ASSESSMENT MANAGEMENT FOR E-LEARNING AT PRIMARY EDUCATION LEVEL

Group Members

Hamad Zaheen (01-131172-007)

Ali Raza (01-131172-037)

Supervisor: Dr. Tamim Ahmad Khan

A Final Year Project submitted to the Department of Software Engineering, Faculty of Engineering Sciences, Bahria University, Islamabad in the partial fulfillment for the award of degree in Bachelor of Software Engineering

June 2021

THESIS COMPLETION CERTIFICATE

Student Name: Ali Raza Enrolment No: 01-131172-037

Student Name:Hamad Zaheen Enrolment No: 01-131172-007

Programme of Study: Bachelor of Software Engineering

Project Title: Assessment Management for E-Learning at Primary Education Level

It is to certify that the above students' project has been completed to my satisfaction and to my belief, its standard is appropriate for submission for evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at ____________ that is within the permissible limit set by the HEC. I have also found the thesis in a format recognized by the department.

Supervisor's Signature: ____________________________

Date: ___________________________ Name: Dr. Tamim Ahmad Khan

CERTIFICATE OF ORIGINALITY

This is certifying that the intellectual contents of the project

Assessment Management for E-Learning at Primary Education Level

are the product of my/our own work except, as cited properly and accurately in the acknowledgements and references, the material taken from such sources as research journals, books, internet, etc. solely to support, elaborate, compare, extend and/or implement the earlier work. Further, this work has not been submitted by me/us previously for any degree, nor it shall be submitted by me/us in the future for obtaining any degree from this University, or any other university or institution. The incorrectness of this information, if proved at any stage, shall authorities the University to cancel my/our degree.

Name of the Student: Ali Raza

Signature: ________________________ Date: __________________

Name of the Student: Hamad Zaheen

Signature: ________________________ Date: __________________

Abstract

COVID-19 has resulted in closure of educational institute across the globe. Due to this education system has changed a lot, and teaching is moved remotely and on digital platforms. With this shift away from the classrooms, some people are questioning whether the online learning will continue post- pandemic, and how will it affect the education systems across the world.

Assessment Management for E-learning at Primary Education Level is a web-based automated assessment system. The purpose of this system is to address the concerns related to online assessment during this pandemic. This application allows teacher to prepare quiz, and students to attend it.

The system will allow teacher to add questions into the databank. While creating the quiz, the system will allow teacher to add questions from databank or add new questions in the quiz. The system will allow teacher to assign quiz by creating a link. The system will allow students to attend quiz via link. The system will generate result for students and teacher after quiz submission.

Dedication

To our parents for their love and support

Acknowledgments

All praises to **Allah Almighty** whose blessings enabled us to complete this Final Year Project. Praise to **Prophet Muhammad (PBUH)** a mentor for all knowledge seekers and the humanity as a whole.

We feel a deep sense of gratitude to our dignified supervisor **Dr. Tamim Ahmad Khan**, for his supervision, constant support, guidance, useful suggestions, consistent encouragement, and dynamic supervision which enabled us to complete this FYP successfully. It is an honor to be his students.

We would like to express our gratitude towards our parents and friends who were our source of encouragement in completion of this project.

Table of Contents

CHAPTER-1

INTRODUCTION

1. Introduction

Assessment Management for E-Learning at Primary Education Level is a web-based application. In this chapter we will discuss our motivation behind the project, the problem statement, and objectives of our project. We will discuss and compare our system with the pre-existing systems. Rest of the document include the literature review, requirements, specifications, design, implementation, and testing of our project.

1.1 Motivation

Due to COVID-19 education has been shifted to digital platforms and E-learning. We found a technological gap between E-learning and online assessment. Our system is a web-based application which will bridge this gap by integrating it with an existing Learning Management System (LMS) or can be used as stand-alone system.

1.2 Problem statement

As COVID-19 has shifted the traditional education system to digital platform, so we need an efficient, easy to use, and reliable assessment system. We need an assessment system allowing online assessment of both MCQ type as well as a rubric based analysis ensuring transparency and reducing plagiarism.

1.3 Objectives

Assessment Management for E-Learning at Primary Education Level is a web-based application which cater the problems related to online assessment. The objectives of this system are:

- A system that records interaction patterns to conduct further research.
- Measuring student learning with item response theory.
- Analyze student performance through assessment.

1.4 Main contributions

- Our assessment system is fully automated, and it eliminate the short comings of existing system. Our system allow teacher to add different type of questions which include Multiple Choice Questions (MCQs), True False, Matching, and Free Text questions.
- We also incorporate Item Response Theory (IRT) in which each question is based on a difficulty level. This feature will allow teacher to assess the student better and identify the capabilities of student.

- The system will also make use of Interaction Pattern Recording in which time spend by student on each individual question is recorded. This will give teacher an idea about where the student faced difficulty and spent most of his time.
- Rubric is also used in Free Text Question in which the student's answer is assessed on basis of a standard answer provided by the teacher.

1.5 Report organization

The structure of the report is discussed below:

- **Chapter 1:** This chapter is of introduction. First of all, we discussed what is project about, the motivation, problem statement, objectives, and main contributions
- **Chapter 2:** Chapter 2 is about Literature Review in which we will discuss the background and previous work done related to our project. We will also discuss short comings of previous projects and how our system incorporates those short comings
- **Chapter 3:** In this chapter we will discuss the specifications of the system. These include the system requirements like functional, non-functional, interface, database, and use -cases etc.
- **Chapter 4:** In this chapter we will discuss system design. This chapter includes system architecture, design approaches, class diagram, sequence diagrams, deployment diagram, and database schema
- **Chapter 5:** This chapter is about system implementation. We will discuss the approaches, methodologies used in system development. We will also discuss the issues encountered
- **Chapter 6:** This chapter is of testing. Which testing techniques were applied, what were the testing techniques that were used, how unit, integration, and system testing was carried out. Discussion related to test cases
- **Chapter 7:** Chapter 7 is about the conclusion, contribution, reflections, and future work of the project.

CHAPTER-2

BACKGROUND /LITERATURE REVIEW

2. Background/Literature Review

Due to COVID-19, education has been shifted to digital platforms and E-learning. There is a technological gap between E-learning and online assessment. There are tons of different assessment systems available, but each assessment system has its own pros and cons. A big problem was if a system has some good features, then it was priced. Our assessment management system primary audience is primary level education, but it can also be used for higher level education assessment as well.

Assessment Management for E-Learning at Primary Education Level is a web-based application which allows us to cater the short comings of existing systems. This system can be used as stand-alone or can be integrated with an existing Learning Management System. (LMS) Given below are the objectives of our system:

- Analyze student performance through assessment
- A system that records interaction patterns to conduct further research
- Measuring student learning with item response theory

1. **Analyze student performance through assessment**

   This assessment management system will authenticate teachers through sign up and log in. Teacher can add four different types of questions. i.e., MCQ, True False, Matching, and Free Text questions. Teacher can create a question bank. System will allow teacher to prepare quizzes in which teacher has the option to add questions from data bank or add new question.

   A rubric is attached with the Free Text question. A rubric is basically the correct answer for a particular free text question. This rubric will be used to automatically mark the student marks on quiz submission.

   After adding quiz in system, a link can be generated for the quiz. Teacher will assign the quiz attempt time and a password will be set for attempting quiz. After attempting quiz, teacher can view the detailed result of students.

2. **A system that records interaction patterns to conduct further research**

   Our system caters the interaction pattern recordings in such a way that when students submit the quiz after attempting, each individual quiz question has a time duration. The time duration shows that how much time is spent by student on a particular question. From the results teacher can predict from these durations is that in which particular questions the student faced difficulties.

3. Measuring student learning with item response theory

By definition “Item response Theory(IRT) is a way to analyze responses to tests or questionnaires with the goal of improving measurement accuracy and reliability.” We integrate IRT in our assessment system in such a way that each question has an attribute named difficulty. The difficulty levels are Easy, Medium, and Hard. After quiz submission by student, the system displays question summary based on question difficulty. The summary contains no. of questions correctly attempted from each difficulty level. By looking at summary we can assume that a student was good in attempting easy difficulty level questions and was not able to correctly attempt medium and hard difficulty level question.

Application Name	Description	Difference
EasyLMS	Allow users to create quizzes based on MCQ, Fill in the blanks, & Free Text question. A link is generated for attempting the quiz	UI is very confusing and sophisticated. It does not have True False & Matching type question. No security or no check or balance on who has attempted quiz
QuestionPro	This web-based assessment application allows user to create quizzes based on MCQ type only. The MCQ type question are further divided into Select one, Select Many, Drop-down menu, and Single Row Text questions. A link is generated for attempting the quiz	This application does not have True False, Matching, and Free Text questions. Application UI is difficult to navigate through. No security mechanism and check and balance system on who can attempt the quiz
ClassMarker	This assessment management system allow user to create quizzes based on MCQ, True False, Matching, and Free Text questions. After creating quiz user has to set a password and time duration for quiz. A person will be able to attend quiz when the quiz password is entered correctly.	The detailed results of individuals students are only available in paid version. There is no feature which can display the individual duration of questions. No rubric can be attached with a Free Text question.

CHAPTER-3

SYSTEM REQUIREMENTS

3. System Requirements

3.1 Functional Requirements

Following are the different functions that the system will perform:

Teacher:

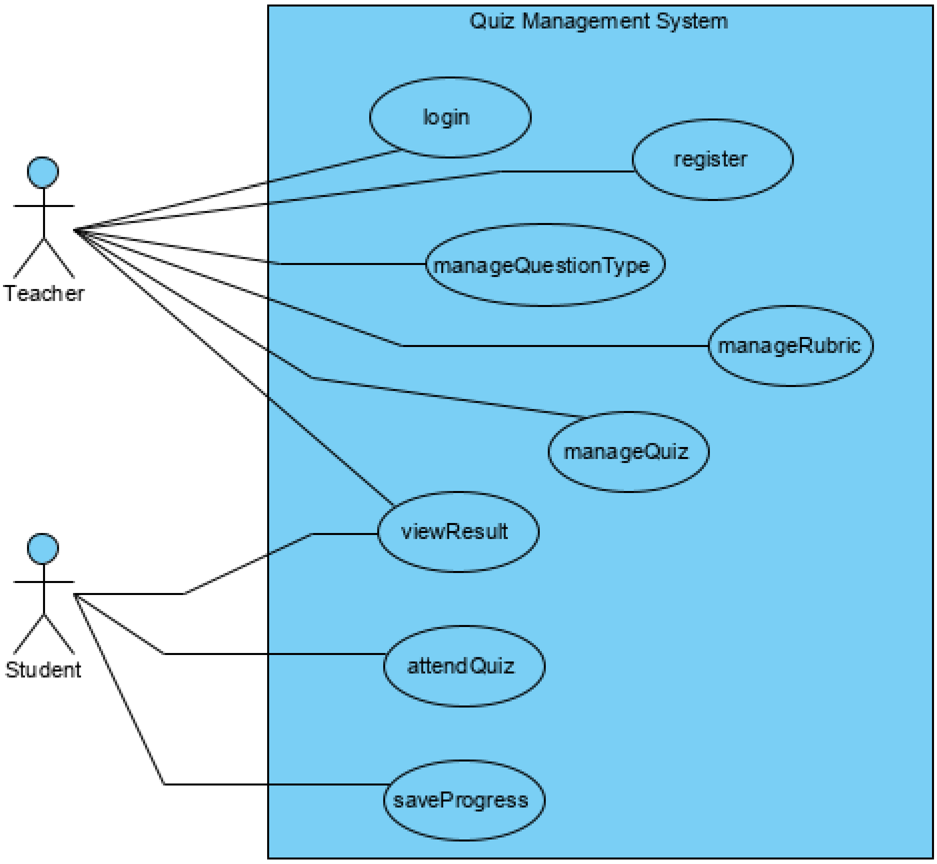
- The system should allow Teacher to register.
- The system should allow teacher to sign in.
- The system should allow Teacher to develop quiz data bank.
- The system should allow Teacher to design Rubrics.
- The system should allow Teacher to select quiz pattern.
- The system should allow Teacher to access dashboard.
- The system should allow Teacher to view time taken by student to solve quiz.
- The system should allow Teacher to view pattern results.
- The system should allow Teacher to shuffle quiz questions (if teacher wants to)
- The system should allow Teacher to set Time limit.

Student:

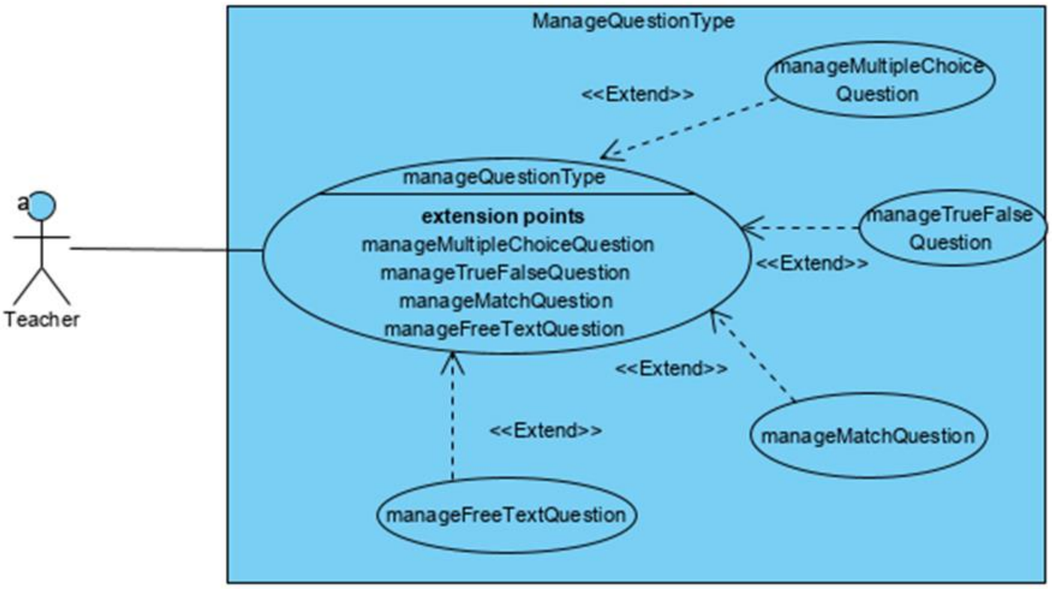
- The system should allow Student to attend quiz.
- The system should allow Student to save quiz progress.
- The system should allow Student to view results.

3.2 Use Cases

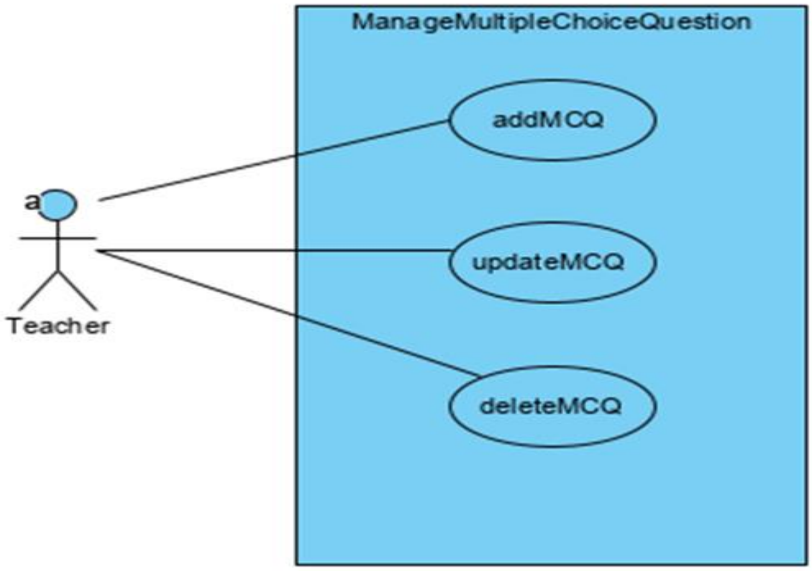
3.2.1 System Level Use Case Diagram



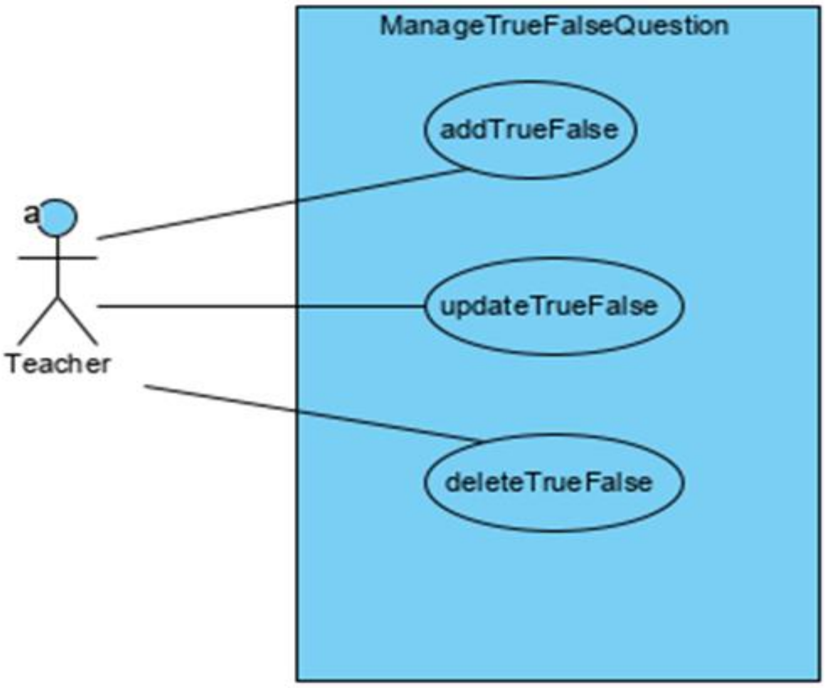
3.2.2 Manage Question Type Use Case Diagram



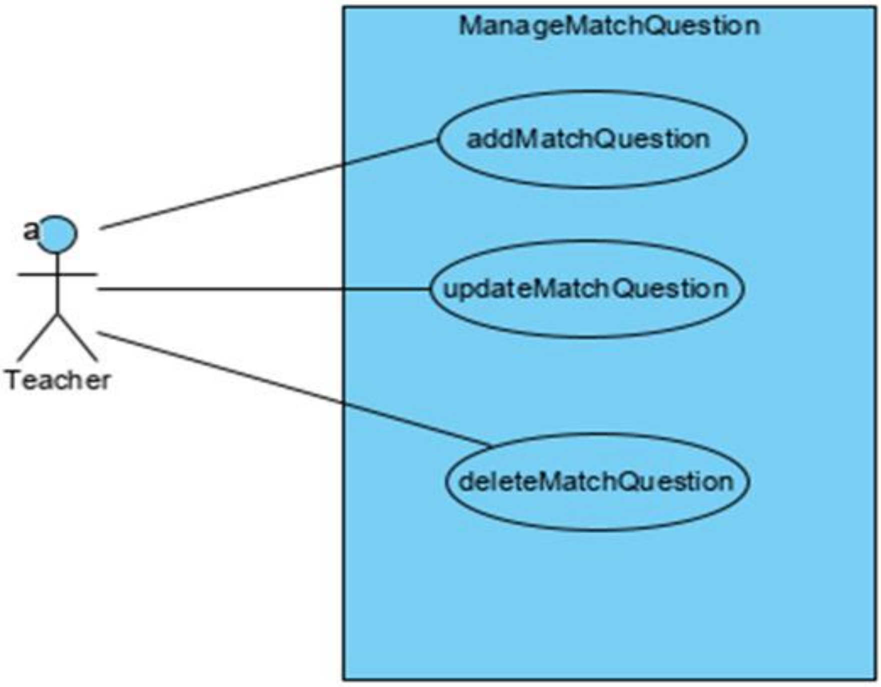
3.2.3 Manage Multiple Choice Question Use Case Diagram



