BSCS-S21-016

03-134181-025 QAIS BIN AZAM

03-134181-033 ZARYAB ILYAS

Advanced Learning and Assessment System

In partial fulfilment of the requirements for the degree of

Bachelor of Science in Computer Science

Supervisor: Zupash Awais

Department of Computer Sciences

Bahria University, Lahore Campus

January 2022

C e r t i f i c a t e

We accept the work contained in the report titled

"Advanced Learning and Assessment System"

written by

QAIS BIN AZAM

ZARYAB ILYAS

as a confirmation to the required standard for the partial fulfilment of the degree of Bachelor of Science in Computer Science.

Approved by:

Supervisor: Zupash Awais

_____ (Signature)

January 26, 2022

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-134181-025	QAIS BIN AZAM	
03-134181-033	ZARYAB ILYAS	

Date : January 26, 2022

Specially dedicated to
my beloved mother and father
(QAIS BIN AZAM)
my beloved mother and father
(ZARYAB ILYAS)

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express our gratitude to our supervisor, Miss ZUPASH AWAIS for her invaluable advice, guidance and her enormous patience throughout the development of the project.

In addition, We would also like to express our gratitude to our loving parents and friends who had helped and given us encouragement.

QAIS BIN AZAM
ZARYAB ILYAS

Advanced Learning and Assessment System

ABSTRACT

World Health Organization has declared Covid-19 as a pandemic that has posed a contemporary threat to humanity. This pandemic has successfully forced the global shutdown of several activities, including educational activities, and this has resulted in tremendous crisis-response migration of universities with online learning serving as the educational platform. Bahria University's existing LMS is failed to provide a solid online assessment mechanism. Considering the flaws of online assessments in already existing learning management system we have designed a system that will make online assessments more credible and transparent.

This web application is developed by using ASP.NET Core MVC 5.0, Bootstrap 4 along with ASP.NET Core Razor pages, Entity Framework Core and MS SQL Server. Using the Agile Model for our web app we have applied the Feature Driven Development (FDD) method. Advanced Learning and Assessment primarily focuses on improving the assessment mechanism only. ALAS has a feature-packed assessment system where teachers can check plagiarism inside the system. There is a built-in text editor to solve exams, a facial recognition system and live video call to ensure either the person who is giving the exam is the university's student or not.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

ALAS	Advance Learning And Assessment System
ASP	Active Server Pages
LMS	Learning Management System
COVID-19	Coronavirus Disease Of 2019
FFD	Feature Driven Development
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
LINQ	Language Integrated Query
ORM	Object-Relational Mapping
SMTP	Simple Mail Transfer Protocol

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

The existing LMS was designed in a way that was effective before COVID-19. After the world is affected by COVID-19 the examination of all educational institutes was shifted online which gave students a chance to cheat easily during the examination. Our proposed system focuses on making the assessment system better and transparent. This assessment system (ALAS) is completely different from the existing Bahria University's LMS. Our main goal is to improve our assessment to this level that cheating could be minimized.

Advanced learning and assessment system has a built-in plagiarism checker. A built-in HTML text editor to solve paper online. ALAS will take the camera permission from the user at the start of the exam and only then the user would be allowed to enter an exam if he/she allows the permission, and the face is recognized by the system. The exam interface contains a text editor and a live video calling window through which students can join the exam room. If students don’t join the live video call, they will be considered to fail even if they submit the exam. A count down timer will be shown during the exam and the content on the HTML text editor will be submitted automatically on time out.

1.2 Problem Statements

The World Health Organization has declared Covid-19 as a pandemic that has posed a contemporary threat to humanity. This pandemic has successfully forced the global shutdown of several activities, including educational activities, and this has resulted in tremendous crisis-response migration of universities with online learning serving as the educational platform. [1] Already existing LMS is failed to provide a solid online assessment mechanism. Considering the flaws of online assessments in already existing learning management systems we have designed a system that will make online assessments more credible and transparent.

1.3 Aims and Objectives

The objectives of our project are shown as follows:

i) To host and track online learning. Providing a virtual hub where learners can access resources remotely.
ii) To manage content and store learners' data.
iii) To create a credible and reliable online assessment mechanism with real-time face recognition and built-in plagiarism detection.
iv) Create online exams and evaluate them within the system.

1.4 Scope of Project

Faculty, Students and Admin have their dashboard. The system has a plagiarism detector. There will be a live video call and real-time face recognition before an exam. Courses, Attendance, Accounts, Exam seating plan, and other things related to administrative content management.

CHAPTER 2

Software Requirement Specifications

2.1 User Classes and Characteristics

The users can be divided into three classes.

- Administrator
- Student
- Teacher

2.1.1 Administrator

The administrator will perform the following activities.

- Sign in / Sign out
- Update Profile
- Change Password
- Manage Users and Roles

2.1.2 Student

Students will perform the following activities

- Registration
- Sign in / Sign out

- Update profile
- Change password
- Give quiz/paper
- View/download course content

2.1.3 Teacher

The teacher will perform the following activities.

- Registration
- Sign in / Sign out
- Update profile
- Change password
- Create assignment /quiz / paper
- Manage course content
- Check Plagiarism / Publish Results

2.2 Operating Environment

The operating environment for the Advance Learning and Assessment system is as listed below.

- MS SQL Server (DB)
- ASP.NET Core Runtime (To Run web app)
- Operating system: Cross-Platform (Windows, Linux, Mac).
- Nodejs (to run faceapi.js)
- Kestrel server/ IIS HTTP

2.3 Design and Implementation Constraints

The solution should be developed using agile methodology. The solution should be cross-platform and work on all popular operating systems. The front end of the application should be user friendly with a responsive design.

2.4 Assumptions and Dependencies

The user needs a web browser and an internet connection to use the system as it is a web application. A slow internet connection or outdated browser may lead to some problems while using the application.

2.5 Use Case Diagrams

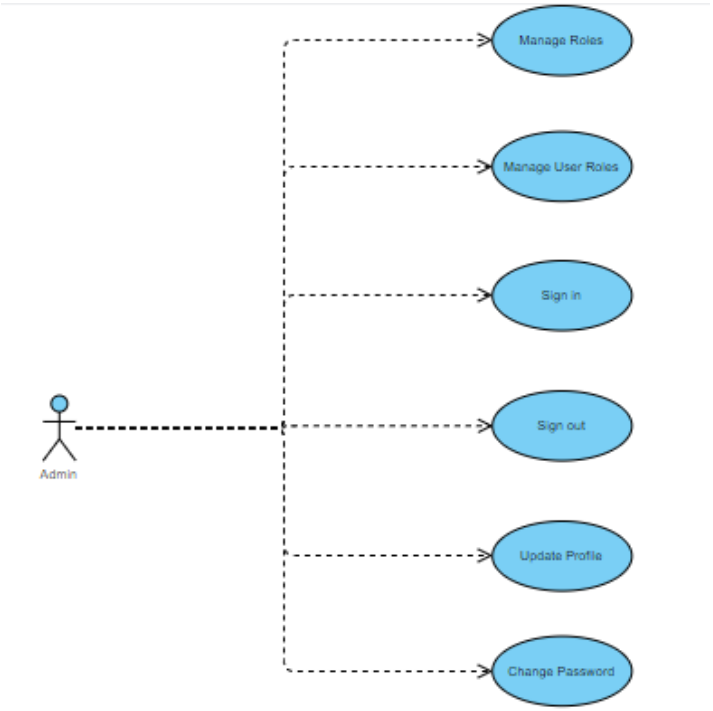

Figure 2. 1 Admin Use Case Diagram

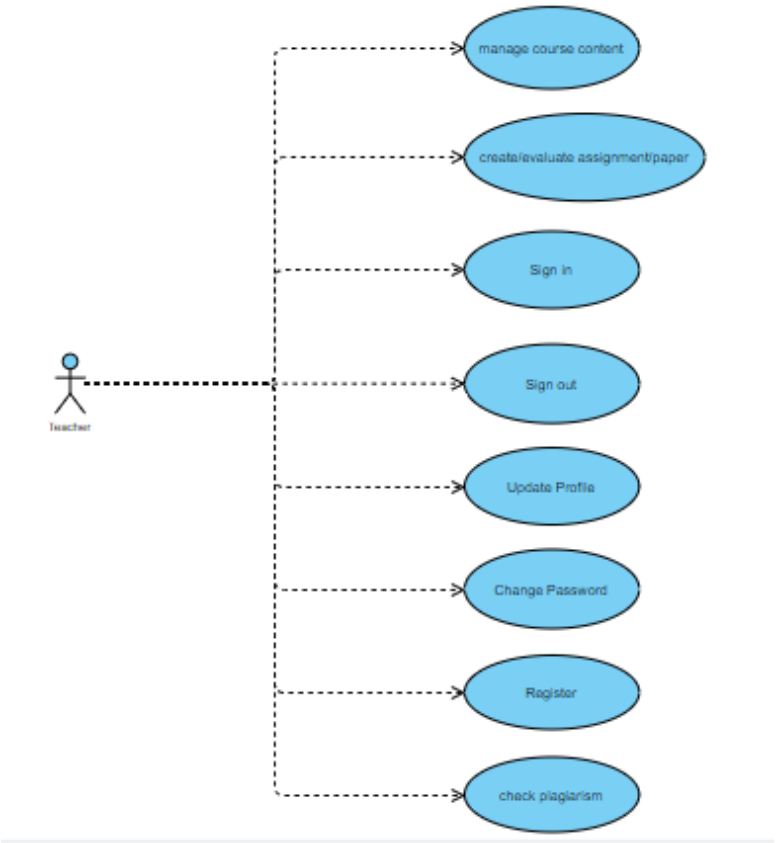

Figure 2. 2 Teacher Use Case Diagram

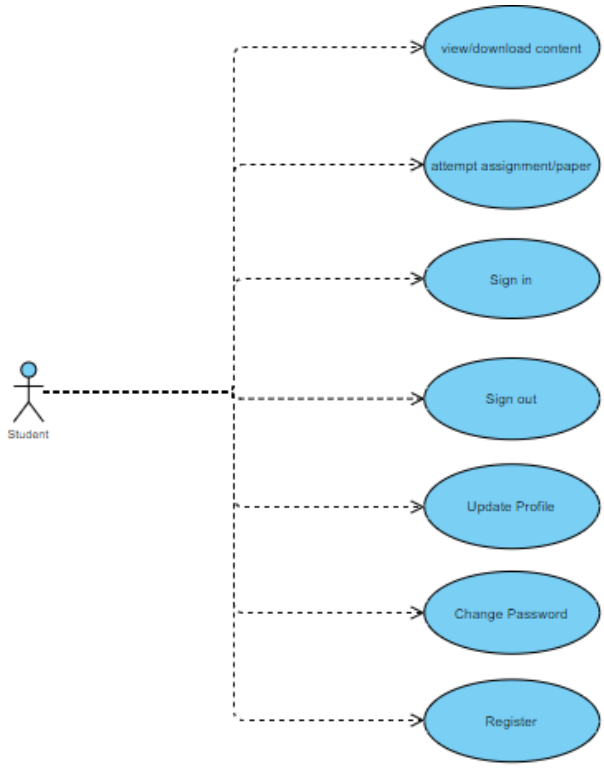

Figure 2. 3 Student Use Case Diagram

Table 2. 1 Admin Sign in U1

	Name	**Sign-in**
1.	**Use-Case ID**	U1
2.	**Objective**	Admin will sign in with the login credentials. (Credentials will be given)
3.	**Priority**	High
4.	**Source**	Qais Bin Azam (Developer)
5.	**Actors**	Administrator
6.	**Flow of Events**	1. Admin open the Browser 2. Enter admin dashboard URL 3. Enter Username and Password 4. Click on Sign-in Button to access the admin panel
6.1	**Basic Flow**	After successful sign-in Administrator will go to U2, U3, U4, U5
6.2	**Alternate Flow(s)**	No alternate flow, Administrator must sign-in to proceed further
6.3	**Exception Flow(s)**	1. Invalid Username 2. Invalid Password
7.	**Includes**	No other use case
8.	**Preconditions**	The administrator must be connected to web service via an internet connection (e.g., LAN, WAN)
9.	**Postconditions**	The administrator will use the Administrator panel
10.	**Notes/Issues**	If the Administrator will sign-in with the right credentials no problem will occur

Table 2. 2 Admin Update Profile U2

	Name	**Update Profile / Change Password**
1.	**Use-Case ID**	U2
2.	**Objective**	In this use case, Admin can update his/her profile and can change the password.
3.	**Priority**	Medium
4.	**Source**	Admin is the main source of this use case. Admin cares most about this functionality.
5.	**Actors**	Admin
6.	**Flow of Events**	1. Login to this Application 2. Select Change Password / Update Profile Option 3. Enter old password 4. Enter new password 5. Re-Enter new password 6. Click the save changes button
6.1	**Basic Flow**	After changing the password Admin will go to U3, U4.
6.2.	**Alternative Flow(s)**	If the Admin does not want to change credentials, he/she will go to U5.
6.3.	**Exception Flow(s)**	1. Invalid old password 2. Miss matches the new password
7.	**Includes**	No
8.	**Preconditions**	Admin must be logged in to perform U2.
9.	**Postconditions**	Credentials are successfully changed.
10.	**Notes/Issues**	No

Table 2. 3 Admin Manage Roles U3

	Name	**Manage Roles**
1.	**Use-Case ID**	U3
2.	**Objective**	In this use case, Admin will manage roles.
3.	**Priority**	High.
4.	**Source**	Admin is the main source of this use case. Admin cares most about this functionality.
5.	**Actors**	Admin
6.	**Flow of Events**	1. Must fulfil U1 2. Click roles tab 3. Manage anything regarding roles
6.1	**Basic Flow**	Admin will be able to manage roles.
6.2.	**Alternative Flow(s)**	If the Admin does not want to manage roles, then, he/she may go to U2, U4, U5.
6.3.	**Exception Flow(s)**	No.
7.	**Includes**	No Includes.
8.	**Preconditions**	Admin must be logged in.
9.	**Postconditions**	Roles will be updated.
10.	**Notes/Issues**	No Notes.

