Non-Functional Requirements

3.2.1 Performance:

1. Throughput: The web application must load completely within 3 seconds, ensuring a responsive and efficient user experience.
2. Concurrency: The video conferencing functionality should support up to 200 users in the same video call concurrently, ensuring seamless and reliable communication.

3.2.2 Accessibility:

Remote Access: The system should offer a web app to enable users to access the platform remotely, ensuring accessibility from various devices.

3.2.3 Customization:

1. Theme Customization: Uers should have the ability to customize the web application's theme, allowing for a personalized visual experience.
2. UI Element Customization: Users must be able to adjust the size of web app elements, providing flexibility in adapting the interface to individual preferences and needs.

3.2.4 Reliability:

Database Update: After a system update, the database must be fully operational within 2 seconds, ensuring minimal downtime and a quick return to full functionality.

3.2.5 Scalability:

Video Conferencing Scalability: The video conferencing feature should be scalable to accommodate a growing user base, with a capacity plan ensuring smooth performance even with increased demand.

3.2.6 Resource Utilization:

Optimized Resource Usage: The system should be optimized to use computing resources efficiently, ensuring that hardware resources are utilized effectively for improved overall performance.

3.2.7 Security:

1. Data Transmission Security: Video conferencing and data transmission should be encrypted to maintain the confidentiality and integrity of user data.
2. Access Control: Robust access controls should be implemented, ensuring that only authorized users have the ability to customize themes or modify application elements.

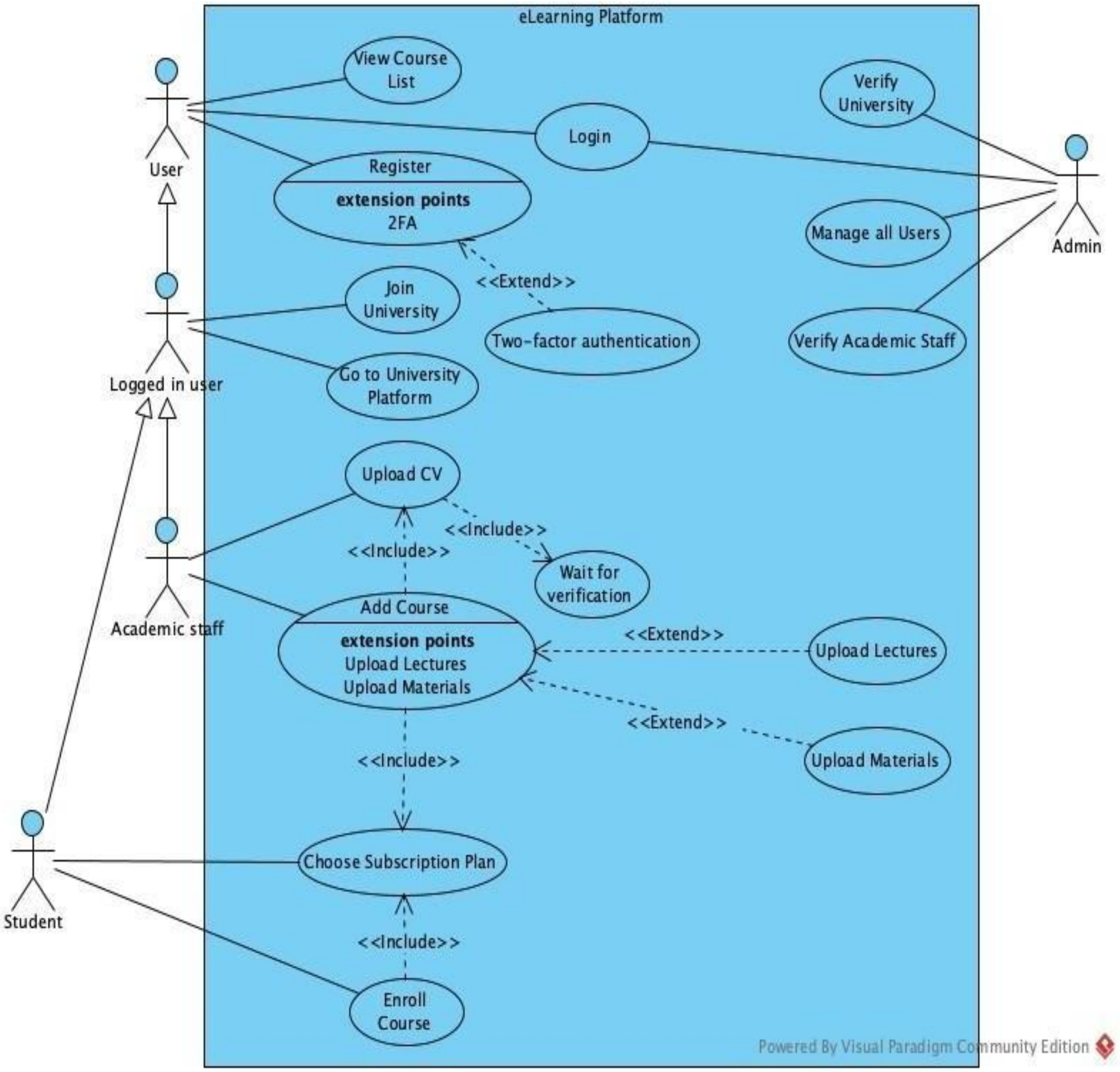
3.2.8 Usability:

User-Friendly Updates: System updates should be designed to minimize disruption, and any changes in the user interface or functionality should be communicated clearly to ensure a smooth transition for users.

3.3 Requirement Analysis

3.3.1 Use Case Diagram:

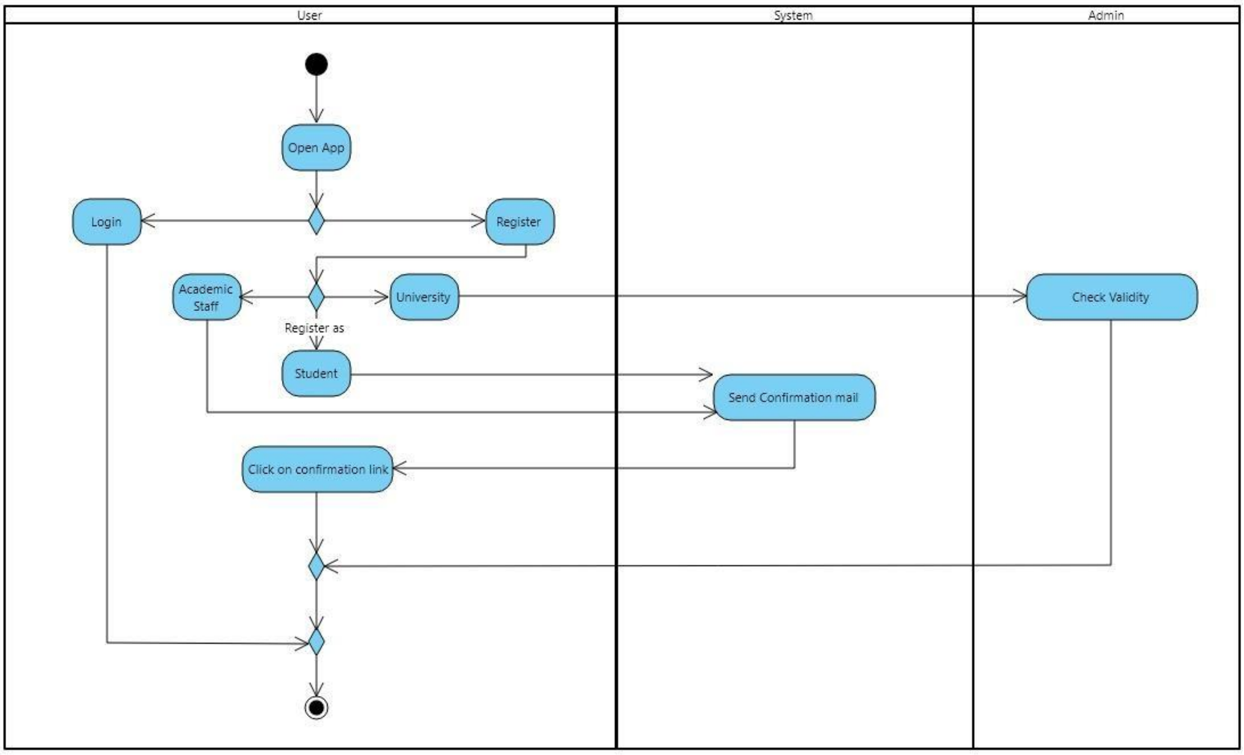
General



Use diagram name	E-Learning Platform
Actors	Users (Students, Academic Staff), Admin
Functionality	1. All users can view the course list. 2. All users can register; non-logged-in users, including admins, can log in. 3. Logged-in users can join universities or switch to the university platform. 4. Academic staff upload CV, awaiting verification. 5. Academic staff can add a course; if so, they must choose a subscription plan and can optionally upload lectures/materials. 6. Students can choose a subscription plan or enroll in a course, then select a plan.
Admin function	1. Admin verifies academic staff and universities. 2. Admin manages all users.
Verification process	1. Academic staff CVs wait for verification. 2. Admin manages the verification process.
Subscription and Enrollment	1. Students choose subscription plans. 2. Students can enroll in courses, selecting a plan.