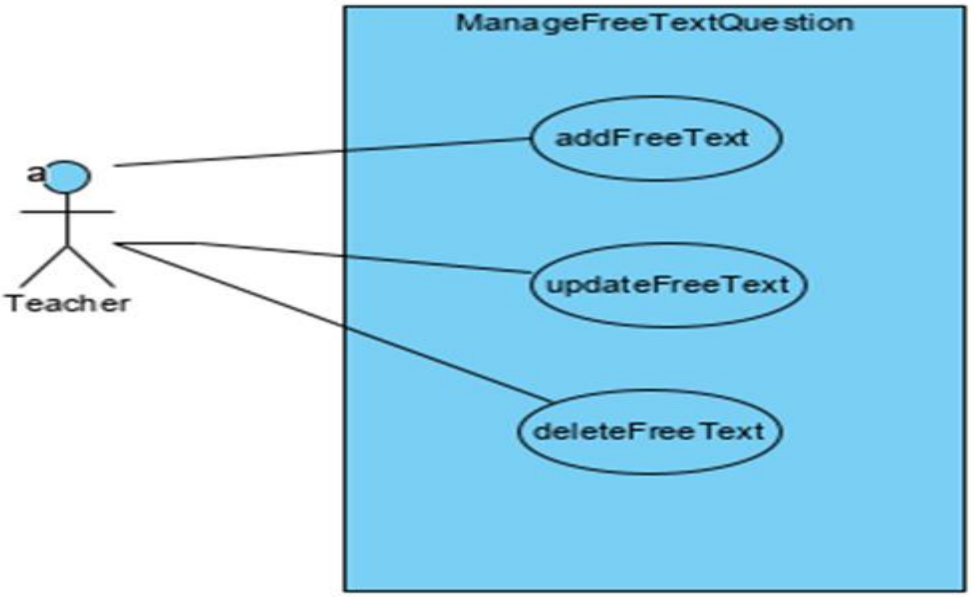
3.2.4 Manage True False Question Use Case Diagram



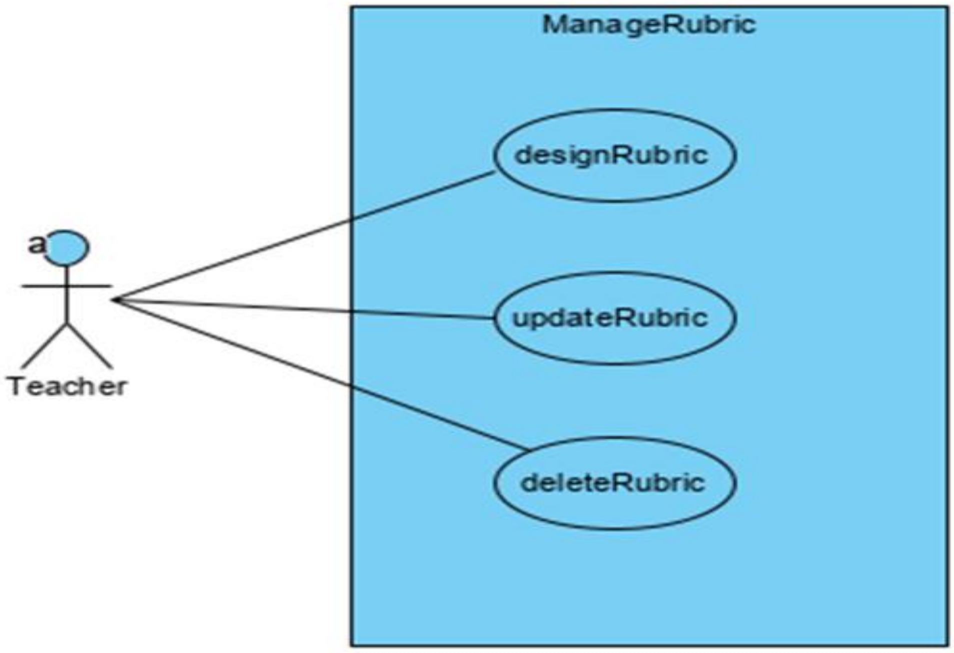
3.2.5 Manage Matching Question Use Case Diagram



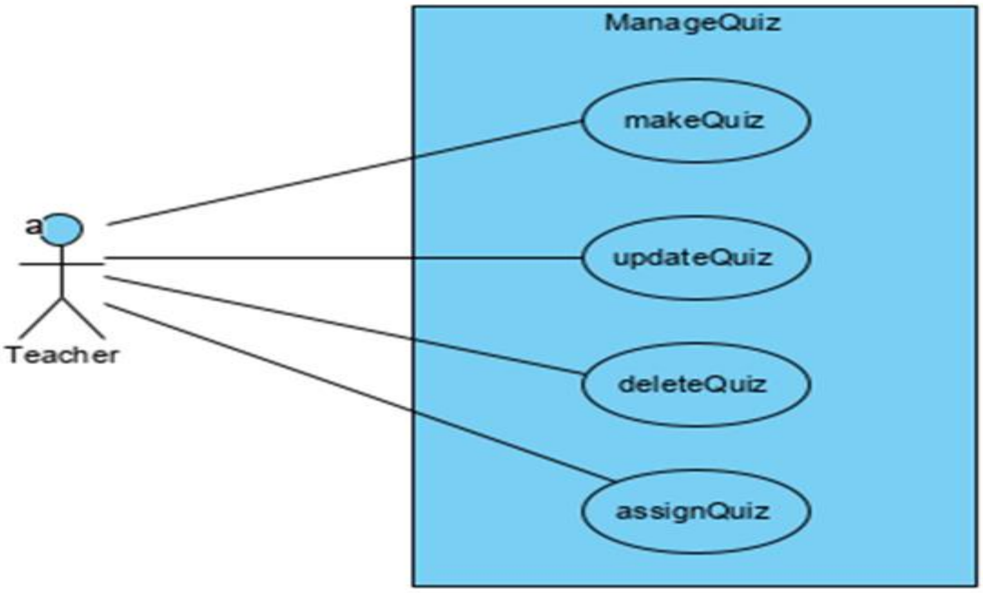
3.2.6 Manage Free Text Question Use Case Diagram



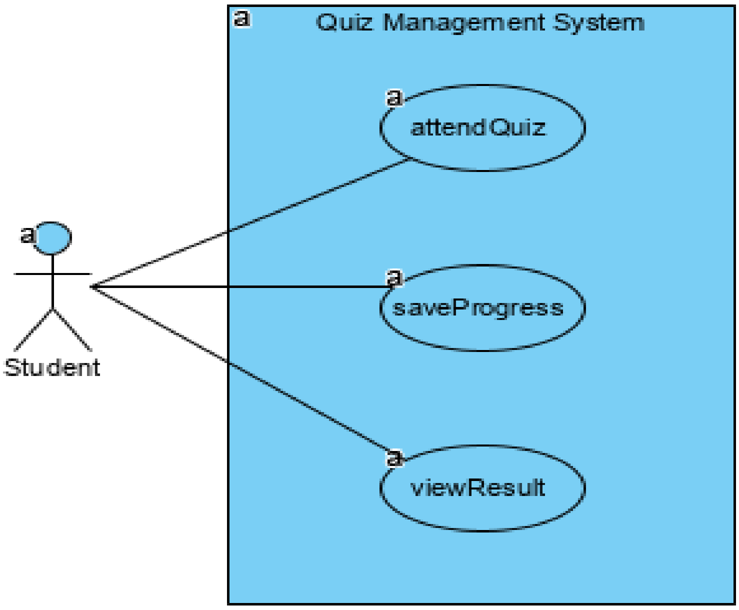
3.2.7 Manage Rubric Use Case Diagram



3.2.8 Manage Quiz Use Case Diagram



3.2.9 Student Use Case Diagram



3.3 System Features

3.3.1 Register

Use case ID:	UC-TE-01	
Use case Name:	Register	
Actor(s):	Teacher	
Basic Flow:	Teacher clicks on register button for registration	
Actor Actions		**System Response**
1. User navigates to Register link and clicks on it		System opens the interface containing the relevant form
2. User enters the required information and clicks on register button		System checks the provided credentials and register user into the system
Alternate Course of Action:		
Actor Actions		**System Response**
		2.a. If provided credentials are incorrect, an error message is displayed on screen for the respective error

3.3.2 Login

Use case ID:	UC-TE-02	
Use case Name:	Login	
Actor(s):	Teacher	
Basic Flow:	Teacher clicks on login link to login	
Actor Actions		**System Response**

1. User navigates to login link and enter credentials	System checks the provided credentials and log in user into the system
Alternate Course of Action:	
Actor Actions	**System Response**
	2.a. If provided credentials are incorrect, an error message is displayed on screen for the respective error

3.3.3 Add Multiple Choice Question

Use case ID:	UC-TE-03
Use case Name:	Add Multiple Choice Question
Actor(s):	Teacher
Basic Flow:	Teacher adds multiple choice question in data bank
Actor Actions	**System Response**
1. User navigates to Add Question link and clicks on it	System opens the interface containing the relevant form
2. User clicks on Select Topic	System displays the available topics from data bank in drop-down menu
3. User clicks on Multiple Choice question link	System opens the interface containing the relevant form
4. Users write question and mark the correct answer for question	System records the performed action
5. User clicks on question point	System displays drop-down menu of available points
6. User clicks on question difficulty level	System displays drop-down menu of Easy, Medium, Hard
7. User selects from randomize answers option	System records the performed action
8. User clicks on save button	System saves question in data bank

3.3.4 Update Multiple Choice Question

Use case ID:	UC-TE-04
Use case Name:	Update Multiple Choice Question
Actor(s):	Teacher
Basic Flow:	Teacher Updates multiple choice question in data bank
Actor Actions	**System Response**
1. User navigates to Update Question link and clicks on it	System opens the interface containing the relevant form
2. User clicks on Select Topic	System displays the available topics from data bank in drop-down menu
3. User clicks on Multiple Choice question link	System opens the interface containing the relevant form
4. Users write question and mark the correct answer for question	System records the performed action

5. User clicks on question point	System displays drop-down menu of available points
6. User clicks on question difficulty level	System displays drop-down menu of Easy, Medium, Hard
7. User selects from randomize answers option	System records the performed action
8. User clicks on save button	System updates question in data bank

3.3.5 Delete Multiple Choice Question

Use case ID:	UC-TE-05
Use case Name:	Delete Multiple Choice Question
Actor(s):	Teacher
Basic Flow:	Teacher Deletes question from data bank
Actor Actions	**System Response**
1. User navigates to Delete Question link and clicks on it	System opens the interface containing the relevant form
2. User clicks on Select Topic	System displays the available topics from data bank in drop-down menu
3. Users select question	System opens the interface containing the relevant form
4. User clicks on Delete button	System deletes the question from data bank

3.3.6 Add True False Question

Use case ID:	UC-TE-06
Use case Name:	Add True False Question
Actor(s):	Teacher
Basic Flow:	Teacher adds True False question in data bank
Actor Actions	**System Response**
1. User navigates to Add Question link and clicks on it	System opens the interface containing the relevant form
2. User clicks on Select Topic	System displays the available topics from data bank in drop-down menu
3. User clicks on True False question link	System opens the interface containing the relevant form
4. Users write question and mark the correct answer for question	System records the performed action
5. User clicks on question point	System displays drop-down menu of available points
6. User clicks on question difficulty level	System displays drop-down menu of Easy, Medium, Hard
7. User clicks on save button	System saves question in data bank

3.3.7 Update True False Question

Use case ID:	UC-TE-07
Use case Name:	Update True False Question
Actor(s):	Teacher
Basic Flow:	Teacher Updates True False question in data bank
Actor Actions	**System Response**
1. User navigates to Update Question link and clicks on it	System opens the interface containing the relevant form
2. User clicks on Select Topic	System displays the available topics from data bank in drop-down menu
3. User clicks on True False question link	System opens the interface containing the relevant form
4. Users write question and mark the correct answer for question	System records the performed action
5. User clicks on question point	System displays drop-down menu of available points
6. User clicks on question difficulty level	System displays drop-down menu of Easy, Medium, Hard
7. User selects from randomize answers option	System records the performed action
8. User clicks on save button	System updates question in data bank.

3.3.8 Delete True False Question

Use case ID:	UC-TE-08
Use case Name:	Delete True False Question
Actor(s):	Teacher
Basic Flow:	Teacher Deletes question from data bank
Actor Actions	**System Response**
1. User navigates to Delete Question link and clicks on it	System opens the interface containing the relevant form
2. User clicks on Select Topic	System displays the available topics from data bank in drop-down menu
3. Users select question	System opens the interface containing the relevant form
4. User clicks on Delete button	System deletes the question from data bank

3.3.9 Add Matching Question

Use case ID:	UC-TE-09
Use case Name:	Add Matching Question
Actor(s):	Teacher
Basic Flow:	Teacher adds Matching question in data bank
Actor Actions	**System Response**
1. User navigates to Add Question link and clicks on it	System opens the interface containing the relevant form
2. User clicks on Select Topic	System displays the available topics from data bank in drop-down menu
3. User clicks on Matching question link	System opens the interface containing the relevant form
4. User selects the match type from available options	System opens the interface containing the relevant form
5. Users write question and add the matching pair	System records the performed action
6. User clicks on question point	System displays drop-down menu of available points
7. User clicks on question difficulty level	System displays drop-down menu of Easy, Medium, Hard
8. User selects from the shuffle mode	System records the performed action
9. User clicks on save button	System saves question in data bank

3.3.10 Update Matching Question

Use case ID:	UC-TE-10
Use case Name:	Update Matching Question
Actor(s):	Teacher
Basic Flow:	Teacher Updates Matching question in data bank
Actor Actions	**System Response**
1. User navigates to Update Question link and clicks on it	System opens the interface containing the relevant form
2. User clicks on Select Topic	System displays the available topics from data bank in drop-down menu
3. User clicks on True False question link	System opens the interface containing the relevant form
4. Users write question and mark the correct answer for question	System records the performed action
5. User clicks on question point	System displays drop-down menu of available points
6. User clicks on question difficulty level	System displays drop-down menu of Easy, Medium, Hard
7. User selects from randomize answers option	System records the performed action
8. User clicks on save button	System updates question in data bank

3.3.11 Delete Matching Question

Use case ID:	UC-TE-11
Use case Name:	Delete Matching Question
Actor(s):	Teacher
Basic Flow:	Teacher Deletes question from data bank

Actor Actions	**System Response**
1. User navigates to Delete Question link and clicks on it	System opens the interface containing the relevant form
2. User clicks on Select Topic	System displays the available topics from data bank in drop-down menu
3. Users select question	System opens the interface containing the relevant form
4. User clicks on Delete button	System deletes the question from data bank

3.3.12 Add Free Text Question

Use case ID:	UC-TE-12
Use case Name:	Add Free Text Question
Actor(s):	Teacher
Basic Flow:	Teacher adds Free Text question in data bank

Actor Actions	**System Response**
1. User navigates to Add Question link and clicks on it	System opens the interface containing the relevant form
2. User clicks on Select Topic	System displays the available topics from data bank in drop-down menu
3. User clicks on Free Text question link	System opens the interface containing the relevant form
4. Users write the question	System records the performed action
5. User clicks on design rubric link	System opens the interface containing the relevant form
6. User design rubric for the question	System records the performed action
7. User clicks on question point	System displays drop-down menu of available points
8. User clicks on question difficulty level	System displays drop-down menu of Easy, Medium, Hard
9. User clicks on save button	System saves question in data bank

3.3.13 Update Free Text Question

Use case ID:	UC-TE-13
Use case Name:	Update Free Text Question
Actor(s):	Teacher
Basic Flow:	Teacher Updates Free Text question in data bank

Actor Actions	**System Response**
1. User navigates to Update Question link and clicks on it	System opens the interface containing the relevant form
2. User clicks on Select Topic	System displays the available topics from data bank in drop-down menu

3. User clicks on True False question link	System opens the interface containing the relevant form
4. Users write question and mark the correct answer for question	System records the performed action
5. User clicks on question point	System displays drop-down menu of available points
6. User clicks on question difficulty level	System displays drop-down menu of Easy, Medium, Hard
7. User selects from randomize answers option	System records the performed action
8. User clicks on save button	System updates question in data bank

3.3.14 Delete Free Text Question

Use case ID:	UC-TE-14
Use case Name:	Delete Free Text Question
Actor(s):	Teacher
Basic Flow:	Teacher Deletes question from data bank
Actor Actions	**System Response**
1. User navigates to Delete Question link and clicks on it	System opens the interface containing the relevant form
2. User clicks on Select Topic	System displays the available topics from data bank in drop-down menu
3. Users select question	System opens the interface containing the relevant form
4. User clicks on Delete button	System deletes the question from data bank

3.3.15 Make Quiz

Use case ID:	UC-TE-15
Use case Name:	Make Quiz
Actor(s):	Teacher
Basic Flow:	Teacher makes quiz and save it into the data bank
Actor Actions	**System Response**
1. User navigates to Make Quiz link and clicks on it	System opens the interface containing the relevant form
2. User Add question from data bank	System displays search bar for searching question through topic name or question text
3. Users select from randomize question option	System records the performed action
4. User set the time limit for quiz	System records the time duration for quiz
5. User clicks on save button	System saves quiz in data bank
Alternate Flow:	
Actor Actions	**System Response**
2.A. User search question by topic name or question text in data bank	System displays error message of no question found

3.3.16 Update Quiz

Use case ID:	UC-TE-16
Use case Name:	Update Quiz
Actor(s):	Teacher
Basic Flow:	Teacher Updates quiz and save it into the data bank
Actor Actions	**System Response**
1. User navigates to Update Quiz link and clicks on it	System opens the interface containing the relevant form
2. User Add question from data bank	System displays search bar for searching question through topic name or question text
3. Users select from randomize question option	System records the performed action
4. User set the time limit for quiz	System records the time duration for quiz
5. User clicks on save button	System saves quiz in data bank
Alternate Flow:	
Actor Actions	**System Response**
2.A. User search question by topic name or question text in data bank	System displays error message of no question found

3.3.17 Delete Quiz

Use case ID:	UC-TE-17
Use case Name:	Delete Quiz
Actor(s):	Teacher
Basic Flow:	Teacher Deletes quiz from the data bank
Actor Actions	**System Response**
1. User navigates to Delete Quiz link and clicks on it	System opens the interface containing the relevant form
2. Users select from the list of quizzes	System records the performed action
3. User clicks on Delete button	System deletes the quiz from data bank

3.3.18 Assign Quiz

Use case ID:	UC-TE-18
Use case Name:	Assign Quiz
Actor(s):	Teacher
Basic Flow:	Teacher assigns the quiz to and share it via link
Actor Actions	**System Response**
1. User navigates to share quiz link and clicks on it	System generates the link for that particular quiz
2. User clicks on share link	System displays the generated link

3.3.19 Design Rubric

Use case ID:	UC-TE-19
Use case Name:	Design Rubric
Actor(s):	Teacher
Basic Flow:	Teachers create assessment criteria for Free Text question by designing a rubric for that particular question
Actor Actions	**System Response**
1. User clicks on design rubric link	System opens the interface containing the relevant form
2. Users design the rubric for question	System records the performed action
3. User clicks on save button	System saves rubric for that particular Free Text question in data bank

3.3.20 Update Rubric

Use case ID:	UC-TE-20
Use case Name:	Update Rubric
Actor(s):	Teacher
Basic Flow:	Teachers create assessment criteria for Free Text question by Updating a rubric for that particular question
Actor Actions	**System Response**
1. User clicks on Update rubric link	System opens the interface containing the relevant form
2. User Updates the rubric for question	System records the performed action
3. User clicks on save button	System saves rubric for that particular Free Text question in data bank

3.3.21 Delete Rubric

Use case ID:	UC-TE-21
Use case Name:	Delete Rubric
Actor(s):	Teacher
Basic Flow:	Teacher Deletes the rubric for Free Text question
Actor Actions	**System Response**
1. User clicks on Delete rubric link	System opens the interface containing the relevant form
2. User selects the rubric for question	System records the performed action
3. User clicks on Deletes button	System deletes rubric for that particular Free Text question from data bank

3.3.22 View Result

Use case ID:	UC-STD-22
Use case Name:	View Result
Actor(s):	Teacher
Basic Flow:	Teachers view the result of a quiz
Actor Actions	**System Response**
1. User navigates to view result	System opens the interface containing the relevant form
2. User clicks on a particular quiz	System displays result details of that particular quiz

3.3.23 Attend Quiz

Use case ID:	UC-STD-01
Use case Name:	Attend quiz
Actor(s):	Student
Basic Flow:	Student attend and submit the quiz
Actor Actions	**System Response**
1. User clicks on start quiz link	System opens the interface containing the relevant form
2. User attends the quiz and click on submit button	System saves quiz. and display the quiz result

3.3.24 Save Progress

Use case ID:	UC-STD-02
Use case Name:	Save Progress
Actor(s):	Student
Basic Flow:	Student will save progress during quiz
Actor Actions	**System Response**
1. User clicks on the start quiz link	System opens the interface containing the relevant form
2. User clicks on save progress button	System saves the student progress till that particular time

3.3.25 View Result

Use case ID:	UC-STD-03
Use case Name:	View Result
Actor(s):	Student
Basic Flow:	Students view the result of a quiz
Actor Actions	**System Response**
1. User navigates to view result	System opens the interface containing the relevant form
2. User clicks on a particular quiz	System displays result details of that particular quiz

3.4 Non-Functional Requirements

3.4.1 Performance Requirements

- System will accommodate 1200 users at a time.
- Average response time of system is 1 second.
- The system response time should not be late than 10 second.

