**SOFTWARE DEVELOPMENT SPECIFICATION**

**Enhancing Engagement in Video Conferencing Based Learning through Predictive Clustering, Targeted Interactions, and Real Time Analytics**

**Tech Snatchers**

|  |  |  |
| --- | --- | --- |
| Name | Registration Number Index | Index Number |
| Shawmica Sivatharan | ICT/2021/119 | 5274 |
| Vimalan Arunpragash | ICT/2021/011 | 5173 |
| Keran Shama Sudharshan | ICT/2021/121 | 5276 |
| Abdul khan Wafry Ahamed | ICT/2021/003 | 5165 |
| Prashanthy Kugathas | ICT/2021/056 | 5217 |

Supervised By: Dr.K.A.S.H.Kulathilake

**Faculty of Applied Sciences**

**Rajarata University of Sri Lanka**

**2020/2021 Batch**

Table OF Contents

[**ABSTRACT** 7](#_Toc219055626)

[**ACKNOWLEDGEMENT** 8](#_Toc219055627)

[**CHAPTER 1 – INTRODUCTION** 9](#_Toc219055628)

[**1.1 Problem to be addressed** 9](#_Toc219055629)

[**1.2 Objectives of the Project** 10](#_Toc219055630)

[**1.3 Deliverables** 11](#_Toc219055631)

[**1.4 System Design Approach** 11](#_Toc219055632)

[**1.4.1 The Product Backlog (PB) Items** 12](#_Toc219055633)

[**1.5 Standards to be followed** 15](#_Toc219055634)

[**1.6 Organization of the SDS** 15](#_Toc219055635)

[**CHAPTER 2 -ARCHITECTURAL DESIGN** 17](#_Toc219055636)

[**2.1 System Architecture** 17](#_Toc219055637)

[**2.2 Object and communication** 18](#_Toc219055638)

[**2.2.1 Register** 18](#_Toc219055639)

[**2.2.2 Login** 19](#_Toc219055640)

[**2.2.3 Session** 20](#_Toc219055641)

[**2.2.4 Course** 21](#_Toc219055642)

[**2.2.5 Question** 22](#_Toc219055643)

[**2.2.6 Analytics** 23](#_Toc219055644)

[**2.2.7 Report** 24](#_Toc219055645)

[**2.2.8 Class Diagram of the System** 25](#_Toc219055646)

[**2.3 State Machines** 27](#_Toc219055647)

[**2.4 Deployment of the System** 28](#_Toc219055648)

[**2.5 Processes and Special Algorithms** 30](#_Toc219055649)

[**2.5.1 WebRTC-Based Network Data Retrieval Algorithm** 30](#_Toc219055650)

[**2.6 Integrating and Developing Tool Kit** 31](#_Toc219055651)

[**CHAPTER 3 - UI DESIGN** 33](#_Toc219055652)

[**3.1 PACT Analysis** 33](#_Toc219055653)

[**3.1.1 People** 35](#_Toc219055654)

[**3.1.2 Activities** 35](#_Toc219055655)

[**3.1.3 Contexts** 36](#_Toc219055656)

[**3.1.4 Technologies** 37](#_Toc219055657)

[**3.2 UI design consideration and approach** 38](#_Toc219055658)

[**3.3 Design tools and Techniques** 39](#_Toc219055659)

[**3.3.1 Design Tools** 39](#_Toc219055660)

[**3.3.2 Techniques** 39](#_Toc219055661)

[**3.4 Hosting / Installation Environment** 50](#_Toc219055662)

[**CHAPTER 4 – DATA MANAGEMENT** 51](#_Toc219055663)

[**4.1 Introduction** 51](#_Toc219055664)

[**4.2 Justification for Selecting MongoDB** 51](#_Toc219055665)

[**4.3 Data Requirements** 51](#_Toc219055666)

[**4.4 Database Design Approach** 52](#_Toc219055667)

[**4.5 Identification of Application Workload** 52](#_Toc219055668)

[**4.6 Collection Design** 52](#_Toc219055669)

[**4.6.1 Users Collection** 52](#_Toc219055670)

[**4.6.2 Courses Collection** 53](#_Toc219055671)

[**4.6.3 Sessions Collection** 53](#_Toc219055672)

[**4.6.4 Zoom Participants Collection** 53](#_Toc219055673)

[**4.6.5 Questions Collection** 54](#_Toc219055674)

[**4.6.6 Quiz Answers Collection** 54](#_Toc219055675)

[**4.6.7 Clusters Collection** 54](#_Toc219055676)

[**4.7 Data Relationship Mapping** 55](#_Toc219055677)

[**4.8 Application of MongoDB Schema Design Patterns** 55](#_Toc219055678)

[**4.9 Indexing Strategy** 55](#_Toc219055679)

[**4.9.1 Common Query Patterns in the System** 55](#_Toc219055680)

[**4.9.2 Indexes Implemented** 56](#_Toc219055681)

[**4.9.3** **Collections & Schema design** 57](#_Toc219055682)

[**4.10 Security Design** 64](#_Toc219055683)

[**4.11 ER Diagram** 65](#_Toc219055684)

[**4.12 Relational Database Mapping** 65](#_Toc219055685)

[**4.12.1 Map Regular Entities** 65](#_Toc219055686)

[**4.12.2 Map Weak Entities** 66](#_Toc219055687)

[**4.12.3 Binary Relationships** 67](#_Toc219055688)

[**4.12.7 Map Aggregation** 68](#_Toc219055689)

[**4.13 Schema Refinement** 68](#_Toc219055690)

[**4.14 Physical Database Design** 69](#_Toc219055691)

[**CHAPTER 5 - RESEARCH DESIGN** 72](#_Toc219055692)

[**5.1 Objective Based Literature Review** 72](#_Toc219055693)

[**5.1.1 Literature Review for Objective – 1** 72](#_Toc219055694)

[**5.1.2 Literature Review for Objective – 2** 73](#_Toc219055695)

[**5.1.3 Literature Review for Objective – 3** 73](#_Toc219055696)

[**5.1.4 Literature Review for Objective – 4** 74](#_Toc219055697)

[**5.2 Formalizing High level Implementation Component** 75](#_Toc219055698)

[**5.2.1 To develop a predictive learner clustering model that groups students based on real time behavioral and performance data** 77](#_Toc219055699)

[**5.2.2 To develop a model that provides personalized feedback based on student clusters** 79](#_Toc219055700)

[**5.3 Data Extractions, Sample Design, Test Datasets, Training Datasets** 81](#_Toc219055701)

[**5.3.1 Data Extractions** 81](#_Toc219055702)

[**5.3.2 Sample Design** 81](#_Toc219055703)

[**5.3.3 Test Datasets and Training Datasets** 82](#_Toc219055704)

[**5.4 Non-functional Aspects** 82](#_Toc219055705)

[**5.5 Validation Methods and Measurements** 84](#_Toc219055706)

[**References** 86](#_Toc219055707)

[**Approval** 87](#_Toc219055708)

LIST OF TABLES

[Table 1:Functional Requirements of the Proposed Project 12](#_Toc219056090)

[Table 2:Scrum Model for the Proposed Project – Implementation 01 13](#_Toc219056091)

[Table 3:Scrum Model for the Proposed Project – Implementation 02 14](#_Toc219056092)

[Table 4:Scrum Model for the Proposed Project – Implementation 03 14](#_Toc219056093)

[Table 5:Integrating and Developing Tool Kit 30](#_Toc219056094)

[Table 6:PACT Analysis 32](#_Toc219056095)

[Table 7: Factors of People 34](#_Toc219056096)

[Table 8:UI DESIGN 39](#_Toc219056097)

[Table 9:Users Collection 56](#_Toc219056098)

[Table 10:Courses Collection 57](#_Toc219056099)

[Table 11:Sessions Collection 59](#_Toc219056100)

[Table 12:Zoom Participants Collection 60](#_Toc219056101)

[Table 13:Clusters Collections 61](#_Toc219056102)

[Table 14:Quiz Answers Collection 62](#_Toc219056103)

[Table 15:Question Collection 63](#_Toc219056104)

LIST OF FIGURES

[Figure 1:Proposed Solution 9](#_Toc219055709)

[Figure 2:System Architecture 18](#_Toc219055710)

[Figure 3: Sequence diagram for Register 19](#_Toc219055711)

[Figure 4:Sequence diagram for Login 20](#_Toc219055712)

[Figure 5:Sequence diagram for Session 21](#_Toc219055713)

[Figure 6:Sequence diagram for Course 22](#_Toc219055714)

[Figure 7:Sequence diagram for Question 23](#_Toc219055715)

[Figure 8:Sequence diagram for Analytics 24](#_Toc219055716)

[Figure 9:Sequence diagram for Report 25](#_Toc219055717)

[Figure 10:Class Diagram 26](#_Toc219055718)

[Figure 11:Live Learning Session State Machine 27](#_Toc219055719)

[Figure 12:User Account State Machine 28](#_Toc219055720)

[Figure 13:component diagram 28](#_Toc219055721)

[Figure 14:component diagram 29](#_Toc219055722)

[Figure 15:Registration Form 41](#_Toc219055723)

[Figure 16:Login Form 41](#_Toc219055724)

[Figure 17:Login Form 41](#_Toc219055725)

[Figure 18:Instuctor dashboard 42](#_Toc219055726)

[Figure 19:Live Session Management Screen 43](#_Toc219055727)

[Figure 20:Course List & Management Options 44](#_Toc219055728)

[Figure 21:Add / Edit Course Details Screen 45](#_Toc219055729)

[Figure 22:Question Bank Management 46](#_Toc219055730)

[Figure 23:Real-Time Question Triggering Panel 47](#_Toc219055731)

[Figure 24:Report Type Selection Screen 48](#_Toc219055732)

[Figure 25:Report Output / Download Screen 49](#_Toc219055733)

[Figure 26:Real-Time Analytics 49](#_Toc219055734)

[Figure 27:Logout Confirmation Screen 50](#_Toc219055735)

[Figure 28:ER Diagram 65](#_Toc219055736)

[Figure 29: Schema Refinement Table 68](#_Toc219055737)

[Figure 30:High level Implementation Component 75](#_Toc219055738)

[Figure 31:Predictive Clustering Model Architecture 77](#_Toc219055739)

[Figure 32:Personalized Feedback Generation Architecture 79](#_Toc219055740)

[Figure 33:Sample Dataset 81](#_Toc219055741)

# **ABSTRACT**

The System Design Specification (SDS) for the real time student engagement and learning analytics system provides an overview of the design and functionality of the proposed platform. This system is designed to support online learning by monitoring student engagement during live Zoom sessions and providing real time feedback to instructors and students.

The need for this system arises from the challenges faced in online education, where instructors find it difficult to identify student engagement levels in real time. This platform helps overcome that problem by using analytics, clustering and real time interaction tools to improve teaching and learning experiences.

The system design includes the overall system architecture, process models, tools and techniques used, data design, user interface and user experience considerations, analytics features and security aspects. The design focuses on ease of use, responsiveness, and real time interaction to support live online classes.

The key components of the system include Zoom integration, real time engagement tracking, analytics dashboards, question triggering mechanisms and report generation features. The system also supports different user roles such as students, instructors and administrators.

In addition, this document describes the hosting and deployment environment of the system, which uses cloud based services to ensure scalability, reliability and continuous availability. This SDS provides a clear understanding of how the system is designed to support effective and interactive online learning.

## **ACKNOWLEDGEMENT**

This project was carried out as a final year project at the Faculty of Applied Sciences, Rajarata University of Sri Lanka. The successful completion of this document would not have been possible without the support and guidance of several individuals.

We would like to express our sincere gratitude to our supervisor, Dr. K.A.S.H. Kulathilake, for his continuous guidance, valuable feedback and encouragement throughout the project. His support helped us to complete this work successfully.

We also thank the lecturers and staff members of the Department of Computing, Faculty of Applied Sciences, Rajarata University of Sri Lanka, for providing the knowledge and resources needed for this project.

Finally, we would like to thank all our group members for their teamwork, dedication, and cooperation in completing the project milestones on time. We hope this project will be useful for future research and development in the field of online learning and educational analytics.

# **CHAPTER 1 – INTRODUCTION**

In this chapter, we introduce the study context and provide an overview of the document structure. Section 1.1 summarizes the problem addressed in the study, while Section 1.2 outlines the objectives that guide the project. Section 1.3 lists the expected project deliverables, followed by Section 1.4, which briefly explains the overall system design approach. Section 1.5 presents the standards and guidelines considered during development, and finally, Section 1.6 describes the organization of the SDS.

The background of this study arises from the upcoming need to strengthen learner interaction and overall learning experience in online lecturing environments. As the shift toward virtual education increases, it can be understood that traditional teaching methods do not always translate effectively into online methods.

## **1.1 Problem to be addressed**

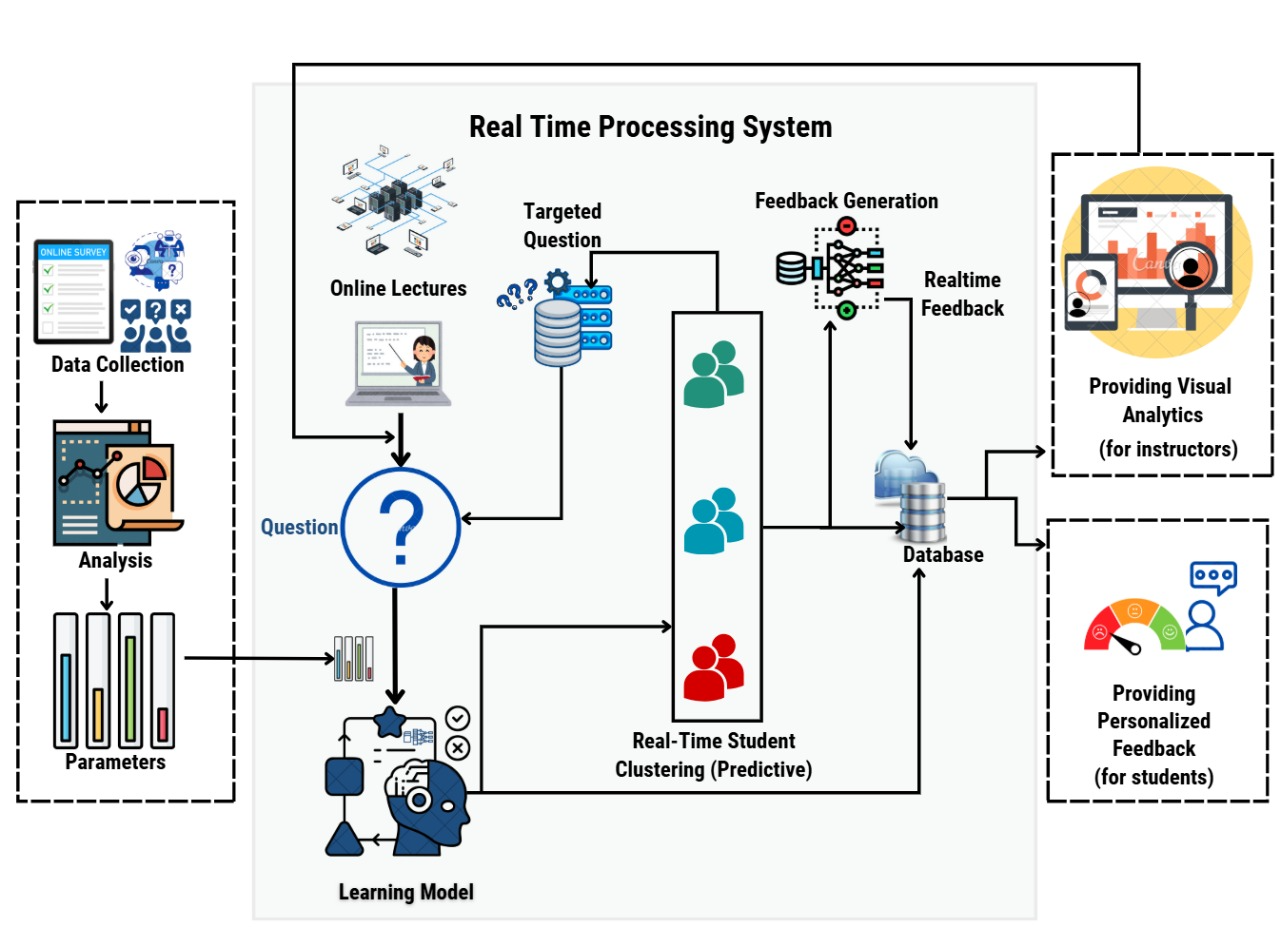


Figure :Proposed Solution

The rapid shift toward online learning has created a learning environment that many students are still continuing to adapt to. Unlike physical classrooms, online platforms often lack the natural interaction, instant feedback and social presence that support learner motivation and engagement. Students frequently experience reduced communication, limited real time interaction, and difficulty maintaining focus during virtual sessions.

Current online learning systems also provide limited visibility into student participation and behaviour. Instructors are unable to accurately observe engagement levels, track interaction patterns, or identify learners who may be struggling during a live session. The absence of face to face cues further reduces opportunities for personalized support, making it challenging to address specific individual learning needs.

The physical presence inside a classroom with a teacher and student often leads to an atmosphere that can’t be replicated through virtual methods. Without continuous interaction, many learners report decreased motivation and a sense of detachment from the learning process.

The physical mode also ensures discipline as students cannot switch off webcams. As well as their may be a technical issues. So, need some tricks to student stay on the lecture. Physical classrooms also allow for teachers to provide more personal attention to each student’s needs. However, interactive online learning methods can help improve student engagement.

As technology continues to influence modern education, it is essential for students and educators to adapt to digital environments that support necessary interaction. There is a growing need for online learning experiences that better engage learners, reflect their behavioural patterns, and provide opportunities for more personalized and interactive participation.

## **1.2 Objectives of the Project**

The following objectives are,

1. to analyze key factors influencing student engagement and learning effectiveness in video conferencing based learning environments.

2. to develop a predictive learner clustering model that groups students based on real time behavioural and performance data.

3. to develop mechanisms for delivering targeted and active interactions that address the specific needs of identified learner clusters.

4. to develop a model that provides personalized feedback based on student clusters.

5. to develop an integrated video conferencing platform that incorporates predictive clustering, active interactions, personalized feedback, and real time analytics into a unified system.

6. to evaluate the system’s performance with respect to question targeting accuracy, clustering prediction effectiveness, feedback relevance, and the impact of real time visualizations.

## **1.3 Deliverables**

Project deliverables are the intangible quantifiable goods or services we deliver as the results of the project.

1. Predictive Clustering Model

2. Question Targeting Module

3. Personalized Feedback Generation Model

4. Database of the System

5. Web application

6. Publication

a) Poster

b) Technical Report

## **1.4 System Design Approach**

The Agile methodology is a flexible and iterative approach to project management, particularly popular in software development but applicable to various industries. It prioritizes adaptability and collaboration, aiming to deliver small, incremental improvements in short iterations rather than one large product release.

Scrum is a popular framework for implementing the Agile methodology in a project, particularly in software development projects. It emphasizes collaboration, transparency and iterative progress.

To obtain the expected outcome of this study, the Scrum framework is utilized as the system design approach. The development process is organized into short, fixed length sprints that enable continuous delivery, early validation and iterative refinement of both system and research components.

### **1.4.1 The Product Backlog (PB) Items**

The product backlog items encompass the tasks necessary to achieve the objectives outlined. Functional requirements, system modules, database design, analytics components and testing activities are considered as product backlog items.

Functional Requirements of the Project

Table :Functional Requirements of the Proposed Project

|  |  |
| --- | --- |
| **PB Item Number** | **Functional Requirement** |
| 1 | User Registration & Login |
| 2 | Course & Session Management |
| 3 | Zoom API Integration |
| 4 | Real Time Interaction Tracking |
| 5 | Predictive Learner Clustering Model |
| 6 | Targeted Question Triggering Module |
| 7 | Personalized Feedback Generation model |
| 8 | Analytics Dashboard |
| 9 | Reports Generation |
| 10 | Database Design & Security |
| 11 | UI Design & Integration |
| 12 | Testing & Validation |
| 13 | Documentation & Publication |

**Implementation 01**

* **Total Points =** 30 points (4 + 4 + 7 + 7 + 4 + 4 respectively in PB items)
* **Estimated Velocity =** 7 points per sprint
* **Time sprints required =** 30/7 ≈ 4 sprints

**Implementation 02**

* **Total Points** = 22 points (3 + 3 + 5 + 4 + 2 + 2 + 3 respectively in PB items)
* **Estimated Velocity** = 6 points per sprint
* **Time sprints required** = 22/6 ≈ 4 sprints

**Implementation 03**

* **Total Points** = 28 points (6 + 6+ 6 + 5 + 5 respectively in PB items)
* **Estimated Velocity** = 7 points per sprint
* **Time sprints required** = 28/7 = 4 sprints

**Scrum Framework**

* **Framework** = Scrum
* **Sprint Length** = 5 Days
* **Total Duration** = 60 Days
* **Number of Sprints** = 12

Implementation 01

Table :Scrum Model for the Proposed Project – Implementation 01

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product Backlog Items (PB Items)** | **Sprint 01 (5 Days)** | **Sprint 02 (5 Days)** | **Sprint 03 (5 Days)** | **Sprint 04 (5 Days)** |
| Foundation & Planning | 100% | – | – | – |
| UI Design & Wireframes | 60% | 30% | 10% | – |
| 30% of Functional Requirements | 30% | 30% | 30% | 10% |
| Database Schema Design | - | 30% | 50% | 20% |
| System Architecture | 100% | – | – | – |
| Zoom API Integration & Develop model wireframes | - | 30% | 30% | 40% |

Implementation 02

Table :Scrum Model for the Proposed Project – Implementation 02

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product Backlog Items (PB Items)** | **Sprint 01 (5 Days)** | **Sprint 02 (5 Days)** | **Sprint 03 (5 Days)** | **Sprint 04 (5 Days)** |
| 70% of Functional Requirements | 50% | 50% | – | – |
| Predictive Learner Clustering Model | 20% | 40% | 40% | – |
| Targeted Question Triggering Module | - | 40% | 60% | - |
| Personalized Feedback Generation Model | - | - | 50% | 50% |
| Attendance & Interaction Tracking | – | 20% | 50% | 30% |
| Raw Engagement Data Storage | – | – | 100% | – |
| Unit Testing | - | - | 50% | 50% |

Implementation 03

Table :Scrum Model for the Proposed Project – Implementation 03

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product Backlog Items (PB Items)** | **Sprint 01 (5 Days)** | **Sprint 02 (5 Days)** | **Sprint 03 (5 Days)** | **Sprint 04 (5 Days)** |
| Intergrate the model | 40% | 40% | 20% | **-** |
| Analytics Dashboard | – | – | 80% | 20% |
| Evaluate and fine tune the application | 20% | 20% | 40% | 20% |
| Acceptance Testing | – | – | 50% | 50% |
| Model Validation | – | – | 50% | 50% |

## **1.5 Standards to be followed**

Standards for coding Coding standards are rules, techniques, and guidelines for creating cleaner, more readable, and more efficient code with minimal bugs and errors.