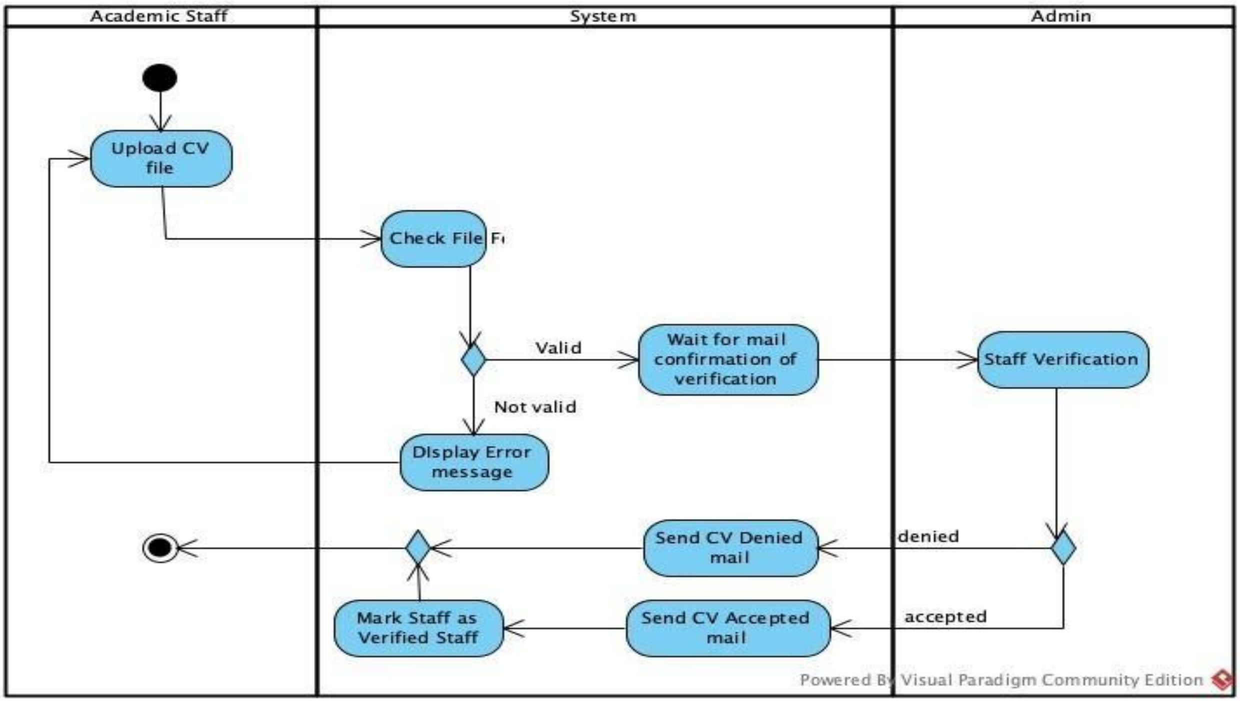
3.3.2 Activity diagrams

1. Register/Login diagram



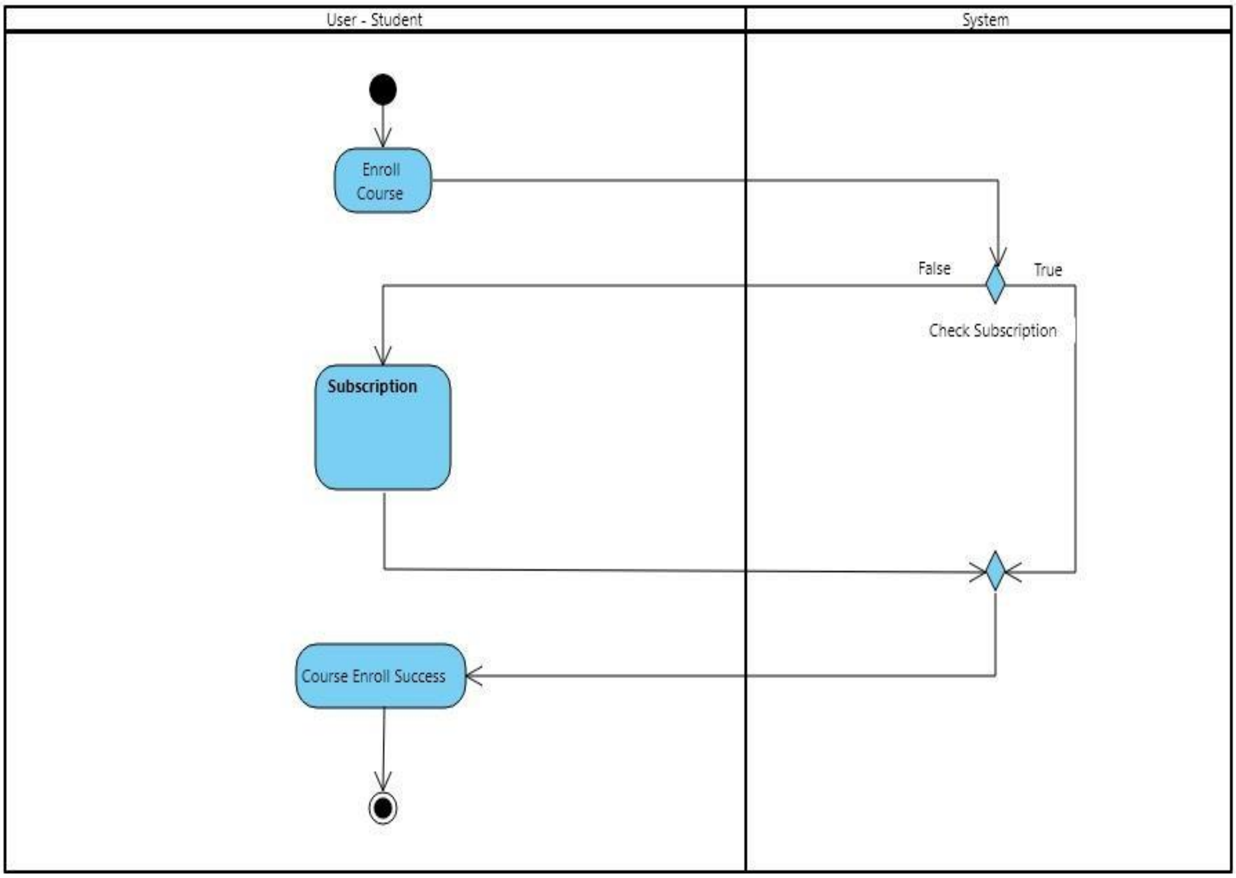
Activity diagram name	Registration/Login
Brief Description	Users have the option to register or log in. If registering, they choose their role (academic staff, student, or university). Each role undergoes verification by the system or administrator.
Actor	Academic Staff, Student, or University
Precondition	None
Basic flow	3. User selects the option to register or log in. 4. If registering, user chooses their role (academic staff, student, or university). 5. System checks are online specific verification requirements. 6. System verifies user details or admin reviews verification. 7. If logging in, user enters credentials. 8. System validates login credentials
Post condition	User is successfully registered, verified, and logged in based on selected options.

2. Upload CV diagram



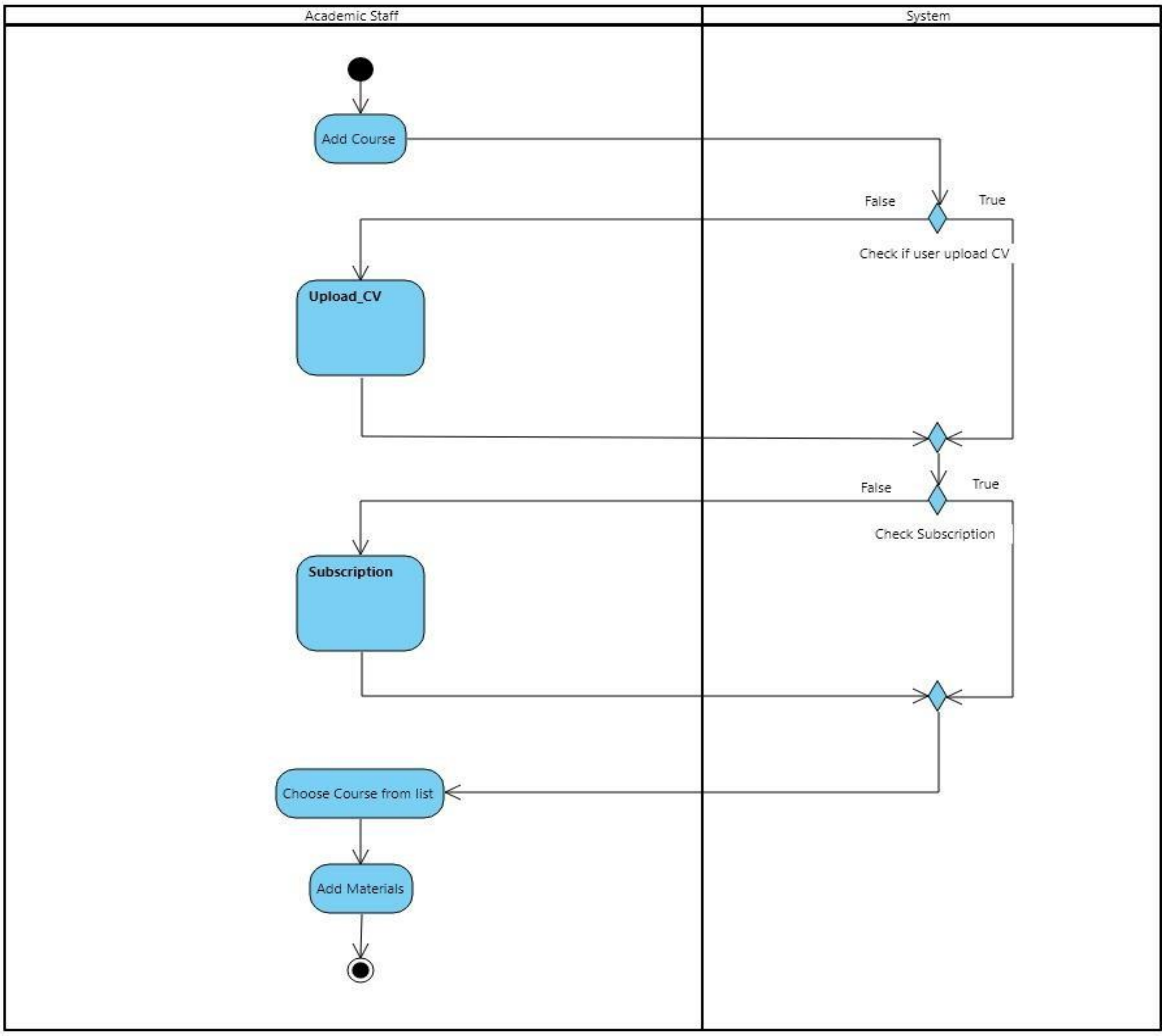
Activity diagram name	Upload CV
Brief Description	Academic staff can upload a CV. The system checks the file type. If valid, the admin is prompted to accept or deny the CV. The system displays an error or success message based on the admin's assessment.
Actor	Academic Staff
Precondition	None
Basic flow	1. Academic staff selects the option to upload a CV. 2. Academic staff uploads the CV file. 3. System checks the file type. 4. If the file type is valid, the admin is prompted for assessment. 5. Admin accepts or denies the CV. 6. If accepted, staff is marked as verified, and an email verification is sent. 7. If denied, a CV denial email is sent. 8. System displays an error or success message based on the admin's decision.
Post condition	CV is accepted or rejected based on admin assessment. If accepted, staff is marked as verified, and an email verification is sent. If denied, a CV denial email is sent.