Table 2. 4 Admin Manage User Roles U4

	Name	**Manage User Roles**
1.	**Use-Case ID**	U4
2.	**Objective**	In this use case, Admin will manage user roles.
3.	**Priority**	High.
4.	**Source**	Admin is the main source of this use case. Admin cares most about this functionality.
5.	**Actors**	Admin
6.	**Flow of Events**	1. Must fulfil U1 2. Click the user tab 3. Select manage from the dropdown 4. Assign roles to users
6.1	**Basic Flow**	Admin will be able to manage user roles.
6.2.	**Alternative Flow(s)**	If the Admin does not want to manage roles, then, he/she may go to U2, U3 and U5.
6.3.	**Exception Flow(s)**	No.
7.	**Includes**	No Includes.
8.	**Preconditions**	Admin must be logged in.
9.	**Postconditions**	User roles will be updated.
10.	**Notes/Issues**	No Notes.

Table 2. 5 Admin Sign out U5

	Name	**Sign out**
1.	**Use-Case ID**	U5
2.	**Objective**	In this use case, Admin will be able to log out.
3.	**Priority**	High
4.	**Source**	Admin cares most about this functionality and is the main source of this use case.
5.	**Actors**	Admin
6.	**Flow of Events**	Must be fulfilled U1.
6.1	**Basic Flow**	Logout from Admin panel.
6.2.	**Alternative Flow(s)**	No.
6.3.	**Exception Flow(s)**	No.
7.	**Includes**	No Includes
8.	**Preconditions**	Admin must be logged in to this application.
9.	**Postconditions**	Logout from Admin panel.
10.	**Notes/Issues**	No Notes.

Table 2. 6 Student Sign-in U6

	Name	**Sign-in**
1.	**Use-Case ID**	U6
2.	**Objective**	In this use case, Students will be able to sign-in
3.	**Priority**	High
4.	**Source**	The student cares most about this functionality and is the main source of this use case
5.	**Actors**	Student
6.	**Flow of Events**	1. Users open the web browser and visit the app URL 2. Select sign-in 3. Enter Username and Password 4. Click on Sign-in Button to sign-in
6.1	**Basic Flow**	After successful sign-in students will go to U8, U9, U10. U11.
6.2	**Alternate Flow(s)**	No alternate flow, Students must sign-in to proceed Further
6.3	**Exception Flow(s)**	1. Invalid Username 2. Invalid Password
7.	**Includes**	No other use case
8.	**Preconditions**	Students must be connected via an internet connection.
9.	**Postconditions**	Students will use the Student panel
10.	**Notes/Issues**	If students will sign in with the right credentials no problem will occur

Table 2. 7 Student Registration U7

	Name	**Registration**
1.	**Use-Case ID**	U7
2.	**Objective**	In this use case, the student will be able to register
3.	**Priority**	High
4.	**Source**	The student cares most about this functionality and is the main source of this use case.
5.	**Actors**	Student
6.	**Flow of Events**	1. User will open the web-app 2. Enter registration 3. Fill out details 4. Click on Register Button to send details to the Administrator
6.1	**Basic Flow**	After successful registration students will go to U8, U9, U10, U11.
6.2	**Alternate Flow(s)**	If the student is already registered, then go to U9.
6.3	**Exception Flow(s)**	Invalid Details
7.	**Includes**	No other use case
8.	**Preconditions**	Students must be connected to an internet connection
9.	**Postconditions**	Students will be registered and will go to the Sign-in page
10.	**Notes/Issues**	If the Student does register with the right credentials no problem will occur

Table 2. 8 Student Update Profile U8

	Name	**Update Profile**
1.	**Use-Case ID**	U8
2.	**Objective**	In this use case, the Student will be able to update profile/change password.
3.	**Priority**	Medium
4.	**Source**	The student cares most about this functionality and is the main source of this use case
5.	**Actors**	Student
6.	**Flow of Events**	1. Must fulfil U6 2. Do the desired changes 3. Save changes
6.1	**Basic Flow**	Students will update their profiles.
6.2	**Alternate Flow(s)**	If the student does not want any changes, they will go to U9, U10, U11
6.3	**Exception Flow(s)**	No exceptions
7.	**Includes**	No other use case
8.	**Preconditions**	Students must be signed in to perform U8
9.	**Postconditions**	Changes saved
10.	**Notes/Issues**	No notes

Table 2. 9 Student Attempt Quiz U9

	Name	**Give Quiz / Paper**
1.	**Use-Case ID**	U9
2.	**Objective**	In this use case, Students will be able to give quizzes or papers
3.	**Priority**	High
4.	**Source**	The student cares most about this functionality and is the main source of this use case
5.	**Actors**	Student
6.	**Flow of Events**	1. Must fulfil U6 2. Go to the Quiz / Paper section 3. Attempt quiz
6.1	**Basic Flow**	Student will give exam / quiz
6.2	**Alternate Flow(s)**	If the Student does not want to give exam, then will go to U8, U10, U11
6.3	**Exception Flow(s)**	No exceptions
7.	**Includes**	No other use case
8.	**Preconditions**	Students must be signed in to perform U9
9.	**Postconditions**	The quiz will be submitted
10.	**Notes/Issues**	No notes

Table 2. 10 Student View/Download Course Content U10

Name	**View / Download course content**
Use-Case ID	U10
Objective	In this use case, Students will be able to view/download the course content.
Priority	High
Source	The student cares most about this functionality and is the main source of this use case
Actors	Student
Flow of Events	1. Must fulfil U6 2. Go to lecture notes tab 3. View / download lecture notes
Basic Flow	Students will view Download lecture notes
Alternate Flow(s)	If the student does not want to view/download notes, then will go to U8, U9, U11
Exception Flow(s)	No exceptions
Includes	No other use case
Preconditions	Students must be signed in to perform U11
Postconditions	Notes will be downloaded
Notes/Issues	No notes

Table 2. 11 Student Sign out U11

	Name	**Sign out**
1.	**Use-Case ID**	U11
2.	**Objective**	In this use case, the Student will be able to log out.
3.	**Priority**	High
4.	**Source**	The student cares most about this functionality and is the main source of this use case.
5.	**Actors**	Student
6.	**Flow of Events**	Must be fulfilled U6 and click the sign-out button.
6.1	**Basic Flow**	Logout from the Student panel.
6.2.	**Alternative Flow(s)**	No.
6.3.	**Exception Flow(s)**	No.
7.	**Includes**	No Includes
8.	**Preconditions**	Students must be logged in to this application.
9.	**Postconditions**	Logout from the Student panel.
10.	**Notes/Issues**	No Notes.

Table 2. 12 Teacher Sign in U12

	Name	**Sign-in**
1.	**Use-Case ID**	U12
2.	**Objective**	The teacher will sign in with the credentials.
3.	**Priority**	High
4.	**Source**	The teacher cares most about this functionality and is the main source of this use case.
5.	**Actors**	Teacher
6.	**Flow of Events**	1. The teacher will open the Browser 2. Go to the login page 3. Enter Username and Password 4. Click on Sign-in Button
6.1	**Basic Flow**	After successful sign-in teacher will go to U14, U15, U16, U17
6.2	**Alternate Flow(s)**	No alternate flow, Teacher must sign-in to proceed further
6.3	**Exception Flow(s)**	1. Invalid Username 2. Invalid Password
7.	**Includes**	No other use case
8.	**Preconditions**	The teacher must be connected to web service via an internet connection (e.g., LAN, WAN)
9.	**Postconditions**	The teacher will use teachers' panel
10.	**Notes/Issues**	If the teacher will sign-in with the right credentials no problem will occur

Table 2. 13 Teacher Registration U13

	Name	**Registration**
1.	**Use-Case ID**	U13
2.	**Objective**	In this case, the teacher will be able to register
3.	**Priority**	High
4.	**Source**	The teacher cares most about this functionality and is the main source of this use case.
5.	**Actors**	Teacher
6.	**Flow of Events**	1. User will open the web-app 2. Enter registration 3. Fill out details 4. Click on Register Button to send details to the Administrator.
6.1	**Basic Flow**	After successful registration students will go to U14, U15, U16, U17.
6.2	**Alternate Flow(s)**	If the Teacher is already registered, then go to U12.
6.3	**Exception Flow(s)**	Invalid Details
7.	**Includes**	No other use case
8.	**Preconditions**	The teacher must be connected to an internet connection
9.	**Postconditions**	The teacher will be registered and will go to the Sign-in page
10.	**Notes/Issues**	If the Teacher does register with the right credentials no problem will occur

Table 2. 14 Teacher Update Profile U14

	Name	**Update Profile**
1.	**Use-Case ID**	U14
2.	**Objective**	In this case, the Teacher will be able to update the profile/change password.
3.	**Priority**	Medium
4.	**Source**	The teacher cares most about this functionality and is the main source of this use case
5.	**Actors**	Teacher
6.	**Flow of Events**	1. Must fulfil U12 2. Do the desired changes 3. Save changes
6.1	**Basic Flow**	The teacher will update the profile.
6.2	**Alternate Flow(s)**	If the student does not want any changes, they will go to U15, U16, U17
6.3	**Exception Flow(s)**	No exceptions
7.	**Includes**	No other use case
8.	**Preconditions**	The teacher must be signed in to perform U14
9.	**Postconditions**	Changes saved
10.	**Notes/Issues**	No notes

Table 2. 15 Teacher Create Quiz/Exam U15

	Name	**Take Quiz / Paper**
1.	**Use-Case ID**	U15
2.	**Objective**	In this case, the Teacher will be able to take quizzes or papers
3.	**Priority**	High
4.	**Source**	The teacher cares most about this functionality and is the main source of this use case
5.	**Actors**	Teacher
6.	**Flow of Events**	1. Must fulfil U12 2. Go to the Quiz / Paper section 3. Create meeting for quiz 4. Create quiz/paper
6.1	**Basic Flow**	Teacher will take exam / quiz
6.2	**Alternate Flow(s)**	If the Teacher does not take exam, then will go to U14, U16, U17
6.3	**Exception Flow(s)**	No exceptions
7.	**Includes**	No other use case
8.	**Preconditions**	The teacher must be signed in to perform U15
9.	**Postconditions**	The quiz will be created for students
10.	**Notes/Issues**	No notes

Table 2. 16 Teacher View/Upload Course Content U16

	Name	**View / Upload course content**
1.	**Use-Case ID**	U16
2.	**Objective**	In this case, the Teacher will be able to upload course content.
3.	**Priority**	High
4.	**Source**	The teacher cares most about this functionality and is the main source of this use case
5.	**Actors**	Teacher
6.	**Flow of Events**	1. Must fulfill U12 2. Go to the lecture notes tab 3. Upload lecture notes
6.1	**Basic Flow**	Teacher will upload lecture notes
6.2	**Alternate Flow(s)**	If the teacher does not want to view / upload notes, then will go to U14, U15, U17
6.3	**Exception Flow(s)**	No exceptions
7.	**Includes**	No other use case
8.	**Preconditions**	The teacher must be signed in to perform U16
9.	**Postconditions**	Notes will be uploaded
10.	**Notes/Issues**	No notes

Table 2. 17 Teacher Check plagiarism U17

	Name	**Check plagiarism**
1.	**Use-Case ID**	U17
2.	**Objective**	In this case, the Teacher will be able to check plagiarism.
3.	**Priority**	Medium
4.	**Source**	The teacher cares most about this functionality and is the main source of this use case
5.	**Actors**	Teacher
6.	**Flow of Events**	1. Must fulfil U12 2. Go to the plagiarism tab 3. Check plagiarism
6.1	**Basic Flow**	Teacher will check plagiarism
6.2	**Alternate Flow(s)**	If the teacher does not want to check plagiarism, then will go to U14, U15, U16
6.3	**Exception Flow(s)**	No exceptions
7.	**Includes**	No other use case
8.	**Preconditions**	The teacher must be signed in to perform U17
9.	**Postconditions**	Plagiarism results will be shown
10.	**Notes/Issues**	No notes

Table 2. 18 Teacher Sign out U18

	Name	**Sign out**
1.	**Use-Case ID**	U18
2.	**Objective**	In this case, the Teacher will be able to log out.
3.	**Priority**	High
4.	**Source**	The teacher cares most about this functionality and is the main source of this use case.
5.	**Actors**	Teacher
6.	**Flow of Events**	Must be fulfilled U12 and click sign out button.
6.1	**Basic Flow**	Logout from the Teacher panel.
6.2.	**Alternative Flow(s)**	No.
6.3.	**Exception Flow(s)**	No.
7.	**Includes**	No Includes
8.	**Preconditions**	The teacher must be logged in to this application.
9.	**Postconditions**	Logout from Teacher’s panel.
10.	**Notes/Issues**	No Notes.