There are some coding standards that we generally follow for our web application.

* Naming conventions - Local variables should name using camel case lettering starting with small letter. (e. g. localData)
* Global variables names should start with a capital letter (e.g. GlobalData). We can use underscore “\_” to create multi words variable names.(e.g.: first\_name)
* Comments in function - Single line comments begin with the character "#". (e.g. #single line comment)
* Version control practices using Git and GitHub to ensure proper tracking of changes, collaborative development, and efficient management of the project code

Documentation Standards

* IEEE reference style

## **1.6 Organization of the SDS**

The rest of the SDS Document will be organized as follows,

1. Architectural Design

The architectural design section outlines the system's architecture, including its objects, communications, processes, and specialized algorithms. In the later section of this chapter, we will provide a brief overview of state machines, tools, techniques, libraries, third party tools, and the implementation environment.

2. UI Design

The third chapter will continue with a description of the PACT (People, Activities, Contexts, and Technologies) analysis, followed by an exploration of UI design considerations and approaches. In the middle part of the chapter, we will delve into design tools, techniques, and templates. Additionally, we will address input, output, and dialogue design aspects. Finally, the chapter will conclude with a discussion on the hosting and installation environment.

3.Data Management

The fourth chapter will commence with a discussion on data requirements. Following this, we will explore the design tools and techniques utilized throughout the project. After, we will delve into conceptual, logical, and physical database designs. Finally, the chapter will conclude with a discussion on schema refinement and database security mechanisms.

4. Research Design

In the fifth chapter, we begin by examining the objectives derived from our literature review and formalizing high-level implementation components. As we delve into the chapter, we focus on the extraction of data, sample design, and the creation of testing and training data sets. Towards the middle of the chapter, we address non-functional aspects, while proposing validation methods and measurements towards the end.

# **CHAPTER 2 -ARCHITECTURAL DESIGN**

This chapter discusses the architectural design of the proposed system. Section 2.1 presents the overall system architecture. Section 2.2 describes the objects and communication flows of the system using sequence diagrams and class diagrams. Section 2.3 outlines the state machines relevant to the system’s behaviour. Section 2.4 explains the deployment structure. Section 2.5 describes the enhanced processes and algorithms used in the system. Finally, Section 2.6 explains the tools, techniques, libraries, and implementation environment used throughout the development.

## **2.1 System Architecture**

For a video conferencing based online lecturing environment enhanced through learner interaction and real time analytics, a 3-tier architecture is the most suitable approach. A 3-tier architecture allows for better separation of concerns and scalability by distributing these functionalities across multiple layers.

Figure 1 system has three main components: a web application, a Backend, and server storage. The web application has a web interface. The web interface is client side and it directly connects to the user. The backend is the server side that connects between web applications and server storage. In server storage has a database. The database is storing and managing the data required by the system. There are two models connected to the database. There are live interact and Predictive Clustering Model, and Personalized Feedback Generation Model. The models play distinct roles in executing specific tasks within the system.

By adopting 3-tier architecture, the online lecturing environment can achieve better scalability, maintainability, and flexibility. The separation of concerns allows individual components to be modified, upgraded, or scaled independently, without disrupting the entire system. This architectural approach also enables the integration of additional technologies or services, for video streaming platforms to enhance the overall learning experience.

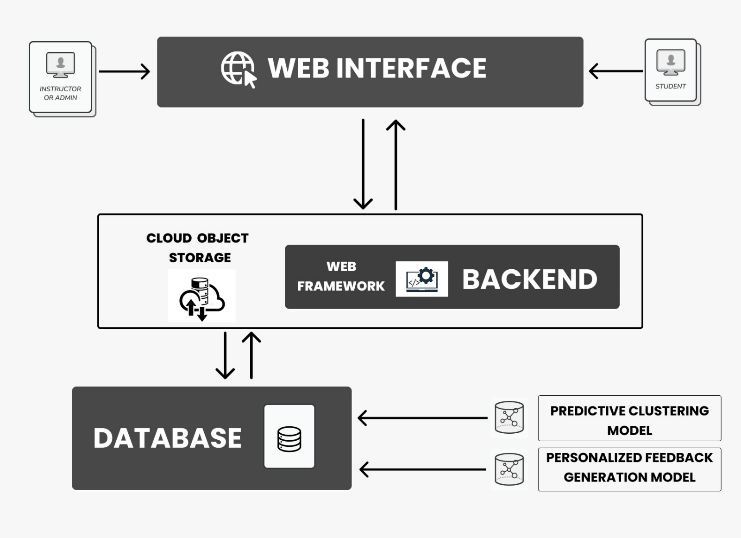


Figure :System Architecture

## **2.2 Object and communication**

### **2.2.1 Register**

The following Figure 3 sequence diagram depicts the interaction between various classes involved in the registration process. The process begins with the formLoad() function being called on the Register, Interface, activating it. The Register\_Interface then communicates with the Register\_Entity to set the registration details, which in turn interacts with the Register\_Handler. The Register\_Handler, along with the Database\_Handler, performs tasks such as loading the driver, creating a database connection, executing a query, and activating the user's account through the Verify\_Account class. At each step, the status and error messages are passed between the class. Once the necessary operations are completed, the Register\_Handler displays a registration message to the Register\_Interface, which is then activated to handle further interactions or display the appropriate message to the user.

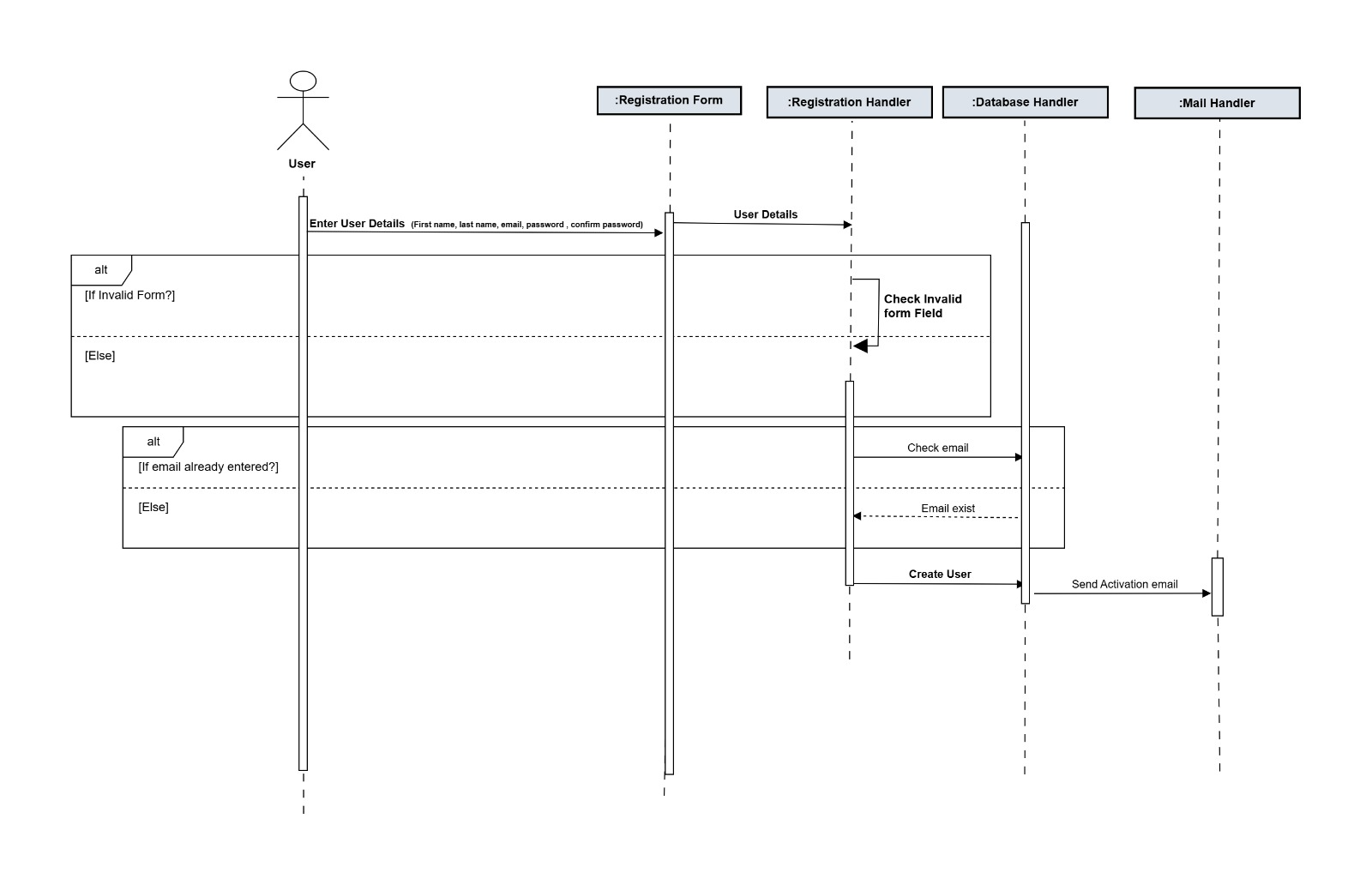


Figure : Sequence diagram for Register

### **2.2.2 Login**

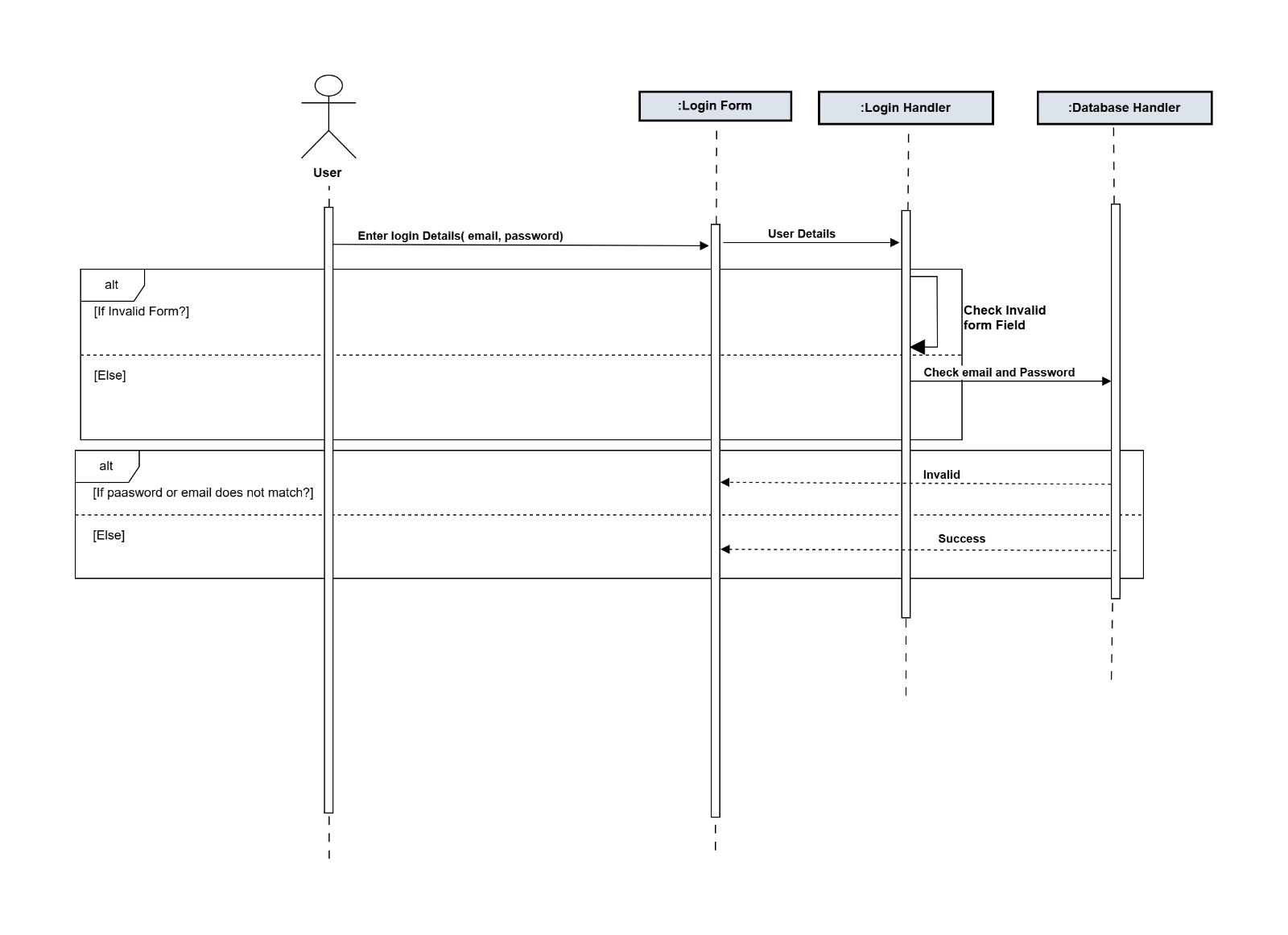
The following Figure 4 sequence diagram illustrates the login process, including the steps for handling forgotten passwords and password resets. Initially, the loginField() function is called on the Login\_Interface, activating it. The Login\_Interface communicates with the Login\_Handler to provide login details. The Login\_Handler interacts with the Database\_Handler to perform tasks such as loading the driver, creating a database connection, and validating the login details. Status and error messages are exchanged between the classes during these operations. If the user has forgotten their password, an alternative path is taken. The Login\_Interface activates the Login\_Handler to request a password hint. The Login\_Handler communicates with the Database\_Handler to validate the hint and sends an appropriate message back to the Login\_Interface. Alternatively, if the user wants to reset their password, the Login\_Handler validates the login details and activates the Reset class. The Reset class interacts with the Database\_Handler to update the password and sends status messages back to the Login\_Interface. At each stage, the Login\_Interface is activated to display relevant messages or handle user interactions. 

Figure :Sequence diagram for Login

### **2.2.3 Session**

The following Figure 5 sequence diagram illustrates the process involved in managing a live session. The interaction begins when the Instructor initiates the startSession() function on the Session\_Interface, activating it. The Session\_Interface communicates with the Session\_Handler, which is responsible for coordinating all operations related to the live session.The Session\_Handler first interacts with the ZoomAPI\_Handler to establish a connection with the Zoom Conferencing API and request the session link. Once the ZoomAPI\_Handler returns the session link, the Session\_Handler distributes it to students through the Session\_Interface.

After receiving the link, students join the session, and the Attendance\_Tracker is activated to record join times and maintain attendance and participation tracking throughout the session. As the live session progresses, interaction events such as chats, reactions, and quiz responses are streamed to the Analytics\_Engine, which processes these events continuously. The Analytics\_Engine sends the processed engagement data to the PredictiveClusteringModel, which updates learner clusters in real time. The updated clusters are then forwarded to the TargetedQuestionModel, which determines and dispatches appropriate questions for each learner group. Simultaneously, the cluster information is passed to the FeedbackModel, which generates real-time personalized feedback for students. Throughout the session, the Database\_Handler is used to store all session-related data, including attendance records, participation metrics, cluster updates, questions delivered, and feedback generated. The Session\_Handler periodically triggers updates to ensure that all information remains synchronized during the ongoing lecture. Once all interactions are completed, the Session\_Interface is activated again to reflect updated analytics and session status, allowing both instructors and students to continue the session with real-time insights and updates provided by the system.

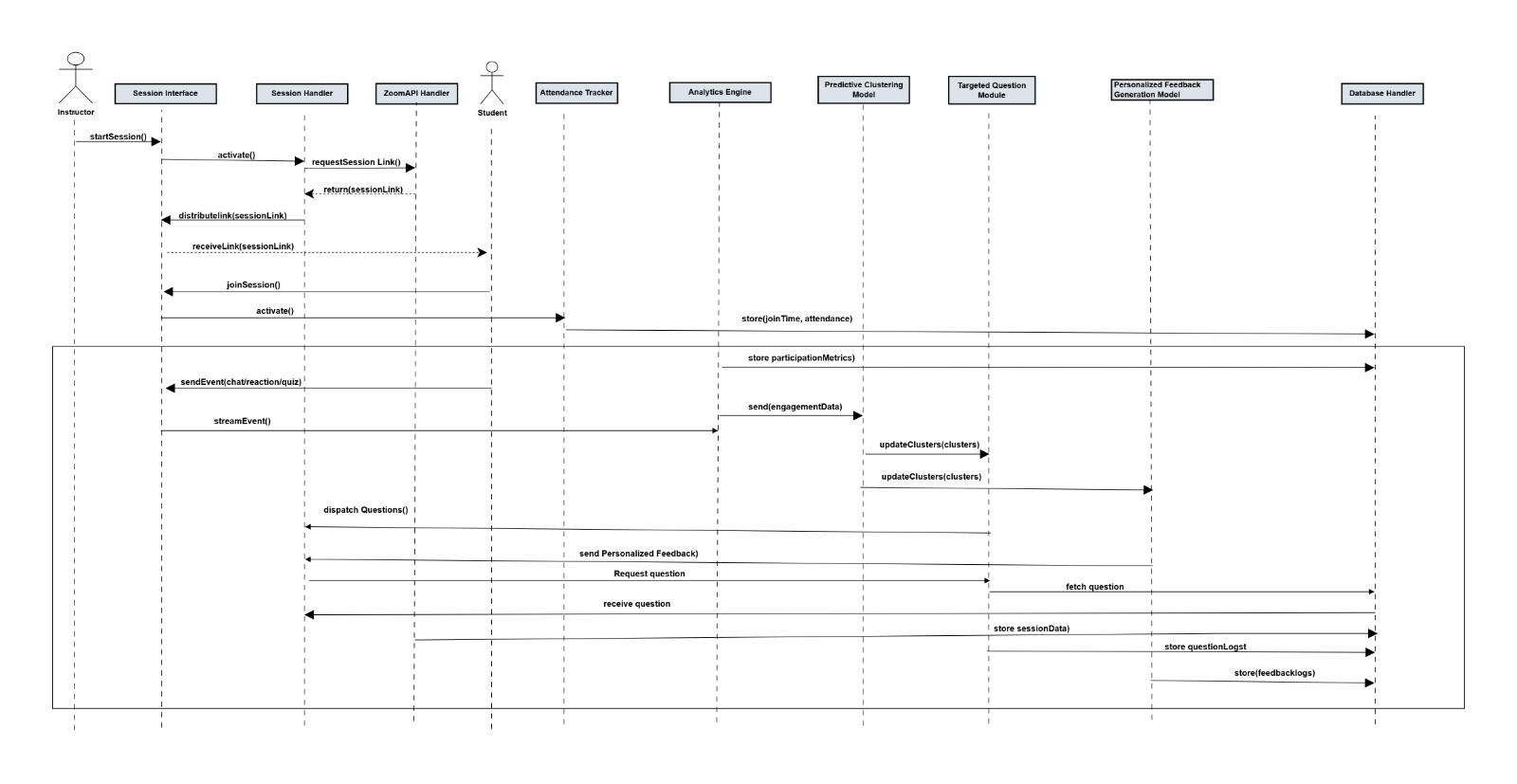


Figure :Sequence diagram for Session

### **2.2.4 Course**

The following Figure 6 sequence diagram illustrates the interactions between the classes involved in managing course information. The process begins when the Instructor or System Admin accesses the course management interface, triggering the loadCourseManagement() function on the Course\_Interface, which becomes active.

The Course\_Interface forwards the entered course details to the Course\_Handler. The Course\_Handler processes the course metadata and prepares it for storage. It then communicates with the Database\_Handler, which handles inserting or updating the course information in the database.For enrollment management, the Course\_Interface invokes functions. These actions are passed to the Enrollment\_Handler, which manages the corresponding enrollment records. The Enrollment\_Handler also interacts with the Database\_Handler to ensure that enrollment updates are stored accurately.

Once all updates course details and enrollment changes are completed, the Course\_Handler notifies the Course\_Interface, which refreshes the display to show the updated course and enrollment information. Throughout this interaction flow, each handler and interface cooperates to maintain consistent and structured course data within the system.

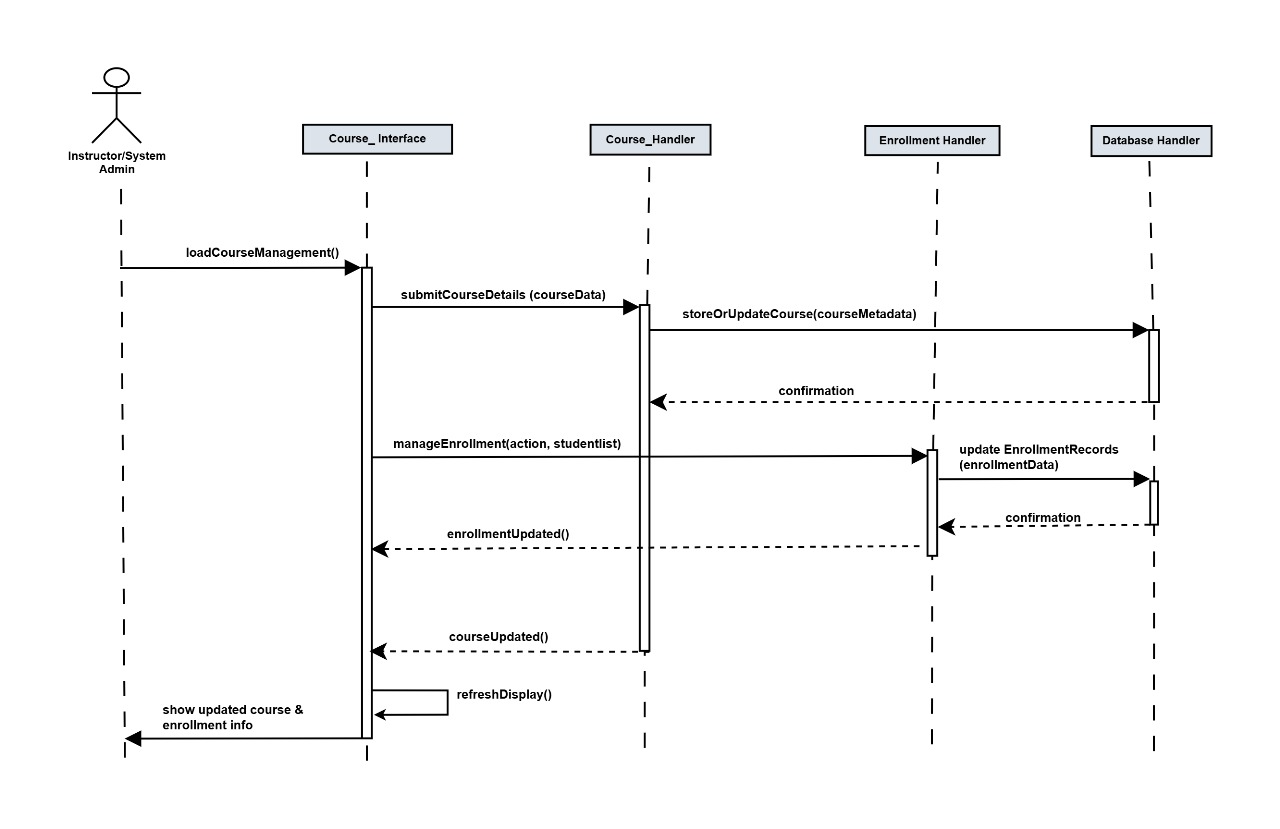


Figure :Sequence diagram for Course

### **2.2.5 Question**

The following Figure 7 sequence diagram illustrates the interaction between the classes involved in managing questions within the system. The process begins when the Instructor or System Admin accesses the question management interface, triggering the loadQuestionBank() function on the Question\_Interface, which becomes active.