3. Enroll course diagram



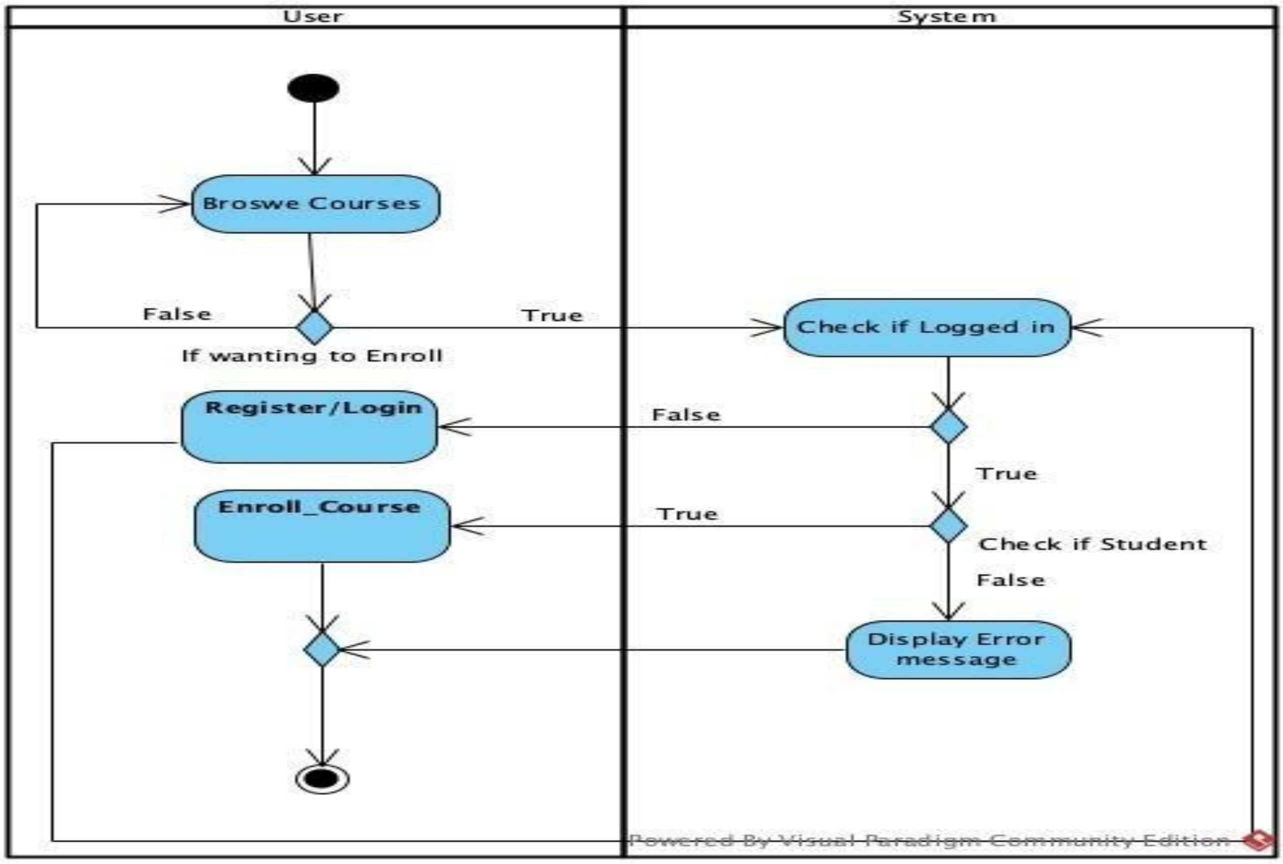
Activity diagram name	Enroll Course
Brief Description	Students can enroll in a course. The system checks if the student has a subscription. If not, they are prompted to subscribe; if yes, the student is successfully enrolled.
Actor	Student
Precondition	Student must be logged in.
Basic flow	1. 1. Student requests course enrollment. 2. System checks for a subscription. 3. If no subscription, prompt student to subscribe. 4. If yes, enroll the student in the course.
Post condition	Student is successfully enrolled if they have a subscription; otherwise, they are prompted to subscribe.

4. Add course diagram



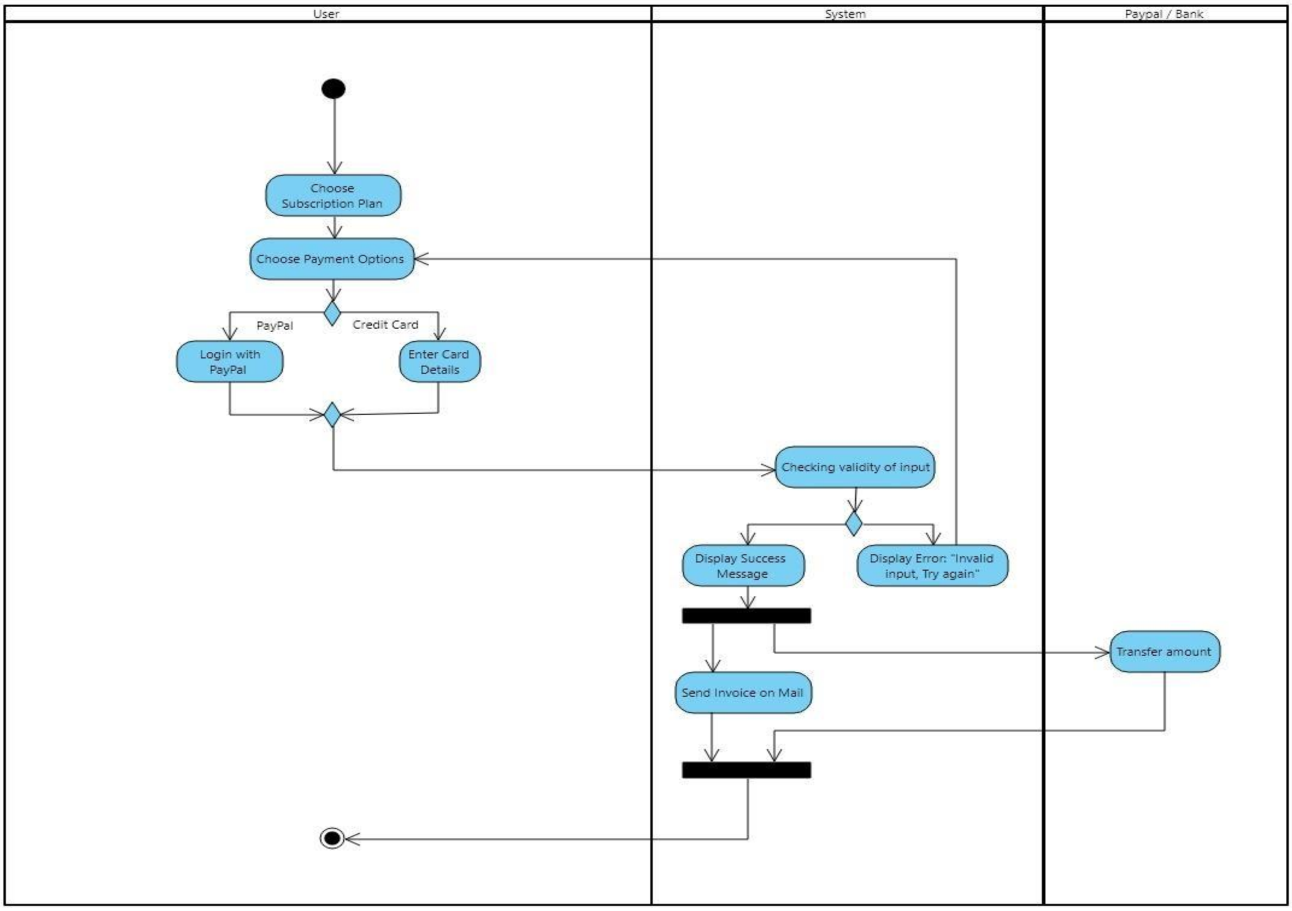
Activity diagram name	Add Course
Brief Description	Academic staff adds a course to the platform.
Actor	Academic Staff
Precondition	Academic staff must be verified.
Basic flow	1. Academic staff selects the option to add a course. 2. Checks whether the academic staff has uploaded the CV 3. Academic staff provides course details. 4. Academic staff selects a subscription plan. 5. Academic staff uploads lectures and materials.
Post condition	Course is added to the platform upon successful completion.

5. Browse courses diagram



Activity diagram name	Browse courses
Brief Description	Users, while browsing courses, may choose to enroll. The system checks if the user is logged in; if not, they are prompted to register or log in before returning to browsing. If logged in, the system verifies if the user is a student; only students can enroll, others receive an error message.
Actor	Students and academic staff,
Precondition	User must be verified.
Basic flow	1. User selects the option to enroll in a course. 2. System checks if the user is logged in. 3. If not logged in, system prompts user to register or login. 4. After registration or login, user returns to browsing. 5. If logged in, system checks if the user is a student. 6. If a student, the system allows enrollment; otherwise, an error message is displayed.
Post condition	User successfully enrolls in the course or receives an error message based on conditions