2.6 Other Non-Functional Requirements

Non-functional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs. Following are some non-functional requirements of the system.

2.6.1 Performance Requirements

The Application should be reliable, robust, and secure and should meet usability as well as availability, safety, efficiency, and flexibility standards.

2.6.2 Safety Requirements

The Application should allow only authorized users to access the system. There should be a proper account registration, sign in, sign out, email verification, and reset password system. The system should prevent SQL injections.

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Design

3.1.1 Domain Model

The domain model is a representation of meaningful real-world concepts pertinent to the domain that need to be modeled in software. The concepts include the data involved in the business and the rules the business uses concerning that data. A domain model leverages the natural language of the domain. Below Fig. shows the domain model of the Advanced Learning and Assessment System.

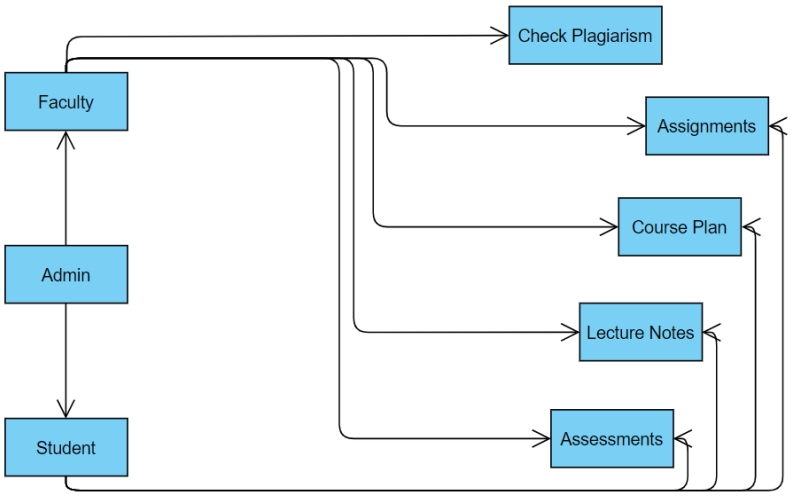

Figure 3. 1 Domain Model Diagram

3.1.2 Sequence Diagram

A sequence diagram, in the context of UML, represents object collaboration and is used to define event sequences between objects for a certain outcome. A sequence diagram is an essential component used in processes related to analysis, design, and documentation. A sequence diagram is also known as a timing diagram, event diagram, and event scenario. The main purpose of a sequence diagram is to define event sequences that result in some desired outcome. The focus is less on messages themselves and more on the order in which messages occur; nevertheless, most sequence diagrams will communicate what messages are sent between a system's objects as well as the order in which they occur.

The diagram conveys this information along the horizontal and vertical dimensions: the vertical dimension shows, top-down, and the time sequence of messages/calls as they occur, and the horizontal dimension shows, left to right, and the object instances that the messages are sent to. The following figures show the Sequence Diagrams of ALAS.