3.4.2 Reliability Requirements

As the software system is of critical nature. So, reliability of the system should not be less than 90 percent.

3.4.3 Security Requirements

- Authorized access for each particular user (log-in)
- System will be protected against session theft and malware attacks (SSL, TLP, TLS)
- Each user will be restricted to their respective functionalities.

3.4.4 Availability Requirements

- The system should be available 24x7.
- The system should achieve an up time of 99.5 %

3.4.5 Portability and Compatibility Requirements

Portability:

- All timestamps recorded by the transaction processing system shall be in UTC (Universal Time Coordinated) when placed into permanent storage.
- The time zone shall be obvious to the user whenever a time element is displayed.

Compatibility:

- System is compatible with all modern browsers.

3.4.6 Usability Requirements

Error ratio of wrong performed actions by user should not exceed 10%

3.4.7 Business Rules

Quiz can only be attended by authorized students.

3.5 Interface Requirements

3.5.1 Hardware Interfaces

There is no specific hardware used in system.

3.5.2 Software Interfaces

The tools and libraries which will be used in development of system are:

- Angular (Front-end)
- SQL Server (database)
- ASP .NET Core (framework for development of web application)

3.5.3 Communications Interfaces

- HTTP
- TCP/IP

3.6 Database Requirements

We used MS SQL for database.

3.7 Project Feasibility

The feasibility of our project is dependent on problem statement. We try to fill the technological gap between E-learning and online assessment. We incorporate the short comings of previous assessment systems. We have classified feasibility into sub-categories; Technical, Operational, and ethical feasibility.

3.7.1 Technical Feasibility

This System is a Web-based Application. We developed the system using Model View Controller (MVC) architecture, so that it can be scalable, separate implement of logics, and easy to maintain and upgrade. This system was developed by keeping in mind the requirements, risks, and security issues related to the system.

3.7.2 Operational Feasibility

This system caters the problems related to operational feasibility in such a way that it can be used as stand-alone system or can be integrated with an existing Learning Management System. (LMS) The system satisfies all the requirements mentioned in requirement document and scope of the project.

3.7.3 Legal and Ethical Feasibility

Our system fulfills its legal and ethical aspects. It cannot be used without proper authorization. The system provides security to the data.

3.8 Conclusion

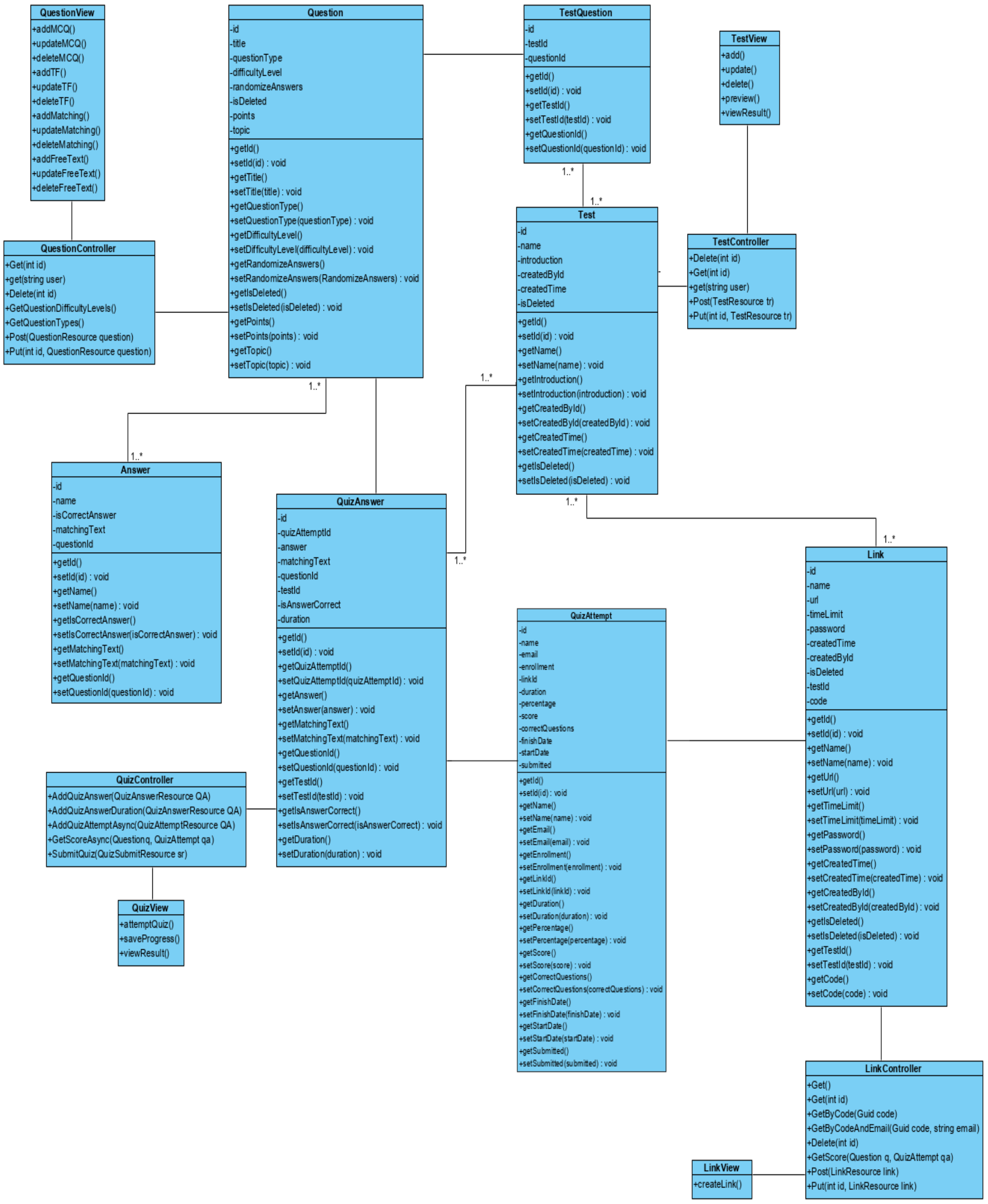
This chapter is related to system requirements of our project. We follow the Software Development Life Cycle (SDLC) for implementation. The development approach that we use was Agile. We started by first identifying the requirements, then followed by use-cases, functional, non-functional, design, and then development. If there was any change in requirements, the steps of Agile methodology were repeated.

CHAPTER-4

SYSTEM DESIGN

4. System Design

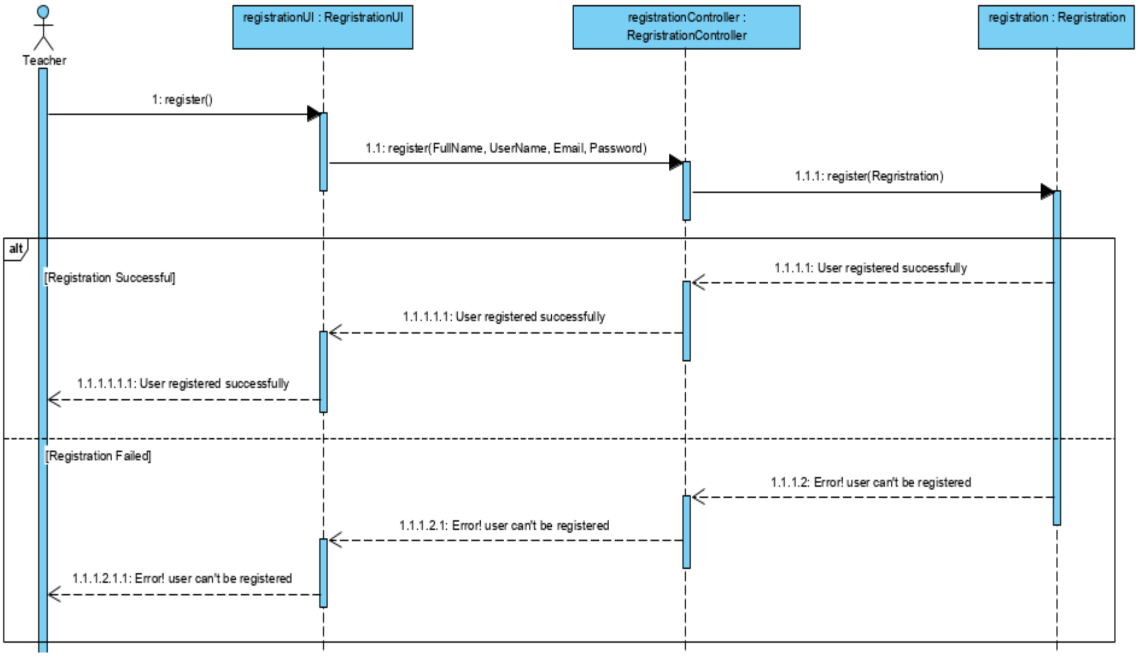
4.1 Logical View



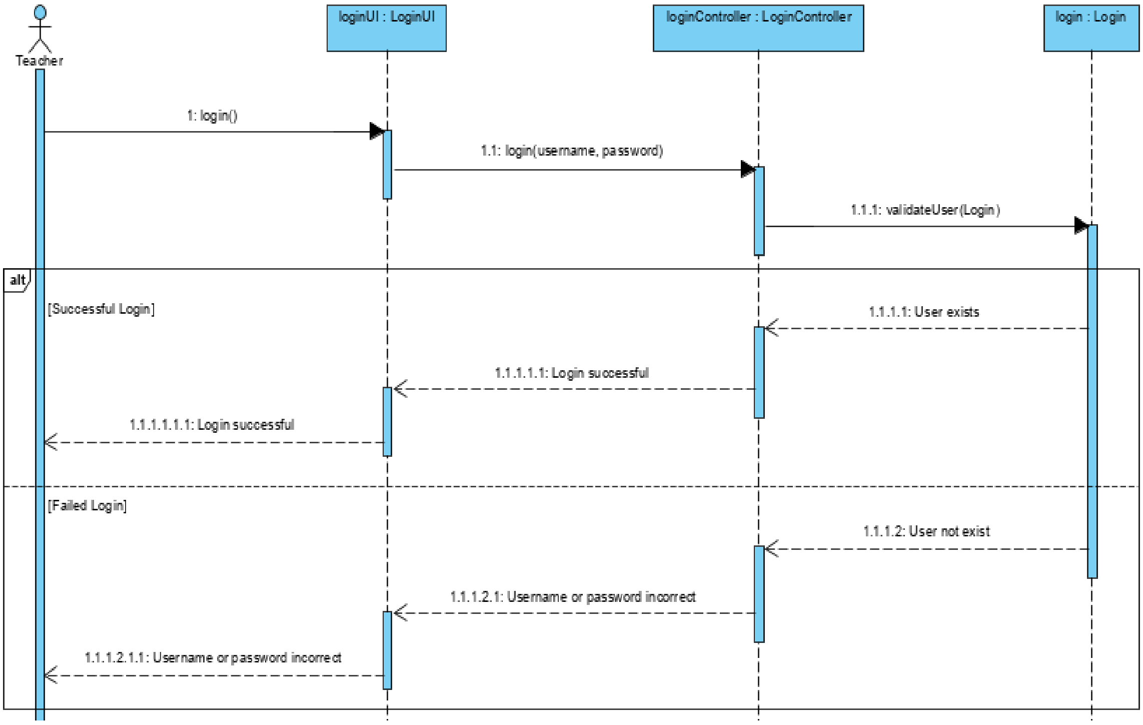
4.2 Dynamic View

4.2.1 Teacher

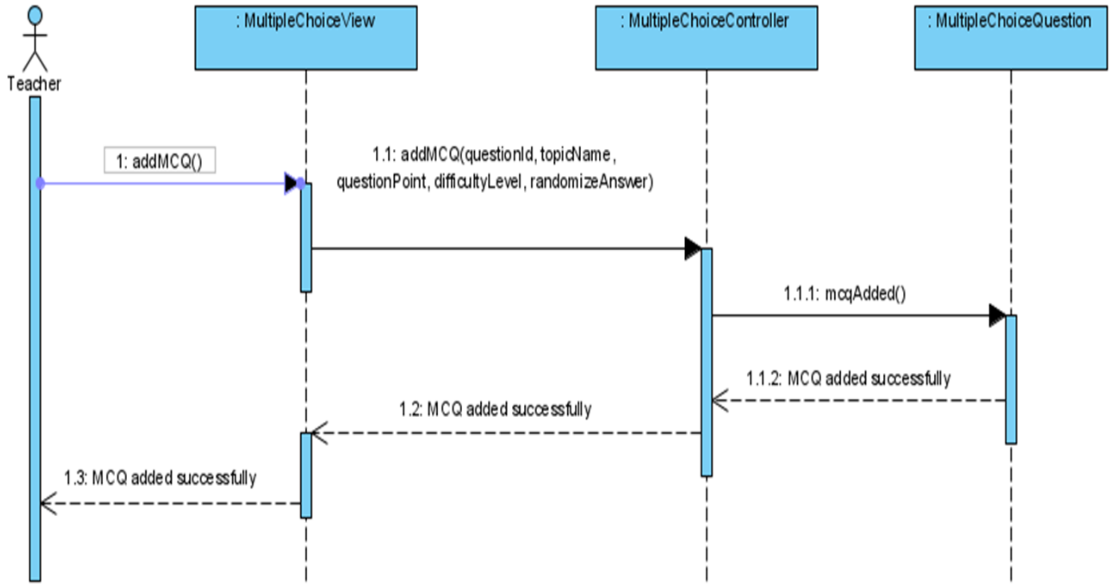
4.2.1.1 Register


Teacher
registrationUI : RegristrationUI
registrationController : RegristrationController
registration : Regristration
1: register()
1.1: register(FullName, UserName, Email, Password)
1.1.1: register(Regristration)
alt
[Registration Successful]
1.1.1.1: User registered successfully
1.1.1.1.1: User registered successfully
1.1.1.1.1.1: User registered successfully
[Registration Failed]
1.1.1.2: Error! user can't be registered
1.1.1.2.1: Error! user can't be registered
1.1.1.2.1.1: Error! user can't be registered

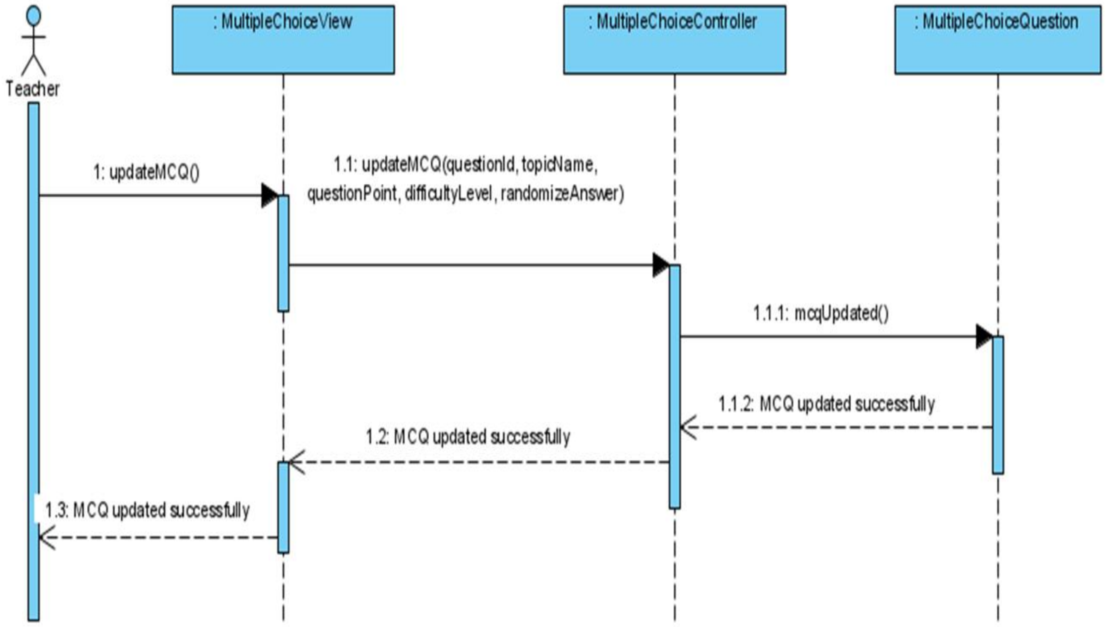
4.2.1.2 Login


Teacher
loginUI : LoginUI
loginController : LoginController
login : Login
1: login()
1.1: login(username, password)
1.1.1: validateUser(Login)
alt
[Successful Login]
1.1.1.1: User exists
1.1.1.1.1: Login successful
1.1.1.1.1.1: Login successful
[Failed Login]
1.1.1.2: User not exist
1.1.1.2.1: Username or password incorrect
1.1.1.2.1.1: Username or password incorrect

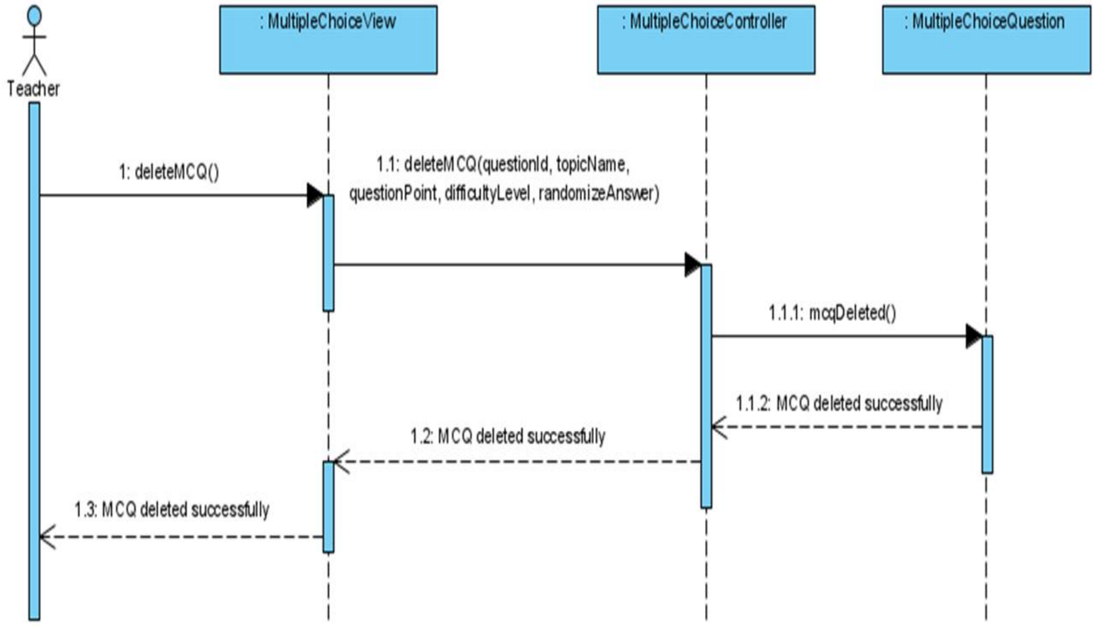
4.2.1.3 Add Multiple Choice Question


Teacher
: MultipleChoiceView
: MultipleChoiceController
: MultipleChoiceQuestion
1: addMCQ()
1.1: addMCQ(questionId, topicName, questionPoint, difficultyLevel, randomizeAnswer)
1.1.1: mcqAdded()
1.1.2: MCQ added successfully
1.2: MCQ added successfully
1.3: MCQ added successfully