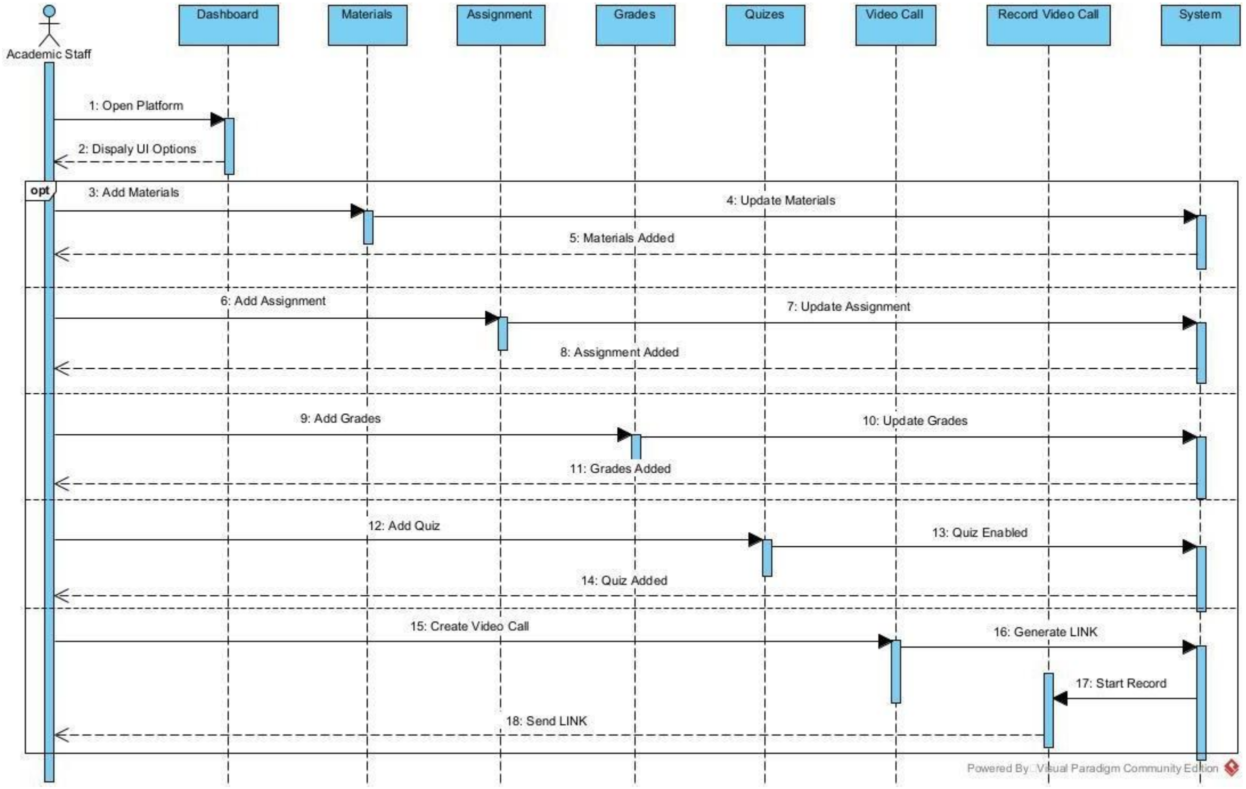
6.Subscription diagram



Activity diagram name	Subscription
Brief Description	Users select a subscription plan, provide payment details, and the system verifies input validity. Upon success, a payment option transfers the amount, and the system sends an invoice via mail. If unsuccessful, an error message is displayed.
Actor	All Users
Precondition	User must be logged in.
Basic flow	1. User selects a subscription plan. 2. User chooses a payment option. 3. User enters payment details. 4. System checks input validity. If valid, a success message is displayed. 5. Payment option transfers the amount and the system sends an invoice via mail
Post condition	User successfully completes subscription selection and payment, receiving a success message and invoice if valid; otherwise, an error message is displayed.

3.3.3 Sequence diagrams

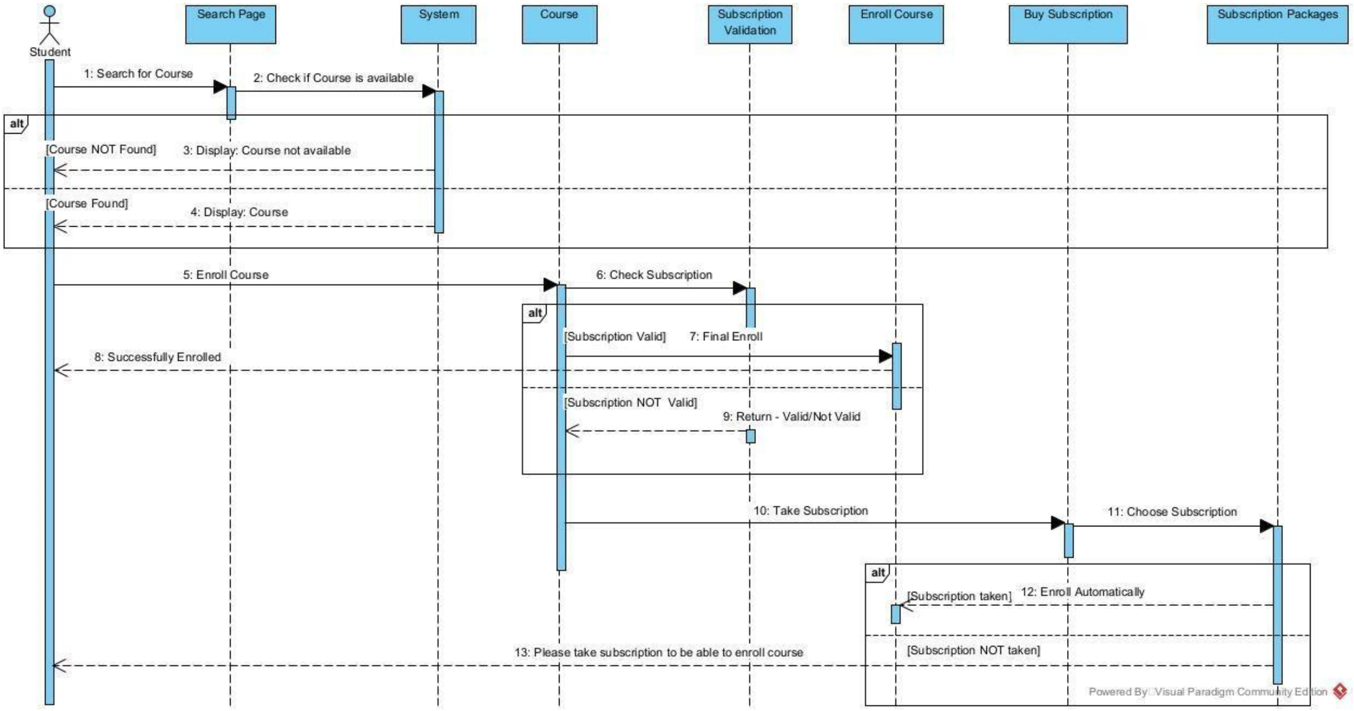
1. Instructor dashboard diagram



Description

On the Instructor dashboard, there are various choices available, including adding materials, assignments, grades, quizzes, or initiating a video call. When a user selects any option except creating a video call, such as adding materials, assignments, grades, or quizzes, the system updates the database and notifies the user of successful completion. However, creating a video call involves the user initiating a call, generating a link, starting recording, and receiving the link back, making it fundamentally distinct from the other options.

2. Student dashboard diagram

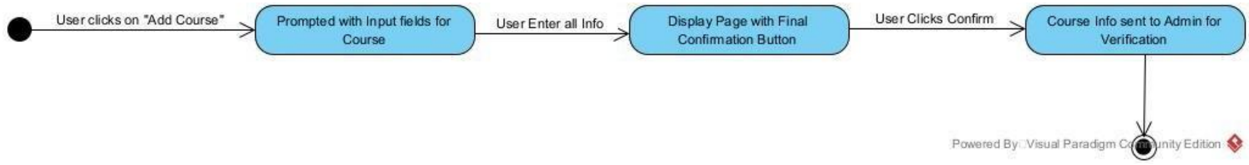


Description:

On the Student dashboard involves the user initially searching for their desired course. Subsequently, the Search Algorithm verifies the course's existence, displaying it to the user. The user then opts to enroll in the course, prompting a check for the validity of their subscription. If the subscription is valid, the user is successfully enrolled; otherwise, they receive a message indicating the subscription is invalid and are automatically redirected to the Subscription Page. If the user chooses to subscribe, they are automatically enrolled; otherwise, they receive a message encouraging them to subscribe for course enrollment.

3.3.4 Behavior state diagrams

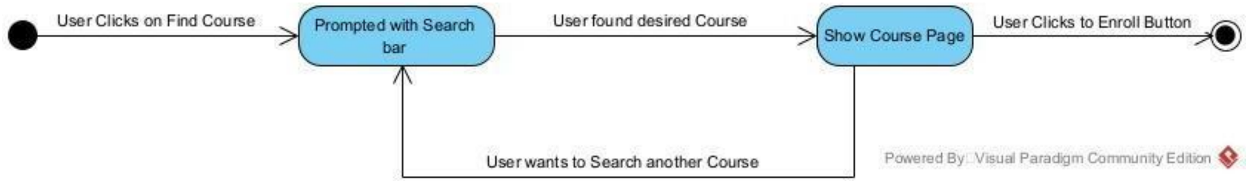
1. Add course diagram


User clicks on "Add Course"
Prompted with Input fields for Course
User Enter all Info
Display Page with Final Confirmation Button
User Clicks Confirm
Course Info sent to Admin for Verification
Powered By: Visual Paradigm Community Edition

Description:

To include a course on the platform, Instructor upon selecting "Add Course," encounters a set of fields to complete, encompassing details like Course Name, Description, Content, Price, etc. After inputting all information and proceeding, an Overview page displays, allowing the user to review the details before confirmation. Upon confirming, the request undergoes verification by the administration, marking the completion of the process.

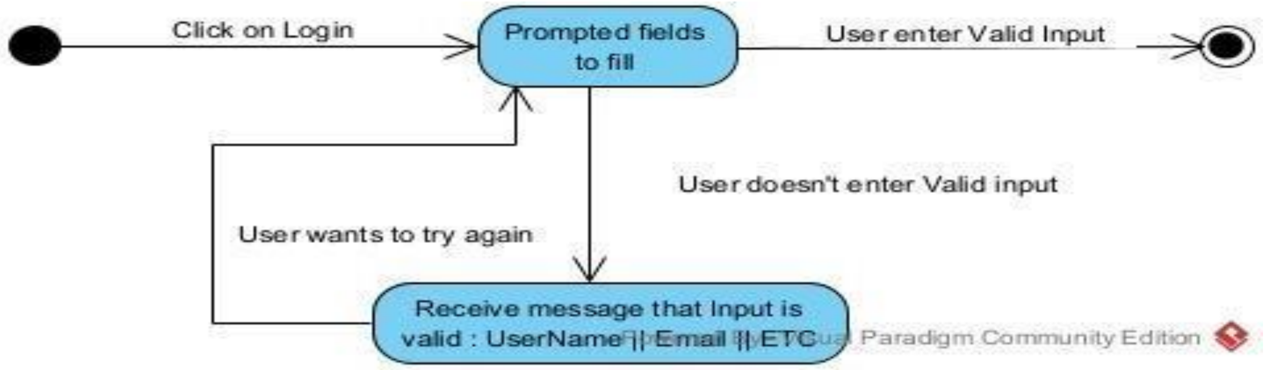
2. Enroll course diagram


User Clicks on Find Course
Prompted with Search bar
User found desired Course
Show Course Page
User Clicks to Enroll Button
User wants to Search another Course
Powered By: Visual Paradigm Community Edition

Description:

When a user intends to enroll in a specific course, they are provided with a search bar to input the desired course name. If the course exists, it is presented to the user, and upon selecting "Enroll," the user successfully enrolls in the course, concluding the process. In case the enrolled course is not found, the user has the option to initiate another search for available courses.