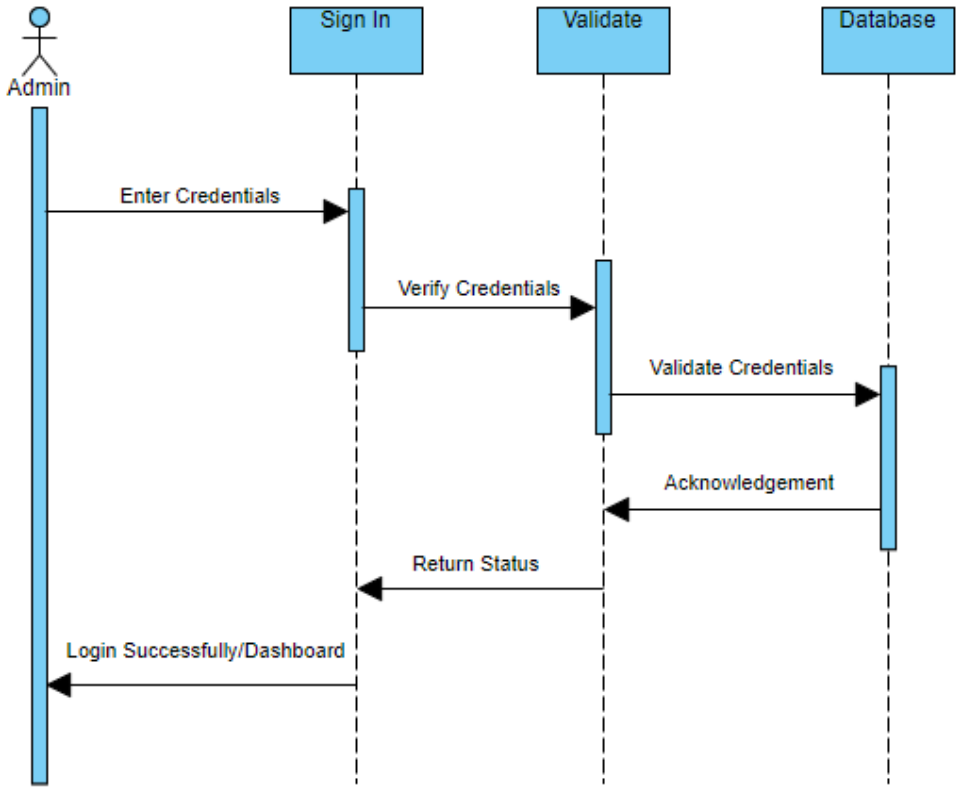

Figure 3. 2 Admin Sign in Sequence Diagram

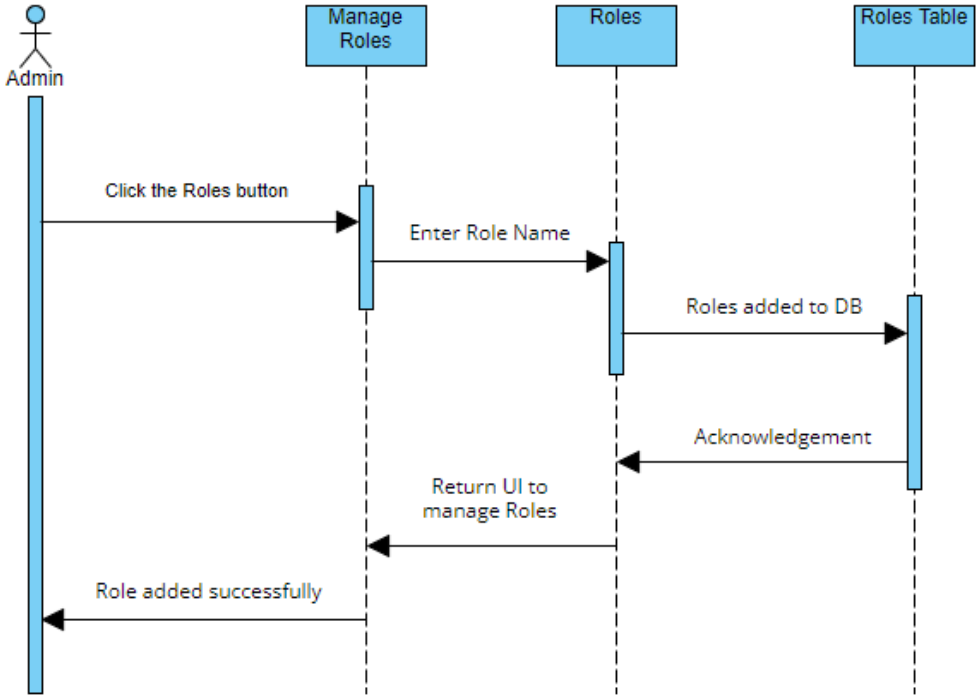

Figure 3. 3 Admin Manage Roles Sequence Diagram

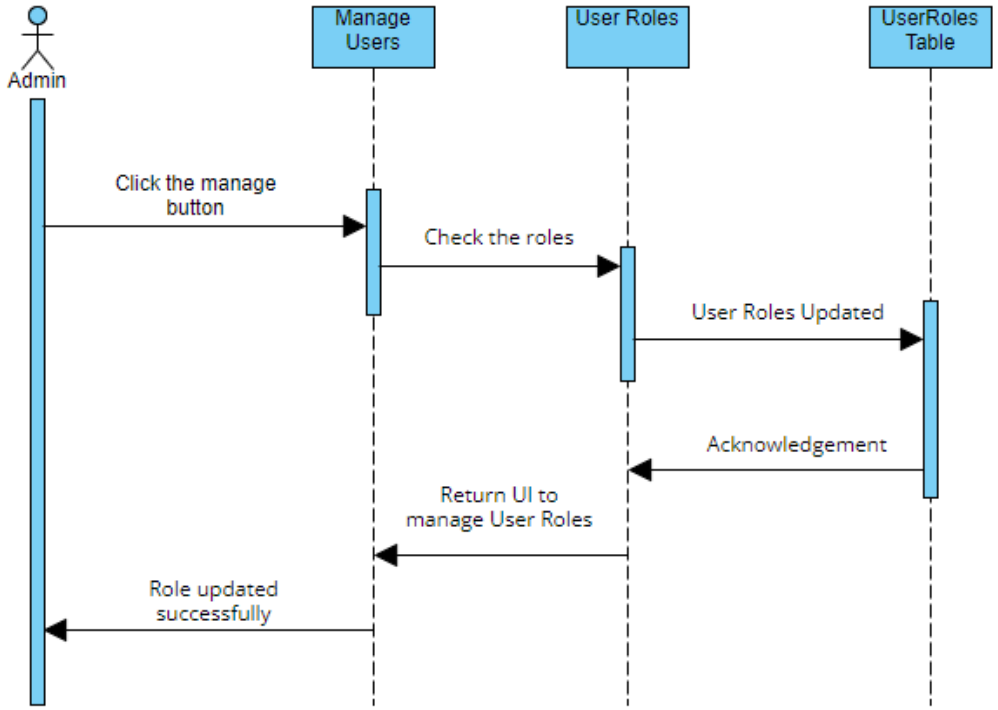

Figure 3. 4 Admin Manage User Roles Sequence Diagram

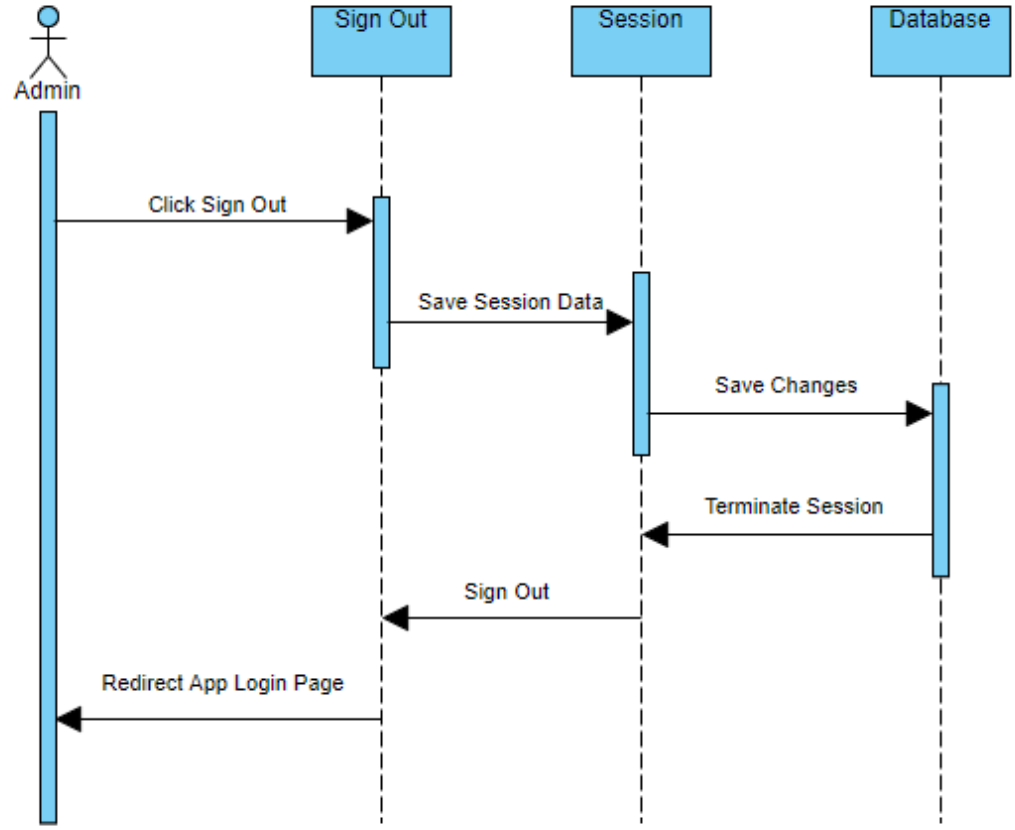

Figure 3. 5 Admin Sign out Sequence Diagram

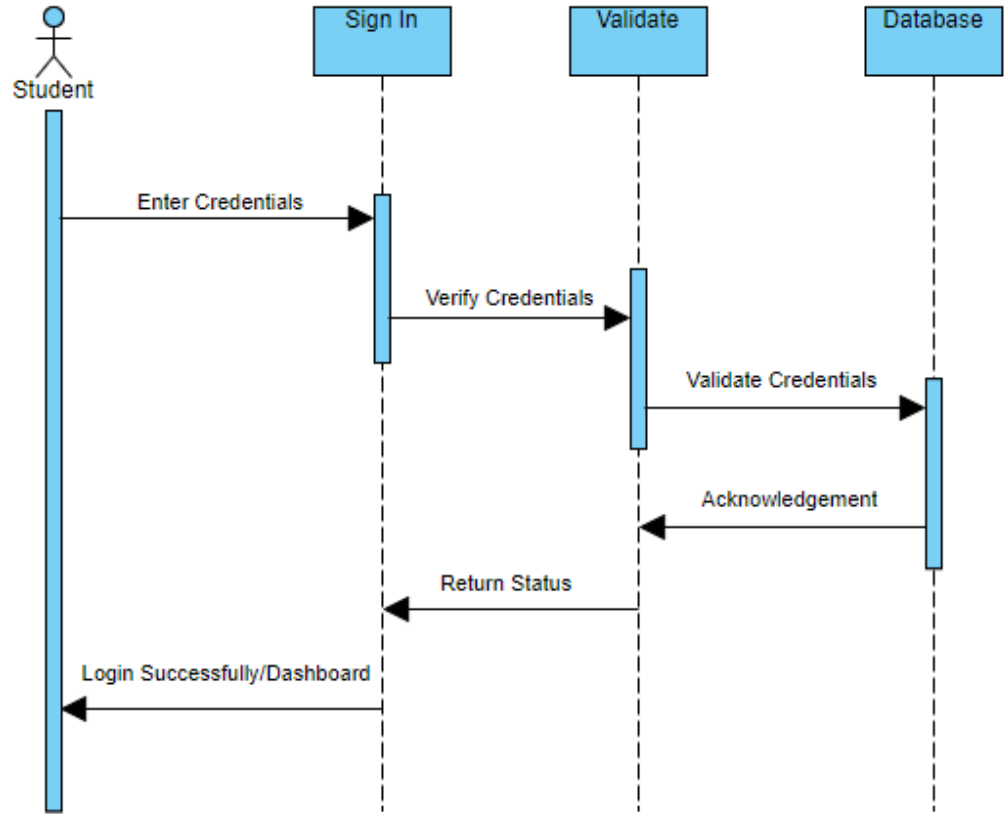

Figure 3. 6 Student Sign in Sequence Diagram

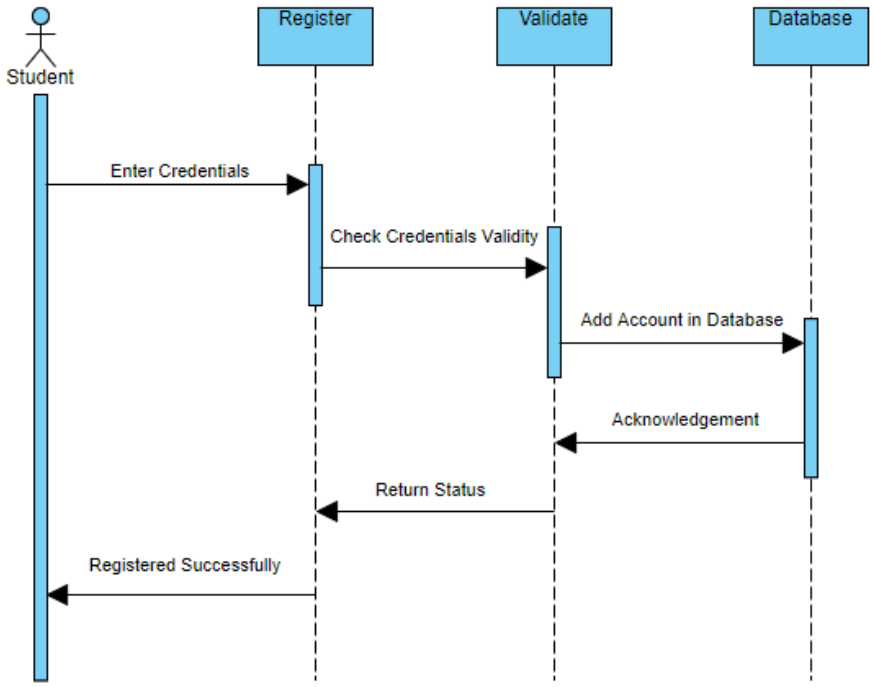

Figure 3. 7 Student Register Sequence Diagram

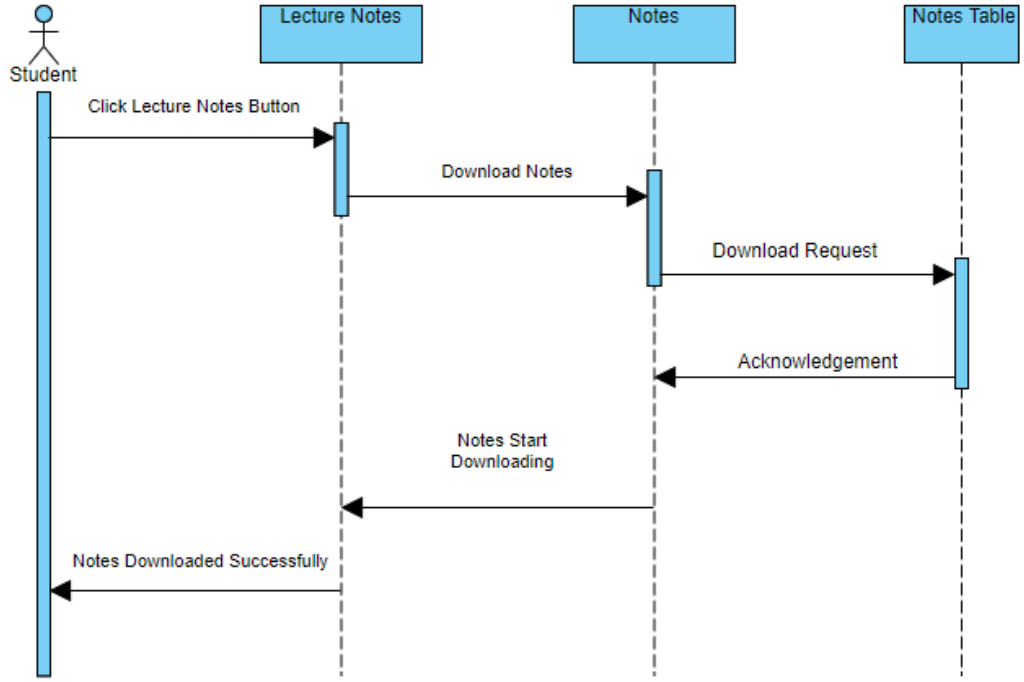

Figure 3. 8 Student Lecture Notes Sequence Diagram

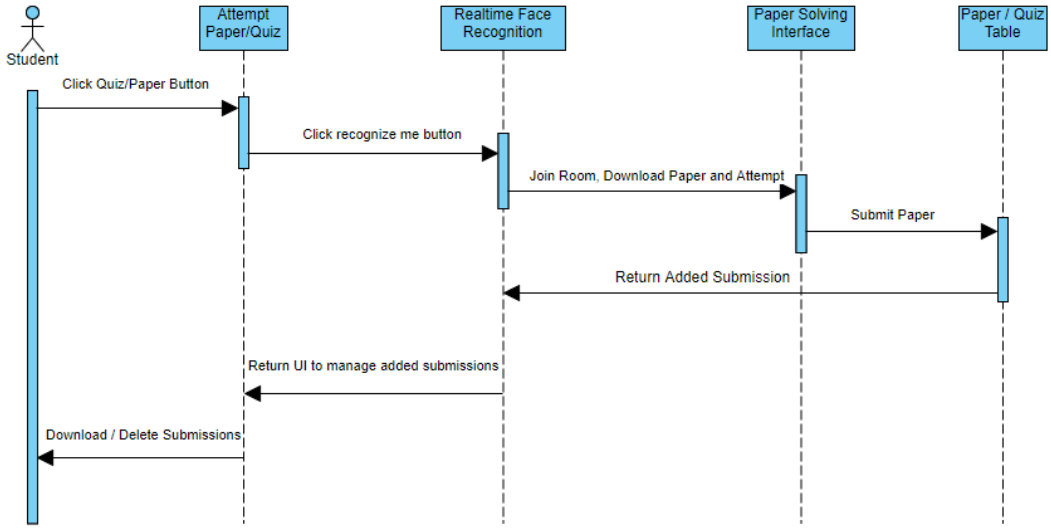

Figure 3. 9 Student Quiz Sequence Diagram

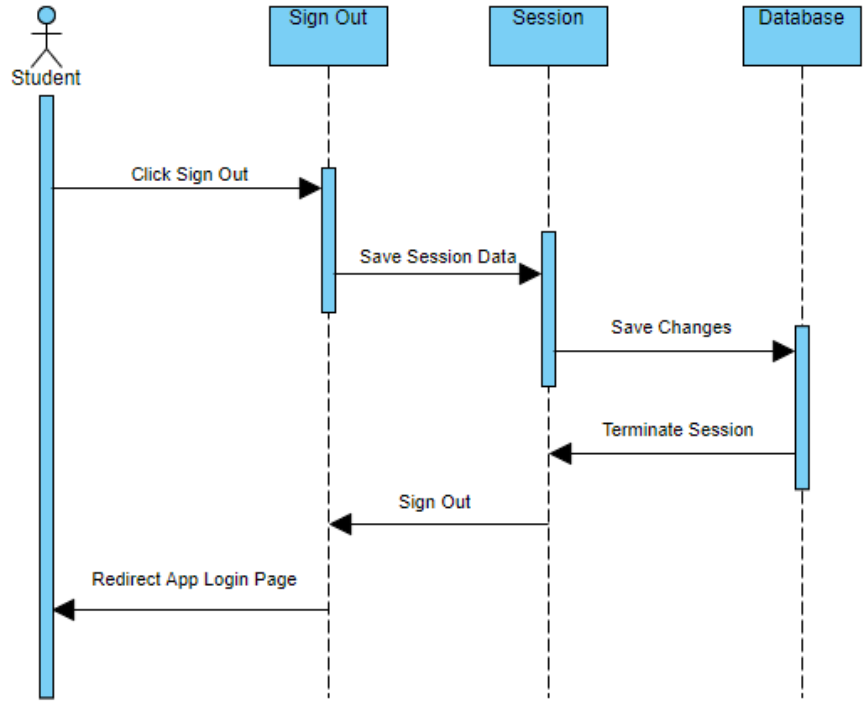

Figure 3. 10 Student Sign out Sequence Diagram

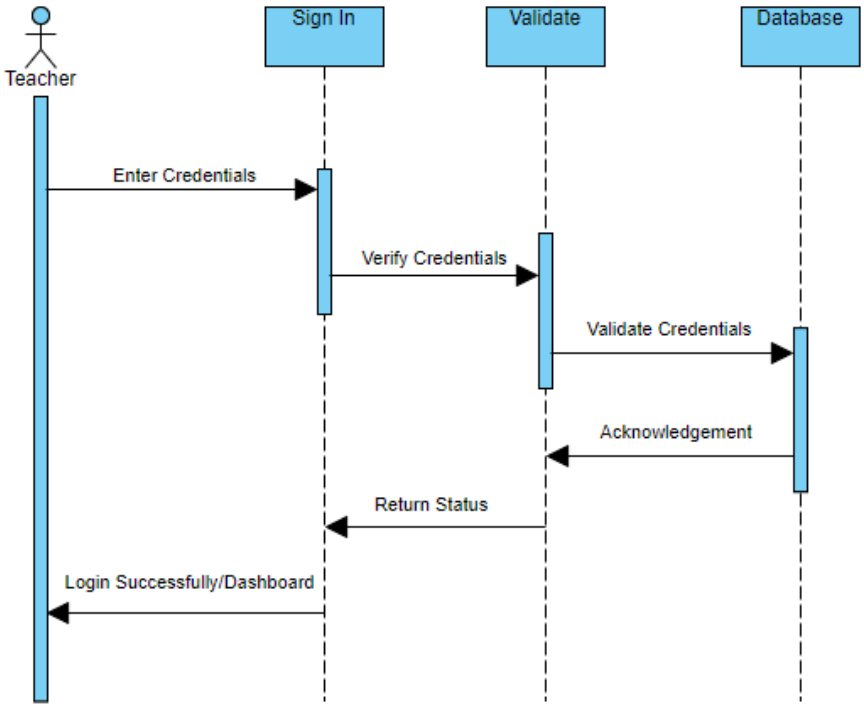

Figure 3. 11 Teacher Sign in Sequence Diagram

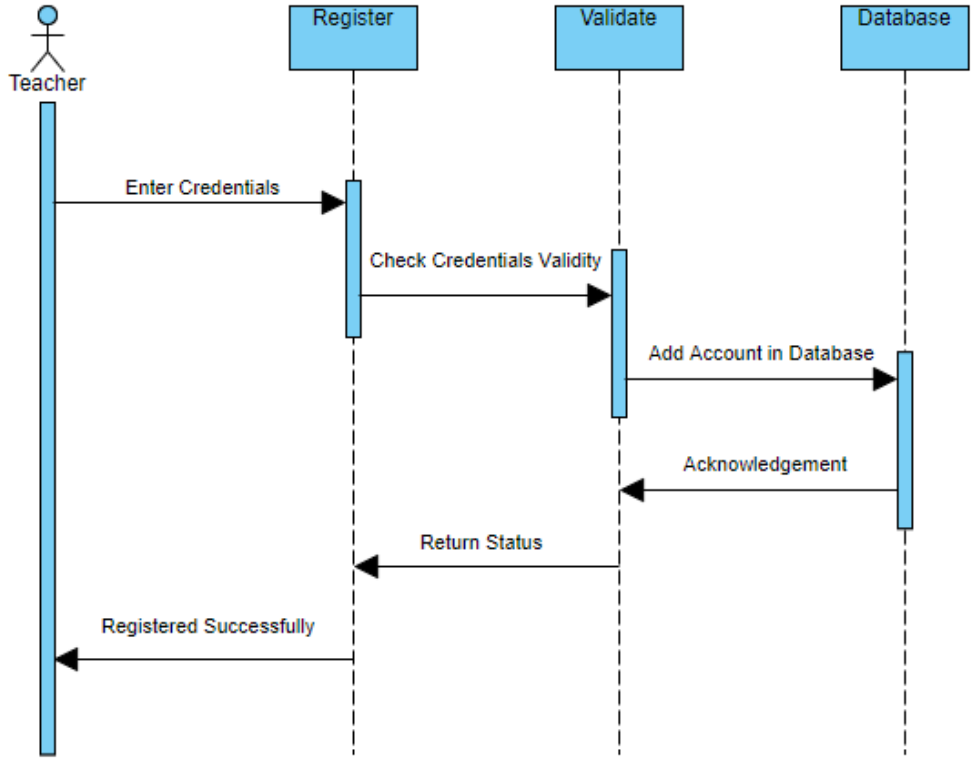

Figure 3. 12 Teacher Register Sequence Diagram