The Question\_Interface receives the question details and forwards them to the Question\_Handler. The Question\_Handler processes the question information and communicates with the Database\_Handler, which stores the question in the database and updates the associated question bank.

During a live session, when a question request is needed, the Session\_Handler invokes a request through the Question\_Interface to fetch appropriate questions. The Question\_Handler retrieves cluster-wise, generic, or random questions by interacting with the Database\_Handler, depending on the needs of the learning model or instructor actions.

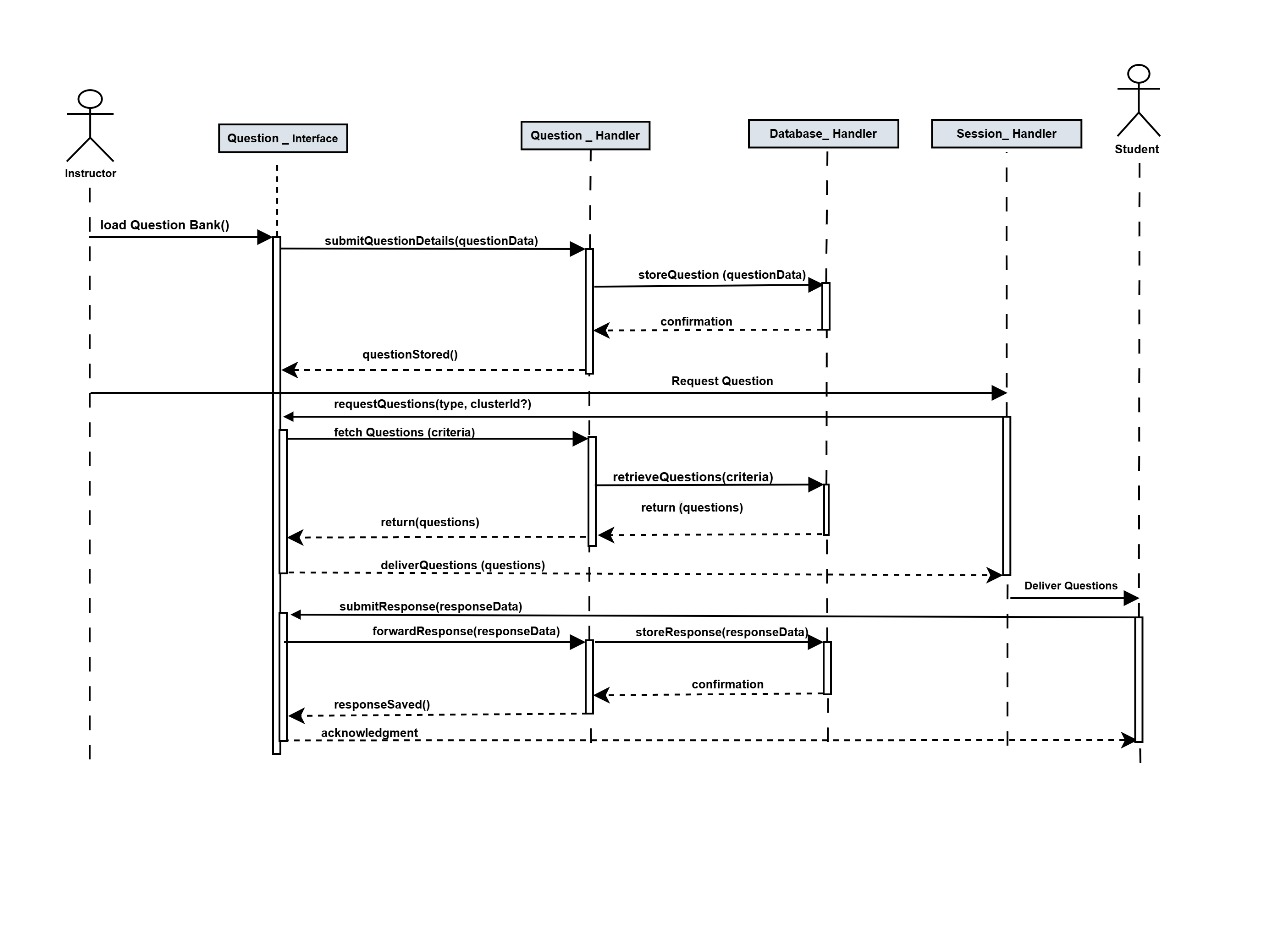
As students respond to the delivered questions, the responses are sent back through the Question\_Interface to the Question\_Handler. The Question\_Handler then stores the collected responses using the Database\_Handler, ensuring that all question-and-answer data is persisted for analytics, model training, and reporting purposes.

Figure :Sequence diagram for Question

### **2.2.6 Analytics**

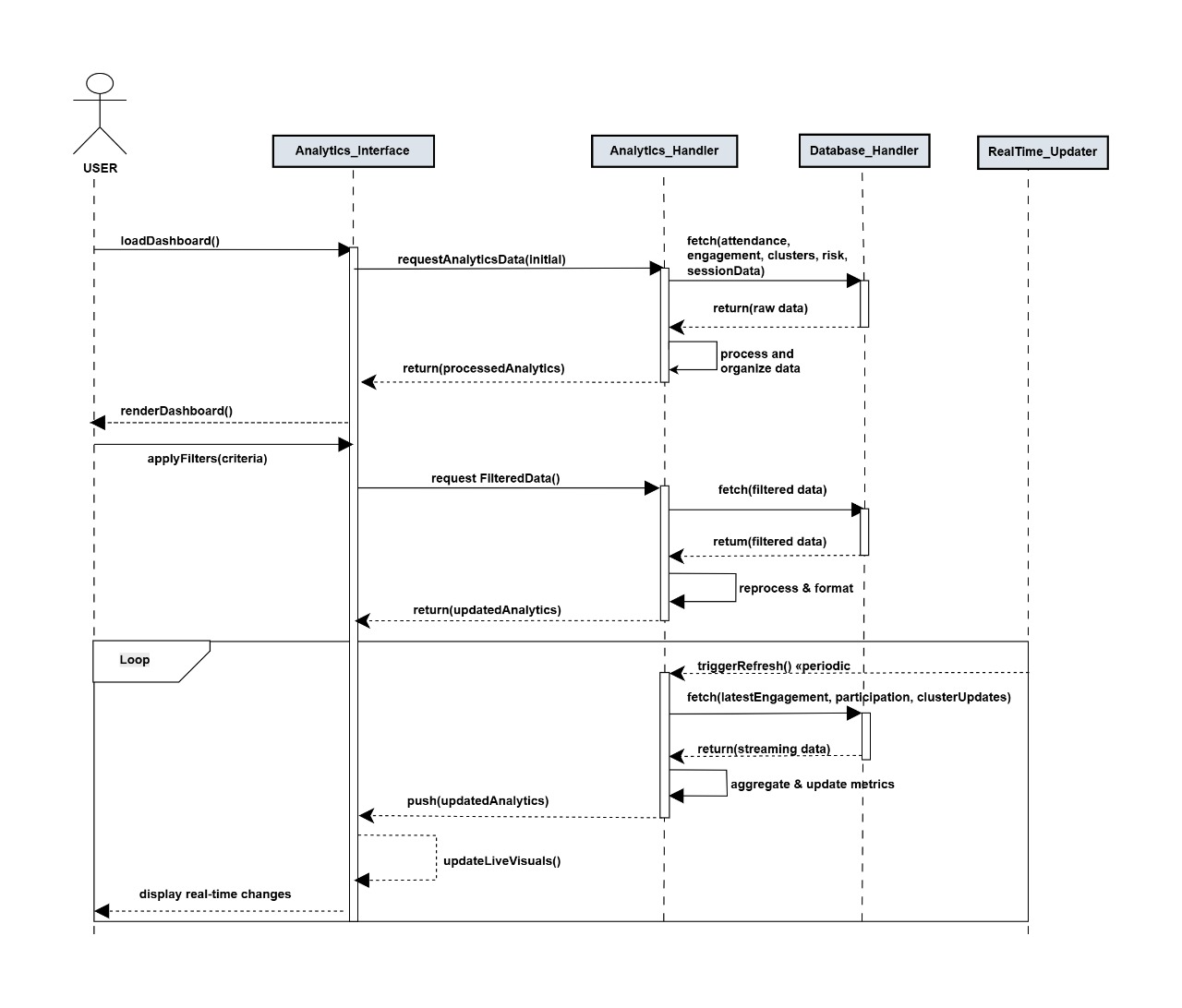
The following Figure 8 sequence diagram illustrates the interaction between the classes involved in displaying analytics for ongoing sessions. The process begins when the User opens the analytics dashboard, activating the loadDashboard() function on the Analytics\_Interface. The Analytics\_Interface sends a request to the Analytics\_Handler, which coordinates the retrieval of necessary data. The Analytics\_Handler communicates with the Database\_Handler to fetch attendance records, engagement metrics, cluster counts, risk indicators, and any session specific data required for generating the analytics view. Once the data is retrieved, the Analytics\_Handler processes and organizes the information. The processed analytics are then sent back to the Analytics\_Interface, where the user can apply additional filters such as course selection, session selection, or time range. These filter requests are forwarded back to the Analytics\_Handler, which requeries the database and returns updated results. During an active session, the RealTime\_Updater periodically triggers the Analytics\_Handler to refresh engagement metrics, participation levels, and cluster updates. The Analytics\_Interface updates the dashboard display continuously, ensuring that instructors and students view the most recent information. 

Figure :Sequence diagram for Analytics

### **2.2.7 Report**

The following Figure 9 sequence diagram illustrates the interaction between the classes involved in generating reports. The process begins when the Instructor or System Admin accesses the reporting interface, triggering the loadReportOptions() function on the Report\_Interface, which becomes active.

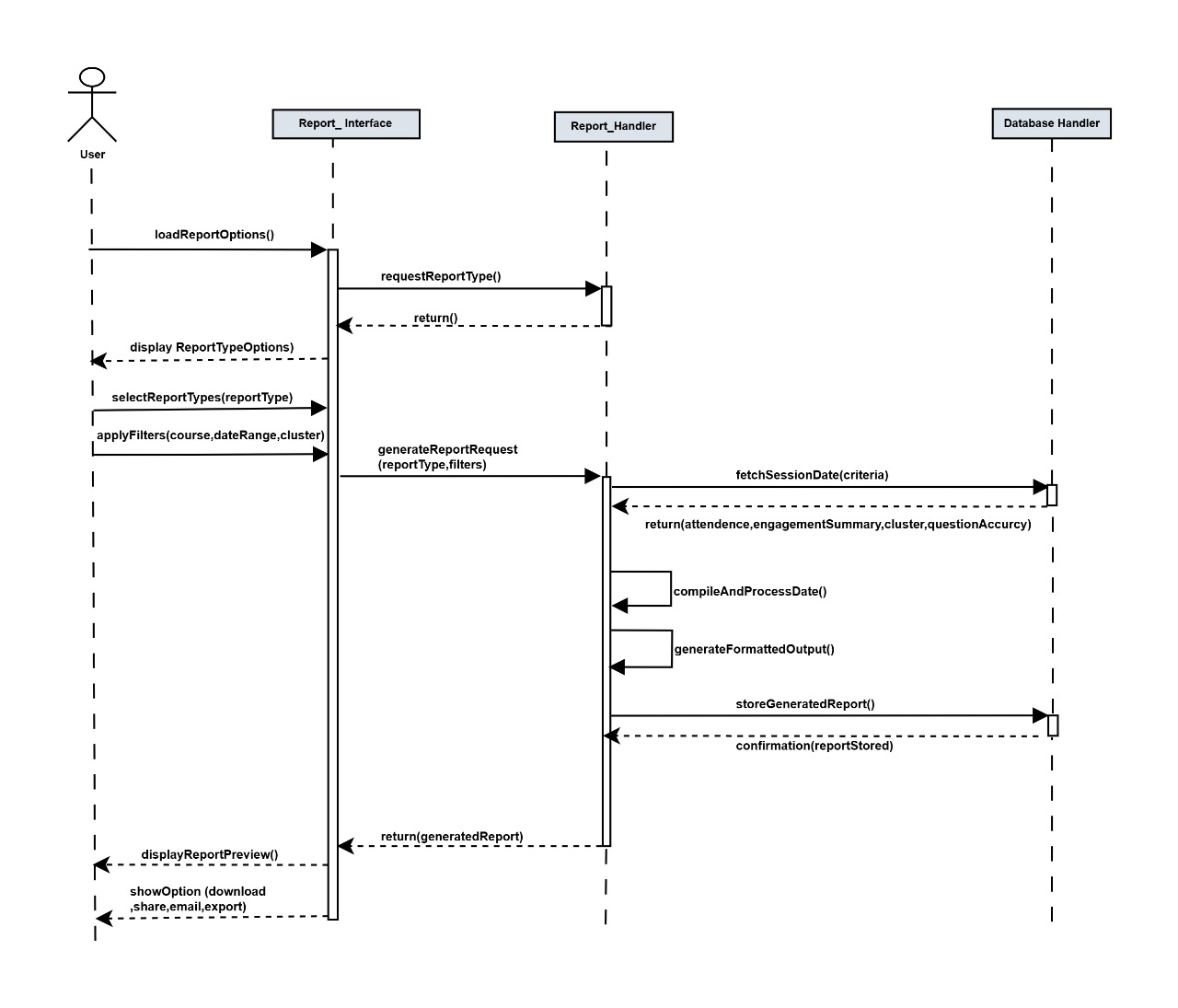
The user selects the desired report type. These selections are forwarded from the Report\_Interface to the Report\_Handler, which manages all report related operations. The user then applies filters, and these filter parameters are also passed to the Report\_Handler. The Report\_Handler communicates with the Database\_Handler to retrieve all relevant session data that meets the selected criteria. After retrieving the required information, the Report\_Handler compiles the dataset, processes the results, and generates the report in the desired format. The generated report is then stored back into the database for future access. Finally, the Report\_Handler returns the compiled report to the Report\_Interface, which provides the user with options to download or share the file. Once the report is displayed, the Report\_Interface remains active to allow the user to generate additional reports or modify filter selections.

Figure :Sequence diagram for Report

### **2.2.8 Class Diagram of the System**

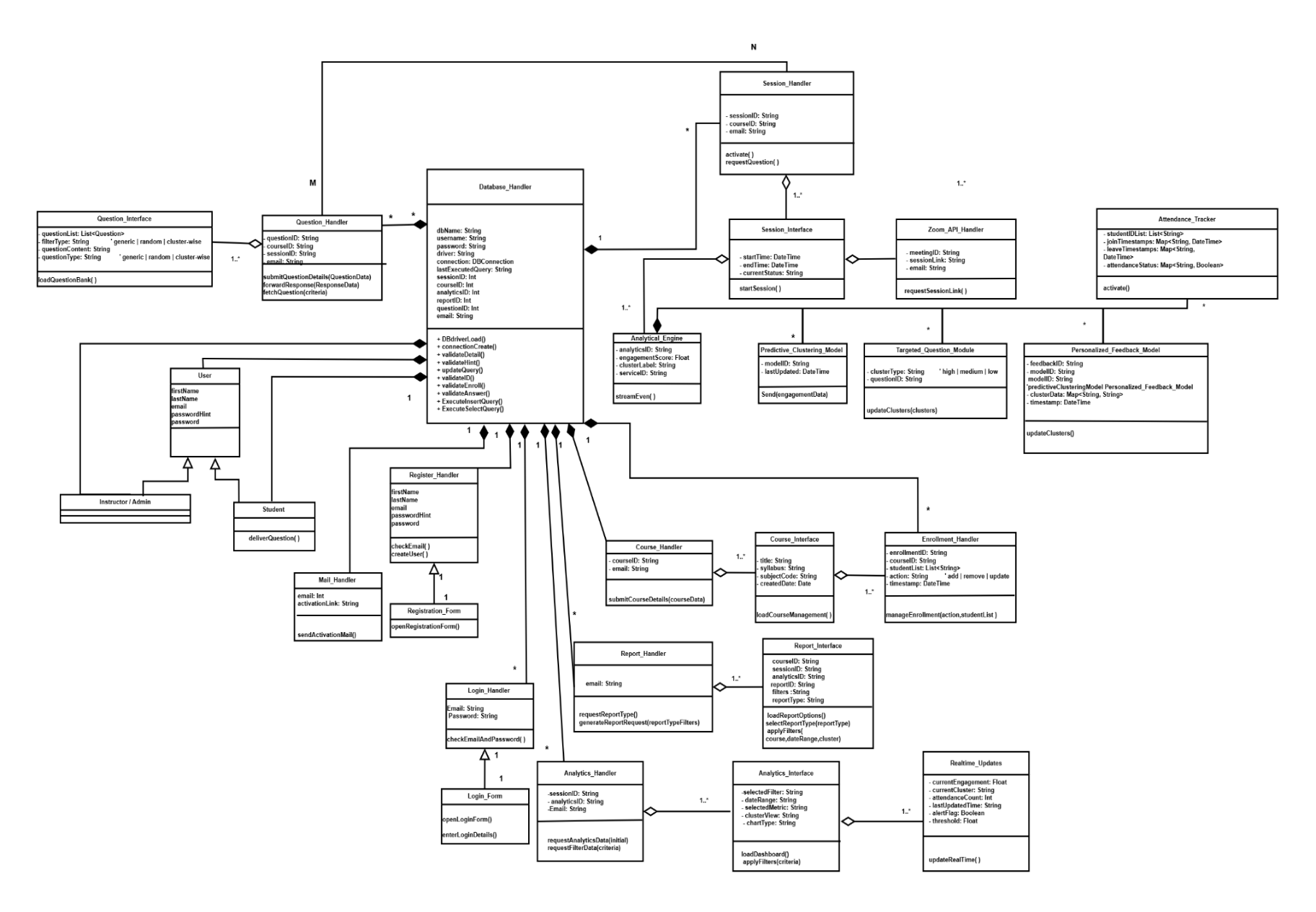


Figure :Class Diagram

## **2.3 State Machines**

In the class interactions described earlier, several objects transition through different states depending on system events and user actions. These state transitions represent how objects behave dynamically during runtime as they interact with other components. The following state machine diagrams illustrate two major processes of the system the live learning session flow and the user account lifecycle.

The following Figure 11 state machine represents the process of a live session within the system. After the session is created, the session state is initialized as Start. When the instructor hosts the session and the connection with the video conferencing platform is established, the session state transitions to Active.

If the instructor does not host the session or the session fails to start, the state transitions to Inactive. At the end of the session, once the instructor ends the meeting, the state returns to Inactive, indicating the session has concluded.

If the session becomes temporarily disconnected due to network or platform interruption and is reconnected, the state transitions again from Inactive back to Active, allowing the live session to continue with real-time analytics and tracking restored.

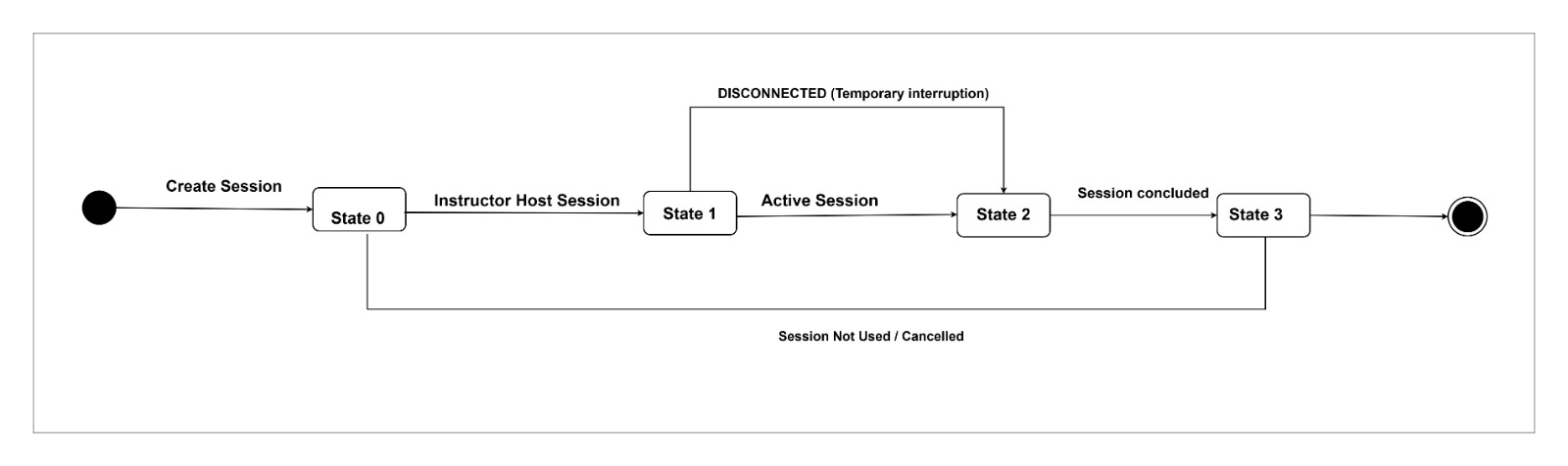


Figure :Live Learning Session State Machine

The following Figure 12 state machine represents the lifecycle of a user account within the system. When a user submits the registration form, the account state is initialized as Not Active, indicating that the user has not yet completed the activation process. When the system sends an activation link to the registered email, the account state transitions to Pending. After the user interacts with the activation link, the system verifies the activation and updates the account state to Active, granting full access to system functionalities.