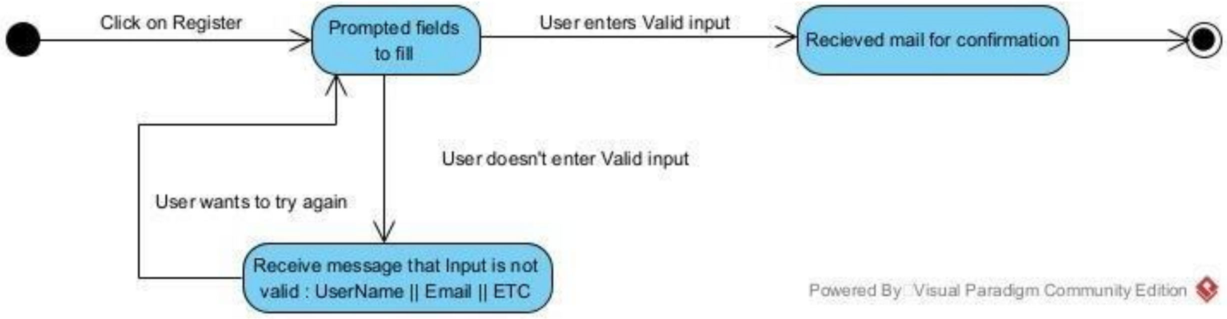
3. Login diagram


Click on Login
Prompted fields to fill
User enter Valid Input
User doesn't enter Valid input
User wants to try again
Receive message that Input is valid : UserName || Email || ETC
Powered By Visual Paradigm Community Edition

Description:

During the Login, the user is offered to enter the Login data, if the data is correct, the user is successfully logged in and this is the end of the process, if some of the data is incorrect, the user also has the opportunity to try again.

4. Register diagram


Click on Register
Prompted fields to fill
User enters Valid input
Recieved mail for confirmation
User doesn't enter Valid input
User wants to try again
Receive message that Input is not valid : UserName || Email || ETC
Powered By Visual Paradigm Community Edition

Description:

In the registration, the user is offered to fill in all the desired fields. If all fields are successfully filled in, the user receives a confirmation mail and this process is complete. If one of the fields is filled in incorrectly, the user receives a message where the error is and has the opportunity to re-enter the data.

4. Formal Technical Review (FTR):

- **Our Team:**

Kavyasri R	(1MS21CS063)
Jyothi Yadav	(1MS21CS056)

- **Reviewer Team:**

Charishma	(1MS21CS040)
Mihika	(1MS21CS075)
Pranay	(1MS21CS097)
Manoj	(1MS21CS072)

- **The Review meeting :** <Link>

- **Reviews given to us:**

Positive Points:

User-Friendly Enroll Process: The enrollment process is straightforward, allowing users to easily search for and enroll in desired courses.

Efficient Course Search: The inclusion of a search bar for finding courses enhances user convenience and makes it easier to locate specific courses.

Clear Confirmation Process: The overview page and confirmation step provide users with a clear and organized summary before finalizing their enrollment, contributing to a transparent user experience.

Areas for Improvement:

Course Discovery Enhancement: Consider improving the course discovery process by providing users with suggestions or categories to explore, especially if the entered course name does not yield results.

Real-time Course Availability: Explore the possibility of implementing real-time updates on course availability to ensure users are informed promptly, enhancing the overall responsiveness of the platform.

Streamlining Verification: Streamline the course enrollment verification process to expedite administrative checks and reduce the time between user confirmation and course access.

User Guidance: Consider incorporating additional guidance or tooltips during the enrollment process to assist users, ensuring a seamless and intuitive experience.

Feedback Mechanism: Implement a feedback mechanism to gather user input on the enrollment process, enabling continuous improvement based on user insight

- **Review given by us:**

Optimizing Search Algorithm: considering to optimize the search algorithm to improve accuracy and offer relevant suggestions, especially when a specific course is not found.

Real-time Updates Implementation: Exploring the possibility of implementing real-time updates on course availability to enhance user experience and provide timely information.

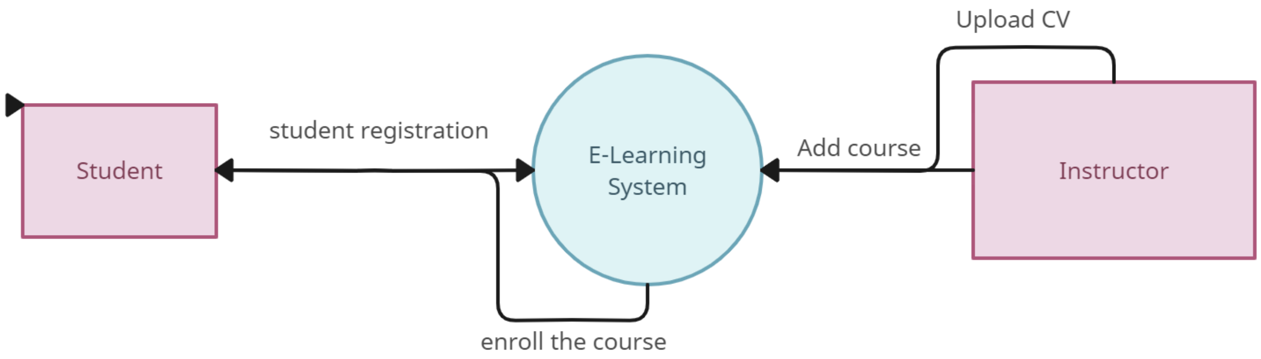
Enhanced User Guidance: Evaluating the need for additional user guidance or tooltips throughout the enrollment process, ensuring a smooth and user-friendly experience.

Feedback Collection Feature: Introducing a mechanism for collecting user feedback during the enrollment process, allowing users to provide insights and suggestions for continuous improvement.

5. Data modelling

Data flow diagram for E-learning management system (LMS) is generally used as a preliminary step towards creating an overview of the E-learning management system project without going into much detail. Data flow diagram in software engineering helps to model how data flows through the information system. This data flow diagram example of E-learning system describes the overall dataflow, processes and external entities. Here, we have 3 levels of DFD Level 0, 1 and 2 DFD.

Level 0 Data Flow Diagram

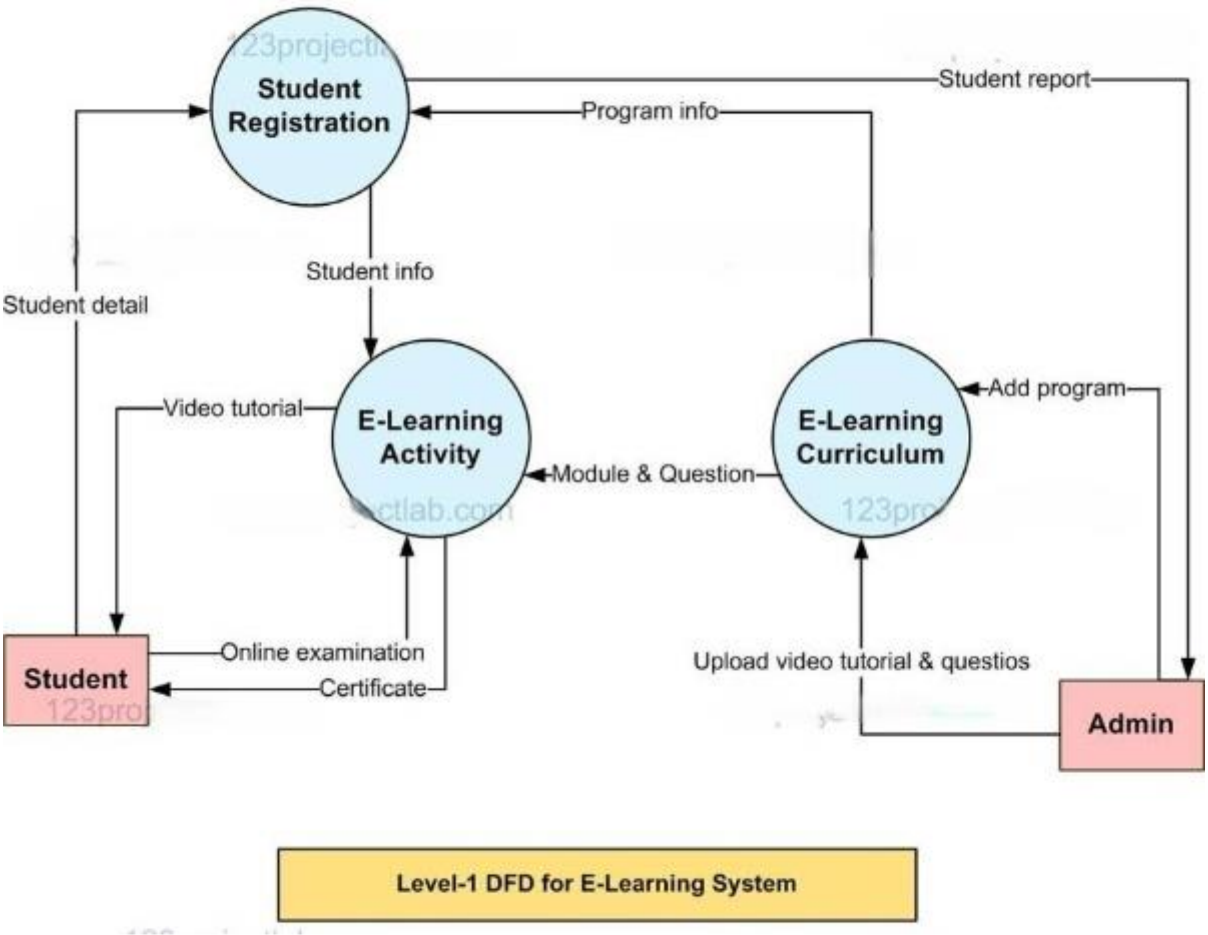


Level 0 DFD is also known as context level DFD. Level 0 DFD of context diagram for the E Learning management system represents the entire system as a single process and its relationship with external entities. This is basically an overview of the organizational system that shows the system boundaries, external entities and its interaction with the system