4.2.1.4 Update Multiple Choice Question

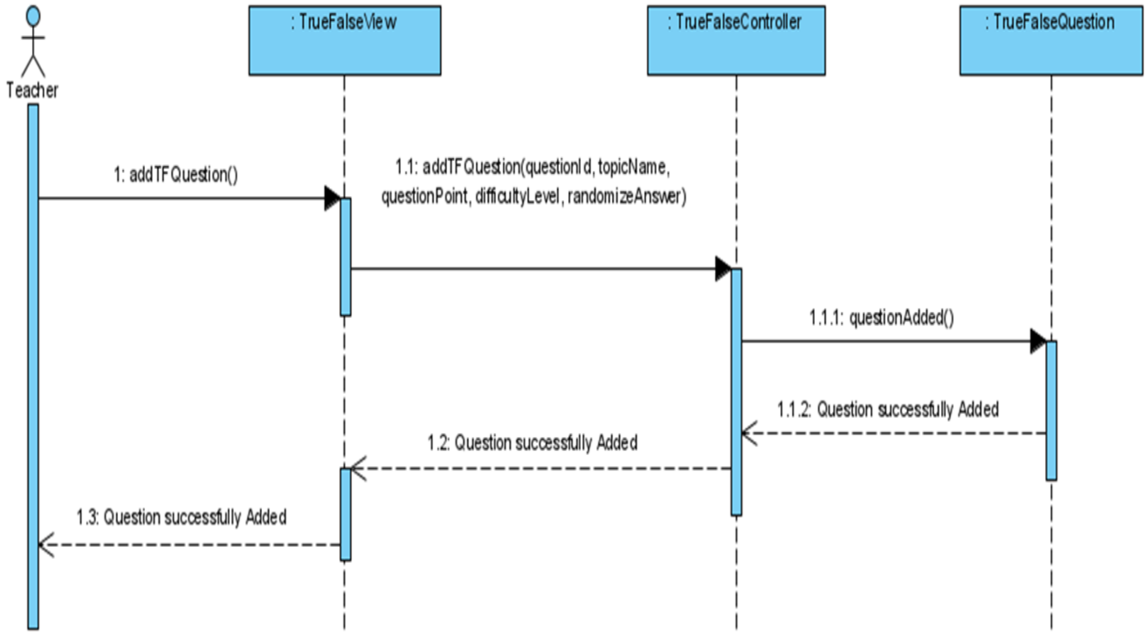

Teacher
: MultipleChoiceView
: MultipleChoiceController
: MultipleChoiceQuestion
1: updateMCQ()
1.1: updateMCQ(questionId, topicName, questionPoint, difficultyLevel, randomizeAnswer)
1.1.1: mcqUpdated()
1.1.2: MCQ updated successfully
1.2: MCQ updated successfully
1.3: MCQ updated successfully

4.2.1.5 Delete Multiple Choice Question



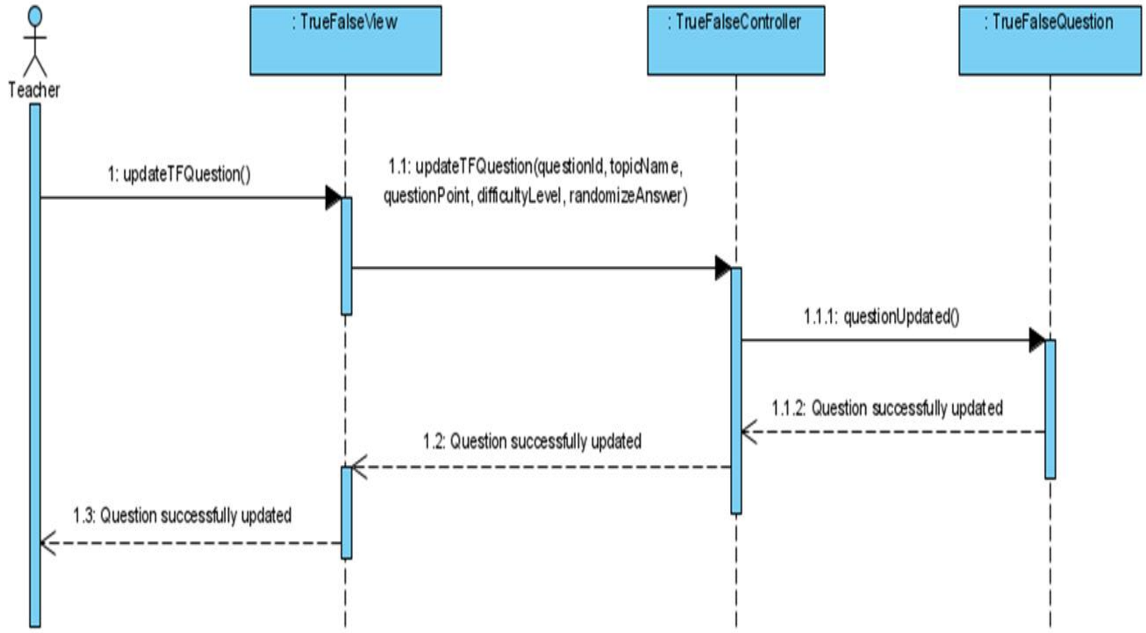
Teacher
: MultipleChoiceView
: MultipleChoiceController
: MultipleChoiceQuestion
1: deleteMCQ()
1.1: deleteMCQ(questionId, topicName, questionPoint, difficultyLevel, randomizeAnswer)
1.1.1: mcqDeleted()
1.1.2: MCQ deleted successfully
1.2: MCQ deleted successfully
1.3: MCQ deleted successfully

4.2.1.6 Add True False Question



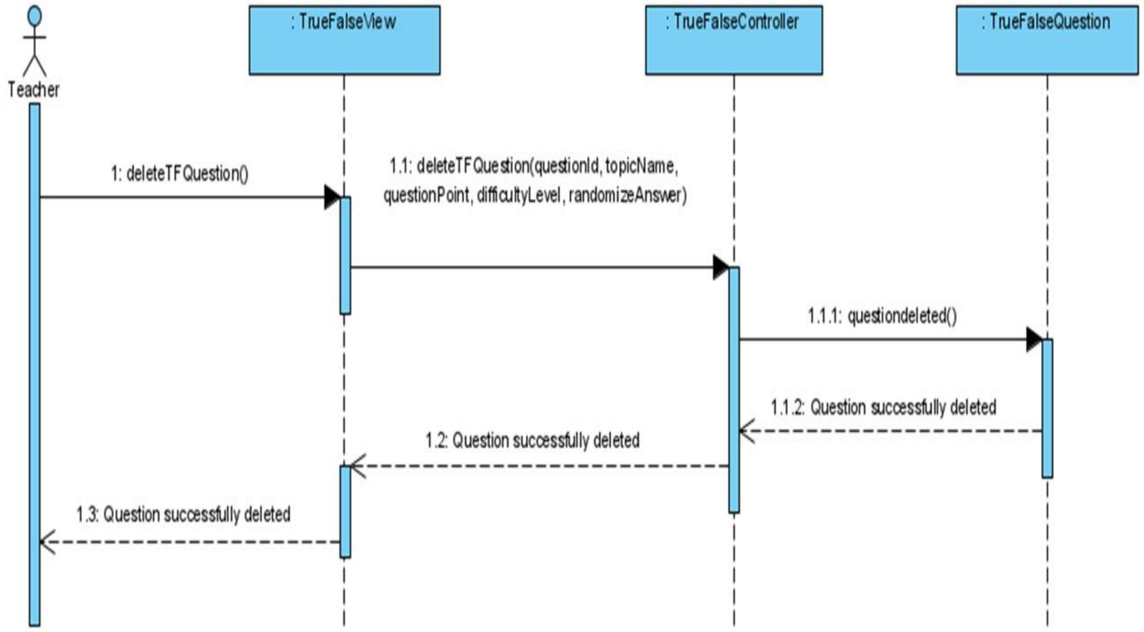
Teacher
: TrueFalseView
: TrueFalseController
: TrueFalseQuestion
1: addTFQuestion()
1.1: addTFQuestion(questionId, topicName, questionPoint, difficultyLevel, randomizeAnswer)
1.1.1: questionAdded()
1.1.2: Question successfully Added
1.2: Question successfully Added
1.3: Question successfully Added

4.2.1.7 Update True False Question



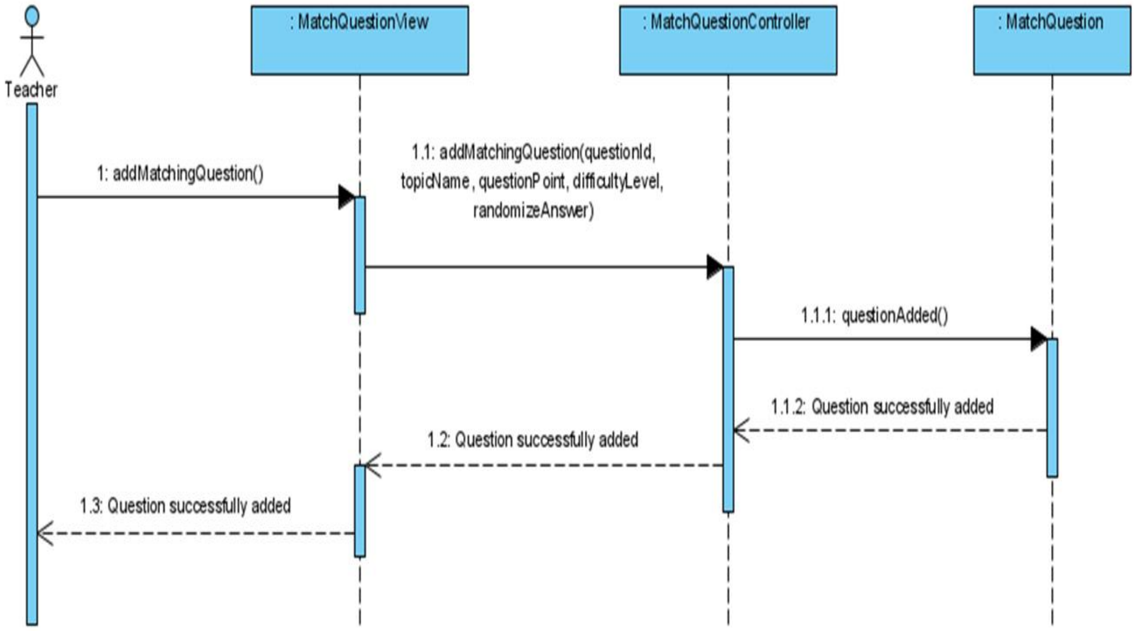
Teacher
: TrueFalseView
: TrueFalseController
: TrueFalseQuestion
1: updateTFQuestion()
1.1: updateTFQuestion(questionId, topicName, questionPoint, difficultyLevel, randomizeAnswer)
1.1.1: questionUpdated()
1.1.2: Question successfully updated
1.2: Question successfully updated
1.3: Question successfully updated

4.2.1.8 Delete True False Question

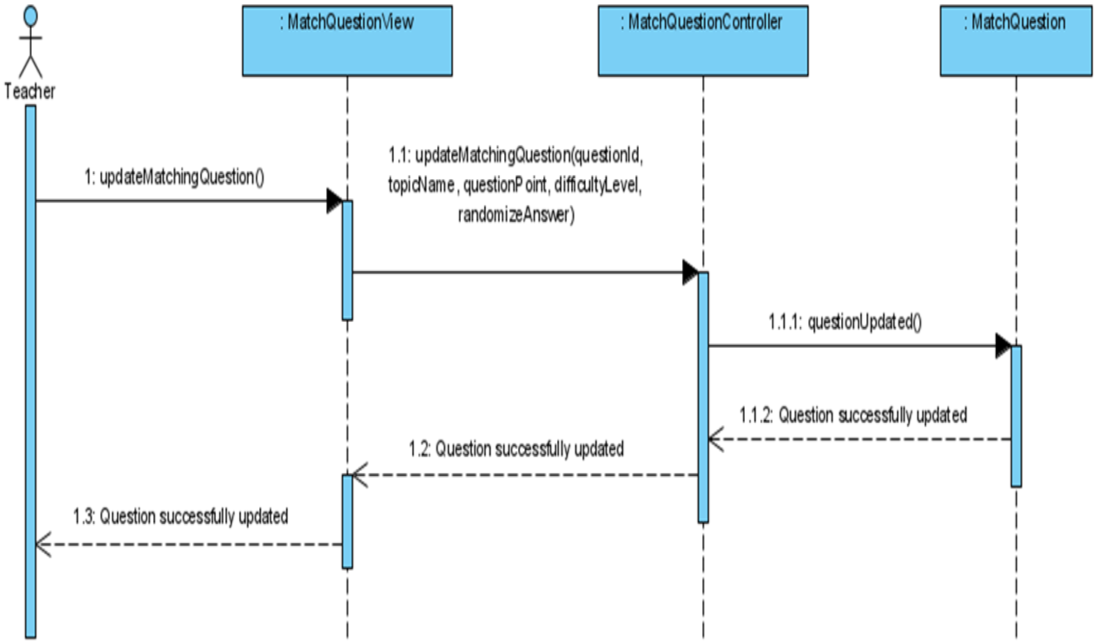


Teacher
: TrueFalseView
: TrueFalseController
: TrueFalseQuestion
1: deleteTFQuestion()
1.1: deleteTFQuestion(questionId, topicName, questionPoint, difficultyLevel, randomizeAnswer)
1.1.1: questiondeleted()
1.1.2: Question successfully deleted
1.2: Question successfully deleted
1.3: Question successfully deleted

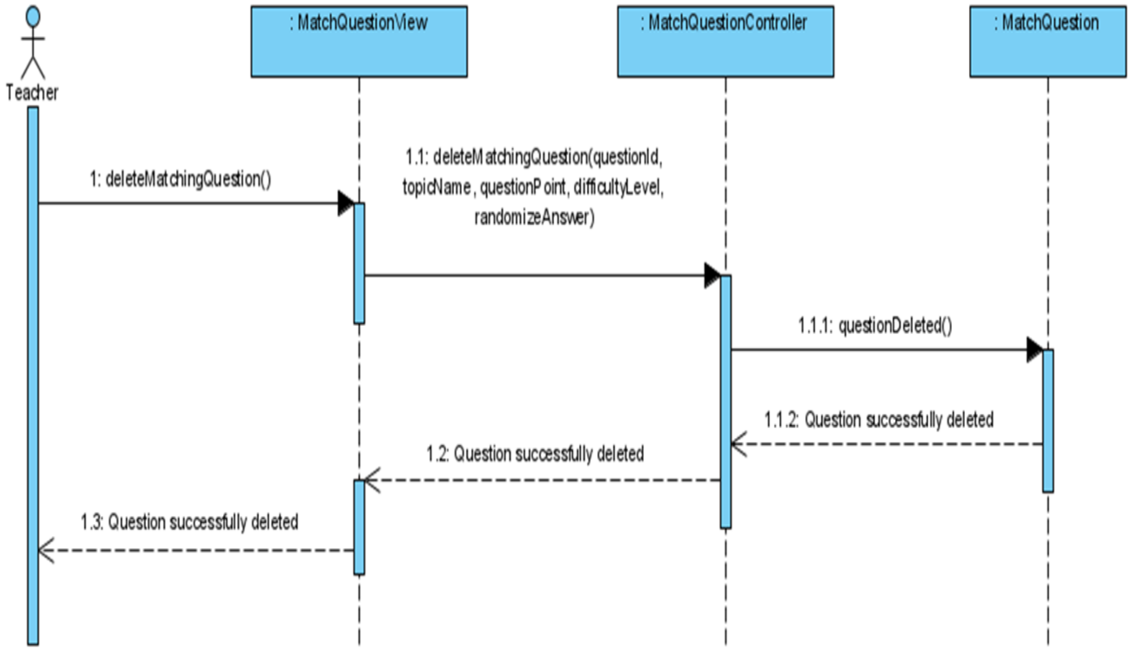
4.2.1.9 Add Matching Question


Teacher
: MatchQuestionView
: MatchQuestionController
: MatchQuestion
1: addMatchingQuestion()
1.1: addMatchingQuestion(questionId, topicName, questionPoint, difficultyLevel, randomizeAnswer)
1.1.1: questionAdded()
1.1.2: Question successfully added
1.2: Question successfully added
1.3: Question successfully added

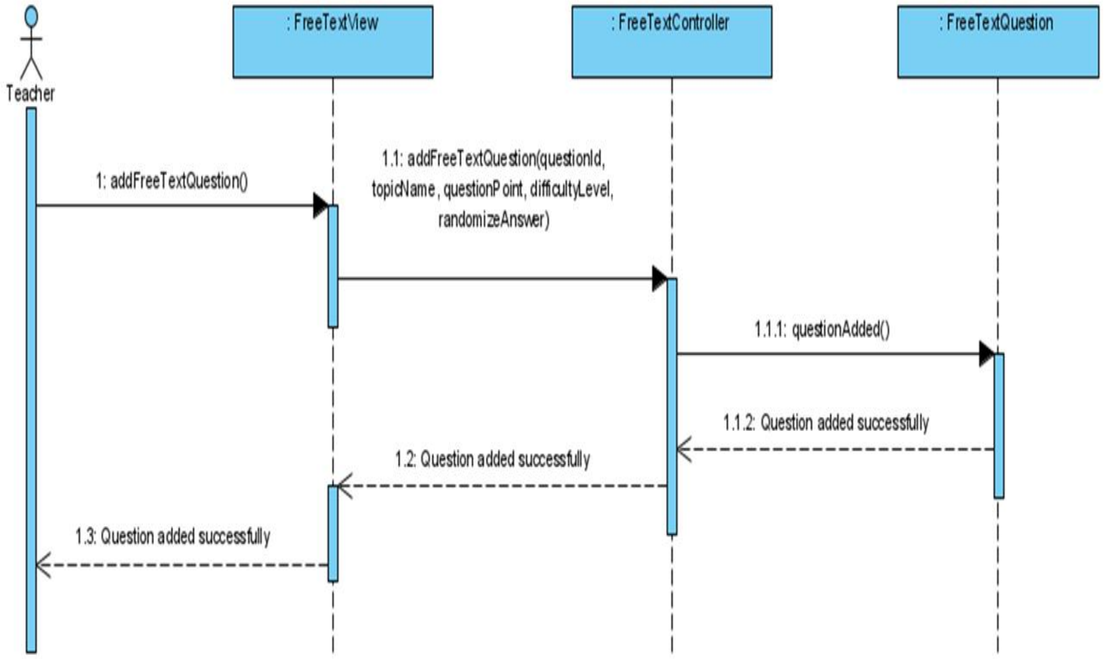
4.2.1.10 Update Matching Question


Teacher
: MatchQuestionView
: MatchQuestionController
: MatchQuestion
1: updateMatchingQuestion()
1.1: updateMatchingQuestion(questionId, topicName, questionPoint, difficultyLevel, randomizeAnswer)
1.1.1: questionUpdated()
1.1.2: Question successfully updated
1.2: Question successfully updated
1.3: Question successfully updated