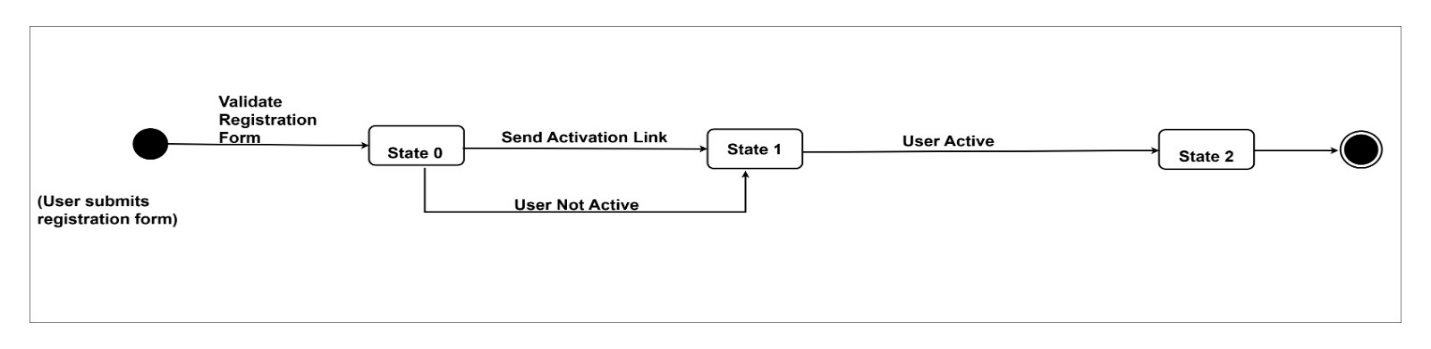
If the user does not activate the account within the expected timeframe, or chooses to deactivate the account later, the state transitions back to Not Active, representing an account that is inactive and restricted from accessing system operations

Figure :User Account State Machine

## **2.4 Deployment of the System**

The following Figure 13 component diagram depicts the system architecture in more detail. The component diagram illustrates the individual components that constitute the system and highlights how they interact with one another. In this system, the architecture is mainly divided into four major components. The Predictive Clustering Model and Personalized Feedback model components represent the machine learning models responsible for processing engagement data, clustering students, and generating interaction outputs. The Web Application component represents the front end platform where instructors and students interact with the system and also we are retrieving data real time data about the meeting details from the zoom API.

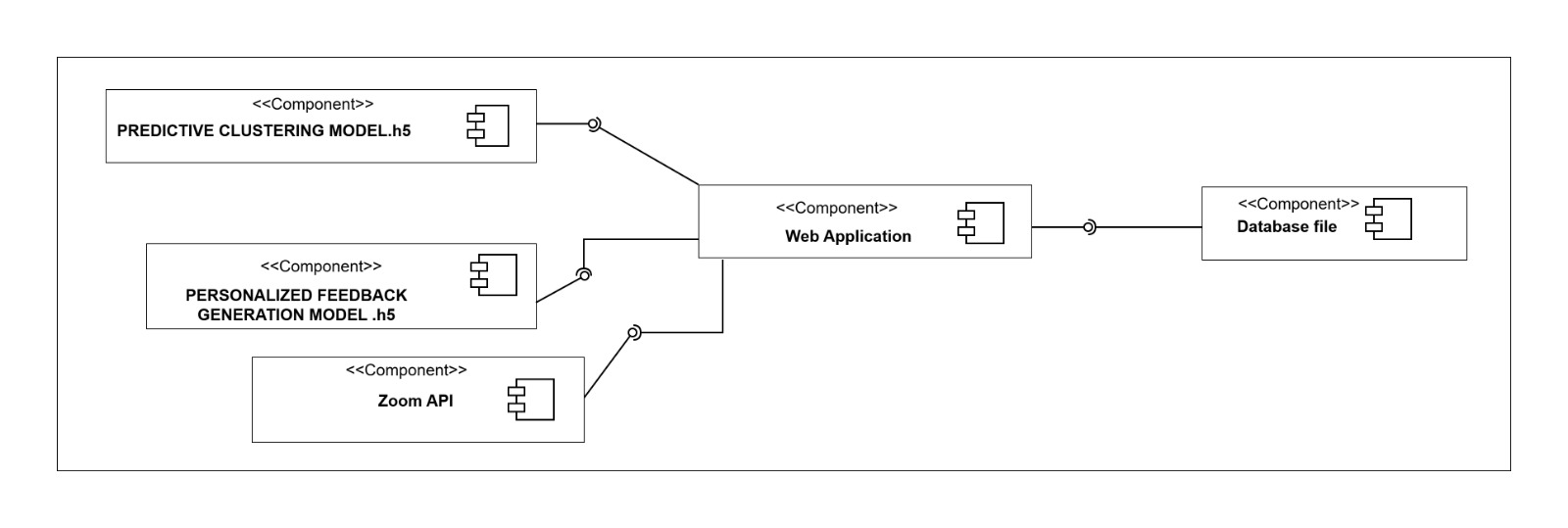
Finally, the Database component is responsible for managing and storing all system data, including session information, user details, analytics records, and model outputs. 

Figure :component diagram

The following Figure 14 deployment diagram is derived from the structure shown in the component diagram. The deployment diagram illustrates how the system is physically deployed within the hardware and network environment.

It shows the distribution of components across servers, including the hosting of the web application on a web server, the execution of backend logic and model processing on an application server, and the storage of data on a dedicated database server. The deployment diagram also reflects the integration with the Zoom Conferencing API and the communication channels used for real-time processing. Together, these diagrams provide a clear representation of how the system operates on its underlying infrastructure.

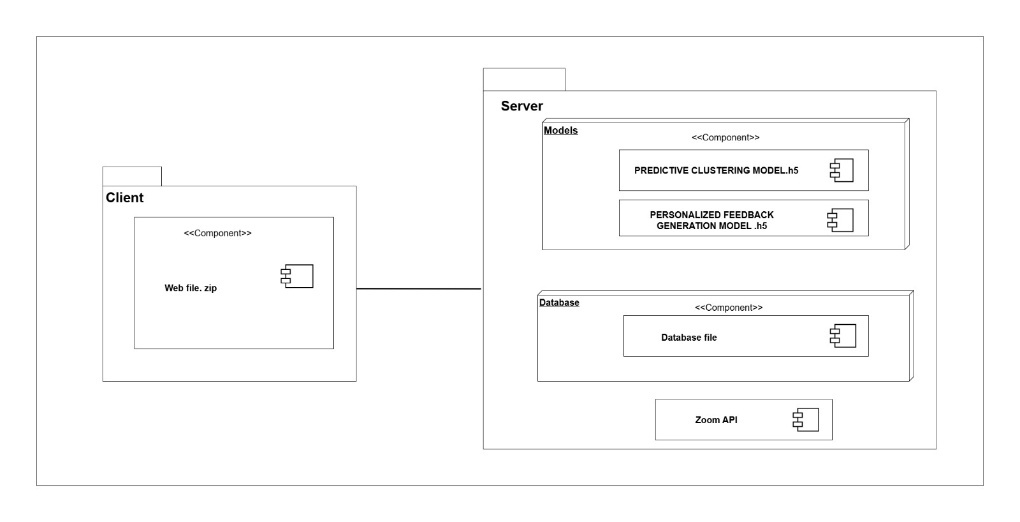


Figure :component diagram

## **2.5 Processes and Special Algorithms**

### **2.5.1 WebRTC-Based Network Data Retrieval Algorithm**

The system uses **WebRTC** to retrieve real time network quality data during live video conferencing sessions. WebRTC provides access to connection statistics such as **latency, jitter, packet loss, round trip time, and bitrate**, which reflect the actual communication quality experienced by learners.

These network metrics are collected continuously during active sessions and combined with learner interaction and performance data. This integration helps the system distinguish between low engagement caused by behavioural factors and reduced participation resulting from network issues.

The processed WebRTC network features are included in the engagement feature set used by the predictive learner clustering model. All collected metrics are stored with session timestamps to support real time analytics and post session reporting. Stay updated with new technologies, pedagogical approaches, and research findings to continually improve the online lecturing experience for learners.

## **2.6 Integrating and Developing Tool Kit**

Table :Integrating and Developing Tool Kit

|  |  |  |
| --- | --- | --- |
|  | | **Purpose** |
| **Tools & Techniques** | **Draw.io** | Draw.io is used to create system diagrams such as flowcharts, UML diagrams, component diagrams, deployment diagrams, and network diagrams. It helps visualize the system architecture clearly. |
| **Figma** | Figma is used for designing UI mockups, wireframes, dashboards, instructor and student interfaces, and interactive prototypes. Its real-time collaboration features support teamwork during the design phase. |
| **Postman** | Postman is used for testing and validating APIs, including backend endpoints and Zoom webhooks. It supports various request formats and provides tools for debugging, variable handling, and authentication testing. |
| **MongoDB Query Language (MQL)** | MQL is used for querying, inserting, updating, and managing data in MongoDB. It helps handle session logs, analytics data, user activity details, cluster histories, and targeted question records efficiently. |
| **Implementation Environment** | **Google Colab** | Google Colab is used for executing Python notebooks, particularly for training and testing machine learning models. It provides powerful cloud-based GPU/TPU resources and seamless integration with TensorFlow, NumPy, and OpenCV. |
| **Web Hosting / Cloud Platform** | Cloud hosting platforms are used for deploying backend services, machine learning models, and databases. They ensure real-time communication, scalability, and continuous availability of APIs. |
| **Visual Studio Code** | VS Code is used for backend and frontend development, implementing APIs, integrating Zoom services, handling real-time data processing, and managing the full-stack development workflow. Its extensions support Python, JavaScript, HTML, and MongoDB. |

# **CHAPTER 3 - UI DESIGN**

User Interface (UI) design focuses on how a digital system looks and how users interact with it. This includes websites, mobile applications and software platforms. The main goal of UI design is to create interfaces that are attractive, easy to use and simple to understand. A good UI helps users complete their tasks smoothly without confusion.

UI design is a collaborative process. It involves understanding user needs, studying user behavior, creating design prototypes, collecting feedback and improving the design step by step. The final objective is to build interfaces that not only look good but also guide users clearly and present information and features in an effective way.

## **3.1 PACT Analysis**

PACT analysis is used in UI design to understand the factors that affect user interaction. PACT stands for People, Activities, Context and Technologies. These four elements help designers create interfaces that match real user needs and usage environments.

Table :PACT Analysis

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Screen** | **Demo-graphics** | **English** | **Skill Level** | **Interaction Level** | **Fill Form** | **Select Option** | **Receive Msg** | **Readable** | **Physical Context** | **Social Context** | **Input** | **Output** | **Navigation** |
| **Login Screen** | Y | Y | Y | Y | Y | Y | Y | Y | Y | N | Y | Y | Y |
| **Register Screen** | Y | Y | Y | Y | Y | Y | Y | Y | Y | N | Y | N | Y |
| **Account Activation Screen** | N | Y | Y | Y | N | N | Y | Y | Y | N | Y | Y | Y |
| **Forgot Password Screen** | Y | Y | Y | Y | Y | Y | Y | Y | Y | N | Y | Y | Y |
| **Dashboard (Instructor/Student)** | Y | Y | Y | Y | N | Y | Y | Y | Y | Y | Y | Y | Y |
| **Manage Live Sessions Screen** | Y | Y | Y | Y | N | Y | Y | Y | Y | Y | Y | Y | Y |
| **Live Session Screen (Zoom Integrated)** | Y | Y | Y | Y | N | Y | Y | Y | Y | Y | Y | Y | Y |
| **Session Summary Screen** | Y | Y | Y | Y | N | Y | Y | Y | Y | N | N | Y | Y |
| **Course Management Screen** | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y | Y |
| **Question Management Screen** | Y | Y | Y | Y | Y | Y | Y | Y | Y | N | Y | Y | Y |
| **Real-Time Question Trigger Screen** | Y | Y | Y | Y | N | Y | Y | Y | Y | Y | Y | Y | Y |
| **Analytics Dashboard Screen** | Y | Y | Y | Y | N | Y | Y | Y | Y | Y | N | Y | Y |
| **Report Generation Screen** | Y | Y | Y | Y | Y | Y | Y | Y | Y | N | Y | Y | Y |
| **Report Preview / Download Screen** | Y | Y | Y | Y | N | Y | Y | Y | Y | N | N | Y | Y |
| **Enroll Session Screen** | Y | Y | Y | Y | Y | Y | Y | Y | Y | N | Y | Y | Y |
| **Invalid Session Screen** | Y | Y | Y | Y | N | N | Y | Y | Y | N | N | Y | Y |
| **Logout Confirmation Screen** | N | Y | Y | Y | N | Y | Y | Y | Y | N | N | Y | Y |

### **3.1.1 People**

This aspect focuses on understanding who the users are, how they behave and what they need when using the interface. It looks at factors such as user background, goals, preferences and skill levels. By clearly understanding these user needs, designers can make better decisions about the layout, features and overall design, resulting in an interface that is easier to use and more effective.

Table : Factors of People

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Actor | Demographics | | Language | Skill Level | Interaction Level |
| Education | Gender |
| Student | Intermediate | Both | English | Good | Normal |
| Instructor/Admin | Advanced | Both | English | Better | High |

### **3.1.2 Activities**

Activities refer to the tasks and actions that users perform while using the interface. When designing the interface, it is important to understand the order of these actions and the goals users want to achieve during a live learning session. By understanding when and how these activities take place, designers can create interfaces that allow users to complete tasks easily, quickly and effectively.

The main activity factors considered in this system are:

**● Fill Out the Form**

Users need to enter information into different input fields. In this system, students, instructors and administrators provide details such as login information, registration data, course details, session information and question content. This activity is mainly related to registration, login, session creation and question management screens.

**● Select the Options**

Users often need to choose from available options. In the learning analytics platform, this includes selecting course types, session IDs, cluster categories, question types (such as generic, random, cluster-based) and analytics filters like date range, session or engagement level. This activity applies to screens where users configure settings, make selections or navigate between features.

**● Receiving the Message**

Users receive system generated messages and notifications. These include confirmations for successful login, session joining notifications, question delivery, real-time feedback and system alerts during live sessions. For example, students may receive targeted questions or feedback, while instructors may receive alerts about students with low engagement.

**● Readability**

Readability ensures that all information displayed on the interface is easy to understand. Since the system shows live analytics, engagement levels, cluster information, attendance records and question prompts, clear presentation is essential for both students and instructors. This factor is important across all interfaces to ensure users can easily read and understand dashboards, charts, feedback messages and instructional content.

### **3.1.3 Contexts**

Context refers to the environment and conditions in which users interact with the interface. Understanding this context is important because it helps designers create interfaces that work well during real time learning sessions. Context affects how users access the system, how they understand information and how they interact with features.

Context can be divided into physical, social and organizational aspects, all of which influence user experience.

**● Physical Environment**

The physical environment describes where users are when they use the system. Since the platform supports real time Zoom integration, users may access it from their homes, classrooms, university labs, or public places. To participate smoothly, users need a stable internet connection, properly working audio and video devices and a reasonably quiet environment.

**● Social Context**

Social context focuses on how users interact with each other during live sessions. In this system, students and instructors communicate through Zoom chat, reactions and collaboration tools. Factors such as cultural background, communication style and classroom behavior influence how users participate. Understanding these social interactions helps designers create interfaces that encourage participation, support collaboration and maintain a positive learning environment.

### **3.1.4 Technologies**

Technologies refer to the tools, devices, platforms and software used in the system. This includes screen responsiveness, compatibility with different devices, interaction methods and integration with external services like Zoom. By understanding these technical aspects, designers can ensure the interface works smoothly across different environments.

**a) Real Time Interaction and Engagement Tools**

The system includes real time features such as live polls, chat responses, reaction tracking and targeted question delivery during Zoom sessions. These tools encourage active participation and provide instant feedback to students during lectures.

**b) Analytics and Visualization Components**

The platform uses analytics dashboards to help instructors monitor student engagement, attendance, performance and cluster changes in real time. Charts, graphs and timeline views make it easier to identify learning patterns and support data driven teaching decisions.

**c) Clear and Intuitive Navigation**

The interface is designed with clear menus, simple labels and logical grouping of features. This allows users to easily find sessions, questions, reports and analytics dashboards, improving overall usability for both students and instructors.

**d) Simplified Input and Output**

The system reduces complexity by providing clear instructions and minimizing unnecessary steps for tasks such as joining sessions, submitting responses and generating reports. Common UI components include

* Text Boxes
* Drop down Menus
* Radio Buttons
* Checkboxes
* Buttons
* Real time Notifications and Alerts

These elements help users interact smoothly with the system on different devices.

## **3.2 UI design consideration and approach**

A consistent color scheme, layout structure, typography and common user interface components were applied across all interface templates. This consistency allows users to quickly become familiar with the system and improves usability by reducing the learning effort required to navigate between different screens.

The interface templates were designed using a user centric approach that prioritizes simplicity, clarity and ease of use. This ensures that users with different technical skill levels, including students and instructors, can interact with the system comfortably and complete tasks without confusion.

Clear visual hierarchy, appropriate spacing and readable fonts were used to ensure that important information such as session details, analytics results, and feedback messages can be easily understood, especially during time sensitive live learning sessions.

Recoverability was incorporated by providing clear error messages, validation feedback, and corrective guidance. These features help users identify mistakes, correct them easily and continue their tasks without frustration or disruption to the learning process.

User guidance was embedded into the interface templates through clear labels, instructions and prompts, particularly in forms and verification processes. This supports first time users and helps reduce input errors during system interaction.

The interface templates were designed according to user roles, ensuring that students, instructors and administrators are presented only with features relevant to their responsibilities. This reduces interface complexity and improves overall usability.

The interface templates follow a responsive design approach to ensure proper display and interaction across different devices and screen sizes. This allows users to access the system effectively from desktops, laptops and other supported devices commonly used in online learning environments.

The interfaces were designed to support real time learning activities by presenting information clearly, minimizing distractions and providing timely visual feedback. This helps users focus on live sessions, analytics and interactions during online lectures.

## **3.3 Design tools and Techniques**

### **3.3.1 Design Tools**

● Figma

Figma is used as the main UI design tool. Its cloud based and collaborative features make it ideal for team based design work. Figma is easy to use, flexible and widely used in UI/UX design across many industries.

### **3.3.2 Techniques**

● Visual Appeal

The interface is designed to be visually attractive and engaging. Appropriate color schemes, visuals and media elements are used to improve the learning experience.

● Wireframing

Low fidelity wireframes are created to plan the basic layout and structure of the system before development starts. This helps visualize the placement of content and UI elements early in the process.

● Information Architecture

Content and navigation are organized clearly and logically so users can find information easily. Good information architecture improves usability and navigation.

● Prototyping

Interactive prototypes are created to simulate user interactions. This helps test design ideas and collect feedback before the final system is developed.