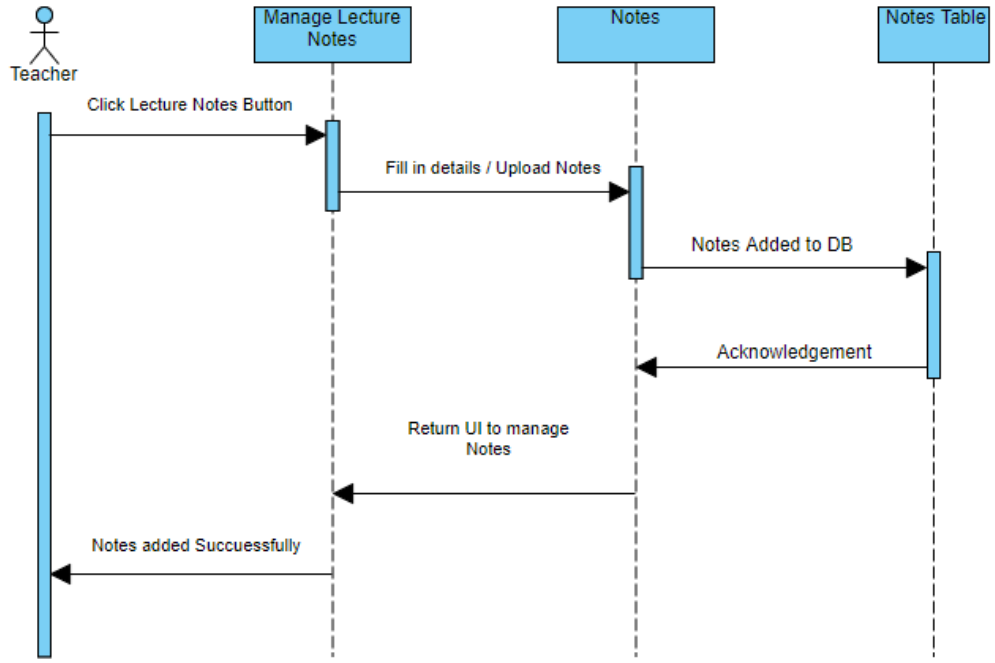

Figure 3. 13 Teacher Lecture Notes Sequence Diagram

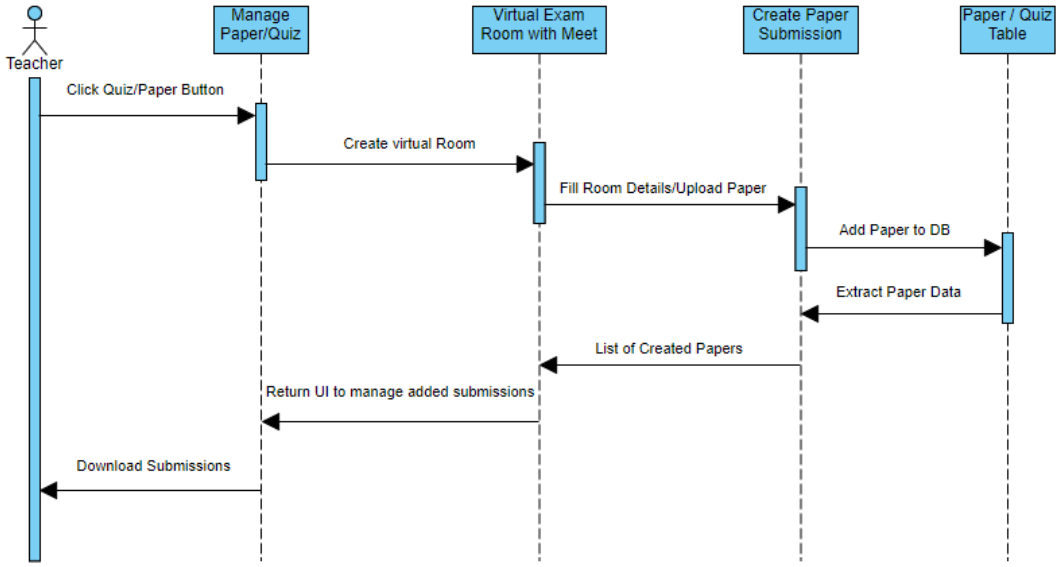

Figure 3. 14 Teacher Quiz Sequence Diagram

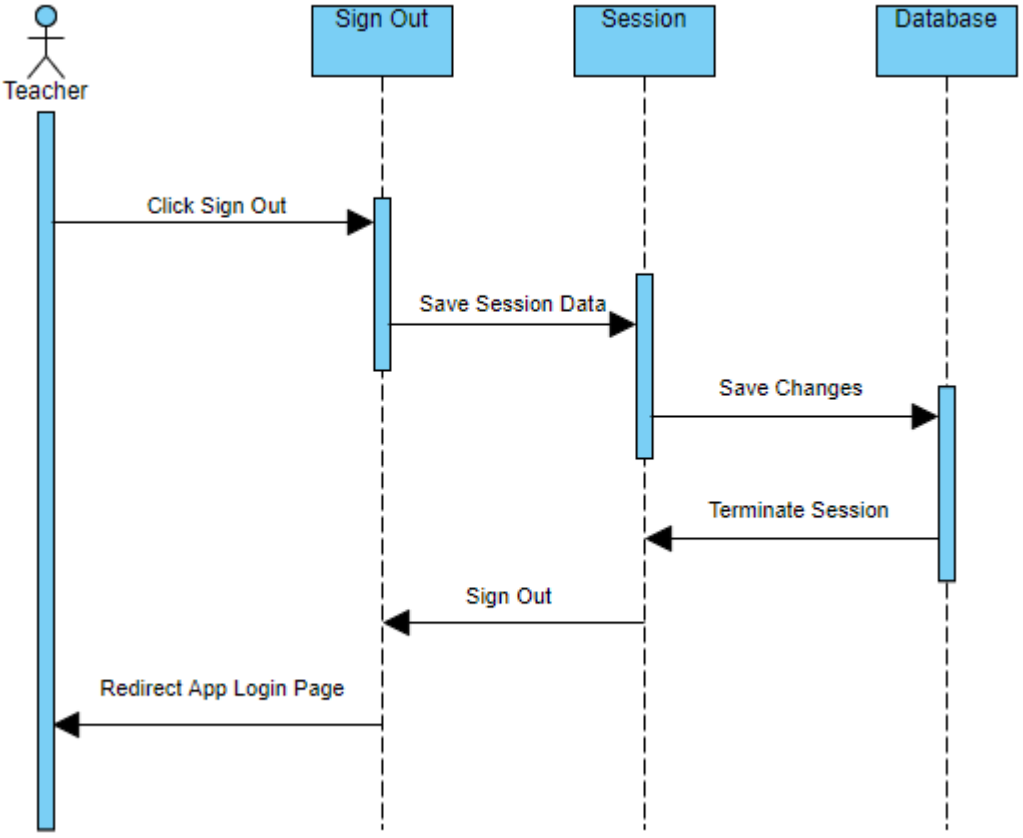

Figure 3. 15 Teacher Sign out Sequence Diagram

3.1.3 Class Diagram

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modelling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.

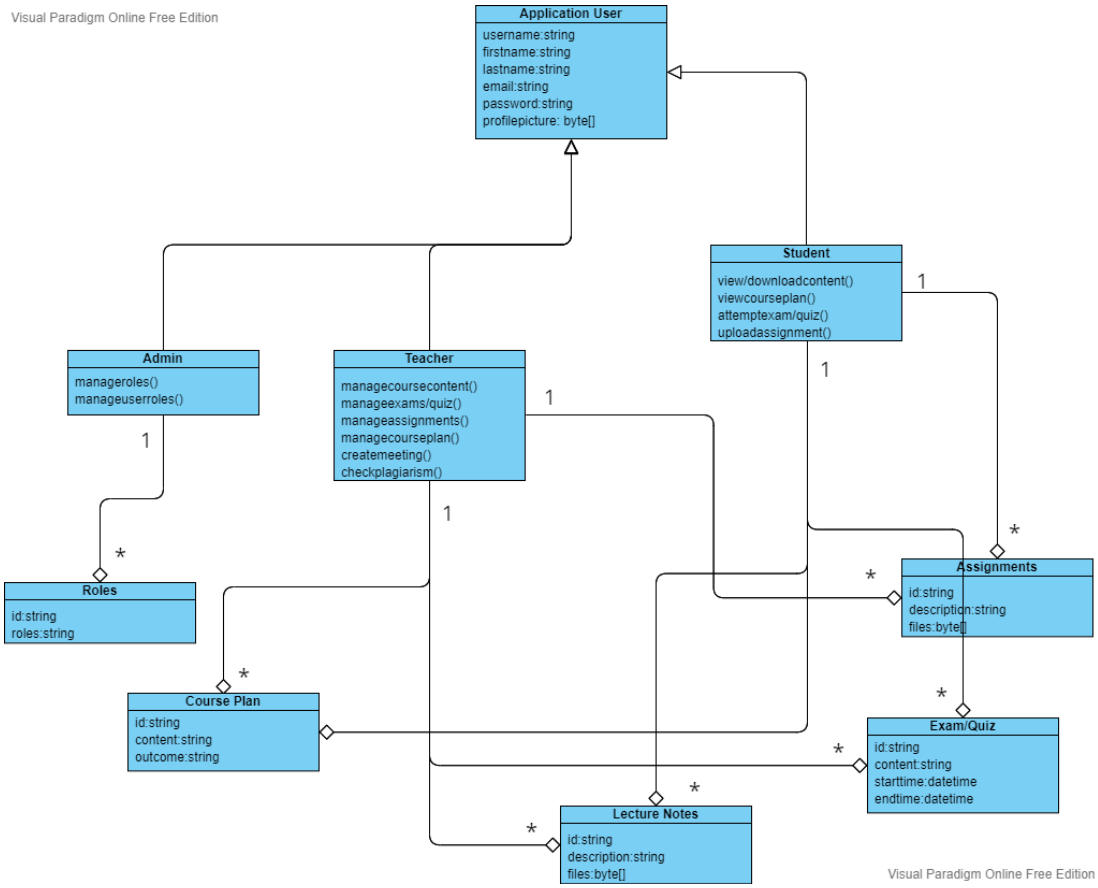

Figure 3. 16 Class Diagram

The class diagram in the figure above describes the classes interacting within the Website. Classes are also showing the functions used in them.

3.1.4 Entity Relationship Diagram

The database is absolutely an integral part of software systems. To fully utilize ER Diagram in database engineering guarantees you to produce high-quality database design to use in database creation, management, and maintenance. An ER model also provides a means for communication.