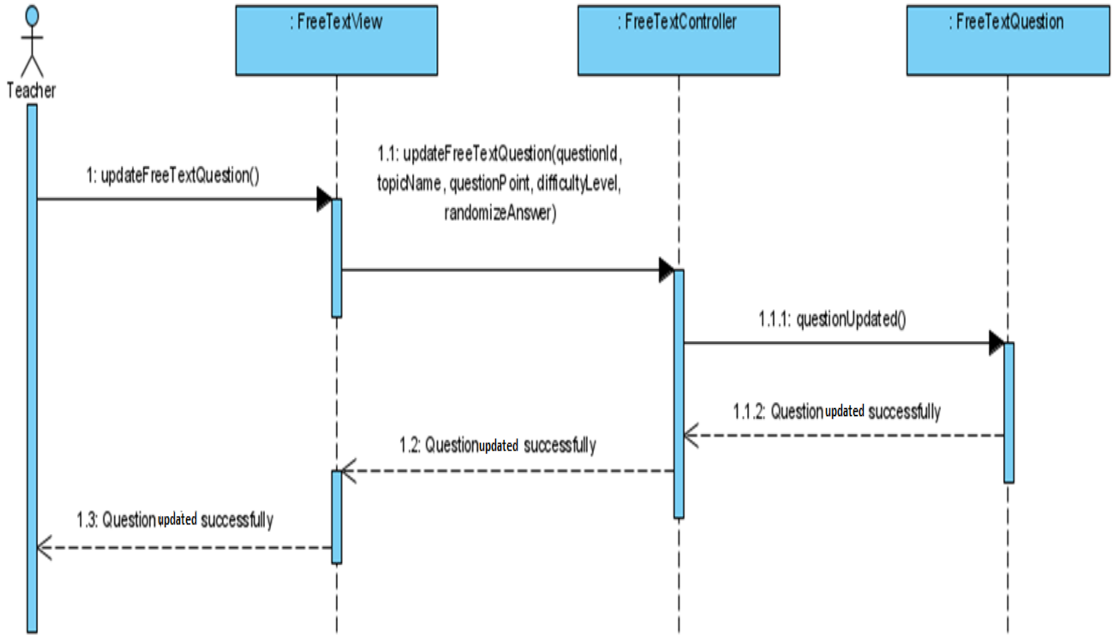
4.2.1.11 Delete Matching Question


Teacher
: MatchQuestionView
: MatchQuestionController
: MatchQuestion
1: deleteMatchingQuestion()
1.1: deleteMatchingQuestion(questionId, topicName, questionPoint, difficultyLevel, randomizeAnswer)
1.1.1: questionDeleted()
1.1.2: Question successfully deleted
1.2: Question successfully deleted
1.3: Question successfully deleted

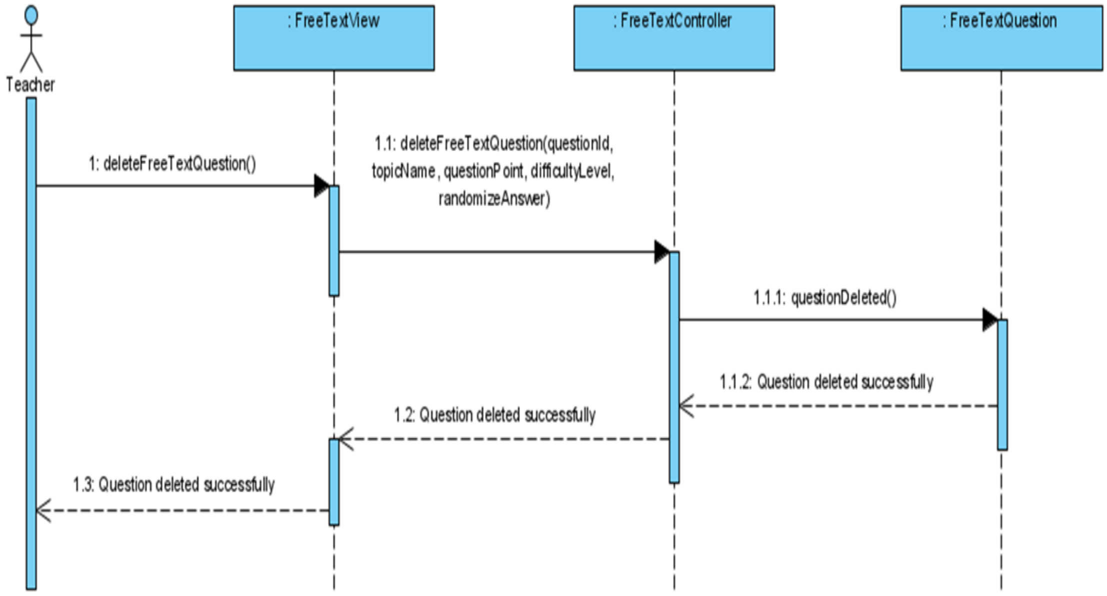
4.2.1.12 Add Free Text Question


Teacher
: FreeTextView
: FreeTextController
: FreeTextQuestion
1: addFreeTextQuestion()
1.1: addFreeTextQuestion(questionId, topicName, questionPoint, difficultyLevel, randomizeAnswer)
1.1.1: questionAdded()
1.1.2: Question added successfully
1.2: Question added successfully
1.3: Question added successfully

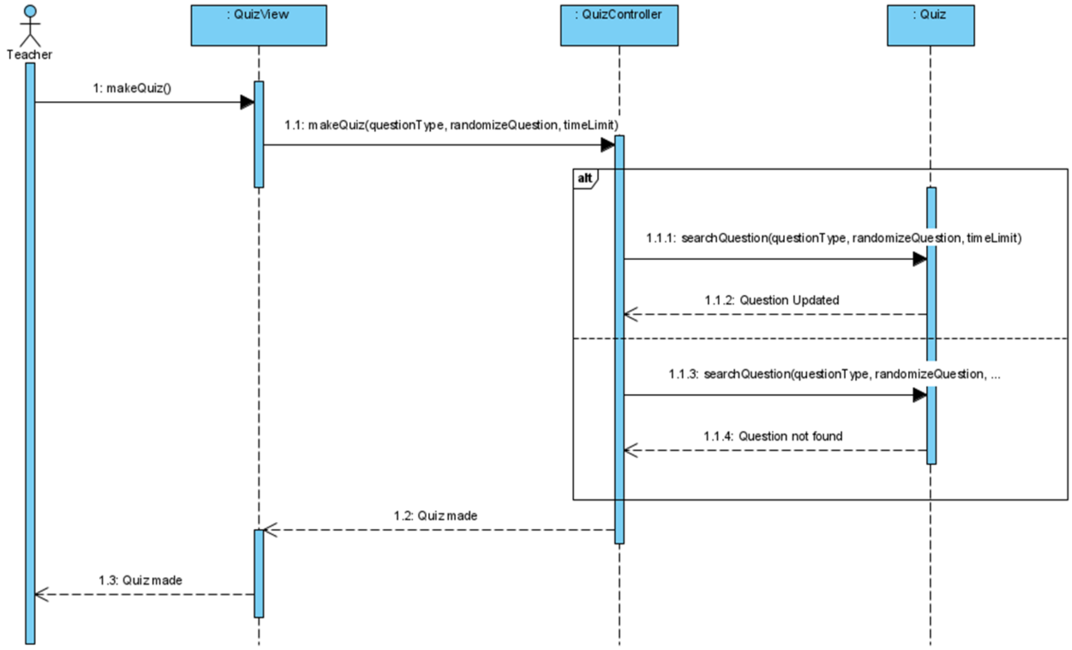
4.2.1.13 Update Free Text Question


Teacher
: FreeTextView
: FreeTextController
: FreeTextQuestion
1: updateFreeTextQuestion()
1.1: updateFreeTextQuestion(questionId, topicName, questionPoint, difficultyLevel, randomizeAnswer)
1.1.1: questionUpdated()
1.1.2: Question updated successfully
1.2: Question updated successfully
1.3: Question updated successfully

4.2.1.14 Delete Free Text Question

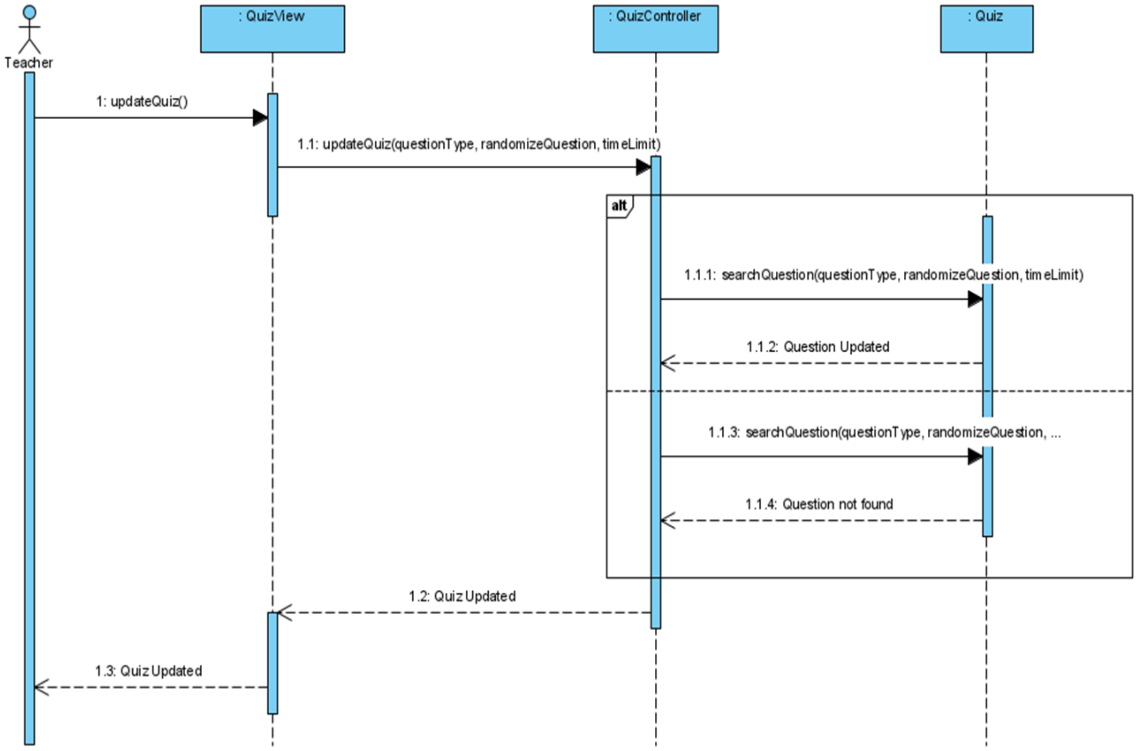

Teacher
: FreeTextView
: FreeTextController
: FreeTextQuestion
1: deleteFreeTextQuestion()
1.1: deleteFreeTextQuestion(questionId, topicName, questionPoint, difficultyLevel, randomizeAnswer)
1.1.1: questionDeleted()
1.1.2: Question deleted successfully
1.2: Question deleted successfully
1.3: Question deleted successfully

4.2.1.15 Make Quiz



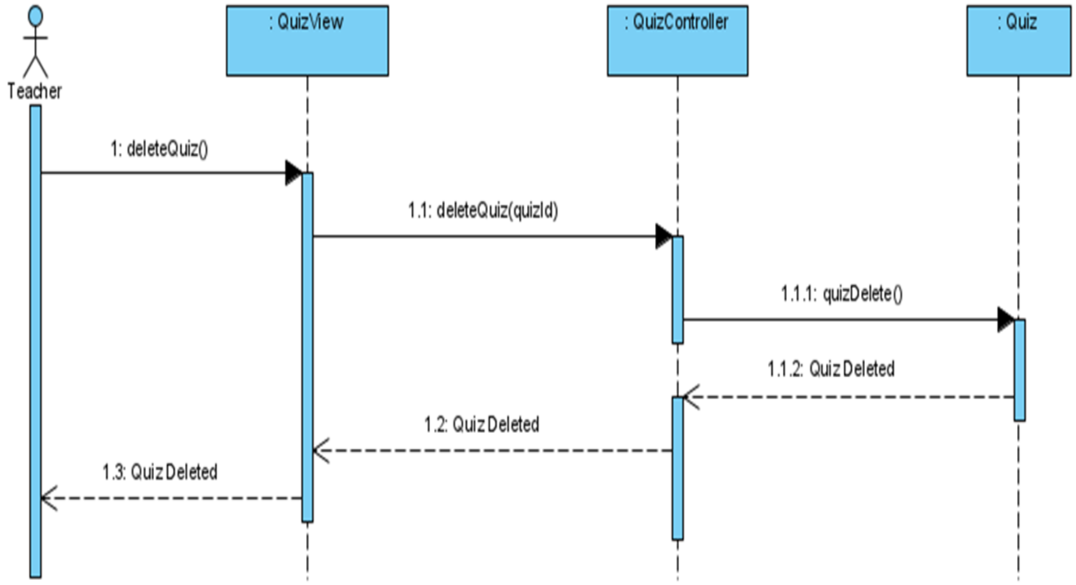
Teacher
: QuizView
: QuizController
: Quiz
1: makeQuiz()
1.1: makeQuiz(questionType, randomizeQuestion, timeLimit)
alt
1.1.1: searchQuestion(questionType, randomizeQuestion, timeLimit)
1.1.2: Question Updated
1.1.3: searchQuestion(questionType, randomizeQuestion, ...
1.1.4: Question not found
1.2: Quiz made
1.3: Quiz made

4.2.1.16 Update Quiz

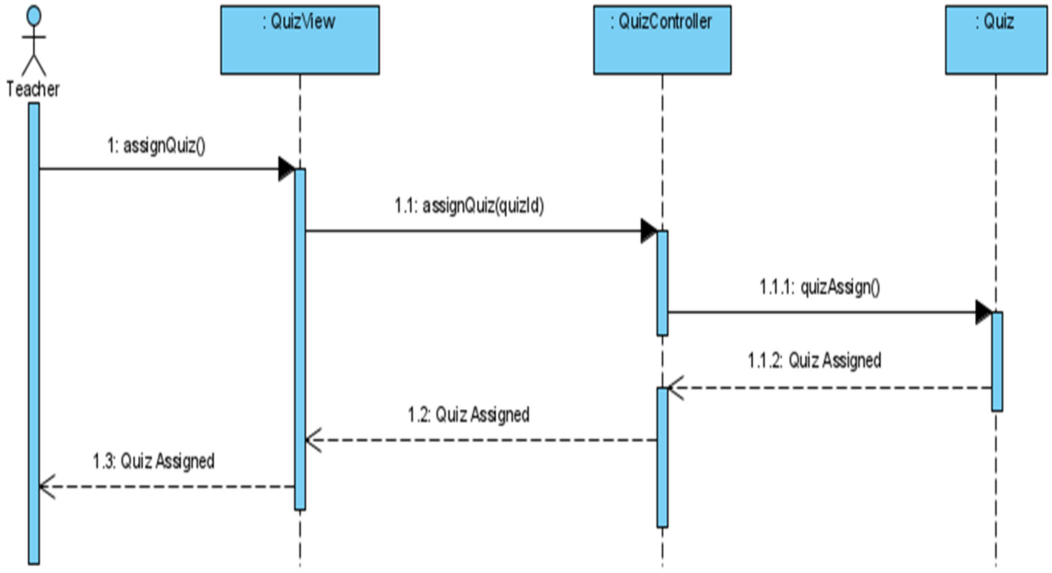


Teacher
: QuizView
: QuizController
: Quiz
1: updateQuiz()
1.1: updateQuiz(questionType, randomizeQuestion, timeLimit)
alt
1.1.1: searchQuestion(questionType, randomizeQuestion, timeLimit)
1.1.2: Question Updated
1.1.3: searchQuestion(questionType, randomizeQuestion, ...
1.1.4: Question not found
1.2: Quiz Updated
1.3: Quiz Updated

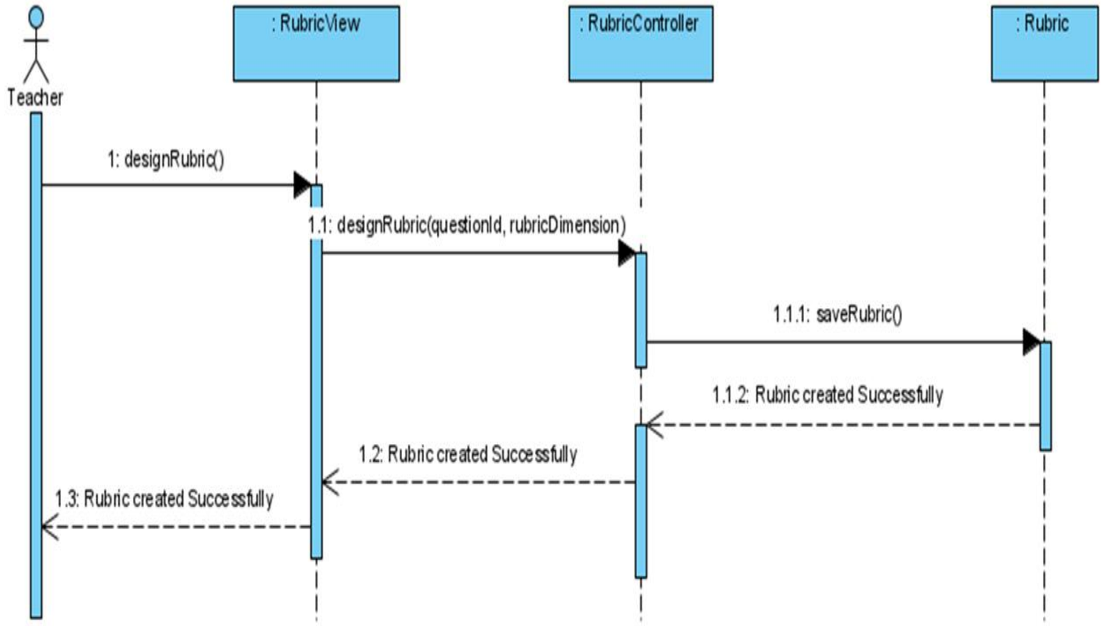
4.2.1.17 Delete Quiz



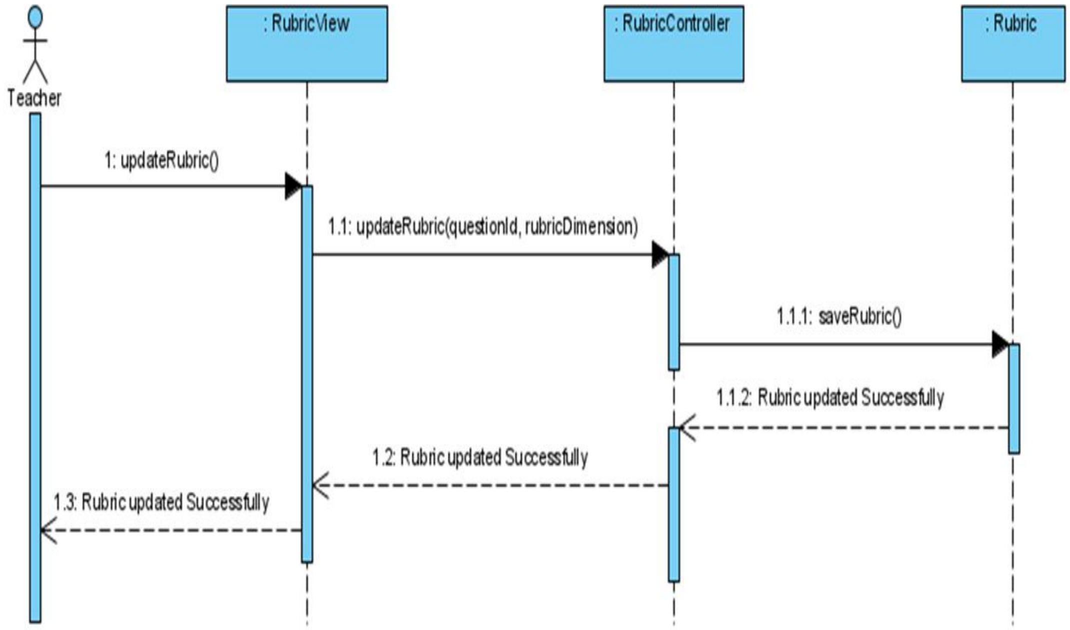
4.2.1.18 Assign Quiz



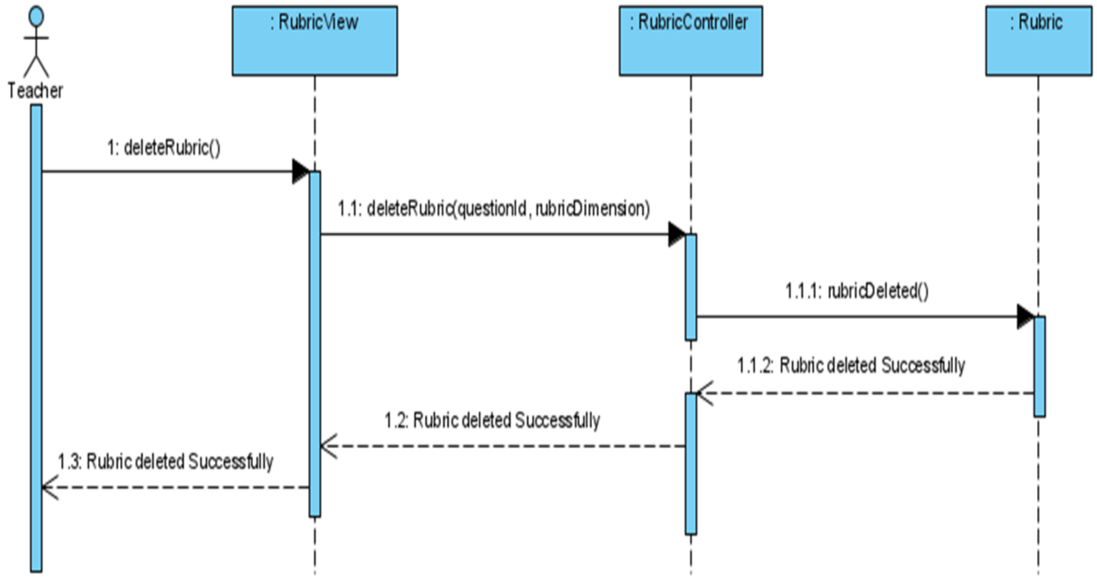
4.2.1.19 Design Rubric


Teacher
: RubricView
: RubricController
: Rubric
1: designRubric()
1.1: designRubric(questionId, rubricDimension)
1.1.1: saveRubric()
1.1.2: Rubric created Successfully
1.2: Rubric created Successfully
1.3: Rubric created Successfully