## 

Table :UI DESIGN

|  |  |  |
| --- | --- | --- |
| **USE CASE** | **INTERFACE NUMBER** | **INTERFACE NAME** |
| Register | Figure 14 | Registration Form |
| Login | Figure 15 | Login – Student / Instructor |
|  | Figure 16 | Student Dashboard |
|  | Figure 17 | Instructor Dashboard |
| Manage Live Sessions | Figure 18 | Live Session Management Screen |
| Course Management | Figure 19 | Course List & Management Options |
|  | Figure 20 | Add / Edit Course Details Screen |
| Question Management | Figure 21 | Question Bank Management |
|  | Figure 22 | Real Time Question Triggering Panel |
| Report Generation | Figure 23 | Report Type Selection Screen |
|  | Figure 24 | Report Output / Download Screen |
| Display Analytics | Figure 25 | Real Time Analytics |
| Logout | Figure 26 | Logout Confirmation Screen |

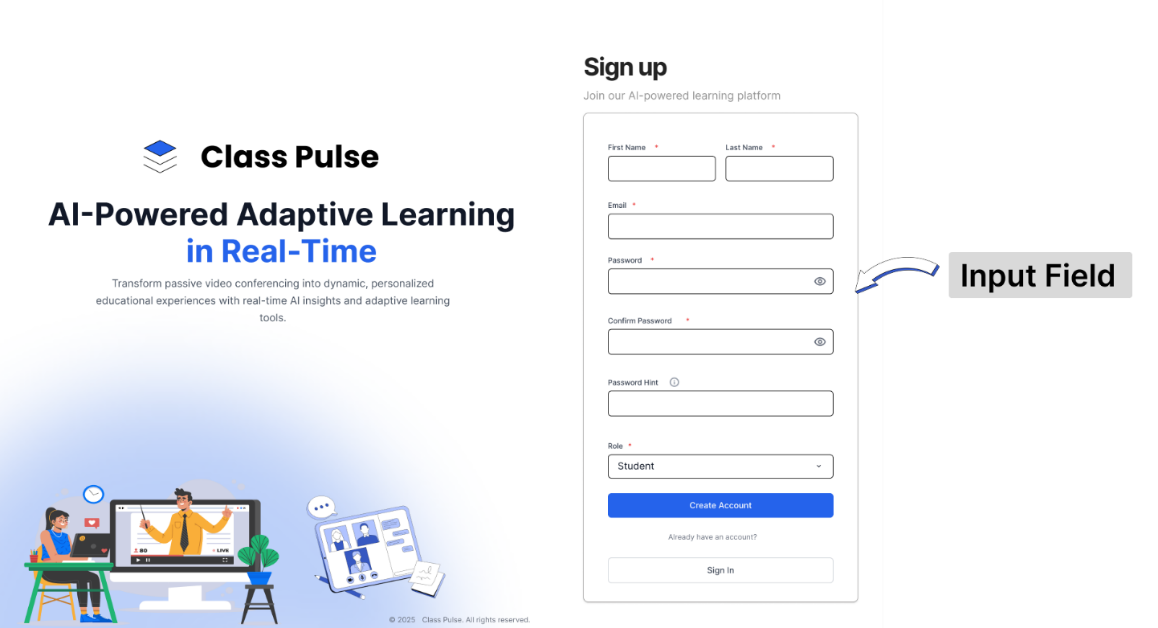


Figure :Registration Form

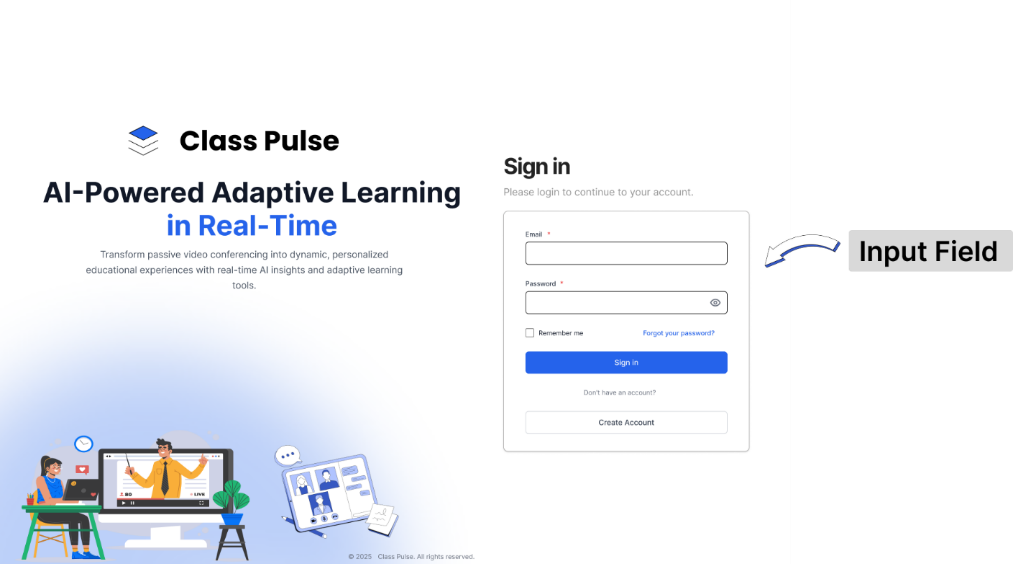
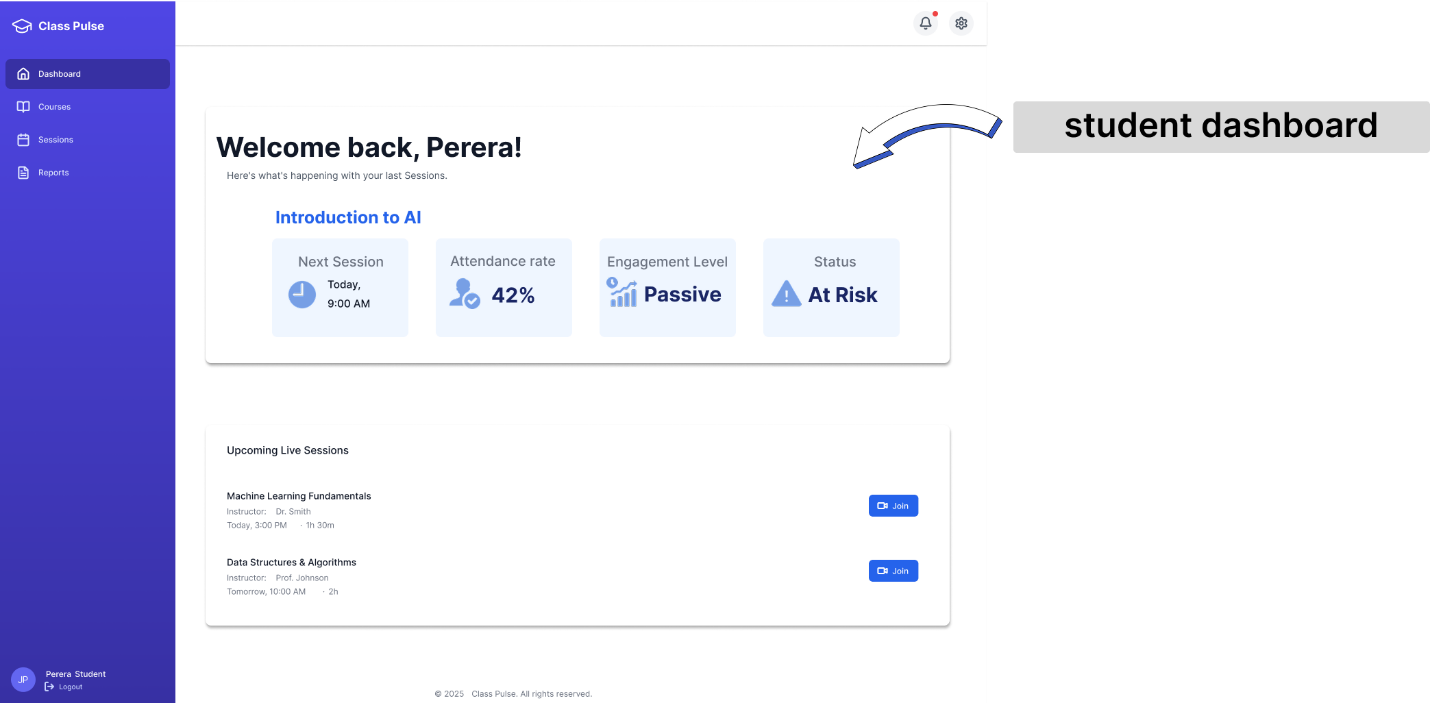


Figure :Login Form

Figure :Login Form



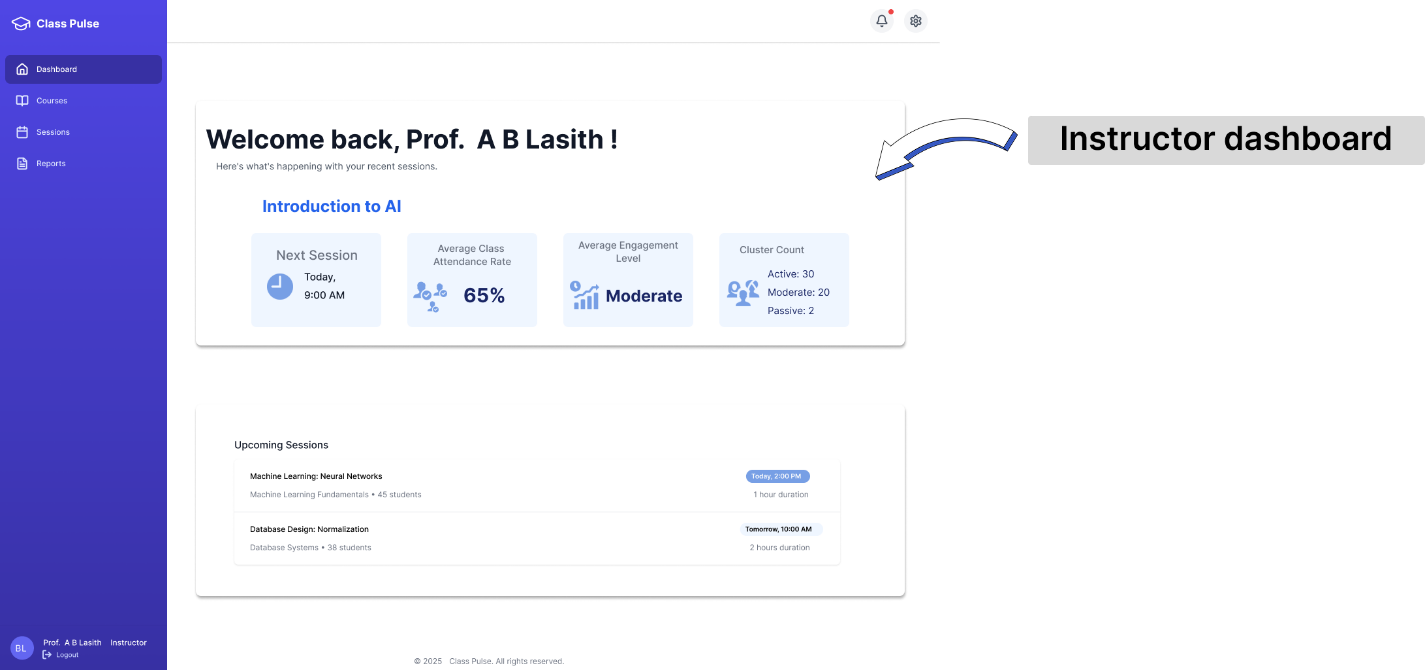


Figure :Instuctor dashboard

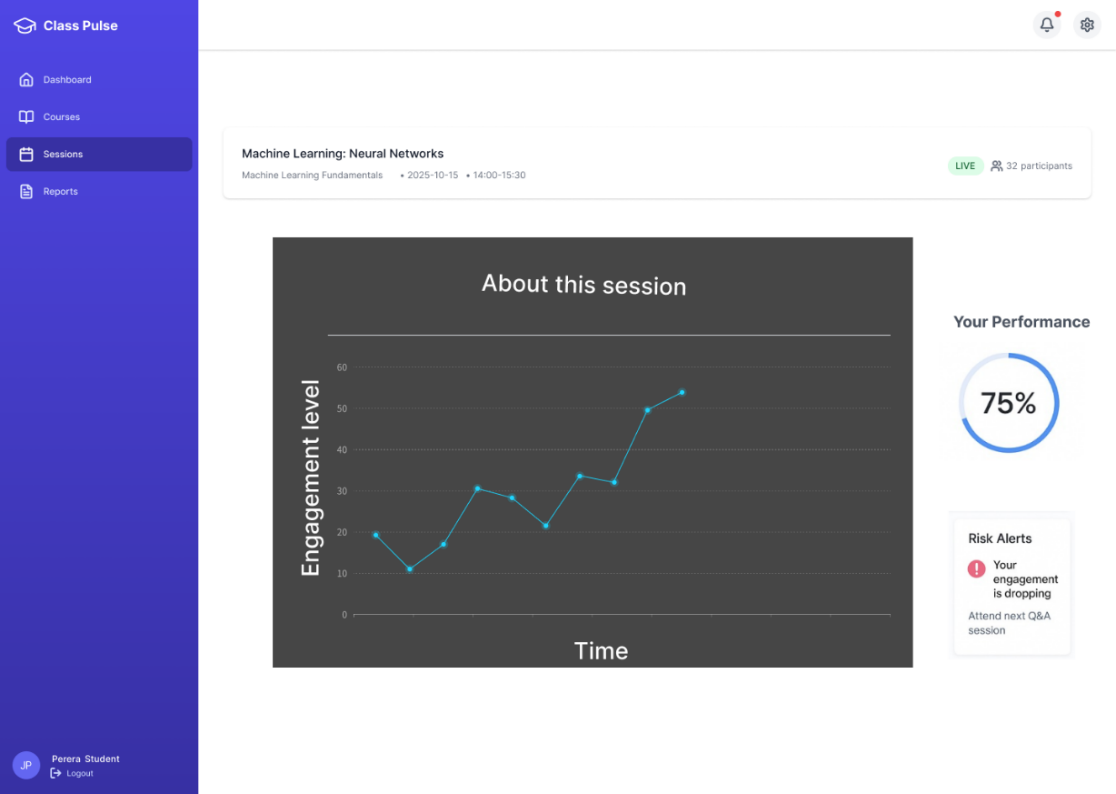
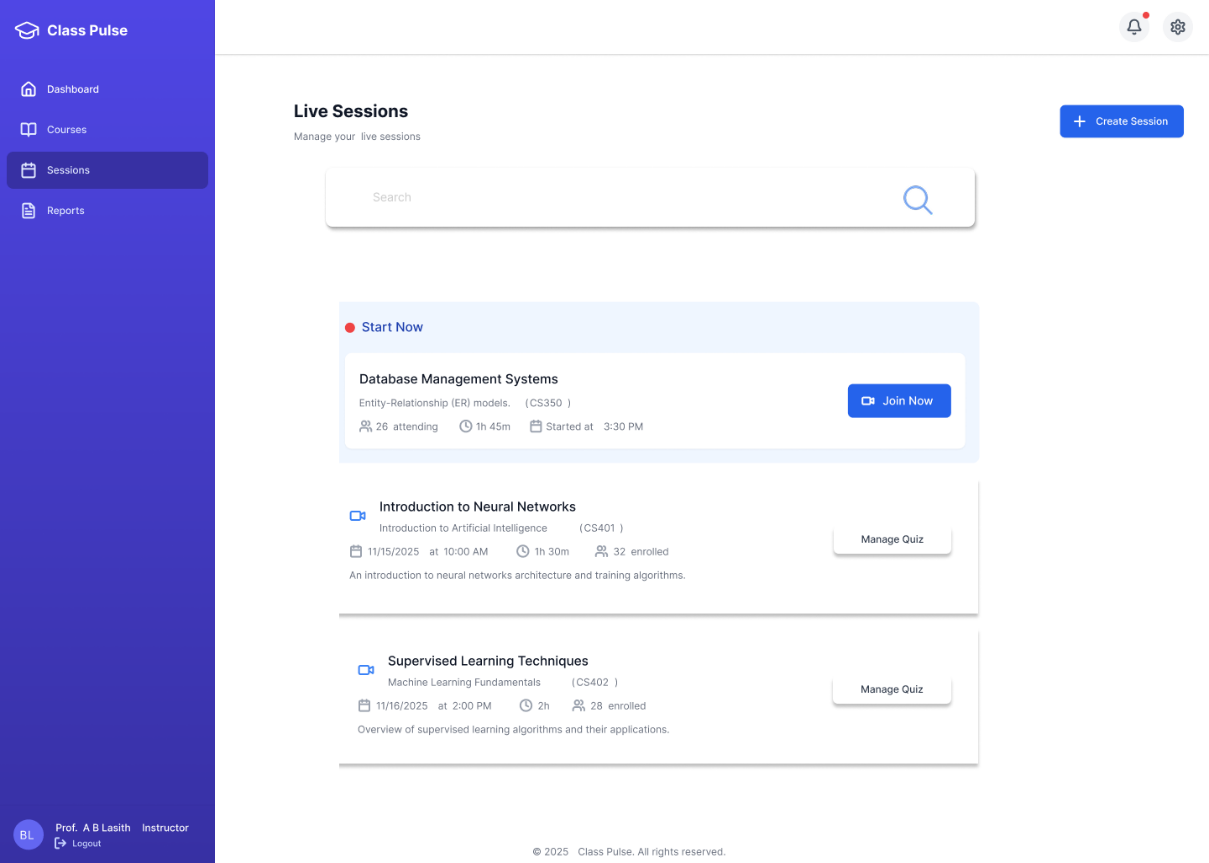


Figure :Live Session Management Screen

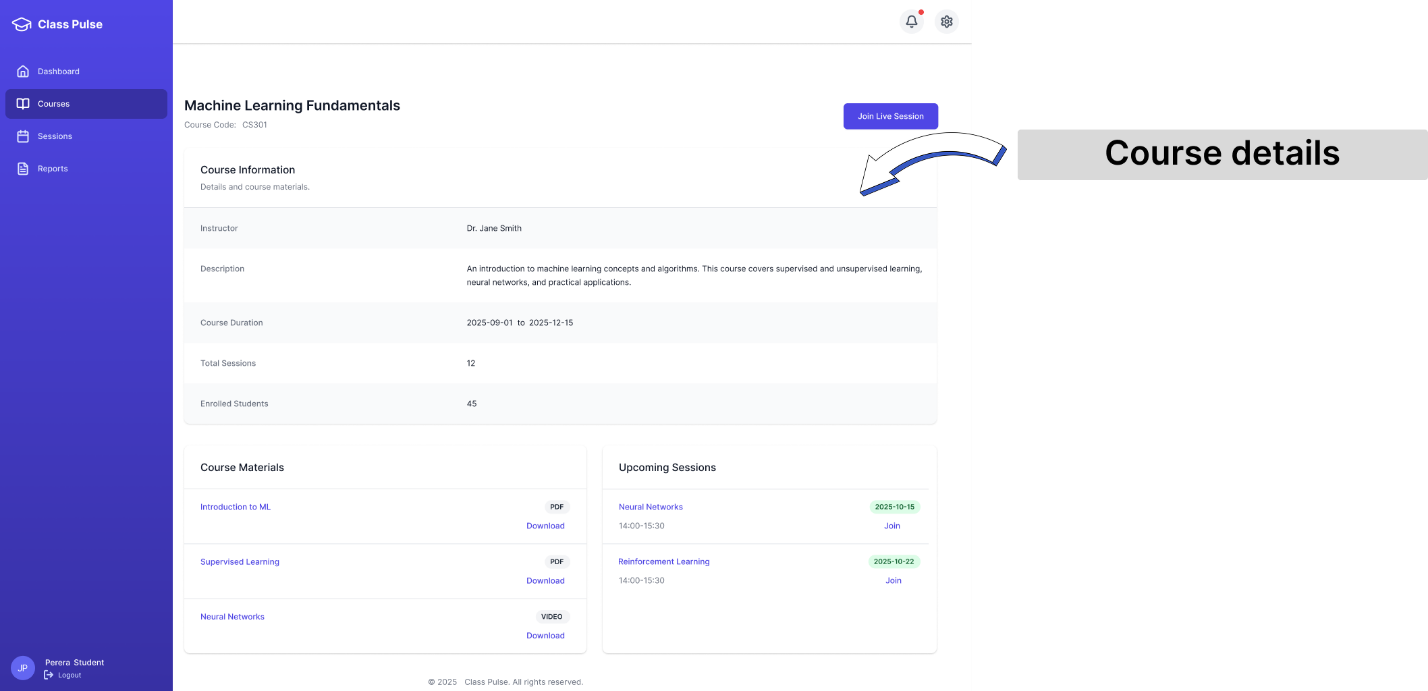
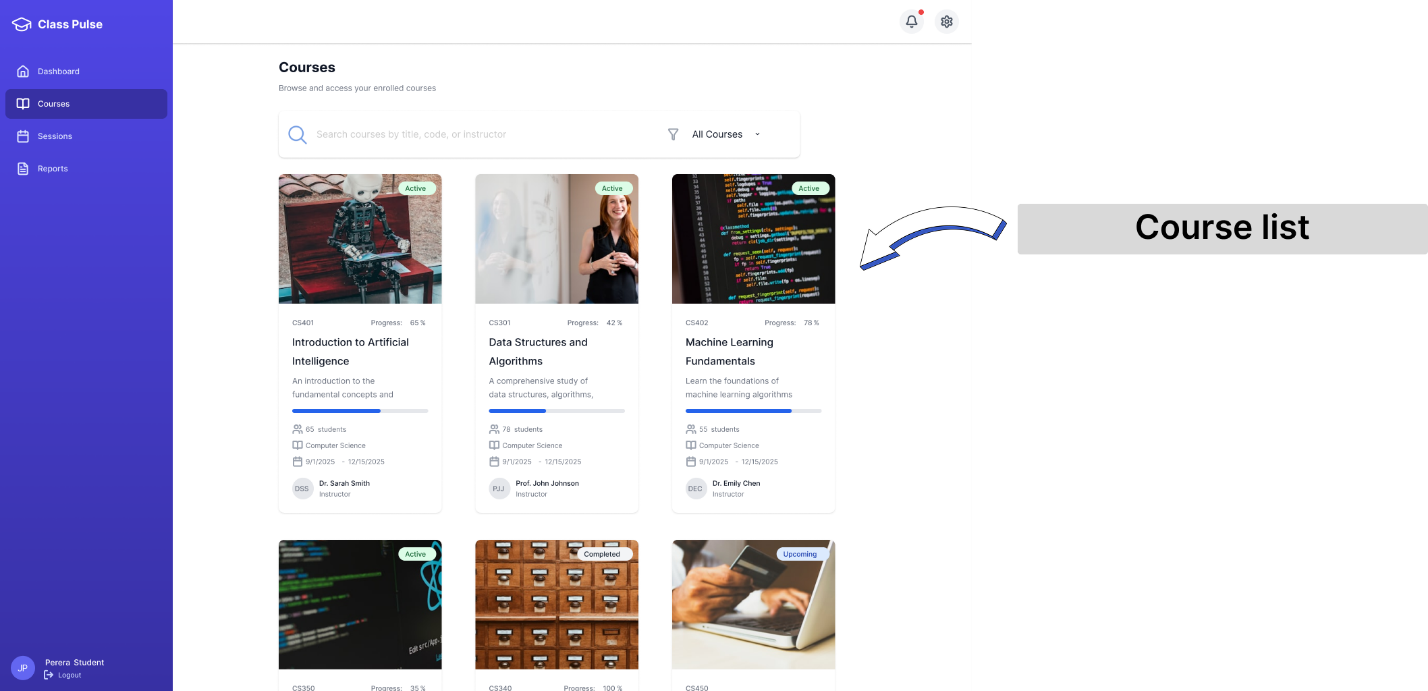