There are two external entities in this context DFD

• Student
• Admin

Level 1 Data Flow Diagram



Level 1 data flow diagram is also a general overview of the proposed system but it goes into more detail than a context diagram. Level 1 DFD for the E- Learning system breaks the single process into the main sub processes. E- Learning system has following sub process in its level 1 DFD

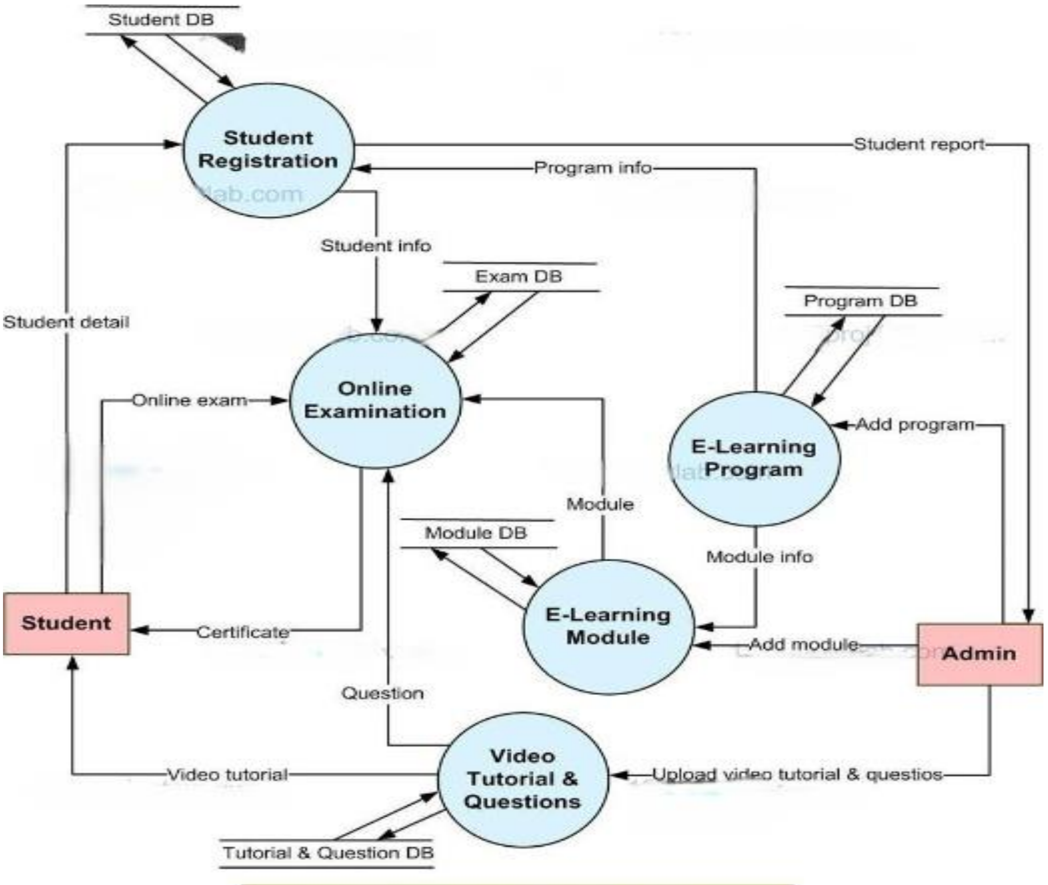
Student Registration: Student registration process stores and maintains records of all the students When a new student registers on the E- Learning platform, this process receives his details and store them in the database This process takes all the program information that are currently being run by the organization This process makes a master record of student information. This process also sends this Information to other processes as required.

E- Learning Curriculum: E- Learning Curriculum process maintains a curriculum of different programs that are being run by the organization. Admin can add new program details or edit any existing program Each program is divided into several modules. Admin can also upload video tutorials and question and answer of all the modules.

E- Learning Activity: E- Learning activity is a process for checking the learning progress of each student through online examination. This process takes student information from the Student Registration process, module and question information from the E- Learning Curriculum process.

Admin can upload video tutorials of different modules. Students can access these video lectures according to their program and learn

Level 2 Data Flow Diagram



Level-2 DFD for E-Learning System

Level 2 data flow diagram further breaks the level 1 process into smaller sub processes. This level also adds information about datastores.

In this level 2 DFD of E- Learning system the "E- Learning Activity process is further divided into

- ONLINE Examination
- Video Tutorial & Questions

E- Learning Curriculum is further divided into

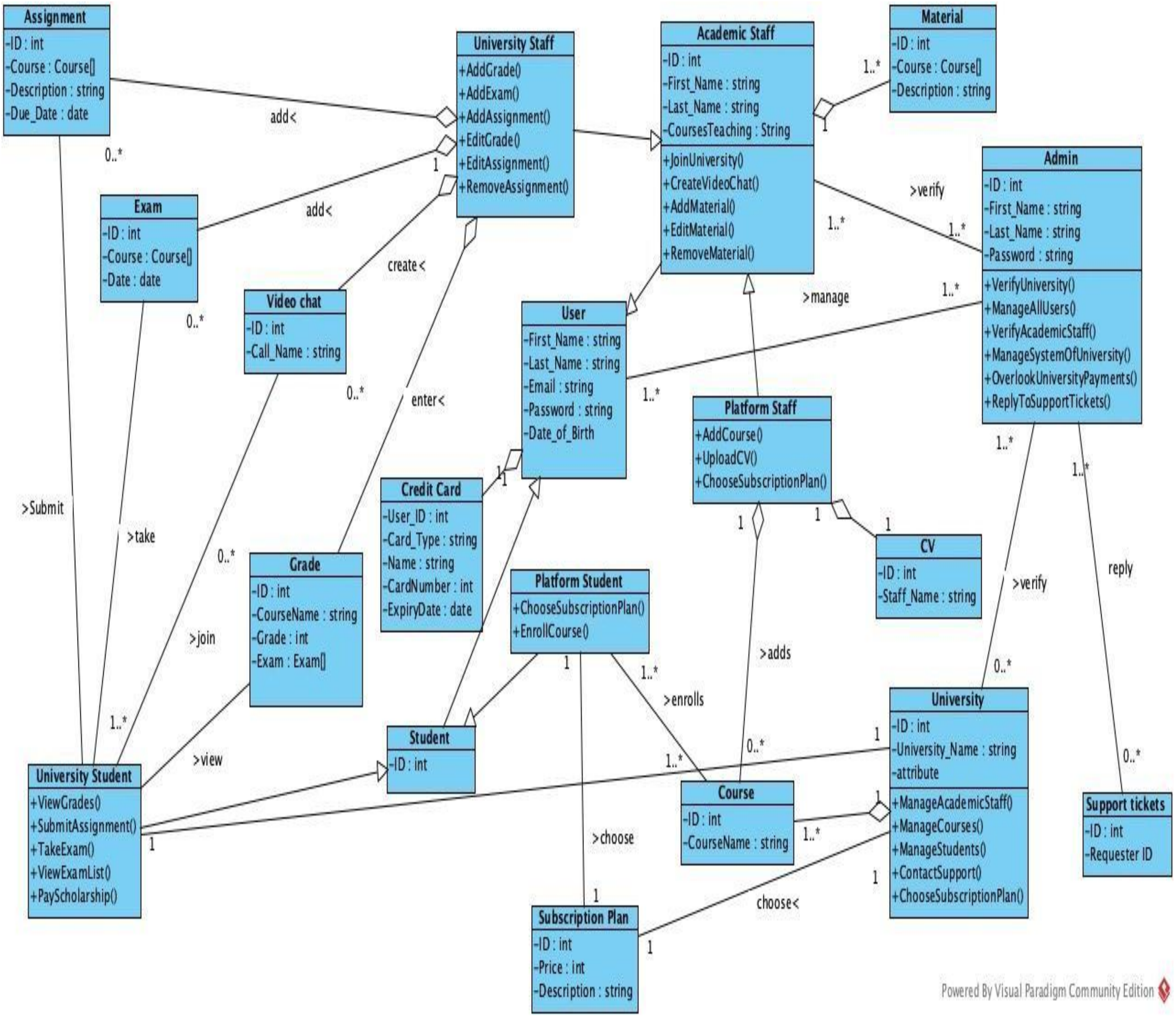
- E- Learning Program
- E- Learning Module