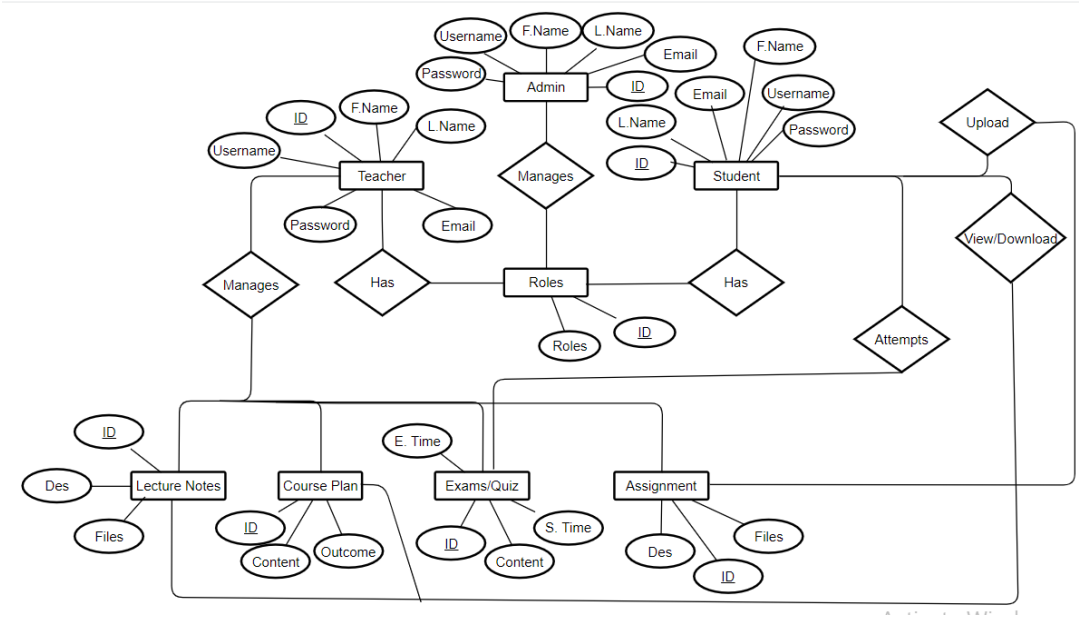

Figure 3. 17 Entity Relationship Diagram

3.2 Methodology

3.2.1 Feature Driven Development

Feature-driven development is an iterative and incremental software development process. It is a lightweight or agile method for developing software. FDD blends several industry-recognized best practices into a cohesive whole. These practices are driven from a client-valued functionality perspective. [2]

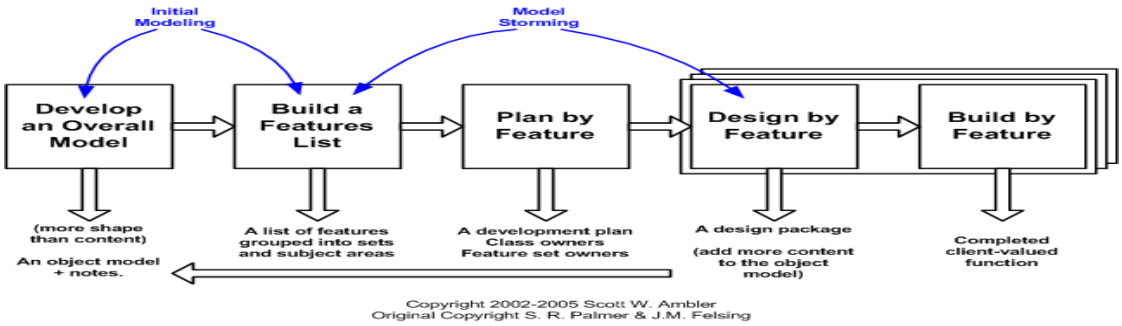

Figure 3. 18 FDD Model

Step 1: Develop an overall model

One of the core principles of feature-driven development is to use domain object modeling. Domain object modeling is a way to represent connected concepts and the relationships between them. During the first step, an outline of the domain model is created. This object model will be the blueprint for the project.

Step 2: Build feature list

The list of features identifies the elements necessary expressed as actions, results, or objects.

Step 3: Plan by feature

Once the feature list is created, you need to determine the order in which they need to be developed and implemented and which team member is responsible for each feature and identify potential risks, dependencies, workload constraints, and other obstacles.

Step 4: Design by feature

A design package is created for each feature. The design phase ends with a design review by the entire team.

Step 5: Build by feature

After the design is approved, now it's time to start writing code. All the items necessary to support the feature design are implemented. These elements can include front-end and back-end elements like user interfaces or database queries. Once everything is completed by individual team members the feature moves onto testing and the process begins again for the next feature on the list.

CHAPTER 4

DATA AND EXPERIMENTS (and/or IMPLEMENTATION)

4.1 Web Development Technologies used for Implementation

We have used the following technologies to develop this web application.

4.1.1 HTML

Hyper-Test Mark-up language is used in creating webpages and web applications along with other languages such as JavaScript and CSS for the responsiveness and beautification of webpages.

4.1.2 CSS

A cascading style sheet is used for the presentation of Web pages including colors, layout, and fonts it allows to adapt the style of presentation for different devices CSS can also be used with other mark-up languages other than HTML.

4.1.3 Bootstrap

Bootstrap is the most popular CSS framework for developing responsive and mobile-first websites. Bootstrap 5 is the newest version of Bootstrap.

4.1.4 JavaScript

JavaScript is a scripting language that allows you to implement complex things on web pages — every time a web page does more than just display static information for you to look at, it mostly displays timely content updates.

4.1.5 LINQ

Language-Integrated Query (LINQ) is the name for a set of technologies based on the integration of query capabilities directly into the C# language. Traditionally, queries against data are expressed as simple strings without type checking at compile time or IntelliSense support. [3]

4.1.6 C#

C# (C-Sharp) is a programming language developed by Microsoft that runs on the .NET Framework. C# is used to develop web apps, desktop apps, mobile apps, games, and much more.

4.1.7 Entity-Framework Core

Entity Framework Core is an open-source ORM framework for .NET Core applications supported by Microsoft. It enables developers to work with data using objects of domain-specific classes without focusing on the underlying database tables and columns where this data is stored. [4]

4.1.8 ASP.NET Core Identity

ASP.NET Core Identity is a fully-featured membership system for creating and maintaining user logins. Using Identity API, you can sign in & sign out users, reset their passwords, lockout users & Implement Multi-Factor Authentication. It can also integrate with external login providers like Microsoft Account, Facebook, Google, etc. [5]

4.1.9 Razor Syntax

Razor is a simple programming syntax for embedding server code in web pages. Razor syntax is based on the ASP.NET and .Net Core framework. [6]

4.1.10 CKEditor

Modern JavaScript rich text editor with a modular architecture. Its clean UI and features provide the perfect WYSIWYG UX for creating semantic content. [7]

4.1.11 Copyleaks API

Copyleaks is a comprehensive anti-plagiarism software that can scan documents, raw text, URLs and connect with APIs. Copyright infringement has become a major issue for publishers, web development companies, SEO agencies, and more. Alongside this, Academic plagiarism has also hit high rates as it has never been easier for students to find content online to copy and paste to pass off as their own. [8]

4.1.12 Face-API.js

JavaScript API for face detection and face recognition in the browser implemented on top of the tensorflow.js core API. [9]

4.1.13 Jitsi Meet External API

Embedding the Jitsi Meet API into your site or app enables you to host and provide secure video meetings with your colleagues, teams, and stakeholders. The Meet API provides a full complement of comprehensive meeting features.

Your Jitsi meetings can be hosted and attended using any device while keeping your data and privacy protected. You can reach your meeting participants anywhere in the world eliminating the need for travel and the associated inconvenience. [10]

4.1.14 Gmail SMTP Server

Google's Gmail SMTP server is a free SMTP service that anyone who has a Gmail account can use to send emails. You can use it with personal emails, or even with your website if you are sending emails for things such as contact forms, newsletter blasts, or notifications. [11]

4.2 Tools Used for Implementation

4.2.1 SSMS

SQL Server Management Studio is a software application first launched with Microsoft SQL Server 2005 that is used for configuring, managing, and administering all components within Microsoft SQL Server.

4.2.2 Visual Studio

Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services, and mobile apps.

CHAPTER 5

RESULTS AND DISCUSSIONS (or USER MANUAL)

5.1 Login

Users can log in with login credentials.

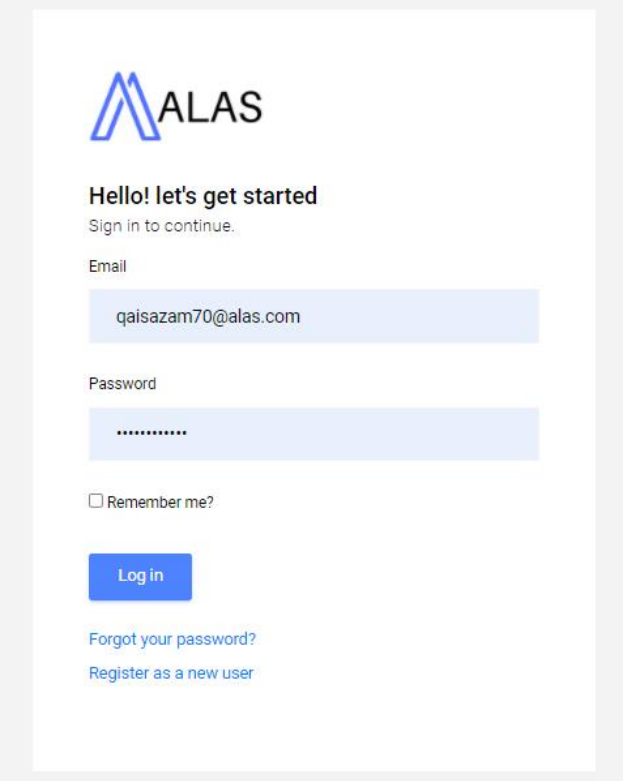

Figure 5. 1 User Login

5.2 Register

New users can register by filling in the details.

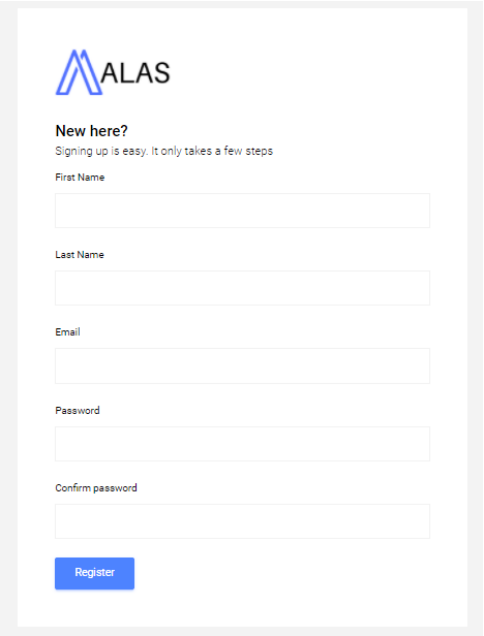

Figure 5. 2 User Register

5.3 Forgot Password

Users can recover passwords through email.

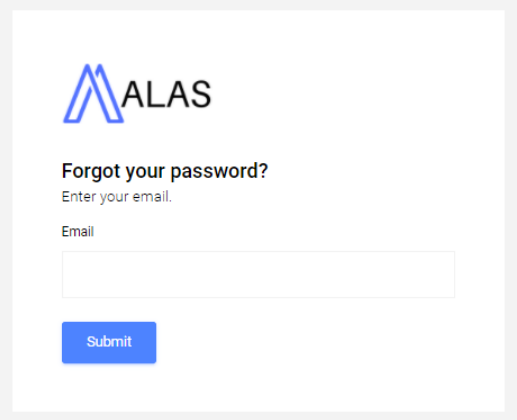

Figure 5. 3 Forgot Password

5.4 Admin Dashboard

This is an admin dashboard where the admin can manage roles and user roles.

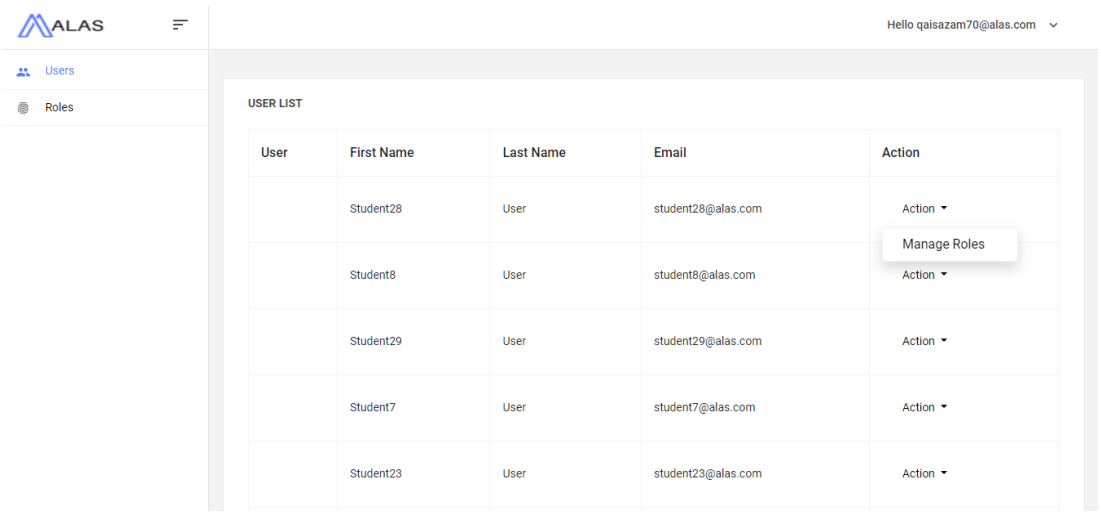

Figure 5. 4 Admin Dashboard

5.5 Teacher Dashboard

This is a teacher dashboard where the teacher can manage course plans, lecture notes, exams/quizzes, Live meetings, and check plagiarism.