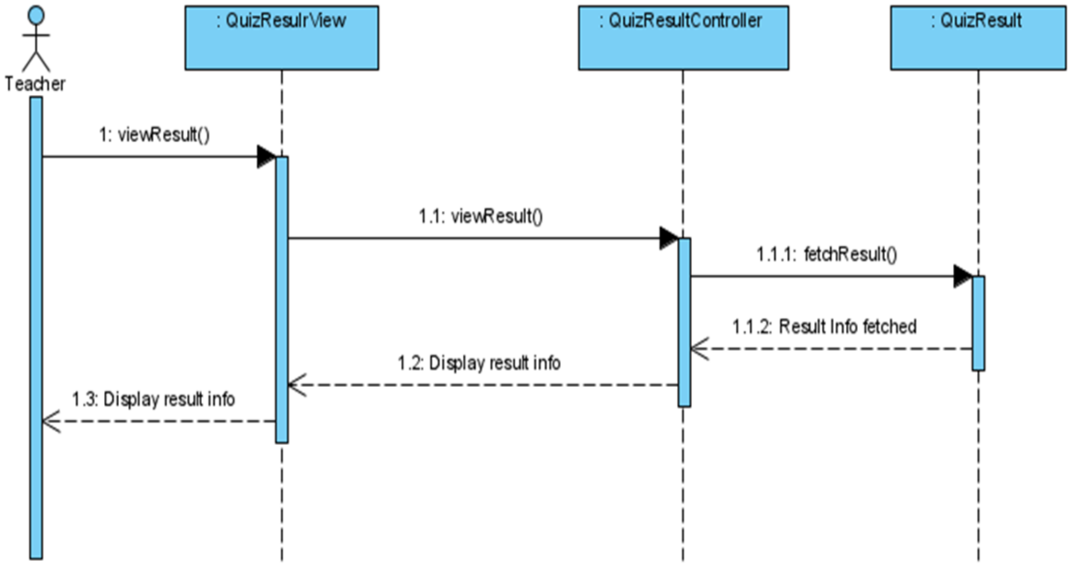
4.2.1.20 Update Rubric


Teacher
: RubricView
: RubricController
: Rubric
1: updateRubric()
1.1: updateRubric(questionId, rubricDimension)
1.1.1: saveRubric()
1.1.2: Rubric updated Successfully
1.2: Rubric updated Successfully
1.3: Rubric updated Successfully

4.2.1.21 Delete Rubric

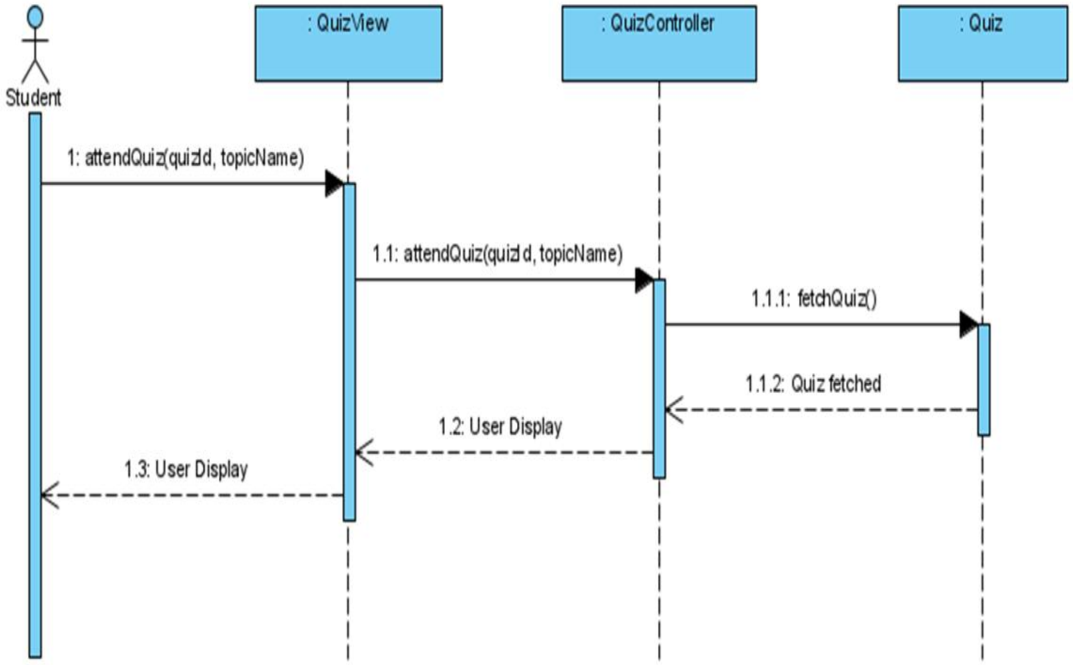

Teacher
: RubricView
: RubricController
: Rubric
1: deleteRubric()
1.1: deleteRubric(questionId, rubricDimension)
1.1.1: rubricDeleted()
1.1.2: Rubric deleted Successfully
1.2: Rubric deleted Successfully
1.3: Rubric deleted Successfully

4.2.1.22 View Result

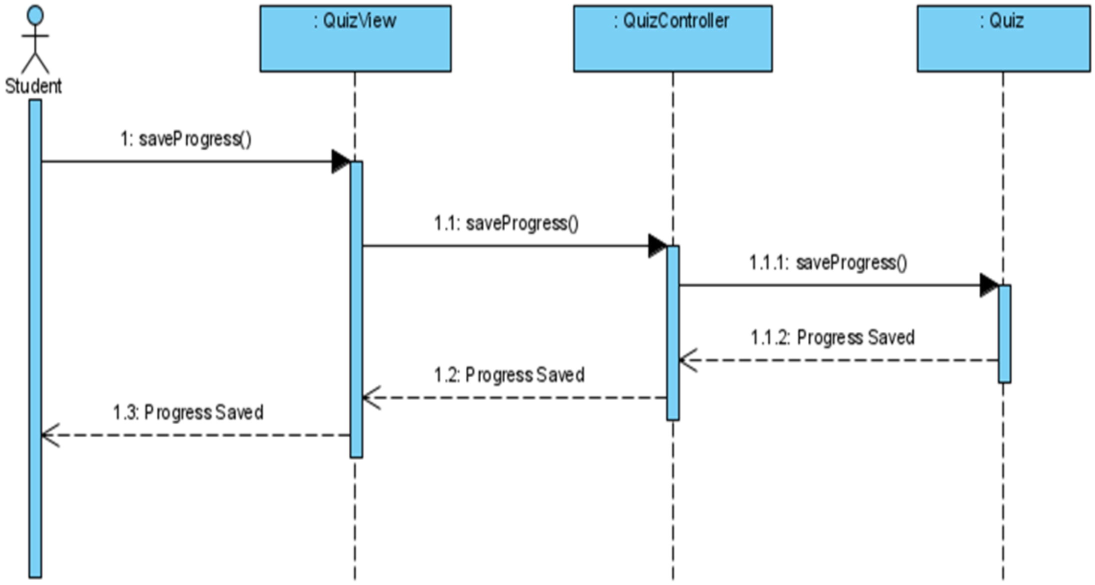


4.2.2 Student

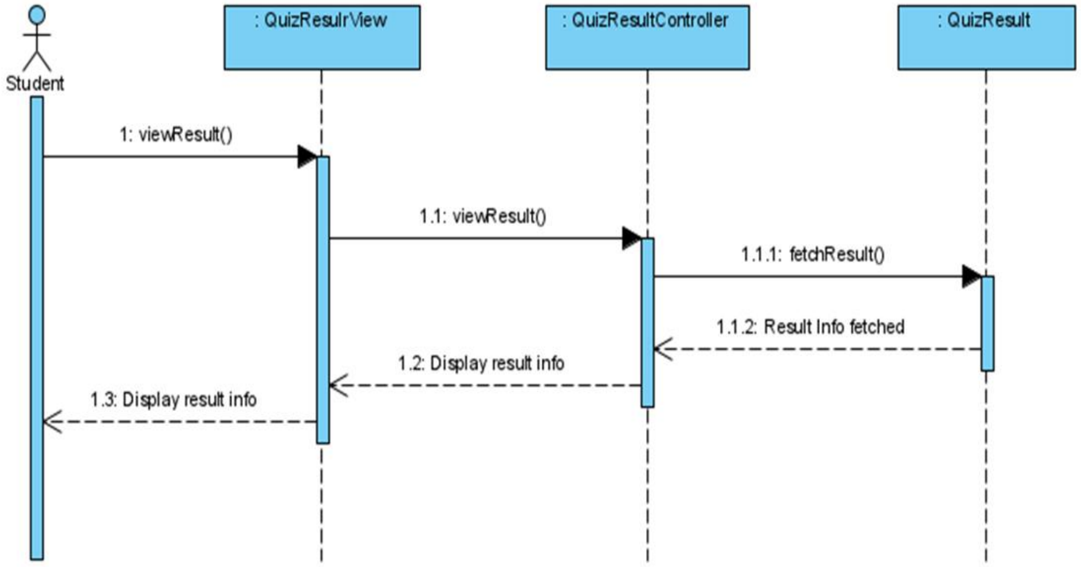
4.2.2.1 Attend Quiz



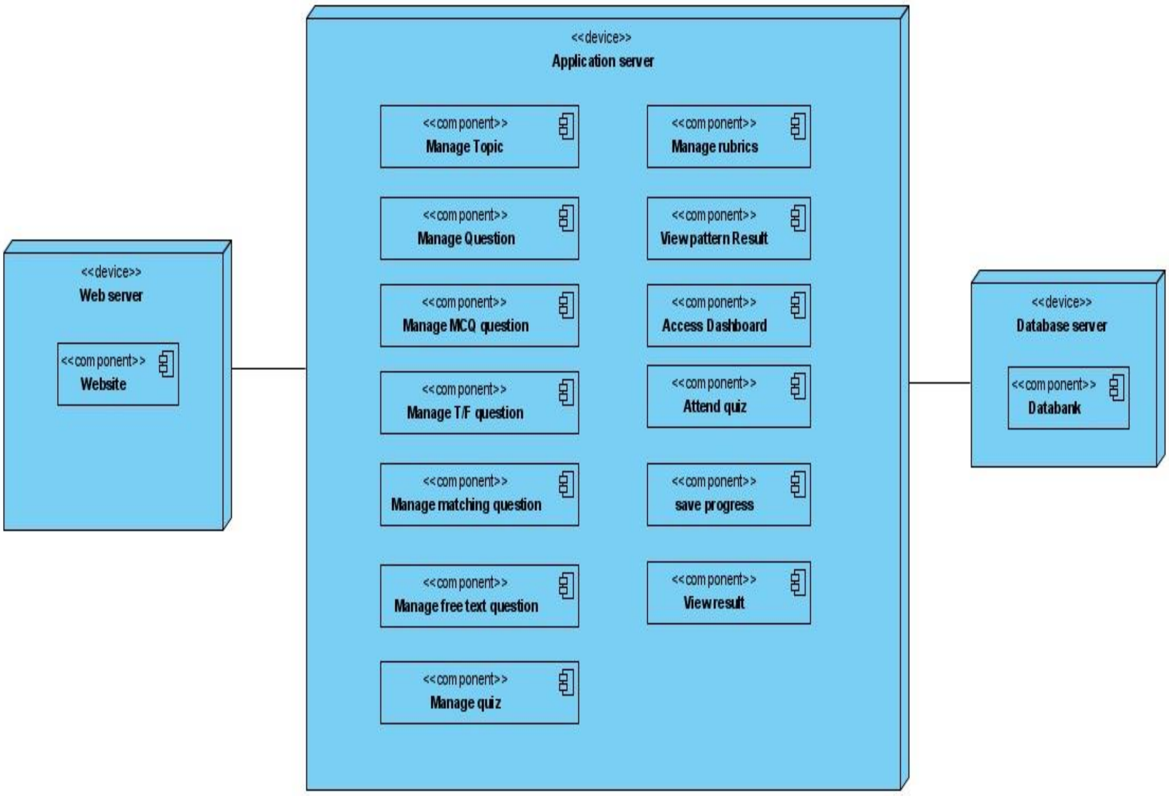
4.2.2.2 Save Progress



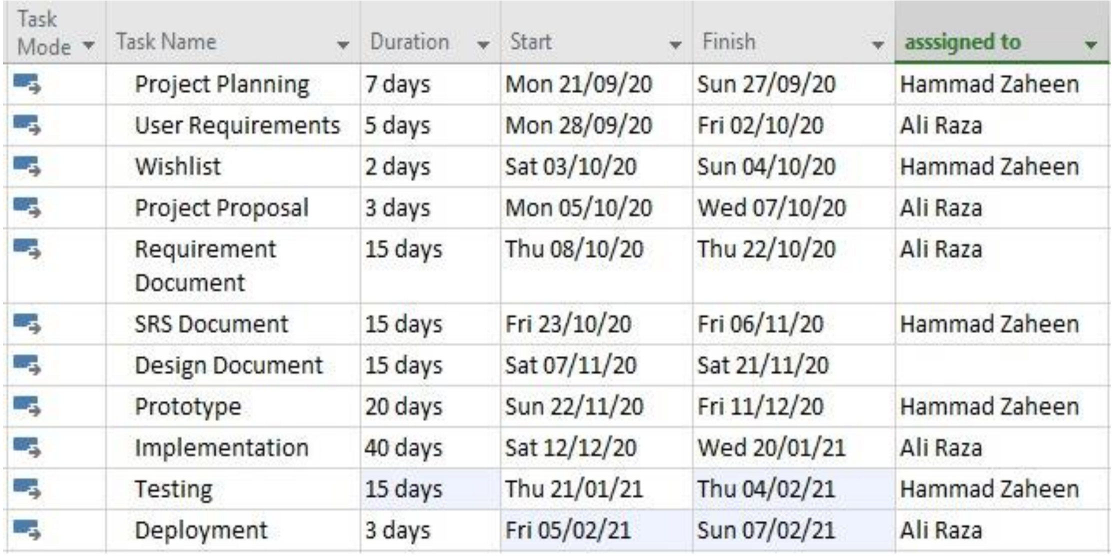
4.2.2.3 View Result



4.3 Deployment View

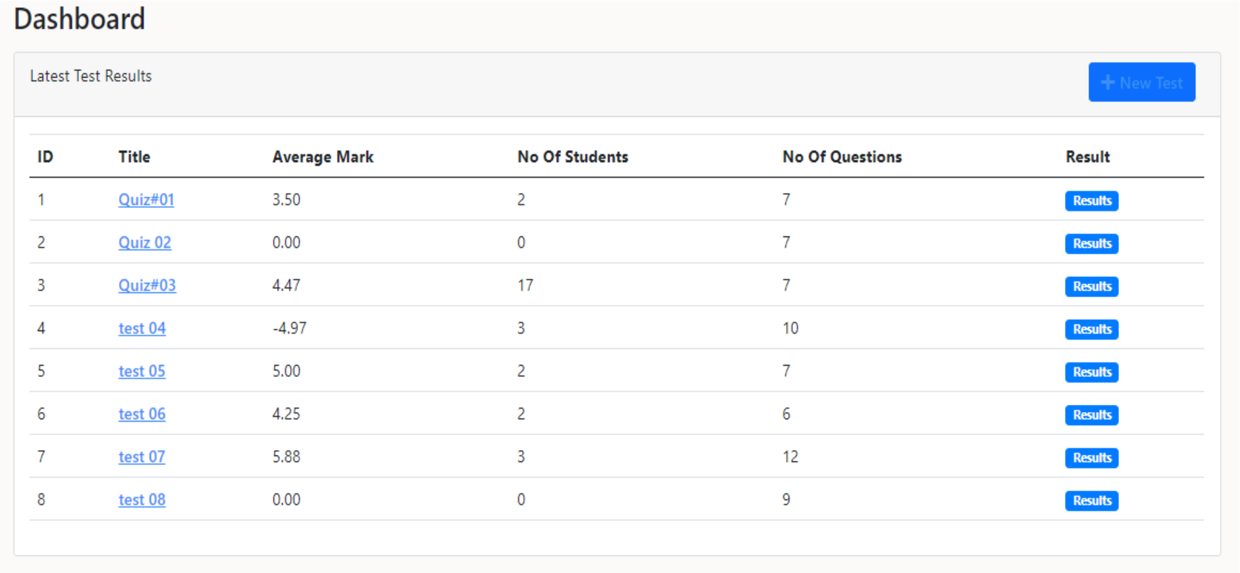


4.4 Development View



Task Mode	Task Name	Duration	Start	Finish	assigned to
	Project Planning	7 days	Mon 21/09/20	Sun 27/09/20	Hammad Zaheen
	User Requirements	5 days	Mon 28/09/20	Fri 02/10/20	Ali Raza
	Wishlist	2 days	Sat 03/10/20	Sun 04/10/20	Hammad Zaheen
	Project Proposal	3 days	Mon 05/10/20	Wed 07/10/20	Ali Raza
	Requirement Document	15 days	Thu 08/10/20	Thu 22/10/20	Ali Raza
	SRS Document	15 days	Fri 23/10/20	Fri 06/11/20	Hammad Zaheen
	Design Document	15 days	Sat 07/11/20	Sat 21/11/20	
	Prototype	20 days	Sun 22/11/20	Fri 11/12/20	Hammad Zaheen
	Implementation	40 days	Sat 12/12/20	Wed 20/01/21	Ali Raza
	Testing	15 days	Thu 21/01/21	Thu 04/02/21	Hammad Zaheen
	Deployment	3 days	Fri 05/02/21	Sun 07/02/21	Ali Raza