Low Level Functionality of E- Learning Management System

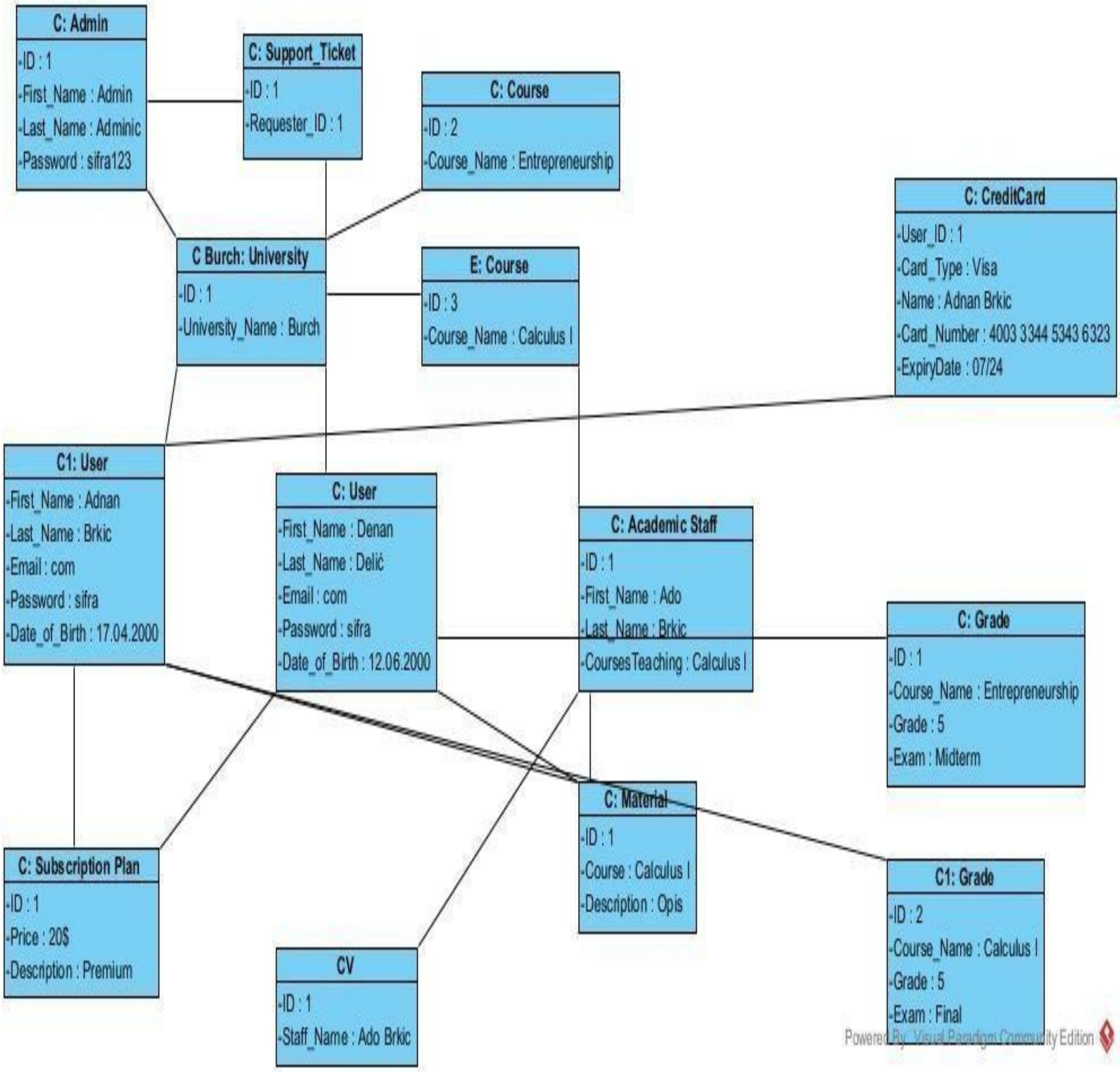
- This is an ONLINE application for distance learning purpose
- There are two main users of the system: Admin and student
- Admin has full control over the system. He can add new program & modules
- Admin can upload video lectures and related question
- Admin can view student report that are currently registered
- Students can register ONLINE on the portal and choose from listed learning programs
- Students can watch video lectures and answer questions at the end of lecture
- This system also conducts ONLINE examination of each program and student have to participate in the examination

6. Object and Class diagrams

Class diagram



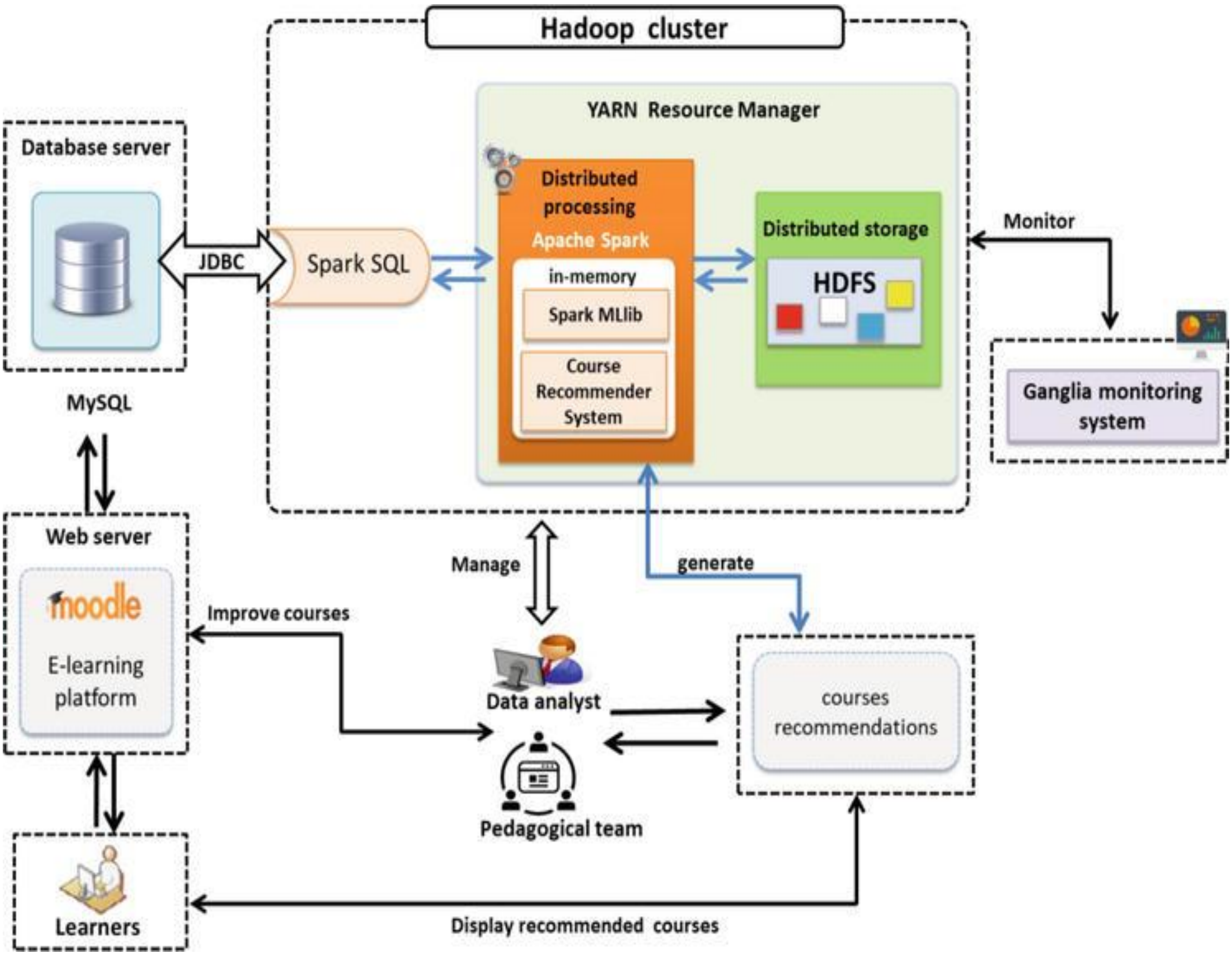
Object diagram



7. Software Architecture

7.1 High Level Design

High-level design for our E- Learning platform involves conceptualizing the overall structure and interactions between major system components. It provides a blueprint outlining the system's architecture, major modules, and their interconnections, offering a strategic overview for development. This phase sets the foundation for the subsequent low-level design and implementation stages.



Why Data-flow architecture?

A data flow architecture is well-suited for our E- Learning platform due to its ability to streamline and automate the flow of information across various functionalities. Just like a well-orchestrated data pipeline, the architecture facilitates seamless data transitions from user interactions, such as registration and course enrollment, through processing and validation steps, ultimately persisting the data in dedicated databases. This approach ensures efficient handling of diverse data sources, supports both real-time and batch processing, and allows for scalability and flexibility in managing the dynamic requirements of an E- Learning environment. The clear delineation of data flow stages, from ingestion to processing and persistence, contributes to a robust and automated system, enhancing the overall performance and reliability of our platform.

Components of the Data Pipeline:

1. Ingestion:

- Description: Ingestion involves collecting data from various sources and bringing it into the platform for further processing.

- Components:
  - User Registration and Login Forms
  - Course Enrollment Credits
  - Subscription Selections
  - Chat Messages
  - CV Files and Course Details

2. Processing:

- Description: Processing encompasses the handling, transformation, and enrichment of data to make it suitable for storage and analysis.

- Components:
  - Registration Controller: Validates and processes user registration data.
  - Course Management Controller: Manages coursonlinerelated operations.
  - Chat Controller: Handles messages in course group chats and private messages.
  - CV Upload Controller: Manages CV file uploads and verification.
  - Subscription Controller: Handles subscription selections and payment processing.

- Framework: Custom Framework for Data Processing: Enables various teams (e.g., Registration, Course Management) to pull and process data according to their needs.