Figure :Course List & Management Options

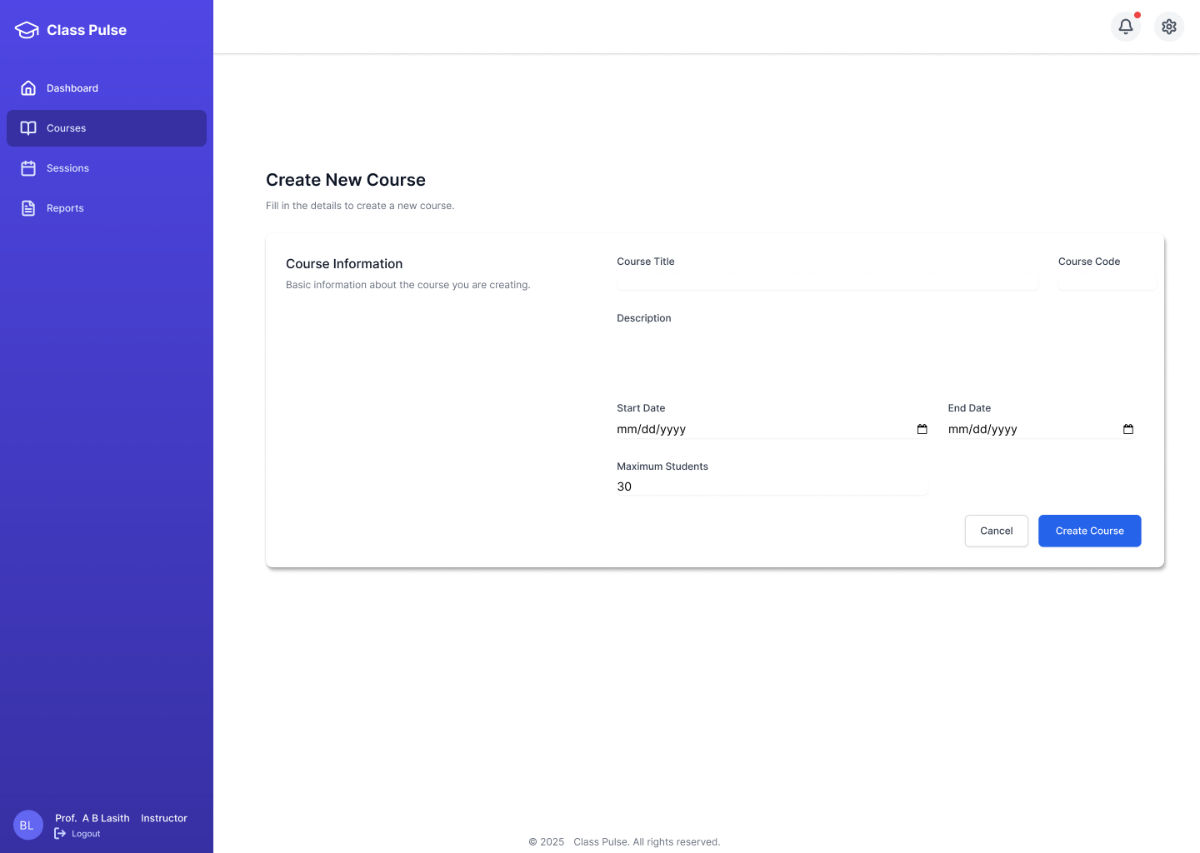


Figure :Add / Edit Course Details Screen

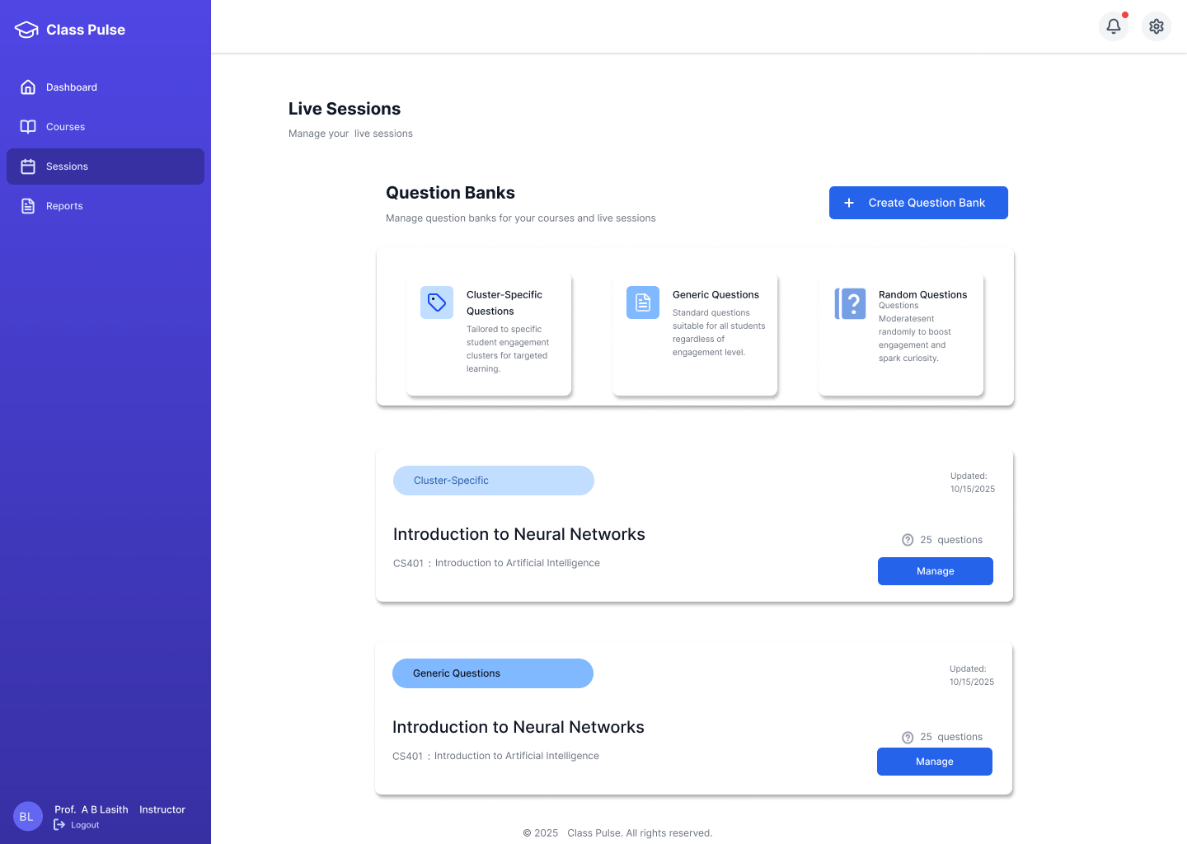
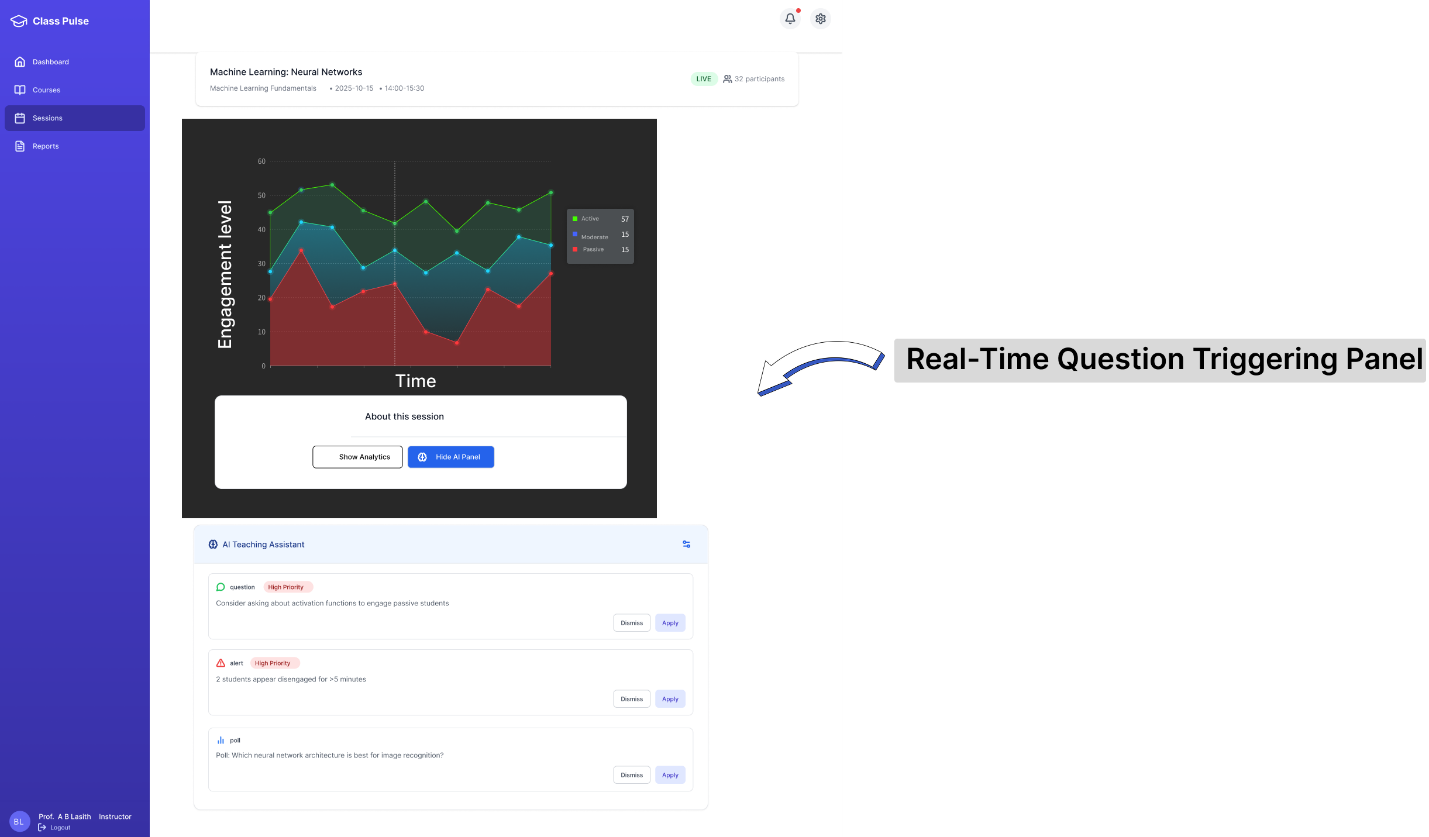


Figure :Question Bank Management



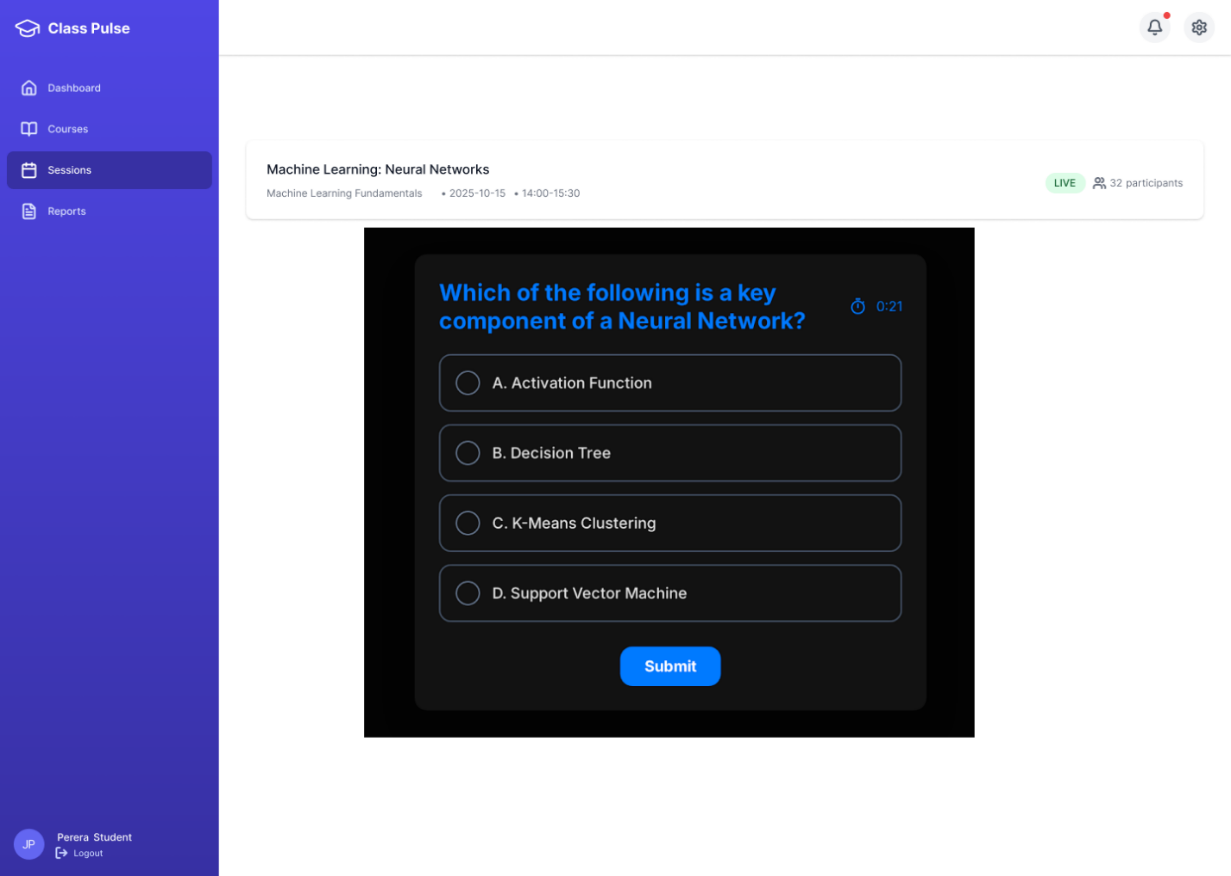
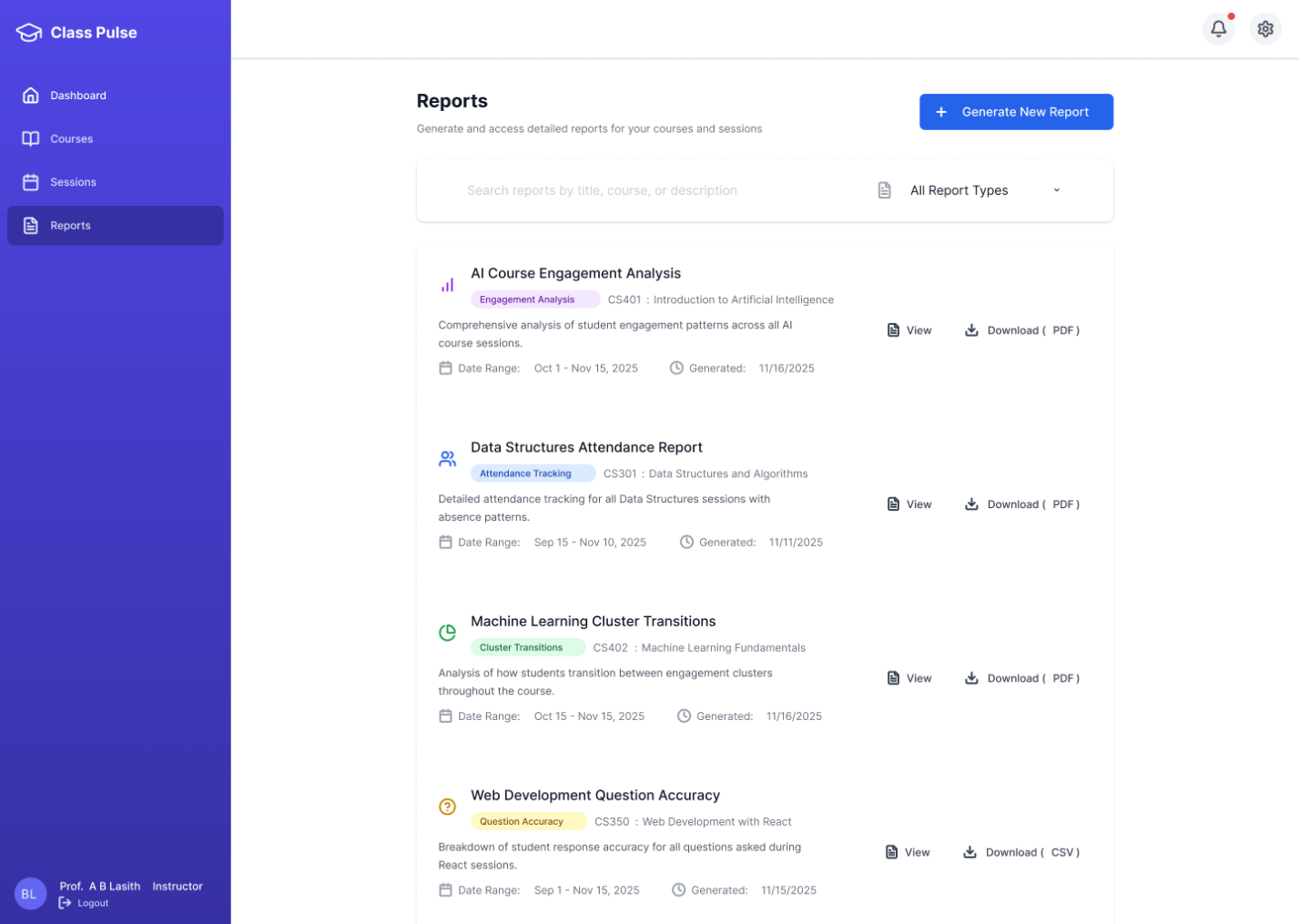