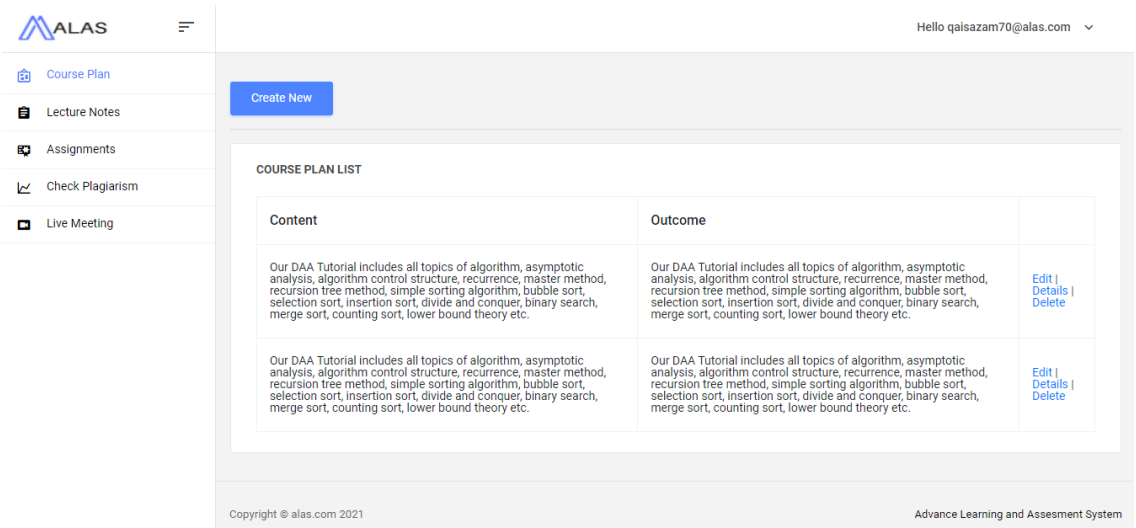

Figure 5. 5 Teacher Dashboard

5.6 Student Dashboard

This is a student dashboard where students can view/download content and submit exams/quiz, and assignments.

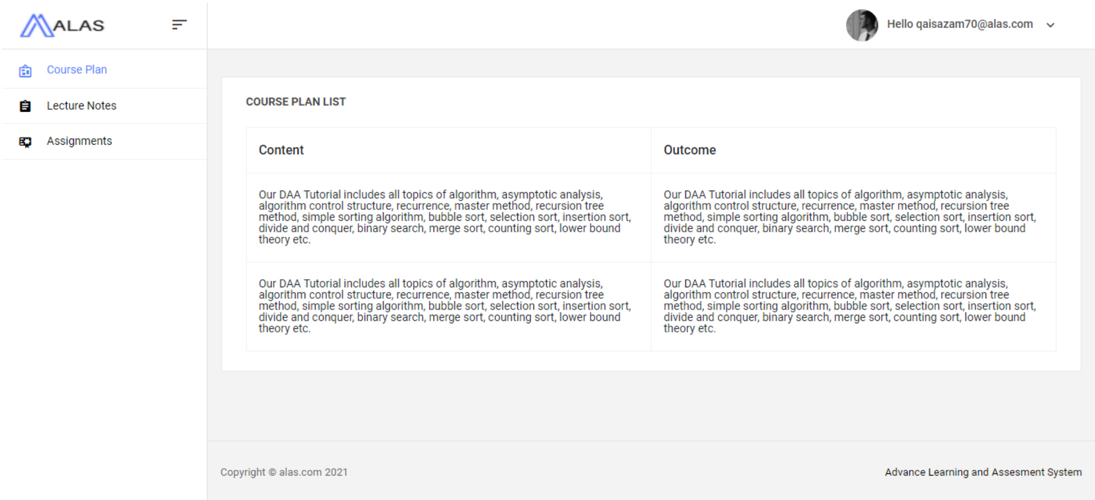

Figure 5. 6 Student Dashboard

5.7 Profile Setting

Users can manage profiles by going into the profile setting.

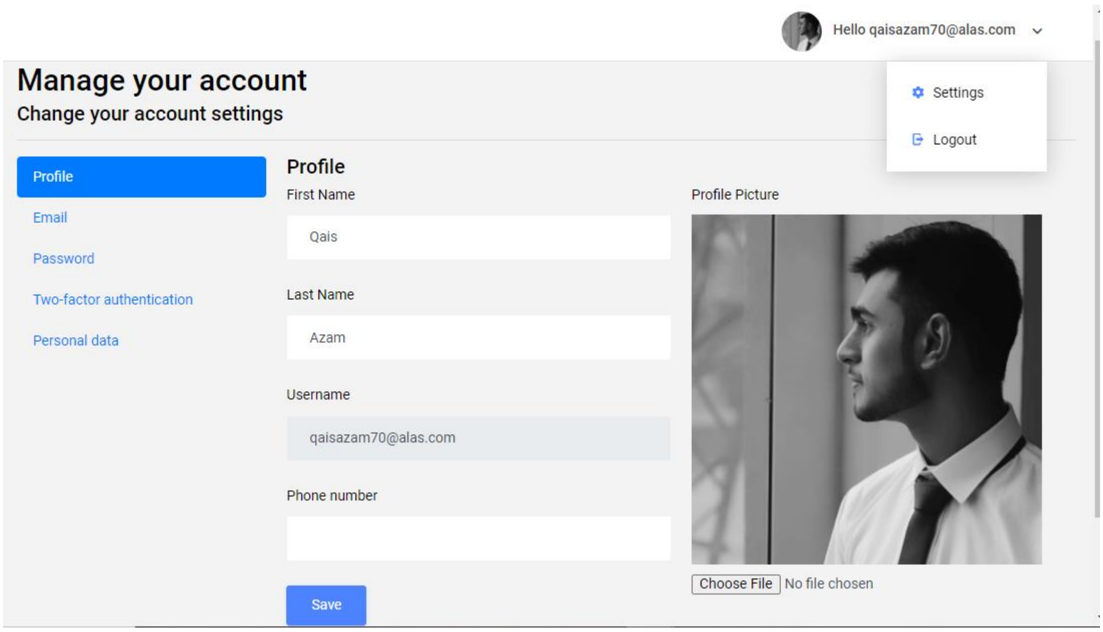

Figure 5. 7 Profile Setting

5.8 User Roles

This is a user role management interface for admin.

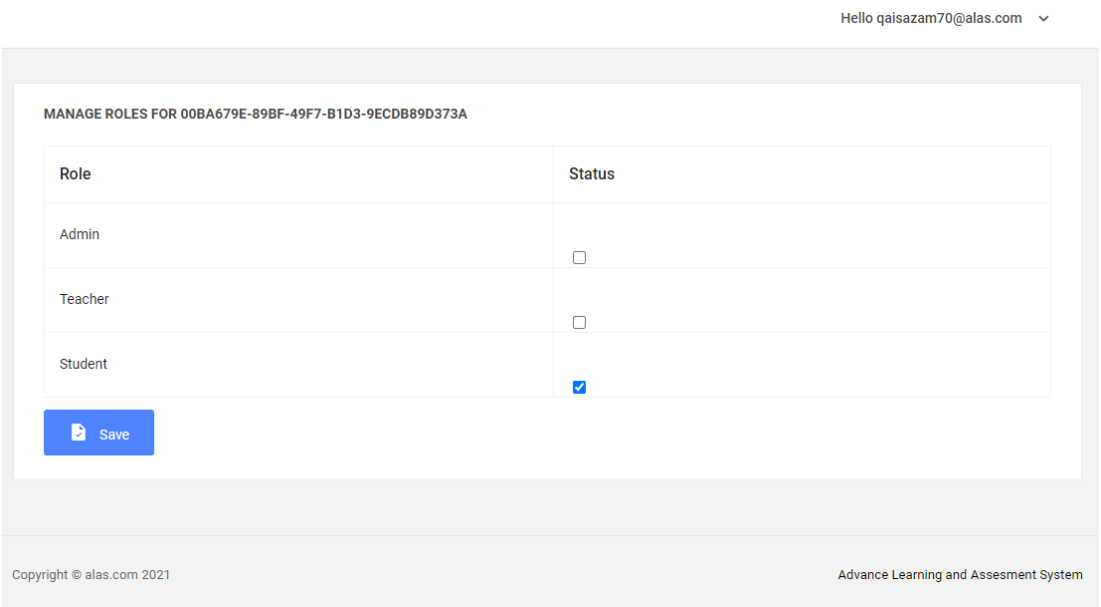

Figure 5. 8 User Roles

5.9 Roles

This is a role management interface for admin.

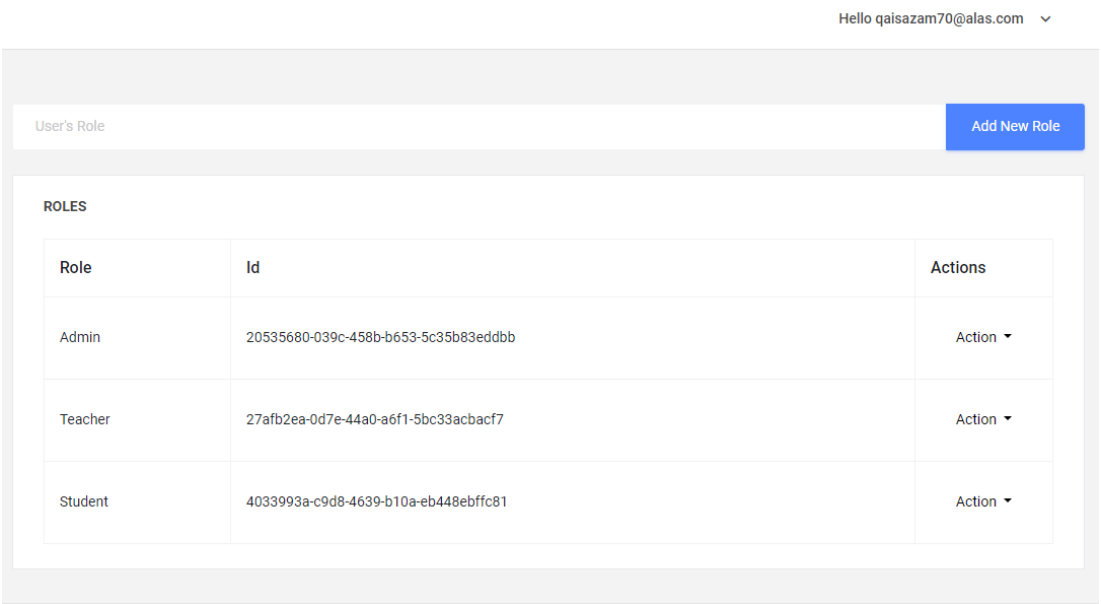

Figure 5. 9 Roles

5.10 Course Plan

Course Plan management interface for the teacher.

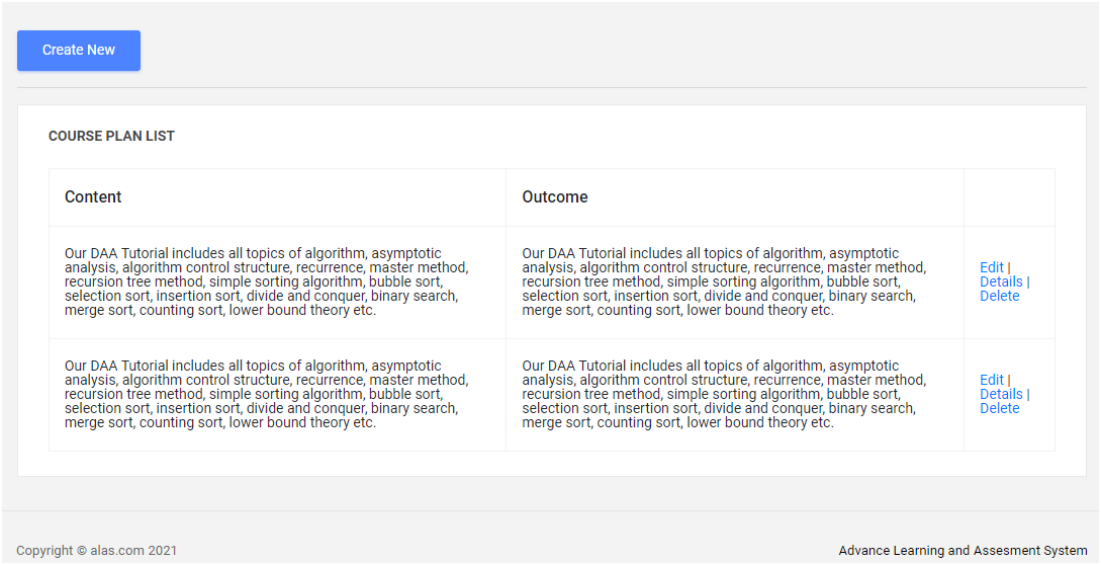

Figure 5. 10 Course Plan

5.11 Lecture Notes

Lecture Notes management interface for the teacher.