3. Persistence:

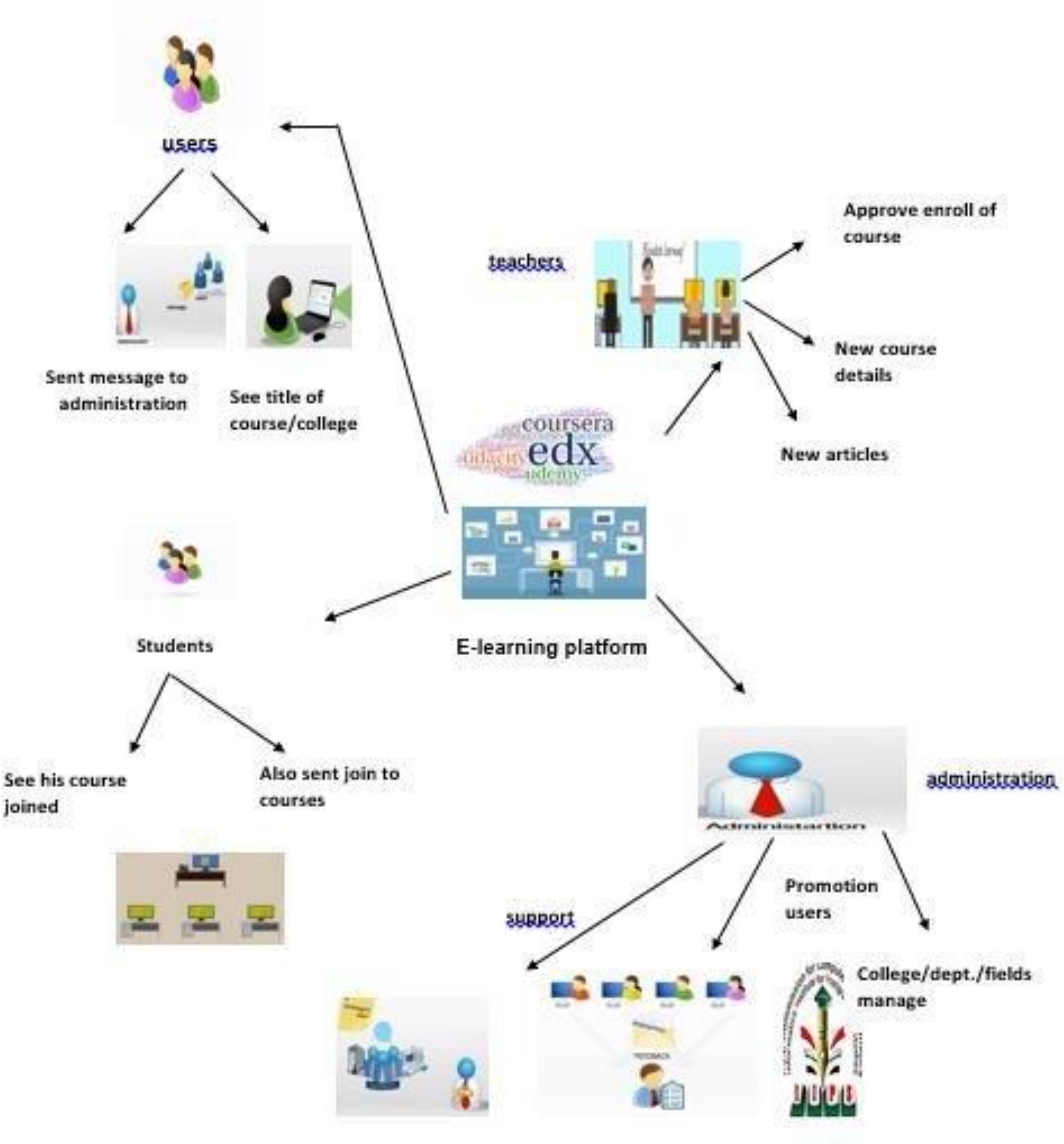
- Description: Persistence involves storing processed data in a reliable and accessible manner for future retrieval.
- Components:
  - User Database: Stores user profiles and authentication information.
  - Course Database: Stores course details, materials, and enrollment information.
  - Chat Database: Stores chat messages and session details.
  - Subscription Database: Stores user subscription details.
  - CV Database: Stores CV files and verification status.
- Tools:
  - Relational Databases for structured data.
  - Document Stores (e.g., MongoDB) for unstructured data.
- Data Retention: Define retention policies based on the type of data (e.g., user profiles retained indefinitely, chat messages retained for a certain period).

Data Flow:

- **User Interaction:** Users interact with the platform through various features like registration, course enrollment, chat, and subscription selection.
- **Ingestion:** Data from user interactions is ingested into the platform, including user registration details, course enrollment actions, chat messages, and subscription selections.
- **Processing**: Controllers process the incoming data based on the specific functionalities (e.g., Registration Controller processes user registrations, Chat Controller processes chat messages).
- **Framework for Processing:**A custom framework allows different teams (e.g., Registration, Course Management) to have ownership of pulling and processing data specific to their functionalities.
- **Spark Processing Engine**: Leverage a powerful processing engine (e.g., Apache Spark) for heavy lifting, cleaning, and transformations on large datasets.
- **Persistence:** Processed data is stored in dedicated databases (e.g., User Database, Course Database) based on the nature of the data.
- **External Services Interaction:** External services (e.g., Payment Gateway) are integrated during the subscription selection process for payment verification.

7.2 Low Level Design

Low-level design for our E- Learning platform entails specifying detailed functionalities, data structures, and algorithms for individual modules. It serves as a granular guide for developers, translating high-level architectural concepts into implementable components, and ensures the precise and efficient implementation of features.



1. User Registration Module:

- Module Design:

  Components:

  - UserRegistrationController
  - UserRegistrationService
  - UserRepository

- Data Structure Design:

  User entity with fields like userId, firstName, lastName, email, password, etc.

- Algorithm Design:
  - Validate user inputs (e.g., email format, password strength).
  - Generate confirmation link and send email for verification.
  - Persist user data in the database.

- Interface Design:

  registerUser(userData: UserData): Result

- Database Schema Design:

  User table with appropriate fields and constraints.

- Security Considerations:
  - Encrypt user passwords using a secure hashing algorithm.
  - Implement account lockout mechanisms for multiple failed login attempts.

2.Course Management Module:

- Module Design:

  Components:

  - CourseManagementController
  - CourseManagementService
  - CourseRepository

- Data Structure Design:
  - Course entity with fields like courseId, courseName, description, etc.
  - Lecture entity with fields like lectureId, title, content, etc.

- Algorithm Design:
  - Create new courses, upload course materials, and schedule lectures.
  - Validate course details before persistence.
  - Retrieve course information for users.

- Interface Design:
  - createCourse(courseData: CourseData): Result
  - uploadCourseMaterials(courseId: string, materials: Material[]): Result
  - scheduleLecture(courseId: string, lectureData: LectureData): Result

- Database Schema Design:
  - Course and Lecture tables with appropriate fields and relationships.

- Security Considerations:
  - Authorize only verified academic staff to add courses.

3. Add Course Module:

- Module Design:

  Components:
  - AddCourseController
  - AddCourseService
  - CourseRepository

- Data Structure Design:
  - Extend Course entity with additional details like courseDetailsFile for course information.

- Algorithm Design:
  - Validate and process course details file.
  - Persist course details and materials in the database.
  - Check academic staff verification status before adding a course.

- Interface Design:
  - addCourse(academicStaffId: string, courseData: CourseData): Result

- Database Schema Design:
  - Extend the Course table with additional fields for course details and materials.

- Security Considerations:
  - Ensure only verified academic staff can add courses.
  - Implement appropriate access controls to manage course additions.

4. Enroll Course Module:

- Module Design:
  Components:
  - EnrollCourseController
  - EnrollCourseService
  - EnrollmentRepository

- Data Structure Design:
  - Enrollment entity with fields like enrollmentId, studentId, courseId, enrollmentDate, etc.

- Algorithm Design:
  - Deduct course enrollment credits from the student's account.
  - Validate the availability of course enrollment credits.
  - Persist enrollment details in the database.

- Interface Design:
  - enrollCourse(studentId: string, courseId: string): Result

- Database Schema Design:
  - Enrollment table with fields for student, course, and enrollment details.

- Security Considerations:
  - Authenticate the student before enrolling in a course.
  - Ensure proper validation to prevent unauthorized enrollments.

5. Upload CV Module:

- Module Design:
  Components:
  - UploadCVController
  - CVVerificationService
  - CVRepository

- Data Structure Design:
  - CV entity with fields like cvId, staffId, fileName, uploadDate, etc.

- Algorithm Design:
    - Validate the CV file format (e.g., PDF).
    - Trigger the verification process by staff and set verification status.
    - Store CV details in the database.

- Interface Design:
    - uploadCV(staffId: string, cvFile: File): Result

- Database Schema Design:
    - CV table with fields for staff, CV details, and verification status.

- Security Considerations:
    - Authenticate the staff before allowing CV uploads.
    - Implement secure file upload mechanisms and validate file types.

8. Testing Phase

1. System Testing:

System Testing for your E- Learning platform is essential to ensure the fully integrated software operates as expected. This phase involves validating end-to-end functionalities, from user registration to course management and chat features. The focus is on assessing overall system behavior and interactions between different components, ensuring a seamless user experience.

2. Integration Testing:

Integration Testing is crucial to verify the smooth interactions between various modules and components within your E- Learning platform. It specifically tests the communication and data flow between functionalities like user registration, course management, and chat features. This ensures that all integrated components work cohesively to deliver a unified and functional system.

3. Validation Testing:

Validation Testing is the final step to confirm that your software meets all specified requirements and is ready for production. This phase involves validating informational, functional, behavioral, and performance aspects. The objective is to ensure that the software aligns with end users' needs, providing a comprehensive assessment of the software's overall quality.

4. Unit Testing:

Unit Testing focuses on verifying the smallest units of software, such as individual modules within your E- Learning platform. By testing internal processing logic and data structures, this phase ensures the correctness of individual functionalities like user registration, course addition, and chat messaging. Unit Testing is crucial for identifying and rectifying errors within the boundaries of each component.

5. Smoke Testing:

Smoke Testing is performed to quickly validate critical functionalities after a software build for your E- Learning platform. This includes executing essential functionalities like login and course access to check for major issues. The primary goal is to ensure that core features are working fine before proceeding with more detailed testing, saving time and resources.

Test cases

Description	State Before	Input Value	Expected Result
Login	User not logged in	Valid credentials (username, password)	Successful login, access to user dashboard
Register	User not registered	User details (name, email, password)	Successful registration, user in the system
Upload CV (instructor)	Academic staff profile	CV file (PDF format)	CV uploaded successfully, pending review
Add Course (instructor)	Academic staff logged in	Course details (name, materials)	Course added, visible in the course list
Enroll Course	Student logged in	Course selection	Successful enrollment, access to course materials
Subscription	User without subscription	Subscription plan selection	Subscription activated, access to features
Add Course (Student)	Student logged in	Course details (name, materials)	Unauthorized action, error message displayed
Upload CV (Student)	Student logged in	CV file (PDF format)	Unauthorized action, error message displayed

9. REFERENCES

1. https://123projectlab.com/data-flow-diagram-for-E- Learning-management-system/

2. https://app.creately.com/d/

3. https://www.researchgate.net/figure/Architecturonlinediagram-of-E- Learning-System-Thonlineapproach-of-architecturonlineis-as-follows-1_fig2_251418341

4. https://polcode.com/resources/blog/how-to-plan-softwaronlinearchitecturonlinefor-E- Learning-platforms/

5. https://arateg.com/blog/how-to-build-an-E- Learning-marketplaconlinelikonlineCoursera

6. https://www.geeksforgeeks.org/differenconlinebetween-high-level-design-and-low-level-design/

7. https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-usonlinecase#:~:text=Use%2Dcase%20diagrams%20describe%20the,how%20the%20system%20operates%20internally.

8. https://www.javatpoint.com/uml-statonlinemachinonlinediagram

9. https://www.freeprojectz.com/dfd/E- Learning-management-system-dataflow-diagram

10. http://worldcomp-proceedings.com/proc/p2011/EEE2784.pdf