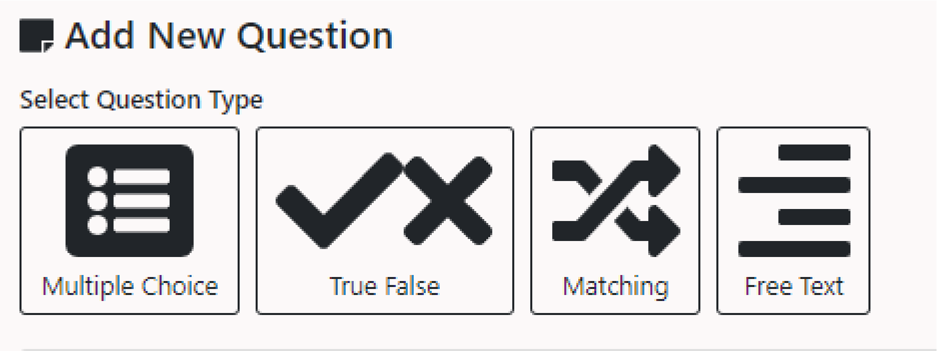
4.5 System Prototype



Dashboard

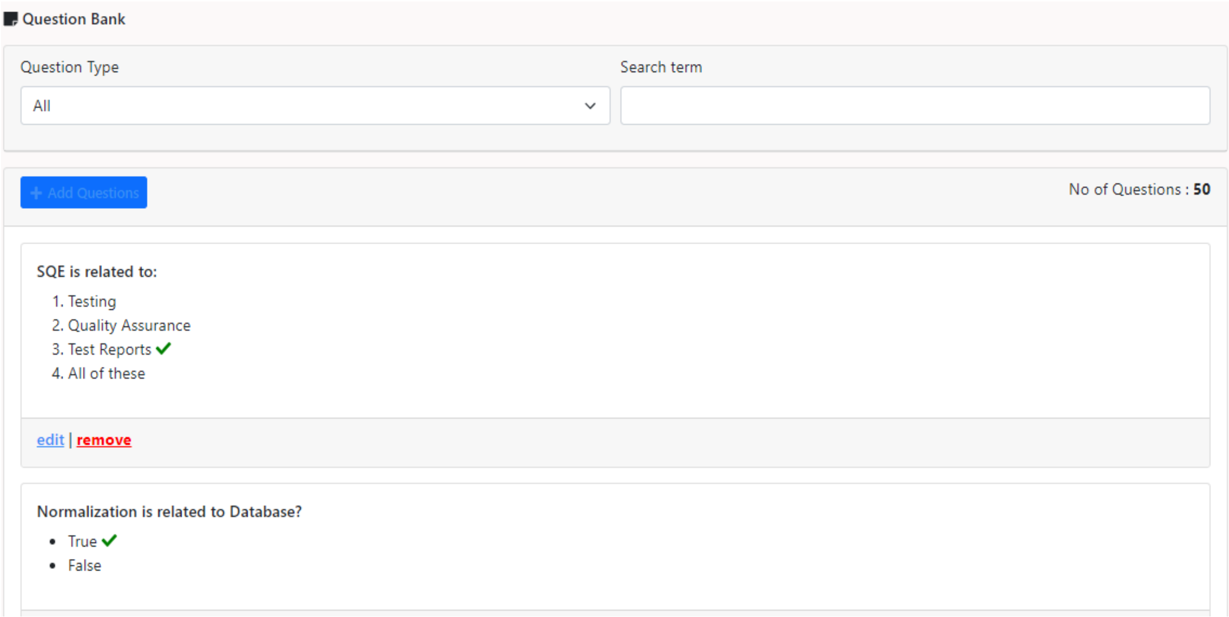
Latest Test Results

ID	Title	Average Mark	No Of Students	No Of Questions	Result
1	Quiz#01	3.50	2	7	Results
2	Quiz 02	0.00	0	7	Results
3	Quiz#03	4.47	17	7	Results
4	test 04	-4.97	3	10	Results
5	test 05	5.00	2	7	Results
6	test 06	4.25	2	6	Results
7	test 07	5.88	3	12	Results
8	test 08	0.00	0	9	Results



Add New Question

Select Question Type



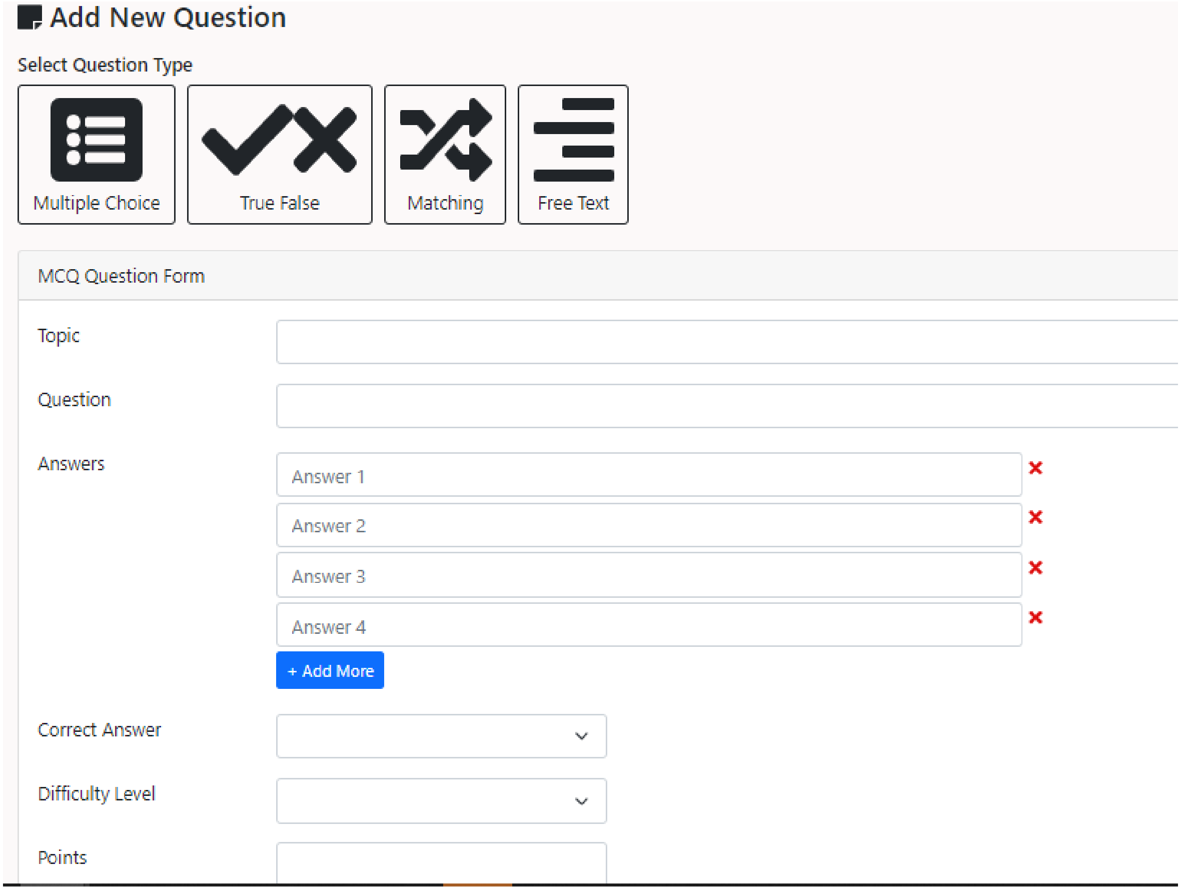
No of Questions : **50**

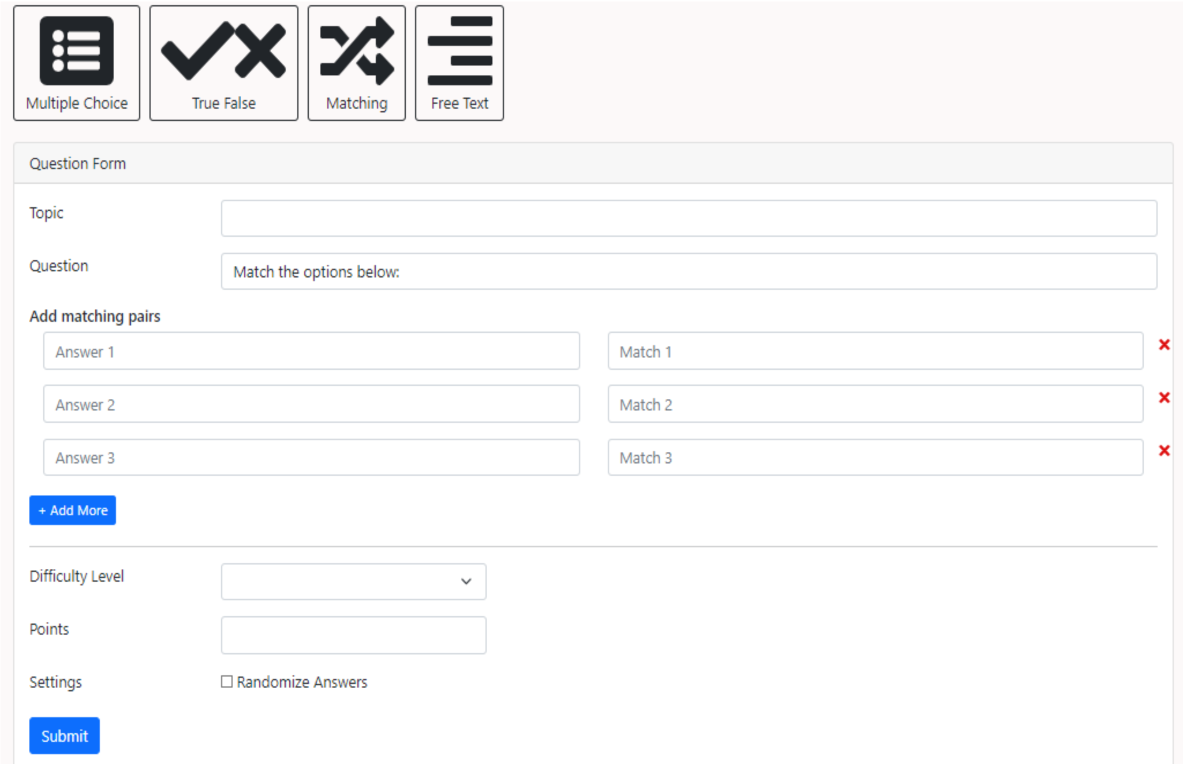
SQE is related to:

1. Testing
2. Quality Assurance
3. Test Reports ✓
4. All of these

Normalization is related to Database?

- True ✓
- False





Question Form

Topic

Question: Match the options below:

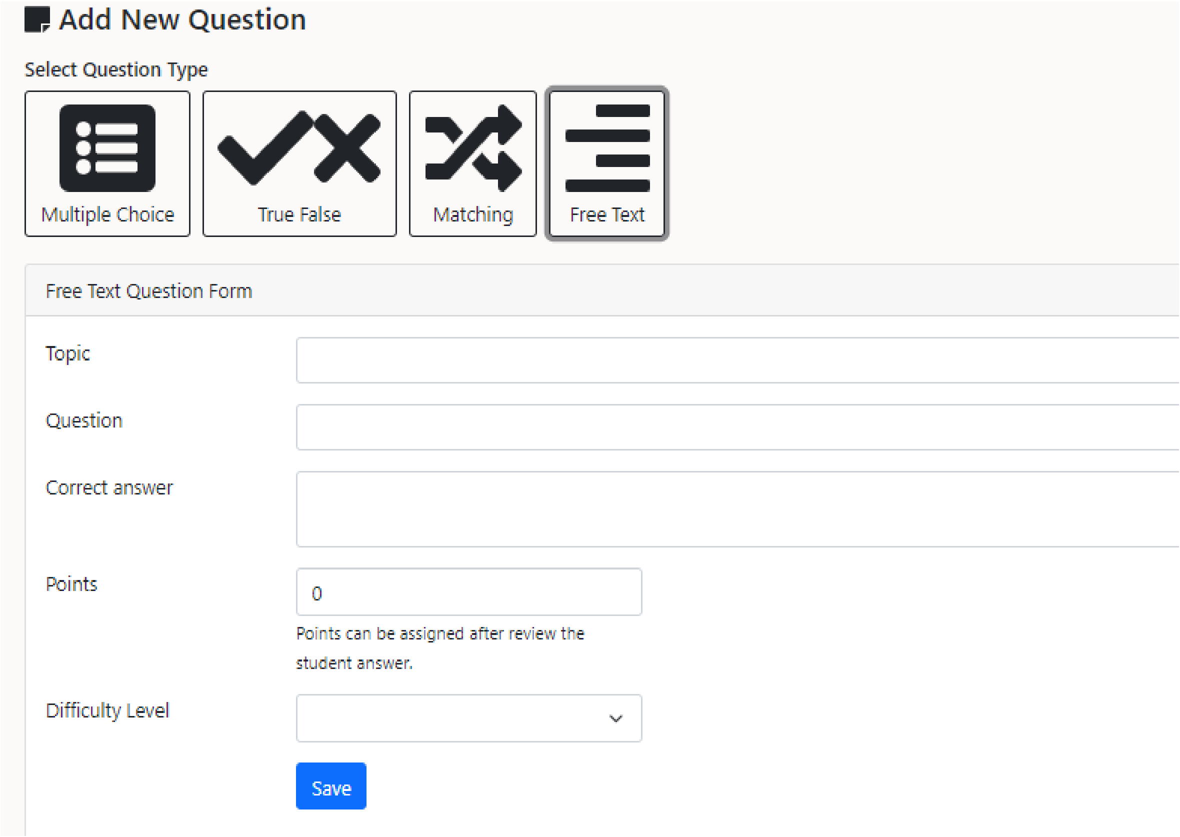
Add matching pairs

Answer 1	Match 1	×
Answer 2	Match 2	×
Answer 3	Match 3	×

Difficulty Level

Points

Settings ☐ Randomize Answers



Add New Question

Select Question Type

Free Text Question Form

Topic

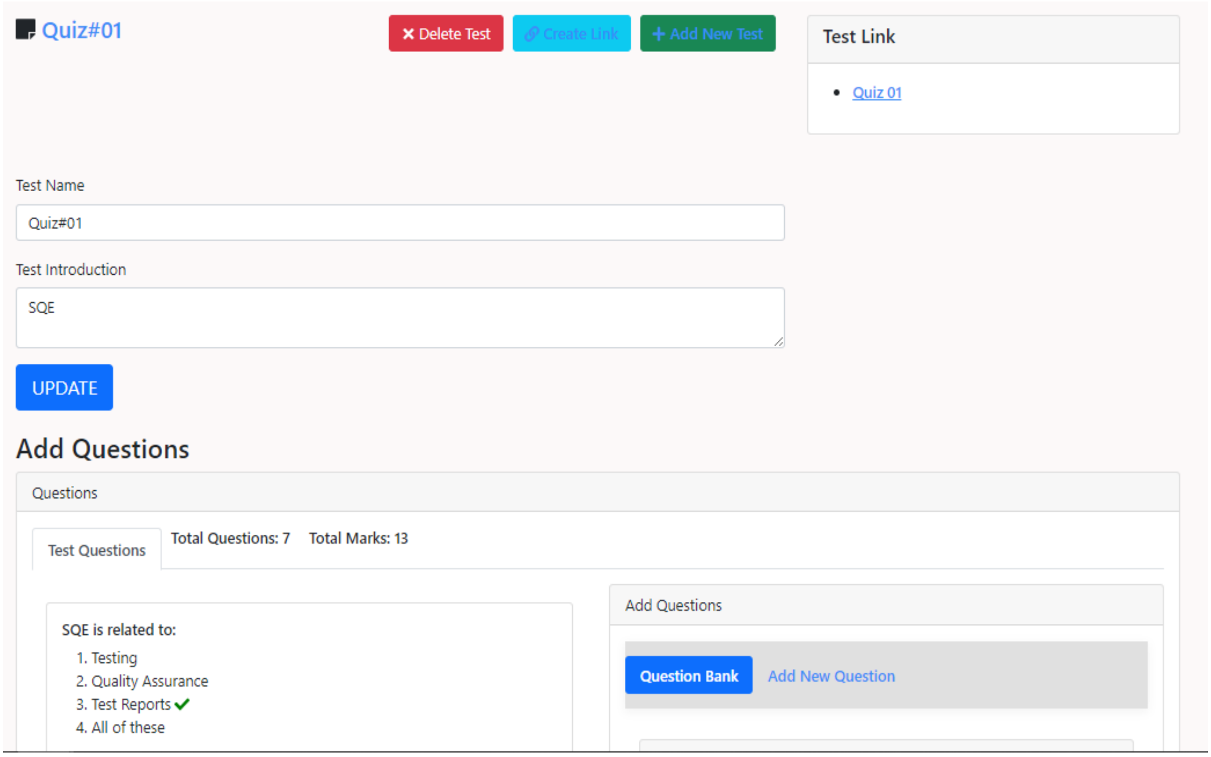
Question

Correct answer

Points: 0

Points can be assigned after review the student answer.

Difficulty Level



Test Name: Quiz#01

Test Introduction: SQE

Total Questions: 7 Total Marks: 13

SQE is related to:

1. Testing
2. Quality Assurance
3. Test Reports ✓
4. All of these

4.6 Conclusion

This chapter is about the design document of project. This chapter contain all the diagrams/figures related to the dynamic view of the system including class, sequence, deployment, and system prototypes. An Agile SDLC was followed here in which the change requests are first referred back to the Software Requirements Specification (SRS) document, and then the changes are incorporated in the dynamic structure of the project.

CHAPTER-5

SYSTEM IMPLEMENTATION

5. System Implementation

5.1 Strategy

Our strategy to complete the project is based on our project's objectives. Our application is web-based. We used Model View Controller (MVC) architectural pattern. Based on our project objectives, our project was divided into three different dimensions/components. The first one is implementation of assessment or quiz management system. The second one is Interaction Pattern Recording, and the third one is Item Response Theory.(IRT)

In order to ensure that all of our project's objectives are implemented, we need to follow Software Development Life Cycle (SDLC) guidelines. We started by defining requirements. These requirements were further translated into use-case diagrams and descriptions. A specification document was prepared which was then extended to the design document. After that the implementation phase was started. We first develop small chunks of functions, test them on unit level and then integrate them to form modules/functionalities.

Now in order to make sure that all of these above-mentioned phases are completed successfully and on time, we develop a work breakdown structure in which all the tasks were assigned a duration. MS Project was used for creation of Work Breakdown Structure. (WBS)

We faced some trivial and challenging tasks during development and production phase. But, due to continuous support and motivation from our supervisor, we successfully implemented our project in given time frame.

5.2 Tools & Technology Used

The technologies that we used for our project are:

- .NET Core (Backend)
- Angular (Front-end)
- MS SQL (database)

The tools that we used throughout our project are:

- Visual Studio
- Visual Studio Code
- MS SQL Management Studio

- Anaconda Navigator
- Jupyter Notebook
- Postman
- GitHub
- Visual Paradigm
- Google Cloud Platform

5.3 Rubric

Rubric by its definition "a scoring guide used to evaluate the quality of students' constructed responses". If we simply put this, rubric is basically a set of criteria used for evaluating assessments.

We incorporate Rubric in our project with Free Text questions. (Descriptive questions) A free text question attribute with the name of ‘correct answer’ is basically the rubric implementation. The instructor has to paste/write the correct answer for any free text question in that part. The correct answer will be used by the system as criteria for evaluating the free text question.

We used a Natural Language Processing (NLP) technique to evaluate the free text question. The technique is *Soft Cosine Similarity*. This technique is a kind of unsupervised learning because it uses a pre-trained dataset for sematic analysis of two sentences/statements. On computing it returns a similarity value (like 0.58) between the two sentences.

5.4 Item Response Theory (IRT)

Item Response Theory is widely used in educational domain for evaluation and calibration in tests and questionnaires. IRT is used to score subjects based on abilities, attitudes, or any other latent traits.

IRT is incorporated in our system in such a way that the question difficulty level is dependent on it. Looking at the detailed result of a student generated by system, the teacher can judge student performance based on question difficulty levels.

5.5 Interaction Pattern Recording

Interaction Pattern Recording are included in quiz management system in such a way that the system records time duration against each question. This noted duration shows how much time a student spends on an individual question. From these individual durations, teacher can predict that in which questions student faced difficulties and took time in solving it.

5.6 Issues during Implementation & Production Phase

5.6.1 Python Integration With .NET Core

There are different techniques and methods available for python code integration with .NET Core. Some of them are:

- IronPython
- Python integration using processes
- Python.NET
- Python tools for visual studio

In our particular case we are using Soft Cosine Similarity technique which using a pre-trained dataset and genism library. We tried above mentioned methods for python integration with .NET Core, but these techniques were not able to solve the problem. The only solution for this problem was API creation for python code in Django or flask framework.