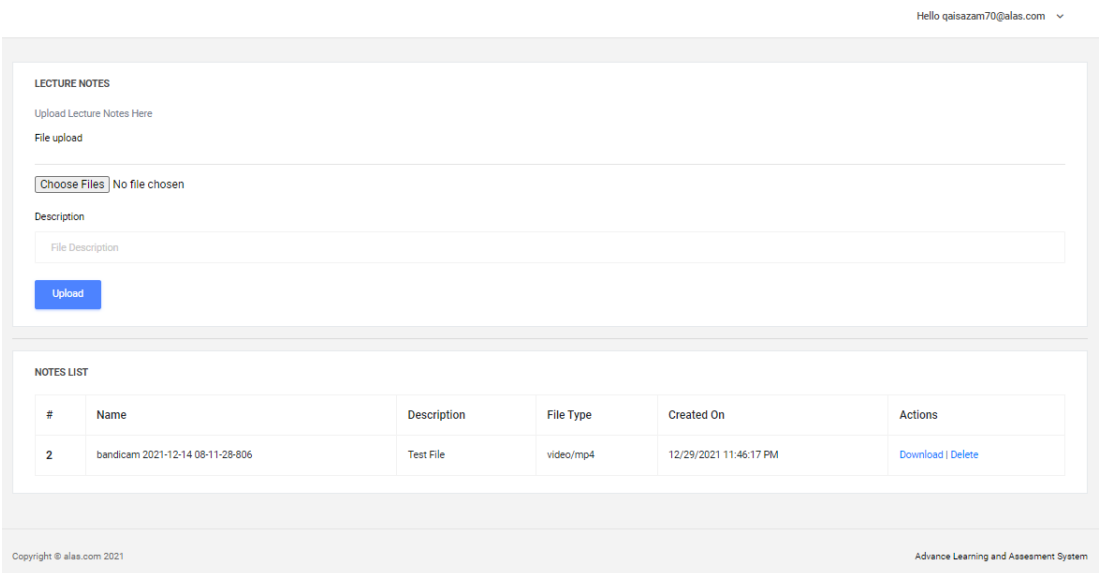

Figure 5. 11 Lecture Notes

5.12 Exam

Exam solving interface for the student.

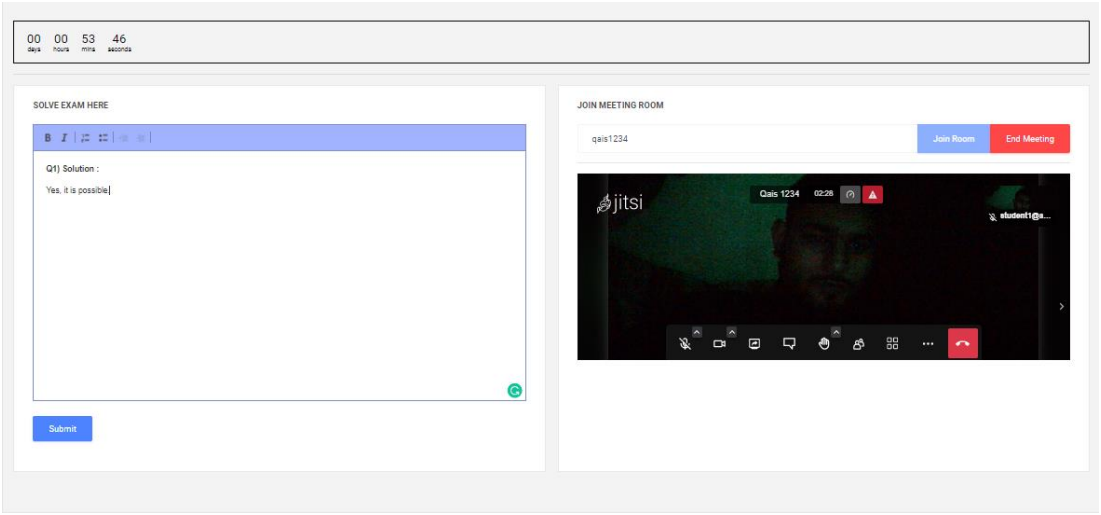

Figure 5. 12 Exam

5.13 Live Meeting

Live meeting interface for the teacher.

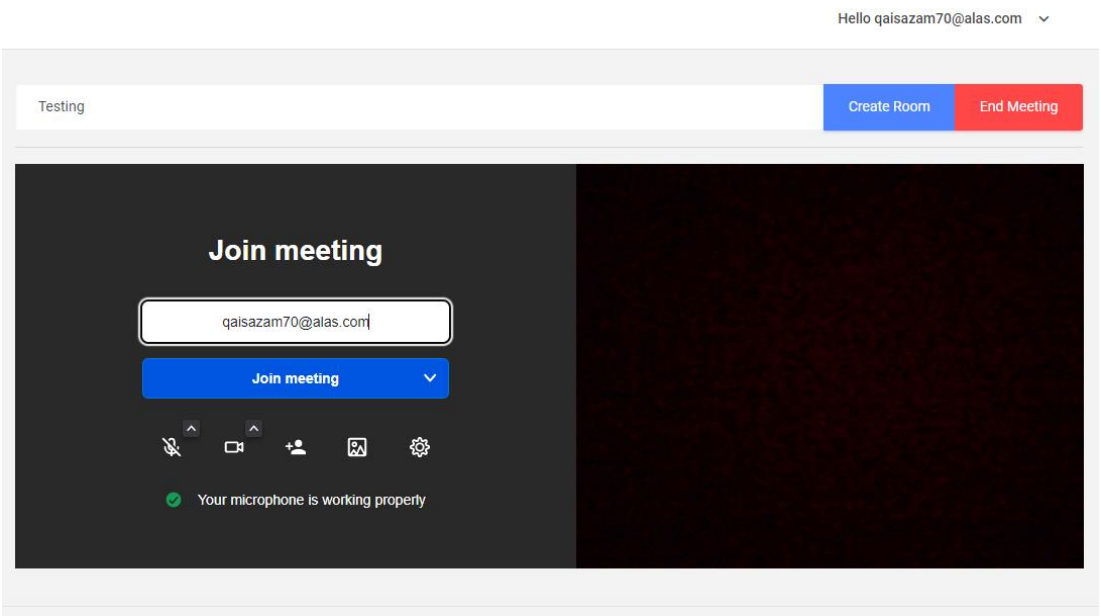

Figure 5. 13 Live Meeting

5.14 Check plagiarism

Enter content to check plagiarism. We are using PrePostSeo to detect plagiarism.

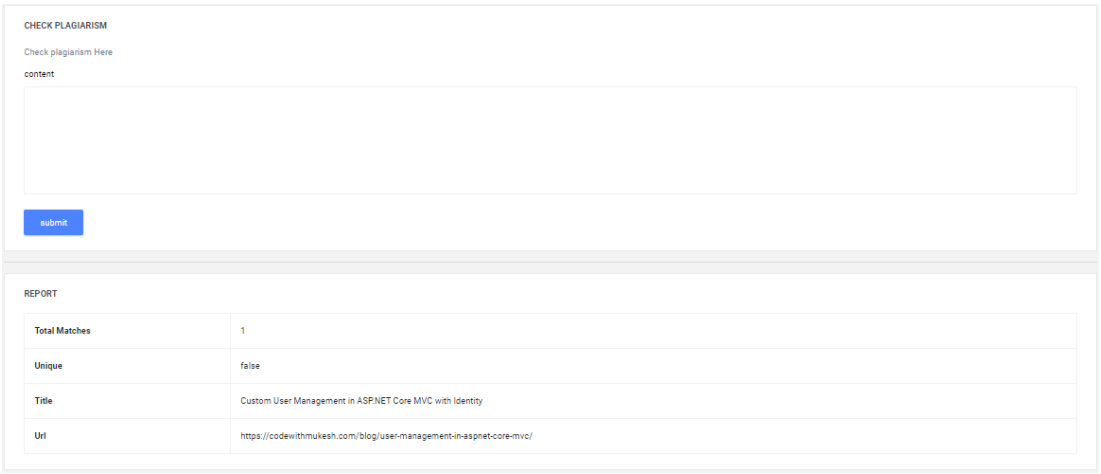

Figure 5. 14 Check Plagiarism

5.15 Facial Recognition

Detecting face using face API.

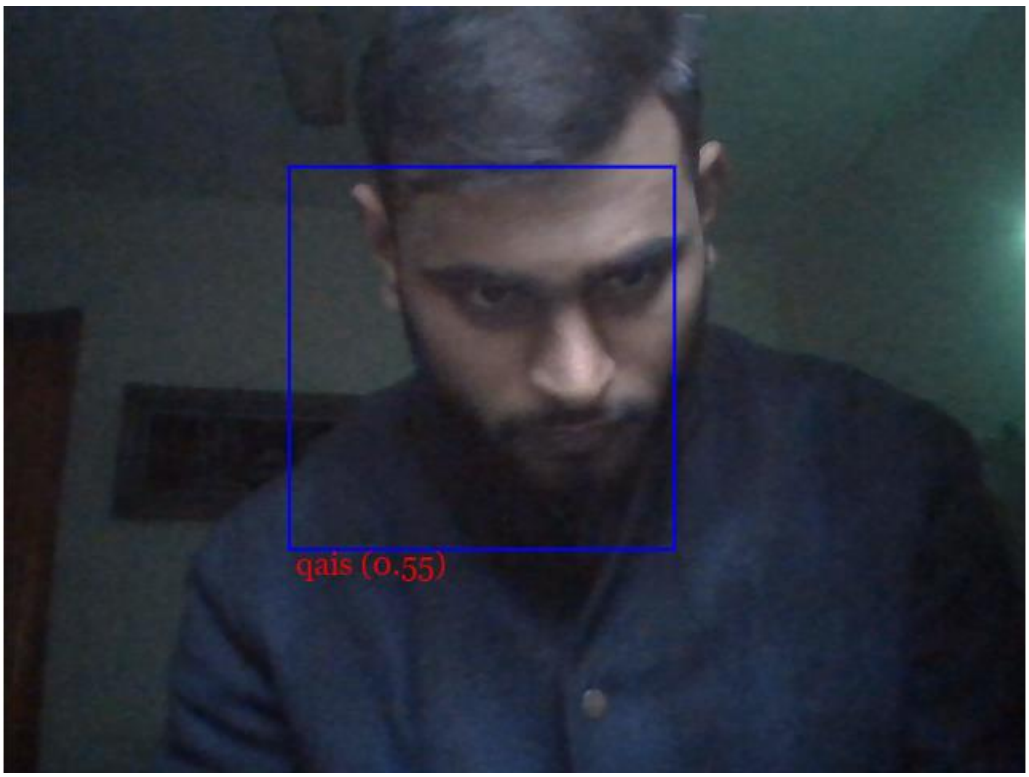

Figure 5. 15 Face Recognition

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

Considering the flaws of online assessments in already existing learning management systems we have designed a system that will make online assessments more credible and transparent. The main aim of this project is to provide a proper assessment system that is useful for both teachers and students. The improvement in the assessment system is achieved to that level where cheating is minimized, and effective online examinations can be conducted more efficiently. The system will have a plagiarism detector and teachers will be able to check plagiarism within the system. There will be real-time face recognition and live video sharing during exams so that teachers can keep an eye on students.

REFERENCES

Electronic Sources from the Internet:

[1] Olasile Babatunde Adedoyin, “Covid-19 pandemic and online learning: the challenges and opportunities”, 2019. Online Available at: https://www.tandfonline.com/doi/full/10.1080/10494820.2020.1813180

[2] Lucid Content Team, “What is feature-driven development?” 2019. Online Available at: https://www.lucidchart.com/blog/why-use-feature-driven-development

[3] Microsoft Docs, “Language Integrated Query (LINQ) (C#)”2021. Online Available at: https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/

[4] Entity Framework Tutorial, “Entity Framework Core” 2021. Online Available at: https://www.entityframeworktutorial.net/efcore/entity-framework-core.aspx

[5] IdentityServer4 “Using ASP.NET Core Identity” 2019. Online Available at: http://docs.identityserver.io/en/latest/quickstarts/6_aspnet_identity.html

[6] W3 Schools “ASP.NET Razor - C# and VB Code Syntax” 2020. Online Available at: https://www.w3schools.com/asp/razor_syntax.asp

[7] Wikipedia “CKEditor” 2021. Online Available at: https://en.wikipedia.org/wiki/CKEditor

[8] Copyleaks “What is Copyleaks Anti-Plagiarism Software?” 2020. Online Available at: https://copyleaks.com/

[9] Github “face-api.js” 2020. Online available at: https://justadudewhohacks.github.io/face-api.js/docs/index.html

[10] Jitsi Meet Handbook “What is Jitsi” 2019 Online available at: https://jitsi.github.io/handbook/docs/intro

[11] Kinsta Blog “How to Use the Gmail SMTP Server to Send Emails for Free” 2020 Online available at: https://kinsta.com/blog/gmail-smtp-server/

APPENDICES

APPENDIX A: Suggested Readings

- Overview of Entity Framework Core Microsoft Docs
- Fundamentals Concepts of Identity Platform Microsoft Docs
- Language Integrated Query Microsoft Docs
- Create a Complex Data Model for an ASP.NET MVC App Microsoft Docs
- Gmail SMTP Configuration Google Docs