Figure :Real-Time Question Triggering Panel



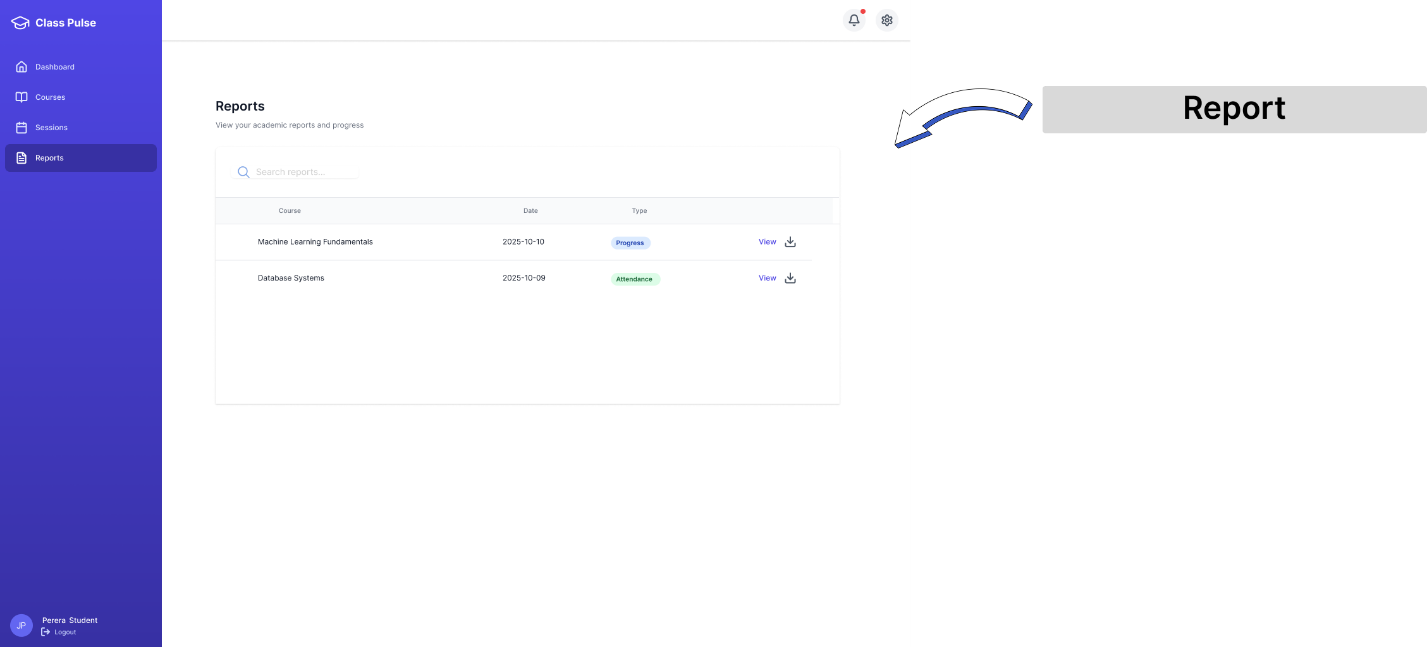


Figure :Report Type Selection Screen

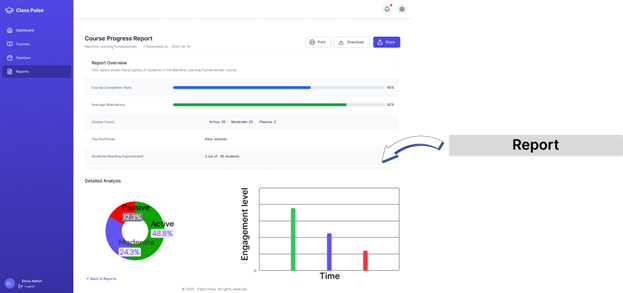


Figure :Report Output / Download Screen

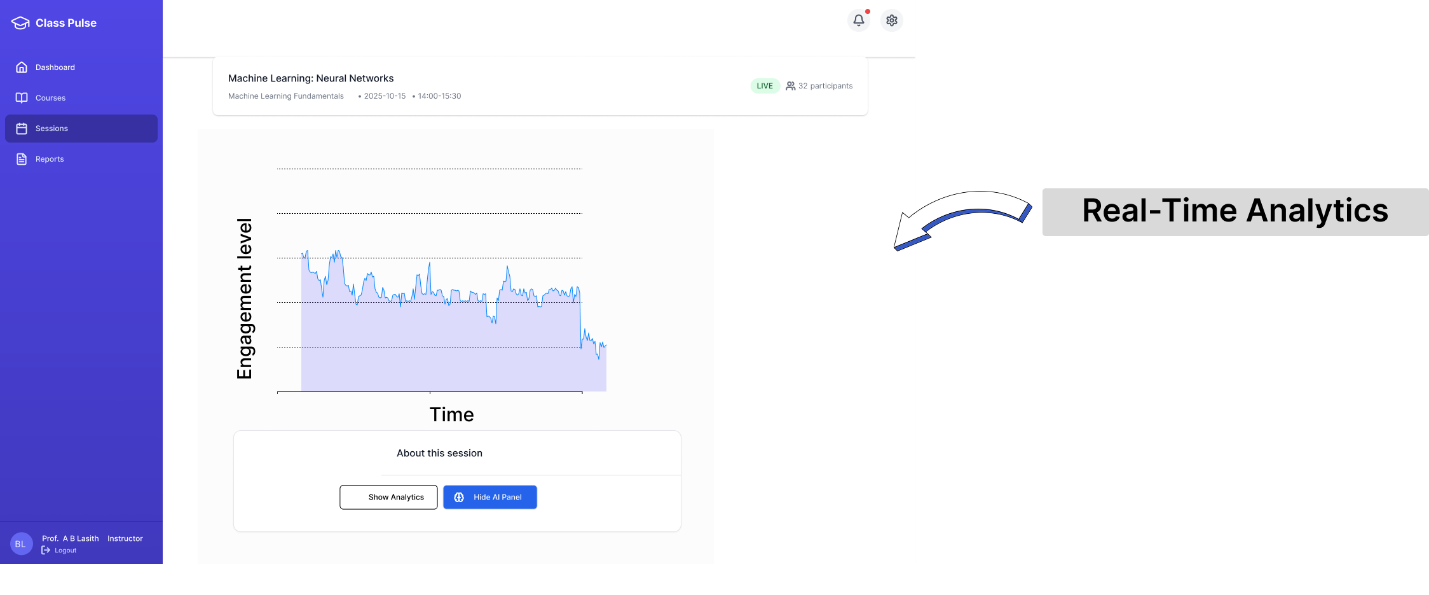


Figure :Real-Time Analytics

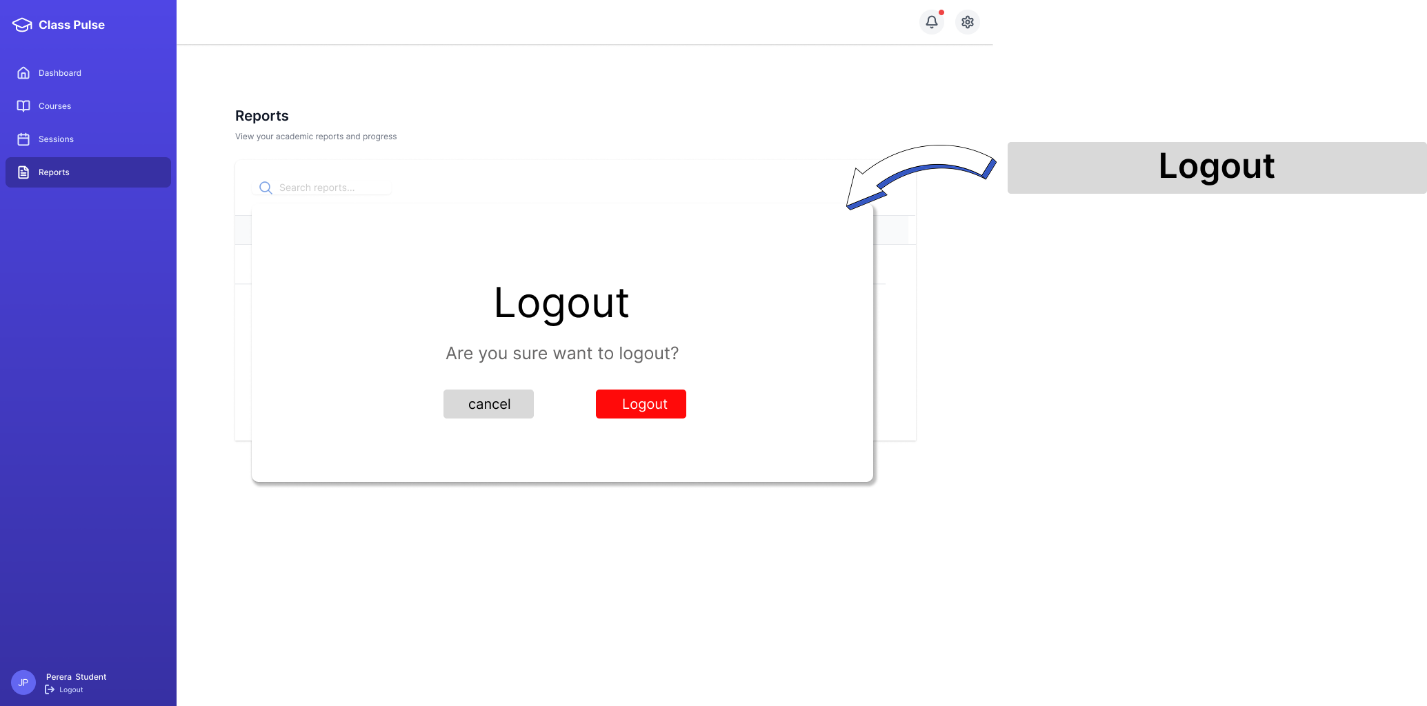


Figure :Logout Confirmation Screen

## **3.4 Hosting / Installation Environment**

The system is hosted on a cloud platform that supports backend services, real time data processing and Zoom API integration. Backend components such as session management, analytics processing and API services are deployed on cloud platforms like AWS or Google Cloud, ensuring scalability, security and reliable performance during live sessions.

Machine learning models used for clustering, question delivery and personalized feedback are also deployed as cloud services, allowing fast responses and continuous availability. The system uses MongoDB Atlas for data storage, which provides secure, highly available cloud based storage for user data, session logs and engagement records.

This cloud based setup ensures smooth operation, easy maintenance and strong support for real time analytics and online learning activities.

# **CHAPTER 4 – DATA MANAGEMENT**

## **4.1 Introduction**

Data management plays a critical role in the proposed system, as it supports real time engagement monitoring, predictive clustering and analytics in video conferencing based learning environments.

During live sessions, the system continuously generates high frequency data such as participant join/leave events, quiz responses, engagement metrics and clustering results. Efficient storage, retrieval and processing of this data are essential to ensure low latency, scalability and reliability.

To meet these requirements, the system adopts MongoDB, a document oriented NoSQL database, as the primary database management system. MongoDB’s flexible schema model, high write throughput and powerful aggregation framework make it well suited for real time analytics and evolving data structures.

## **4.2 Justification for Selecting MongoDB**

MongoDB was selected over traditional relational databases due to the following system specific considerations

* The system processes real time, high volume event data generated during live Zoom sessions.
* Data structures such as engagement events, network metrics, and clustering outputs evolve during experimentation and model refinement.
* Frequent aggregation operations are required to compute engagement levels, student clusters and session analytics.
* Avoidance of complex multi table joins is necessary to maintain low latency for dashboards.

## **4.3 Data Requirements**

The system manages multiple categories of data, including

* User and instructor information
* Course and enrollment details
* Live session metadata
* Zoom participant events
* Student engagement and quiz interaction data
* Engagement clustering and prediction results

## **4.4 Database Design Approach**

The database design follows a workload driven MongoDB schema design process, consisting of

1. Identification of frequent read and write operations
2. Mapping of data relationships based on access patterns
3. Selection of embedding or referencing strategies
4. Application of MongoDB schema design patterns
5. Index design based on common queries

## **4.5 Identification of Application Workload**

The primary workload of the system includes

* High frequency write operations for Zoom participant events and quiz answers
* Continuous updates to engagement scores and clustering results
* Frequent read operations for real time dashboards
* Periodic analytical queries for session summaries and reports

## **4.6 Collection Design**

### **4.6.1 Users Collection**

Stores user profile and authentication information.

**Key Attributes**

* User ID
* Name
* Email
* Role (Student / Instructor)
* Account creation timestamp

This collection is referenced by courses, sessions, and quiz answers.

### **4.6.2 Courses Collection**

Stores information related to learning courses.

**Key Attributes**

* Course title and description
* Instructor details
* Category and difficulty level
* Enrollment key
* Enrolled student identifiers
* Course status and timestamps

A subset of enrolled student details is stored to support quick access without additional lookups.

### **4.6.3 Sessions Collection**

Stores metadata for live learning sessions.

**Key Attributes**

* Session title and course information
* Instructor details
* Scheduled date and duration
* Zoom meeting identifiers and URLs
* Session status and timestamps

Session identifiers are referenced across engagement, clustering, and quiz collections.

### **4.6.4 Zoom Participants Collection**

Stores real time participant join and leave events received via Zoom webhooks.

**Key Attributes**

* Meeting ID
* Participant name and email
* Join and leave timestamps
* Event type (joined / left)

This collection supports time based analytics and attendance tracking.

### **4.6.5 Questions Collection**

Stores quiz questions used during sessions.

**Key Attributes**

* Question text
* Answer options
* Correct answer index
* Difficulty level
* Category and time limit

Questions are referenced by quiz answer documents.

### **4.6.6 Quiz Answers Collection**

Stores student responses to quiz questions.

**Key Attributes**

* Question ID
* Student ID
* Session ID
* Selected answer
* Time taken
* Network performance metrics
* Timestamp

This collection is optimized for high frequency write operations.

### **4.6.7 Clusters Collection**

Stores engagement clustering results generated by analytics models.

**Key Attributes**

* Cluster name and description
* Engagement level
* Prediction state
* Student count
* Visual attributes (colour)
* Associated session ID

Cluster data supports real time monitoring of student engagement states.

## **4.7 Data Relationship Mapping**

Data relationships are designed based on access patterns

* One-to-many relationships exist between sessions and participants, quiz answers, and engagement events.
* Many-to-many relationships exist between students and courses.
* Session centric data is grouped logically to support analytics.

## **4.8 Application of MongoDB Schema Design Patterns**

The following MongoDB schema design patterns are applied

* Embedded Pattern for session centric analytics data
* Reference Pattern for users, questions, and sessions
* Subset Pattern for enrolled student and instructor details
* Computed Pattern for engagement levels and cluster predictions
* Bucket Pattern for time based Zoom and engagement events

## **4.9 Indexing Strategy**

To maintain low latency during live sessions, the proposed system applies an indexing strategy based on the actual query patterns performed on the stored MongoDB documents such as sessions, courses, clusters, quiz answers, and Zoom participant events. Because the system generates high frequency writes (join/leave events, quiz answers, engagement updates), indexes are created only on fields that are repeatedly used for filtering, lookup, sorting, and aggregation.

### **4.9.1 Common Query Patterns in the System**

The most frequent database operations include

* Retrieving a session using Zoom meeting identifiers.
* Loading all clustering results for a session using sessionId in the Clusters collection.
* Fetching quiz answers for a given session and student using sessionId and studentId in the Quiz Answers collection.
* Filtering Zoom participant events for a meeting and ordering them by time using meeting\_id and timestamp in the Zoom Participants collection.
* Loading course data by instructor using instructorId and verifying enrollment keys using enrollmentKey.
* Retrieving questions by category or difficulty when generating targeted questions.

### **4.9.2 Indexes Implemented**

Based on the above patterns, the following indexes are defined.

#### **(A) Sessions Collection**

* Single field index on zoomMeetingId

#### **(B) Clusters Collection**

* Single field index on sessionId
* Compound index on { sessionId, engagementLevel }

#### **(C) Quiz Answers Collection**

* Compound index on { sessionId, studentId }
* Single field index on questionId
* Single field index on timestamp

#### **(D) Zoom Participants Collection**

* Compound index on { meeting\_id, timestamp}

#### **(E) Courses Collection**

* Single field index on instructorId
* Single field unique index on enrollmentKey

#### **(F) Questions Collection**

* Single field indexes on category and difficulty

### **Collections & Schema design**

1. **Users Collection**

Table :Users Collection

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Description** |
| \_id | ObjectId | Yes | Unique user identifier |
| firstName | String | Yes | User first name |
| lastName | String | Yes | User last name |
| email | String | Yes | Registered email address |
| password | String | Yes | Hashed user password |
| role | String | Yes | User role (student / instructor / admin) |
| status | Number | Yes | Account status (1 = active, 0 = inactive) |
| createdAt | Date | Yes | Account creation timestamp |
| updatedAt | Date | Yes | Last account update timestamp |

1. **Courses Collection**

Table :Courses Collection

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Description** |
| \_id | ObjectId | Yes | Unique course ID |
| title | String | Yes | Course title |
| description | String | Yes | Course description |
| instructorId | ObjectId | Yes | Reference to instructor user |
| instructorName | String | Yes | Instructor name |
| instructorEmail | String | Yes | Instructor email |
| category | String | Yes | Course category |
| duration | String | Yes | Course duration |
| level | String | Yes | Course level (Beginner / Intermediate / Advanced) |
| thumbnail | String / Null | No | Course thumbnail image |
| syllabus | Array | No | Course syllabus items |
| maxStudents | Number | Yes | Maximum enrollment limit |
| status | String | Yes | Course status (draft / published) |
| enrolledStudents | Array | Yes | List of enrolled student IDs |
| enrolledStudentDetails | Array | No | Embedded student enrollment info |
| enrollmentKey | String | Yes | Enrollment access key |
| enrollmentKeyActive | Boolean | Yes | Enrollment key status |
| createdAt | Date | Yes | Course creation date |
| updatedAt | Date | Yes | Course update date |

1. **Sessions Collection**

Table :Sessions Collection

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Description** |
| \_id | ObjectId | Yes | Unique session ID |
| title | String | Yes | Session title |
| course | String | Yes | Course name |
| courseCode | String | Yes | Course code |
| instructor | String | Yes | Instructor name |
| date | String | Yes | Session date |
| time | String | Yes | Session start time |
| duration | String | Yes | Session duration |
| status | String | Yes | Session status (upcoming / live / completed) |
| participants | Number | Yes | Current participant count |
| expectedParticipants | Number | Yes | Expected participants |
| engagement | Number | Yes | Engagement score |
| recordingAvailable | Boolean | Yes | Recording availability |
| zoomMeetingId | String | Yes | Zoom meeting ID |
| join\_url | String | Yes | Zoom join URL |
| start\_url | String | Yes | Zoom host start URL |
| createdAt | Date | Yes | Session creation timestamp |

1. **Zoom Participants Collection**

Table :Zoom Participants Collection

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Description** |
| \_id | ObjectId | Yes | Unique event ID |
| event | String | Yes | Zoom event type (joined / left) |
| meeting\_id | String | Yes | Zoom meeting ID |
| meeting\_uuid | String | Yes | Zoom meeting UUID |
| topic | String | Yes | Meeting topic |
| user\_id | String | Yes | Zoom user ID |
| name | String | Yes | Participant name |
| email | String | Yes | Participant email |
| join\_time | Date | No | Participant join time |
| leave\_time | Date | No | Participant leave time |
| timestamp | Date | Yes | Event received timestamp |

1. **Clusters Collection**

Table :Clusters Collections

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Description** |
| \_id | ObjectId | Yes | Cluster identifier |
| id | String | Yes | Cluster label ID |
| name | String | Yes | Cluster name |
| description | String | Yes | Cluster description |
| studentCount | Number | Yes | Number of students in cluster |
| engagementLevel | String | Yes | Engagement level (Passive / Moderate / Active) |
| color | String | Yes | Cluster visualization color |
| prediction | String | Yes | Engagement trend prediction |
| students | Array | No | Student IDs in cluster |
| sessionId | String | Yes | Related session ID |

1. **Quiz Answers Collection**

Table :Quiz Answers Collection

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Description** |
| \_id | ObjectId | Yes | Quiz answer ID |
| questionId | ObjectId | Yes | Question reference |
| answerIndex | Number | Yes | Selected answer index |
| timeTaken | Number | Yes | Time taken to answer (seconds) |
| studentId | ObjectId | Yes | Student ID |
| sessionId | String | Yes | Session ID |
| timestamp | Date | Yes | Answer submission time |
| networkStrength | Object | No | Network quality metrics |
| networkStrength.quality | String | No | Network quality (good / average / poor) |
| networkStrength.rttMs | Number | No | Network RTT (ms) |
| networkStrength.jitterMs | Number | No | Network jitter (ms) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Required** | **Description** |
| \_id | ObjectId | Yes | Unique question ID |
| question | String | Yes | Question text |
| options | Array | Yes | List of possible answer options |
| correctAnswer | Number | Yes | Index of the correct option (0-based) |
| difficulty | String | Yes | Difficulty level (easy / medium / hard) |
| category | String | Yes | Question category or topic |
| tags | Array | No | Additional tags related to the question |
| timeLimit | Number | Yes | Time limit to answer the question (in seconds) |
| createdAt | Date | Yes | Question creation timestamp |

1. **Question Collection**

Table 15:Question Collection

## 

## **4.10 Security Design**

Database security design comprises a range of security control mechanisms to protect the database and underlying infrastructure from threats arising both externally and internally. Since the proposed system handles sensitive data such as user information, session analytics, engagement scores, and quiz responses, multiple security controls are implemented to enhance database security.

**Access Control**

The following access control mechanisms are implemented to ensure the confidentiality and integrity of sensitive system data

* Valid credentials such as a verified email address and password are required for any user to access the system.
* Role based access control is enforced to restrict system functionality based on user roles (Student, Instructor).
* Only instructors are allowed to create courses, schedule sessions, view engagement analytics, and monitor student clusters.
* Students are permitted to view only their own quiz responses, engagement data, and session participation details.
* Students are not allowed to access other students’ personal or performance related information.
* Administrative operations such as system monitoring and configuration are restricted to authorized system components only.

**Authentication**

The system uses secure authentication mechanisms to verify user identity

* User email addresses are verified during the registration process through an email verification mechanism to ensure account authenticity.
* During each login, users are required to provide their verified email address and password, which are validated before granting access to the system.
* Authentication tokens are used to maintain secure user sessions and prevent unauthorized access.

**Backups**

Data backups are a critical component of the database security design. To ensure data availability and recovery

* All important system data, including user information, session records, engagement analytics, and quiz responses, are regularly backed up.
* Backup data is stored in a separate database instance or secure storage location to protect against data loss, system failure, or accidental deletion.
* Backup and monitoring procedures ensure that the system can be restored quickly in the event of a failure.

## **4.11 ER Diagram**

Figure :ER Diagram

## **4.12 Relational Database Mapping**

### **4.12.1 Map Regular Entities**

• User Strong Entity

Figure: User strong entity

USER  
( id, email, first\_name, last\_name, role, created\_at, last\_login, is\_active )

Primary Key: id

• Course Strong Entity

Figure: Course strong entity

COURSE  
( id, course\_code, course\_name, description, instructor\_id, instructor\_name, semester, year, credits, status, enrolled\_count )

Primary Key: id

• Question Strong Entity

Figure: Question strong entity

QUESTION  
( id, question\_text, question\_type, difficulty, course\_id, created\_by, correct\_answer, options )

Primary Key: id

• Quiz Answers Strong Entity

Figure: Quiz Answers strong entity

QUIZ\_ANSWERS  
( id, session\_id, student\_id, question\_id, answer\_index, is\_correct, time\_taken, network\_quality, answered\_at )

Primary Key: id

The QUIZ\_ANSWERS entity stores individual quiz responses submitted by students during live sessions.

• Reports Strong Entity

Figure: Reports strong entity

SESSION\_REPORTS  
( id, session\_id, session\_title, course\_name, course\_code, user\_id, user\_name, session\_date, session\_status, total\_participants, total\_questions\_asked, average\_quiz\_score, highly\_engaged\_count, moderately\_engaged\_count, at\_risk\_count, report\_type )

Primary Key: id

### **4.12.2 Map Weak Entities**

There are no weak entities identified in the EER diagram. All entities are uniquely identifiable using preserved MongoDB identifiers.

### **4.12.3 Binary Relationships**

**• Map Binary One-to-Many (1 : N) Relationship**

➤ Course and Reports

Mapped Relations

COURSE (id, instructor\_id)  
Primary Key: id

SESSION\_REPORTS (id, user\_id)  
Primary Key: id

Each course can generate multiple session reports over time.

➤ Question and Quiz Answers

Mapped Relations

QUESTION (id, question\_text)  
Primary Key: id

QUIZ\_ANSWERS (id, question\_id)  
Primary Key: id

Each question can receive multiple quiz answers from different students.

➤ User and Quiz Answers

Mapped Relations

USER (id)  
Primary Key: id

QUIZ\_ANSWERS (id, question\_id)  
Primary Key: id

Each user can submit multiple quiz answers across different sessions.

➤ Course and Question

Mapped Relations

COURSE (id, instructor\_id)  
Primary Key: id

QUESTION ( id, course\_id )  
Primary Key: id

Each course can contain multiple questions, but each question belongs to exactly one course.

**• Map Binary Many-to-Many (M : N) Relationship**

➤ User and Question (Submits Relationship)

Mapped Relations

USER (id)  
Primary Key: id

QUESTION (id, course\_id)  
Primary Key: id

QUIZ\_ANSWERS (id, question\_id)  
Primary Key: id

This relationship is resolved through the QUIZ\_ANSWERS entity.

### **4.12.7 Map Aggregation**

There are no aggregation relationships explicitly modeled in the EER diagram. Aggregated metrics are stored directly within the session reports backup table.

## **4.13 Schema Refinement**

Schema refinement ensures structural consistency between MongoDB and MySQL.  
Although the backup database is denormalized, this design is intentional to preserve historical data states. All entities maintain functional dependencies and avoid update anomalies, making further normalization unnecessary.

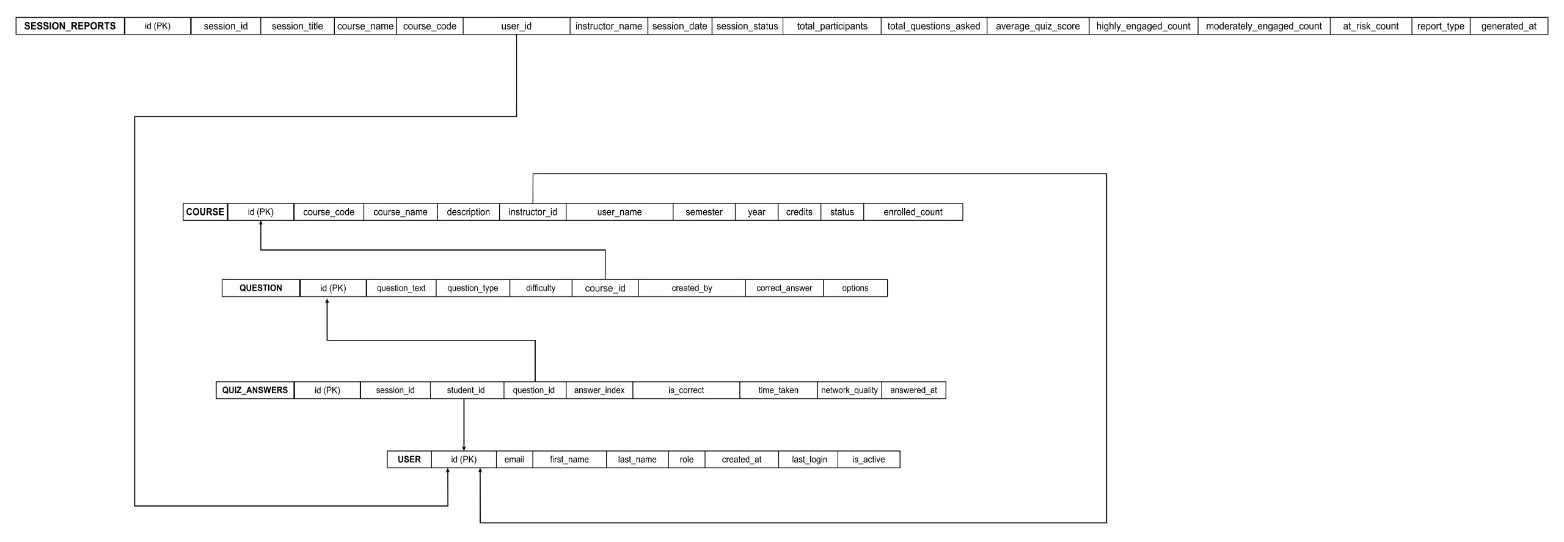


Figure : Schema Refinement Table

## **4.14 Physical Database Design**

The following tables reflect the actual implemented backup schema.

1. USERS\_BACKUP Table

USERS\_BACKUP

(

id INT,

id VARCHAR(100),

email VARCHAR(255),

first\_name VARCHAR(100),

last\_name VARCHAR(100),

role VARCHAR(50),

created\_at DATETIME,

last\_login DATETIME,

is\_active TINYINT,

full\_document JSON,

backed\_up\_at DATETIME

)

2. COURSES\_BACKUP Table

COURSES\_BACKUP

(

id INT,

id VARCHAR(100),

course\_code VARCHAR(50),

course\_name VARCHAR(255),

description TEXT,

instructor\_id VARCHAR(100),

instructor\_name VARCHAR(255),

semester VARCHAR(50),

year INT,

credits INT,

status VARCHAR(50),

enrolled\_count INT,

created\_at DATETIME,

backed\_up\_at DATETIME

)

3. QUESTIONS\_BACKUP Table

QUESTIONS\_BACKUP

(

id INT,

id VARCHAR(100),

question\_text TEXT,

question\_type VARCHAR(50),

difficulty VARCHAR(50),

course\_id VARCHAR(100),

created\_by INT,

correct\_answer INT,

options JSON,

tags JSON,

full\_document JSON,

created\_at DATETIME,

backed\_up\_at DATETIME

)

4. QUIZ\_ANSWERS\_BACKUP Table

QUIZ\_ANSWERS\_BACKUP

(

id INT,

id VARCHAR(100),

session\_id VARCHAR(100),

student\_id VARCHAR(100),

question\_id VARCHAR(100),

answer\_index INT,

is\_correct TINYINT,

time\_taken DECIMAL(5,2),

network\_quality VARCHAR(50),

answered\_at DATETIME,

full\_document JSON,

backed\_up\_at DATETIME

)

5. SESSION\_REPORTS\_BACKUP Table

SESSION\_REPORTS\_BACKUP

(

id INT,

id VARCHAR(100),

session\_id VARCHAR(100),

session\_title VARCHAR(255),

course\_name VARCHAR(255),

course\_code VARCHAR(50),

user\_id VARCHAR(100),

user\_name VARCHAR(255),

session\_date DATE,

session\_status VARCHAR(50),

total\_participants INT,

total\_questions\_asked INT,

average\_quiz\_score DECIMAL(5,2),

highly\_engaged\_count INT,

moderately\_engaged\_count INT,

at\_risk\_count INT,

report\_type VARCHAR(50),

generated\_at DATETIME,

full\_document JSON,

backed\_up\_at DATETIME

)

# **CHAPTER 5 - RESEARCH DESIGN**

This chapter outlines the project objectives based on literature review and the project approach according to the project plan. Sampling methods, data collection methods and data analysis methods are described in this chapter. Objectives of our project are specified based on literature review in section 5.1. Formalizing high level implementation components are described under 5.2 section and Data preparation and training datasets in 5.3. Finally, the proposed validation methods and measurements are described in section 5.5.

## **5.1 Objective Based Literature Review**

### **5.1.1 Literature Review for Objective – 1**

**Analyze key factors influencing student engagement and learning effectiveness in video conferencing based learning environments.**

Research consistently shows that student engagement in online and video conferencing environments is highly determined by behavioural patterns, interaction frequency, and responsiveness. Ayouni et al. highlight that analysing live behavioural data is essential to understand engagement levels in virtual learning environments [1]. Their work shows that participation patterns, activity density, and responsiveness can be used as reliable indicators of learner engagement. Subramaniya and Wang show that behavioural features such as login timing, interaction counts, and response delays can be used to predict student engagement using machine learning methods [2]. However, traditional approaches often struggle to capture dynamic, real time fluctuations during live sessions.

Clustering techniques are used to group learners based on similar behavioural characteristics. Johnston et al. demonstrate that clustering helps identify engagement patterns and classify learners into meaningful groups such as high, medium, and low engagement clusters [3]. Pathak, Stynes, and Palani further support this by proposing clustering methods specifically designed to detect low engagement students in virtual learning environments [4].

In addition to behavioural analytics, studies highlight the importance of learner feedback. Christopher Dann et al. show that student feedback provides valuable qualitative insights that help identify emotional and cognitive engagement factors not visible through activity logs alone [6].Mahafdah et al. reinforce that combining behavioural signals with AI driven analysis leads to more accurate understanding of learner performance and engagement patterns [5].

Overall, the literature suggests that identifying engagement levels requires the integration of behavioural data, clustering methods, and learner feedback. However, existing systems often lack real time detection and integration with live video conferencing platforms. And our project aims to address these gaps.

### **5.1.2 Literature Review for Objective – 2**

**to develop a predictive learner clustering model that groups students based on real time behavioural and performance data.**

Research shows that clustering is an effective method for grouping learners based on behavioural and performance data. In [4] they demonstrate that activity based clustering can identify low engagement learners in virtual environments.in [3] they show that clustering models help categorize students into meaningful engagement groups in online platforms.

To enhance this, in our project predictive modelling is used to forecast changes in engagement. In [2] they highlight that behavioural signals such as interaction frequency and response time can be used to predict engagement levels using machine learning.In another study they emphasize the importance of analysing real time behavioural indicators when assessing engagement in online learning [1]. Together, these studies indicate that combining behavioural clustering with predictive modelling is suitable for developing a real time learner clustering model for video conferencing based learning environments.

### **5.1.3 Literature Review for Objective – 3**

**to develop mechanisms for delivering targeted and active interactions that address the specific needs of identified learner clusters**.

Targeted and active interactions in online learning environments depend on accurately identifying learner engagement patterns and delivering appropriate questions. Research shows that non targeted interaction methods often fail to maintain student engagement during virtual sessions, as all learners receive the same type of questions based on their engagement level [1]. Studies state that real time behavioural data such as participation frequency, responsiveness and activity density can be used to differentiate learners and support targeted interventions [2].

Clustering techniques give a foundation for adaptive interaction, as they allow learners to be grouped according to similar behavioural characteristics. Research papers demonstrate that clustering can identify low engagement learners who may require additional support or simpler interactive activities [4]. Interpretable clustering methods further support the process by enabling clear differentiation of learner groups and informing the design of cluster appropriate interactions [3].

Overall, the existing research validates the use of behavioural clustering combined with real time analytics to deliver targeted and active interactions. It helps to match the specific needs of different learner groups in a live video conferencing environment.

### **5.1.4 Literature Review for Objective – 4**

**to develop a model that provides personalized feedback based on student clusters.**

Personalized feedback in online learning environments requires models capable of interpreting learner behaviour and generating adaptive responses. Research shows that traditional systems often fail to provide realtime adaptive feedback, resulting in reduced student engagement and limited instructional support [7]. Studies emphasize that behavioural indicators such as participation levels, response accuracy, and interaction timing play a critical role in determining the type of feedback students require during virtual sessions [2].

Machine learning approaches are used to analyze learner behaviour and generate feedback signals. Research evidence indicates that combining behavioural data with predictive modelling can produce more accurate insights about a learner’s performance and progress [5]. This helps for a personalized feedback mechanism which adapts to the changing of engagement patterns.

Clustering based approaches also contribute to feedback personalization by grouping learners with similar behavioural characteristics, enabling instructors to deliver cluster appropriate feedback [4]. Interpretable clustering further enhances this process by making feedback decisions clearer and more transparent for educational use [3].

Research additionally highlights the importance of integrating qualitative feedback. Such qualitative insights strengthen the overall feedback mechanism and help address cognitive and emotional dimensions of learning that behavioural logs alone cannot capture [6].

Overall, the literature indicates that combining real time behavioural analytics, clustering techniques, and predictive modelling creates an effective foundation for generating personalized feedback in video conferencing based learning environments.

## **5.2 Formalizing High level Implementation Component**



Figure :High level Implementation Component

The system is implemented as a full stack web application integrated with real time video conferencing services and a cloud hosted database. The database acts as the central repository for the system’s three core components such as Predictive Learner Clustering Model, Targeted Question Triggering Module, and Personalized Feedback Generation Model. These components altogether support dynamic engagement monitoring with prediction, adaptive interaction delivery, and personalized feedback within live learning sessions.

These three intelligent components interact with the database to store, retrieve, and update all information required for real time decision making. The Predictive Learner Clustering Model processes behavioural and performance data such as interaction events, quiz responses and reaction counts to assign students into engagement clusters. These cluster labels and engagement insights are continuously updated during the session and stored in the database.

The Targeted Question Triggering Module retrieves questions from the stored question, which contains cluster wise, generic and random questions. Based on the student's current cluster and engagement behaviour, the module selects suitable questions and directs them to the respective student interfaces. Both the question selections and the students' responses are recorded in the database to support analytics and intervention strategies.

The Personalized Feedback Generation Model analyzes cluster transitions, answer accuracy, participation frequency, and behavioural indicators to generate individualized feedback messages for each student. These feedback messages are saved in the database and delivered to learners instantly during the session.

Additional system components store Zoom session details, join/leave events, attendance logs, and time-based engagement metrics. These stored data points are used to generate real time analytics, including engagement trends, cluster transitions, participation rates, and risk alerts, which are displayed through dashboards accessible to instructors and administrators.

By integrating all models and module with a centralized cloud database, the system ensures efficient data flow, real time responsiveness, and scalability. This architecture enables instructors to monitor student engagement, deliver adaptive question interactions, and provide personalized feedback effectively during live video conferencing based learning sessions.

### **5.2.1 To develop a predictive learner clustering model that groups students based on real time behavioral and performance data**

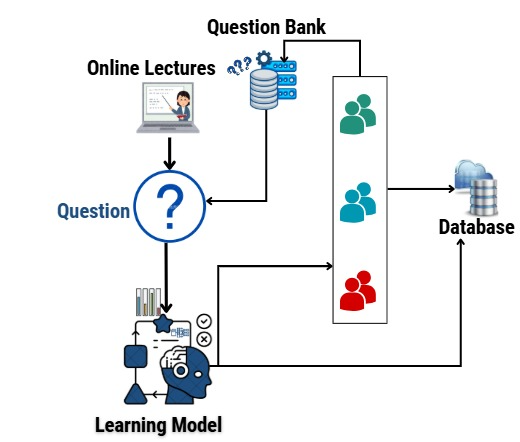
****

Figure :Predictive Clustering Model Architecture

* Data Preparation

We will collect real-time behavioral data from students participating in video conferencing-based learning sessions. This includes network quality metrics from Zoom API and student performance metrics from our integrated quiz platform.

* Data Preprocessing

Raw network metrics and quiz responses will undergo preprocessing to ensure consistency. This includes handling missing values, normalizing metrics to comparable scales using Min Max scaling, and calculating derived features. Network metrics will be aggregated over 5 second windows to capture temporal patterns. Feature vectors will be standardized to have zero mean and unit variance for optimal K-Means performance.

* Model Architecture

We may consider K-Means clustering algorithm for real time student grouping. The model receives six input features and assigns each student to one of three clusters representing engagement levels (0- Passive, 1- Moderate, 2- Active).

The K-Means algorithm operates by initializing three cluster centroids using the k-means++ initialization method to avoid poor clustering solutions. Each student's feature vector is assigned to the nearest centroid based on Euclidean distance. Centroids are then recomputed as the mean of all points assigned to that cluster. This process iterates until convergence or a maximum iteration limit is reached.

* Model Compilation

K-means applies Euclidean distance to assign learner feature vectors to clusters and iteratively updates centroids to improve clustering quality. To ensure stability and reproducibility, the model is executed with controlled initialization and convergence settings. The optimization objective is to minimize the within cluster sum of squares (WCSS), resulting in compact and well separated engagement clusters.

* Model Training

The model will be trained on historical data collected from pilot sessions. Training involves multiple runs with different initial centroid positions to find the global optimum. The elbow method will be used to validate that three clusters is optimal. The training process minimizes intra cluster variance while maximizing inter cluster separation. Converged centroids will be saved for real time inference.

* Model Evaluation

Clustering quality will be evaluated using Silhouette Score and Davies Bouldin Index. We will also compute cluster stability by measuring how consistently students are assigned to the same cluster across multiple time windows. Average silhouette width will be monitored during training.

* Model Optimization

We will optimize the model by experimenting with different distance metrics, adjusting the number of initialization runs, and implementing feature scaling techniques. We will apply Principal Component Analysis (PCA) if feature correlation exceeds 0.7. The optimal number of clusters will be validated using the elbow method and silhouette analysis.

### **5.2.2 To develop a model that provides personalized feedback based on student clusters**

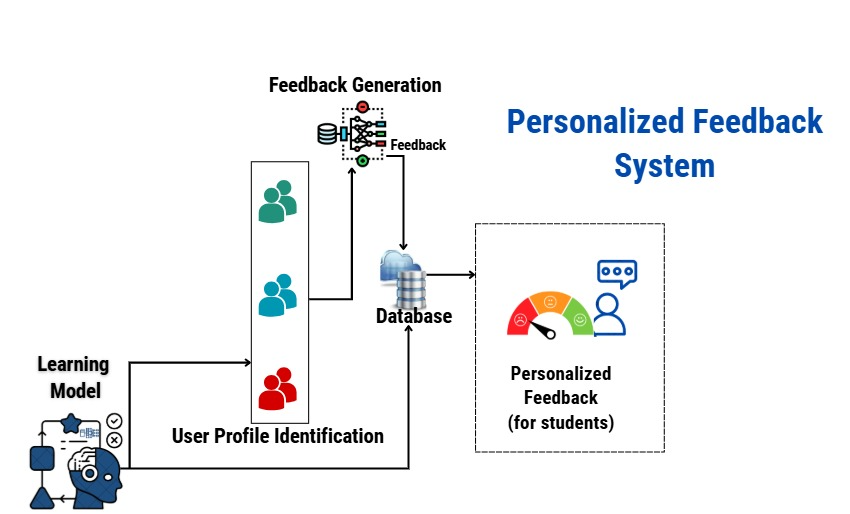


Figure :Personalized Feedback Generation Architecture

This model will be the continuity of previous model. So, at the beginning, the resume data was collected and preprocessing was performed. Finally, the clusters and other features also has been derived.

* Model Architecture

We will implement a Sequence to Sequence LSTM model for personalized feedback generation. The model follows an encoder decoder architecture with attention mechanism.

Encoder

* Input layer accepting concatenated feature vectors
* Embedding layer converting features to dense representations
* Two LSTM layers processing sequential context
* Bidirectional processing to capture both forward and backward dependencies
* Attention layer focusing on relevant context aspects

Decoder

* Initial state from encoder's final hidden state
* Two LSTM layers generating output sequence
* Attention mechanism over encoder states for each decoding step
* Time distributed dense layer with softmax activation for word prediction
* Model Compilation

The model is trained using categorical cross entropy loss, computed at each time step to evaluate prediction performance. An adaptive optimization algorithm is employed to update model parameters efficiently, along with a learning rate scheduling strategy to support stable convergence. Gradient clipping is applied to prevent unstable updates during training. To improve learning stability, teacher forcing is used during training with a controlled probability. The model is evaluated using accuracy as the primary performance metric.

* Model Training

Training will use 70% of data for training, 15% for validation, and 15% for testing. We will implement mini batch training with a specific batch size. Early stopping will be triggered when validation loss doesn't improve for certain epochs.

* Model Evaluation

The model is evaluated on a separate test set to measure its generalization to unseen student data**.** Key metrics such as accuracy, F1 score, or BLEU score are used to assess prediction quality. Evaluation ensures that the model delivers reliable and contextually appropriate personalized feedback for different student clusters under varying conditions.

* Model Optimization

We will optimize the model’s performance through hyperparameter tuning and architecture adjustments. Additionally, fine tuning the model’s parameters and refining the training process will be conducted to enhance its accuracy and efficiency in generating personalized feedback from student engagement data

## **5.3 Data Extractions, Sample Design, Test Datasets, Training Datasets**

### **5.3.1 Data Extractions**

#### We will gather real time behavioral and interaction data directly from students participating in video conferencing based learning sessions. The primary source of data will be our integrated Quiz Delivery Platform, which will be used during live Zoom classes to administer targeted questions and collect responses.

### **5.3.2 Sample Design**

#### Our sample will consist of student participants from an educational center. Participants will engage in a series of live Zoom lectures where the integrated quiz platform will be actively used.

#### 

Figure :Sample Dataset

### **5.3.3 Test Datasets and Training Datasets**

The collected dataset of student interactions and behavioral logs will be partitioned to train and test our machine learning models. The samples are separated in a ratio of 70:30 and 70% are used to train the model and 30% are used to test the model.

## **5.4 Non-functional Aspects**

* Efficiency

The research emphasizes the efficiency of machine learning and data processing techniques including K-Means, LSTM, and predictive algorithms in enabling real time learner analysis and personalized feedback during live sessions. These techniques are evaluated based on their computational requirements, inference speed, and effectiveness in processing dynamic, high frequency interaction data within a video conferencing environment.

The Predictive Learner Clustering Model is optimized for incremental updates, processing streaming engagement metrics such as response times, interaction frequency, and quiz accuracy with minimal latency to support timely cluster reassignments. The Personalized Feedback Generation Model leverages efficient encoder decoder architectures with attention mechanisms to generate relevant feedback without significant processing delays.

#### Scalability

The system architecture, integrating a web application, Zoom API, cloud hosted database (MongoDB) and machine learning services, is designed for scalability. The cloud based deployment allows the system to handle increasing numbers of concurrent users, live sessions and data volumes. The database schema and API design support horizontal scaling to manage growing records of user interactions, session logs and analytics data efficiently. This ensures consistent performance during peak usage, such as when multiple classes use the system simultaneously.

#### Reliability

The reliability of the system is critical for maintaining trust during live educational sessions. The Predictive Clustering Model incorporates robust data validation, preprocessing pipelines and fallback mechanisms to ensure consistent cluster assignments even with incomplete or noisy real time data. The integration with Zoom API includes error handling and reconnection protocols to maintain session continuity. Database operations employ transaction management and regular backups to prevent data loss. System reliability is measured through uptime metrics, error rates and the consistency of analytics outputs.

#### Usability

The system prioritizes usability through an intuitive instructor dashboard and student interface designed using user centered principles. The dashboard presents real time analytic, clusters, engagement scores, risk alerts through clear visualizations that require minimal training to interpret. For students, the quiz interface is non intrusive, with clear prompts and immediate feedback. The entire system aims to enhance, not hinder, the teaching and learning experience by providing valuable insights and interactions without adding complexity to the video conferencing workflow.

#### Maintainability

The codebase is structured following modular design patterns and coding standards (consistent naming, documentation, version control using Git). The separation of components frontend, backend, models, database allows for independent updates and troubleshooting. Comprehensive logging and monitoring tools are integrated to facilitate quick diagnosis of issues. This ensures the system can be easily maintained, updated, and extended with new features over time.

## **5.5 Validation Methods and Measurements**

#### **5.5.1 Quantitative Evaluation Methods**

Quantitative validation is used to objectively assess the performance and effectiveness of the proposed research modules and the overall system. The Predictive Learner Clustering Model is evaluated using clustering quality and stability metrics such as Silhouette Score, Davies Bouldin Index, and Calinski Harabasz Index to measure cluster separation, compactness, and variance distribution. Cluster stability is analyzed using cluster purity to ensure consistent student grouping across time windows. Performance related measures such as inference latency and model convergence rate are also monitored to verify real time feasibility during live sessions.

The Targeted Question Triggering Module is evaluated using precision, recall, and F1 score to measure the accuracy of question delivery to intended learner clusters. Engagement impact is assessed by comparing engagement scores before and after targeted interventions, along with improvements in student response rates and reductions in average response time, indicating the effectiveness of targeted questioning in increasing participation.

The Personalized Feedback Generation Model is evaluated using language quality and relevance metrics such as BLEU and ROUGE L scores to ensure grammatical correctness and semantic alignment with expert feedback templates. Feedback accuracy and generation latency are also measured to confirm that feedback is both contextually appropriate and delivered within acceptable real time constraints.

At the system level, quantitative validation includes measurements of end to end processing latency, system throughput, API response time, and database query performance. Resource utilization metrics such as CPU, memory, and network usage are monitored under varying loads to evaluate scalability and system stability during concurrent live sessions.

#### **5.5.2 Qualitative Evaluation Methods**

Qualitative validation focuses on understanding user experience, pedagogical value, and practical effectiveness of the system in real learning environments. Instructor feedback is collected through structured surveys and interviews to assess dashboard usability, clarity of analytics, and the usefulness of clustering insights and targeted interventions in instructional decision making. Student feedback is gathered to evaluate perceived engagement, relevance of personalized feedback, and the degree to which system interactions support learning without disrupting session flow.

Observational studies are conducted during live sessions to examine behavioral changes such as participation frequency, responsiveness, and interaction patterns following system interventions. Session recordings are analyzed to identify temporal engagement fluctuations and to correlate system triggered actions with observed behavioral responses. Expert reviews by educational technology specialists and UI/UX professionals are used to evaluate usability, accessibility, and cognitive load of the system interfaces.

# **References**

[1] S. Ayouni et al., *“A new ML-based approach to enhance student engagement in online environment,”* 2021.

[2] K. Subramaniya and W. Wang, *“Machine Learning Based Engagement Prediction for Online Courses,”* 2025.

[3] Laura J. Johnston et al., *“Uncovering Student Engagement Patterns in Moodle with Interpretable Machine Learning,”* 2024.

[4] P. Pathak, P. Stynes, and K. Palani, *“Clustering Techniques to Identify Low-engagement Student Levels,”* 2021.

[5] R. Mahafdah et al., *“Enhancing e-learning through AI: advanced techniques for optimizing student performance,”* 2024.

[6] P. R. Christopher Dann et al., *“Making sense of student feedback and engagement using artificial intelligence,”* 2024.

[7]M. N. Hadian, I. Murfidah, and E. S. W. Jannah*, “Analysis of the Impact of Artificial Intelligence and Gamification in Enhancing Student Engagement in Virtual Learning Environments,” 2024*.

# **Approval**

Signature of the team members:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Registration Number | Index Number | Signature |
| Shawmica S | ICT/2021/119 | 5274 |  |
| Keran Shama S | ICT/2021/121 | 5276 |  |
| Prashanthy K | ICT/2021/056 | 5217 |  |
| A k Wafry Ahamed | ICT/2021/003 | 5165 |  |
| V Arunpragash | ICT/2021/011 | 5173 |  |

Date: ………………………

Recommendation of the supervisor(s)

Name: ………………………

Department/Organization:………………………

Signature:………………….