5.6.2 Python API hosting Issues

We create Application Programming Interface (API) for python code on flask framework. The next task was to host the API onto server. Our hosting service has no support of flask framework, so we had to find other hosting services for our flask API.

The first hosting service was PythonAnywhere. This python hosting service was unable to incorporate the ‘gensim’ library. The second hosting service was Heroku. Heroku was not incorporating NLTK (Natural Language Toolkit) suite of libraries which is basic for NLP.
Finally, we were able to host flask API on Google Cloud Platform.

5.7 Methodologies

The system was developed using iterative development. The focus was to build and test smaller modules and then move onto integration and system implementation. Performance was the key outcome along each stage/phase.

5.8 System Architecture

System architecture used is Model-View-Controller(MVC). MVC supports rapid and parallel development. MVC architecture allow user to have separate layers of concerns like View for UI, Model for data, and Controller for application logic. One major advantage of using this architecture pattern is improved performance, code readability and scalability.

Model:

The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data.

View:

The View component is used for all the UI logic of the application.

Controller:

Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output.

CHAPTER-6

SYSTEM TESTING

6. System Testing

6.1 Test Strategy

Testing is an important phase of the software development life cycle. Purpose of testing is to ensure that the system feature matches the requirements. Testing objectives include bug identification and resolution and identification of missing requirements in contrast to actual requirements.

Test strategy of this system include:

- Test Schedule (WBS for testing)
- Test Development (test case preparation)
- Test Execution
- Defect Management
- Delivery

6.2 Unit Testing

Unit testing is focused on the smallest unit of system. The smallest unit can be a function or any other smaller unit. Unit testing is carried out by programmer to ensure that the smaller module or function works fine and is free from any logical error.

6.3 Component Testing

Component testing is also known as module testing. Component testing is performed after unit testing. In component testing instead of targeting a smaller module, a separate class, object, or module is tested which may consist of many smaller units.

6.4 Integration Testing

Integration testing is performed to ensure that when smaller units are combined together, it works fine. In integration testing individual classes or modules are merged together and tested. The purpose is to check the system program structure.

6.5 System Testing

System testing is performed to ensure that the system works fine as a whole. In system testing we check functionality, security, and portability. Blackbox testing technique is used for system testing.

6.6 Test Cases

6.6.1 Register User Test Cases

Use Case ID	**UC-TE-01**
Test Case ID	TC-01
Scenario ID	1-1
Scenario/Condition	Successfully Register
Action	Clicks on the sign-up link
Input	Username (Ali), Email (Ali@...), Password (Ali123@)
Pre-Condition	None
Expected Output	Registration successful
Post-Condition	Toast message appears (User Registered Successfully)
Pass/Fail Criteria	Account successfully created and user can sign in with these credentials
Remark	Positive test case

Use Case ID	**UC-TE-01**
Test Case ID	TC-01
Scenario ID	1-2
Scenario/Condition	User Already exists, sign up failed
Action	Clicks on the sign-up link
Input	Username (Ali), Email (Ali@...), Password (Ali123@)
Pre-Condition	None
Expected Output	User already exists
Post-Condition	Toast message appears (User already exist)
Pass/Fail Criteria	Does not allow the user to sign up and user can try with different username

Remarks	Negative test case

6.6.2 Login Test Cases

Use Case ID	**UC-TE-02**
Test Case ID	TC-02
Scenario ID	2-1
Scenario/Condition	Successful Log in
Action	Enter user credentials and clicks on log in button
Input	Username (Ali), Password (Ali123@)
Pre-Condition	User must be registered
Expected Output	Log in Successful
Post-Condition	Display the system home screen
Pass/Fail Criteria	User successfully able to log into the system
Remarks	Positive test case

Use Case ID	**UC-TE-02**
Test Case ID	TC-02
Scenario ID	2-2
Scenario/Condition	Log In failed
Action	Enter incorrect username and clicks on log in button
Input	Username (Ali Raza), Password (Ali123@)
Pre-Condition	User must be registered
Expected Output	Log in failed
Post-Condition	Toast message appears (Username or password is incorrect)
Pass/Fail Criteria	Account successfully created and user can sign in with these credentials

Remarks	Negative test case

Use Case ID	**UC-TE-02**
Test Case ID	TC-02
Scenario ID	2-3
Scenario/Condition	Log In failed
Action	Enter incorrect username, password and clicks on log in button
Input	Username (Ali Raza), Password (Ali123456)
Pre-Condition	User must be registered
Expected Output	Log in failed
Post-Condition	Toast message appears (Username or password is incorrect)
Pass/Fail Criteria	Account successfully created and user can sign in with these credentials
Remarks	Negative test case

6.6.3 Add MCQ Test Cases

Use Case ID	**UC-TE-03**
Test Case ID	TC-03
Scenario ID	3-1
Scenario/Condition	Question added successfully
Action	User clicks on question bank and then clicks on add-question button to add question
Input	User adds topic name question statement, choices, Correct choice, points, difficulty level
Pre-Condition	User must be logged in and on add question page
Expected Output	Question added successfully

Post-Condition	Adds the question in data bank and display the add-question page to user to add more questions
Pass/Fail Criteria	Adds question into the data bank
Remarks	Positive test case

Use Case ID	**UC-TE-03**
Test Case ID	TC-03
Scenario ID	3-2
Scenario/Condition	Failed to add the question
Action	User clicks on question bank and then clicks on add-question button to add question
Input	User adds topic name choices, Correct choice, points, difficulty level
Pre-Condition	User must be logged in and on add question page
Expected Output	Question failed to add
Post-Condition	Display failed message to asks user to add all the requirements for the question
Pass/Fail Criteria	Unable to add question in question bank
Remarks	Negative test case

6.6.4 Update MCQ Test Cases

Use Case ID	**UC-TE-04**
Test Case ID	TC-04
Scenario ID	4-1
Scenario/Condition	Question updated successfully
Action	User clicks on question bank and then clicks on edit link on any question
Input	User updates the correct option from a to d

Pre-Condition	User must be logged in and on edit question page
Expected Output	Question updated successfully
Post-Condition	Updates the question in data bank
Pass/Fail Criteria	Updates question in the data bank
Remarks	Positive test case

Use Case ID	**UC-TE-04**
Test Case ID	TC-04
Scenario ID	4-2
Scenario/Condition	Question updated failed
Action	User clicks on question bank and then clicks on edit link on any question
Input	User updates the correct option and leaves it empty
Pre-Condition	User must be logged in and on edit question page
Expected Output	Question failed to update
Post-Condition	Updates the question in data bank
Pass/Fail Criteria	Unable to Update question in the data bank
Remarks	Negative test case

6.6.5 Delete MCQ Test Case

Use Case ID	**UC-TE-05**
Test Case ID	TC-05
Scenario ID	5-1
Scenario/Condition	Question Deleted successfully
Action	User clicks on question bank and then clicks on remove link on any question
Input	User clicks on remove link to delete any question

Pre-Condition	User must be logged in and on question bank page
Expected Output	Question deleted successfully
Post-Condition	Question deleted
Pass/Fail Criteria	Deletes the question from data bank
Remarks	Positive test case

6.6.6 Add True False Question Test Cases

Use Case ID	**UC-TE-06**
Test Case ID	TC-06
Scenario ID	6-1
Scenario/Condition	Question added successfully
Action	User clicks on question bank and then clicks on add-question button to add question
Input	User adds topic name question statement, select true or false, points, difficulty level
Pre-Condition	User must be logged in and on add question page
Expected Output	Question added successfully
Post-Condition	Adds the question in data bank and display the add-question page to user to add more questions
Pass/Fail Criteria	Adds question into the data bank
Remarks	Positive test case

Use Case ID	**UC-TE-06**
Test Case ID	TC-06
Scenario ID	6-2
Scenario/Condition	Failed to add the question
Action	User clicks on question bank and then clicks on add-question button to add question

Input	User adds topic name choices, Correct choice, points, difficulty level
Pre-Condition	User must be logged in and on add question page
Expected Output	Question failed to add
Post-Condition	Display failed message to asks user to add all the requirements for the question
Pass/Fail Criteria	Unable to add question in question bank
Remarks	Negative test case

6.6.7 Update True False Question Test Cases

Use Case ID	**UC-TE-07**
Test Case ID	TC-07
Scenario ID	7-1
Scenario/Condition	Question updated successfully
Action	User clicks on question bank and then clicks on edit link on any question
Input	User updates the correct option from true to false
Pre-Condition	User must be logged in and on edit question page
Expected Output	Question updated successfully
Post-Condition	Updates the question in data bank
Pass/Fail Criteria	Updates question in the data bank
Remarks	Positive test case

Use Case ID	**UC-TE-07**
Test Case ID	TC-07
Scenario ID	7-2
Scenario/Condition	Question updated failed

Action	User clicks on question bank and then clicks on edit link on any question
Input	User updates the correct option and leaves it empty
Pre-Condition	User must be logged in and on edit question page
Expected Output	Question failed to update
Post-Condition	Updates the question in data bank
Pass/Fail Criteria	Unable to Update question in the data bank
Remarks	Negative test case

6.6.8 Delete True False Question Test Case

Use Case ID	**UC-TE-08**
Test Case ID	TC-08
Scenario ID	8-1
Scenario/Condition	Question updated failed
Action	User clicks on question bank and then clicks on edit link on any question
Input	User updates the correct option and leaves it empty
Pre-Condition	User must be logged in and on edit question page
Expected Output	Question failed to update
Post-Condition	Updates the question in data bank
Pass/Fail Criteria	Unable to Update question in the data bank
Remarks	Negative test case

6.6.9 Add Matching Question Test Cases

Use Case ID	**UC-TE-09**
Test Case ID	TC-09
Scenario ID	9-1

Scenario/Condition	Question added successfully
Action	User clicks on question bank and then clicks on add-question button to add question
Input	User adds topic name question statement, adds matching columns, points, difficulty level
Pre-Condition	User must be logged in and on add question page
Expected Output	Question added successfully
Post-Condition	Adds the question in data bank and display the add-question page to user to add more questions
Pass/Fail Criteria	Adds question into the data bank
Remarks	Positive test case

Use Case ID	**UC-TE-09**
Test Case ID	TC-09
Scenario ID	9-2
Scenario/Condition	Failed to add the question
Action	User clicks on question bank and then clicks on add-question button to add question
Input	User adds topic name, matching columns, points, difficulty level
Pre-Condition	User must be logged in and on add question page
Expected Output	Question failed to add
Post-Condition	Display failed message to asks user to add all the requirements for the question
Pass/Fail Criteria	Unable to add question in question bank
Remarks	Negative test case

6.6.10 Update Matching Question Test Cases

Use Case ID	**UC-TE-10**
Test Case ID	TC-10
Scenario ID	10-1
Scenario/Condition	Question updated successfully
Action	User clicks on question bank and then clicks on edit link on any question
Input	User updates the correct matching columns
Pre-Condition	User must be logged in and on edit question page
Expected Output	Question updated successfully
Post-Condition	Updates the question in data bank
Pass/Fail Criteria	Updates question in the data bank
Remarks	Positive test case

Use Case ID	**UC-TE-10**
Test Case ID	TC-10
Scenario ID	10-2
Scenario/Condition	Question updated failed
Action	User clicks on question bank and then clicks on edit link on any question
Input	User updates the correct matching columns and leaves it empty
Pre-Condition	User must be logged in and on edit question page
Expected Output	Question failed to update
Post-Condition	Updates the question in data bank
Pass/Fail Criteria	Unable to Update question in the data bank
Remarks	Negative test case

6.6.11 Delete Matching Question Test Case

Use Case ID	**UC-TE-11**
Test Case ID	TC-11
Scenario ID	11-1
Scenario/Condition	Question Deleted successfully
Action	User clicks on question bank and then clicks on remove link on any question
Input	User clicks on remove link to delete any question
Pre-Condition	User must be logged in and on question bank page
Expected Output	Question deleted successfully
Post-Condition	Question deleted
Pass/Fail Criteria	Deletes the question from data bank
Remarks	Positive test case

6.6.12 Add Free Text Question Test Cases

Use Case ID	**UC-TE-12**
Test Case ID	TC-12
Scenario ID	12-1
Scenario/Condition	Question added successfully
Action	User clicks on question bank and then clicks on add-question button to add question
Input	User adds topic name question statement, adds rubric, points, difficulty level
Pre-Condition	User must be logged in and on add question page
Expected Output	Question added successfully
Post-Condition	Adds the question in data bank and display the add-question page to user to add more questions

Pass/Fail Criteria	Adds question into the data bank
Remarks	Positive test case

Use Case ID	**UC-TE-12**
Test Case ID	TC-12
Scenario ID	12-2
Scenario/Condition	Failed to add the question
Action	User clicks on question bank and then clicks on add-question button to add question
Input	User adds topic name, rubric, points, difficulty level
Pre-Condition	User must be logged in and on add question page
Expected Output	Question failed to add
Post-Condition	Display failed message to asks user to add all the requirements for the question
Pass/Fail Criteria	Unable to add question in question bank
Remarks	Negative test case

6.6.13 Update Free Text Question Use Cases

Use Case ID	**UC-TE-13**
Test Case ID	TC-13
Scenario ID	13-1
Scenario/Condition	Question updated successfully
Action	User clicks on question bank and then clicks on edit link on any question
Input	User updates the question statement and saves it
Pre-Condition	User must be logged in and on edit question page
Expected Output	Question updated successfully

Post-Condition	Updates the question in data bank
Pass/Fail Criteria	Updates question in the data bank
Remarks	Positive test case

Use Case ID	**UC-TE-13**
Test Case ID	TC-13
Scenario ID	13-2
Scenario/Condition	Question updated failed
Action	User clicks on question bank and then clicks on edit link on any question
Input	User updates the question statement and leaves it empty
Pre-Condition	User must be logged in and on edit question page
Expected Output	Question failed to update
Post-Condition	Updates the question in data bank
Pass/Fail Criteria	Unable to Update question in the data bank
Remarks	Negative test case

6.6.14 Delete Free Text Question Test Case

Use Case ID	**UC-TE-14**
Test Case ID	TC-14
Scenario ID	14-1
Scenario/Condition	Question Deleted successfully
Action	User clicks on question bank and then clicks on remove link on any question
Input	User clicks on remove link to delete any question
Pre-Condition	User must be logged in and on question bank page
Expected Output	Question deleted successfully

Post-Condition	Question deleted
Pass/Fail Criteria	Deletes the question from data bank
Remarks	Positive test case

6.6.15 Add Quiz Test Cases

Use Case ID	**UC-TE-15**
Test Case ID	TC-15
Scenario ID	15-1
Scenario/Condition	Quiz added Successfully
Action	After logging into the system user clicks on new quiz button
Input	User enters quiz name and description and add different questions
Pre-Condition	User must be logged in into the system
Expected Output	Quiz added successfully
Post-Condition	Display the quiz with different option to preview or generate link for the quiz
Pass/Fail Criteria	Quiz added into the system
Remarks	Positive test case

Use Case ID	**UC-TE-15**
Test Case ID	TC-15
Scenario ID	15-2
Scenario/Condition	No questions added into the quiz
Action	User tries to add a quiz without adding any question into the quiz
Input	User enters quiz name and description
Pre-Condition	User must be logged in into the system

Expected Output	Failed to add the quiz
Post-Condition	Display a message that you must add questions into the quiz to save the quiz
Pass/Fail Criteria	System does not allow the user to add the quiz
Remarks	Negative test case

6.6.16 Update Quiz Test Cases

Use Case ID	**UC-TE-16**
Test Case ID	TC-16
Scenario ID	16-1
Scenario/Condition	Update quiz to add or remove a question from the quiz
Action	On system home screen user clicks on any quiz to edit the quiz
Input	User removes a question that was in the quiz
Pre-Condition	User must be logged in and on edit quiz page
Expected Output	Question updated successfully
Post-Condition	Updates the quiz in the system
Pass/Fail Criteria	Successfully updates the quiz
Remarks	Positive test case
Use Case ID	UC-TE-16
Test Case ID	TC-16
Scenario ID	16-2
Scenario/Condition	Failed to update quiz
Action	User edits the quiz and leaves the quiz name field empty
Input	User updates the quiz name
Pre-Condition	User must be logged in and on edit quiz page

Expected Output	Update failed
Post-Condition	Fails the quiz update until user fills quiz name field
Pass/Fail Criteria	Failed to update quiz
Remarks	Negative test case

6.6.17 Delete Quiz Test Case

Use Case ID	**UC-TE-17**
Test Case ID	TC-17
Scenario ID	17-1
Scenario/Condition	Quiz Deleted successfully
Action	User clicks on any Quiz on the home screen and then clicks on delete quiz button
Input	User clicks on quiz to go to the quiz page and then clicks delete quiz button
Pre-Condition	User must be logged in and has created a quiz
Expected Output	quiz deleted successfully
Post-Condition	quiz deleted
Pass/Fail Criteria	Deletes the quiz from the system
Remarks	Positive test case

6.6.18 Assign Quiz Test Cases

Use Case ID	**UC-TE-18**
Test Case ID	TC-18
Scenario ID	18-1
Scenario/Condition	Quiz assigned successfully
Action	User clicks on generate link and enter the required details and clicks on save button
Input	User enters reference name, duration, and quiz password

Pre-Condition	User must be logged in and has created a quiz
Expected Output	quiz assigned successfully
Post-Condition	quiz assigned
Pass/Fail Criteria	Quiz is assigned to students
Remarks	Positive test case

Use Case ID	**UC-TE-18**
Test Case ID	TC-18
Scenario ID	18-2
Scenario/Condition	Failed to assign quiz
Action	User clicks on generate link and and leave reference name and quiz password fields empty
Input	User enters quiz duration only
Pre-Condition	User must be logged in and has created a quiz
Expected Output	Failed to assign quiz
Post-Condition	quiz not assigned
Pass/Fail Criteria	Quiz is not assigned to students
Remarks	Negative test case

6.6.19 Attempt Quiz Test Cases

Use Case ID	**UC-STD-01**
Test Case ID	STD-01
Scenario ID	1-1
Scenario/Condition	Quiz attempted successfully
Action	User enter quiz credentials and clicks on start button
Input	User enters name, email, enrollment, and quiz password
Pre-Condition	User must be on attempt quiz page

Expected Output	Quiz Attempted Successfully
Post-Condition	Quiz submitted successfully
Pass/Fail Criteria	Student successfully attempted and submitted quiz
Remarks	positive test case

Use Case ID	**UC-STD-01**
Test Case ID	STD-01
Scenario ID	1-2
Scenario/Condition	Failed to attempt quiz
Action	User enter quiz credentials and clicks on start button
Input	User enters name, email, enrollment, and wrong quiz password
Pre-Condition	User must be on attempt quiz page
Expected Output	Quiz password is incorrect
Post-Condition	User is on quiz attempt page
Pass/Fail Criteria	System displays error message for wrong quiz password
Remarks	negative test case

CHAPTER-7

CONCLUSION

7. Conclusion

Assessment Management for E-Learning at Primary Education Level is a web application which allows teacher to assess student on basis of quizzes. This system allow teacher to create and assign quizzes to students. Teacher can create data bank for question. The system allow teacher to add four different types of questions in quizzes. The system allow teacher to make use of Rubric, IRT, and Interaction Pattern to assess and judge student performance. This system can be used as a stand-alone or can be integrated with an existing Learning Management System.

This project report document explains all the phases of software development life cycle. The detail of each phase is discussed chapter wise from requirements to specification document, and then to design, implementation, and testing documents. This document serves as technical document which was used by designers and developers during design and implementation phases.

7.1 Contributions

All the requirements related to technical and non-technical constraints are covered in this project report. This web application tries to solve the problems related to online assessment which is a technological gap between E-learning and assessments.

7.2 Reflections

This assessment management system can be used as stand-alone or can be integrated with an existing LMS. The project was divided into three broad objectives. Each objective has some feature of system attached to it. The general approach we used for project implementation was related to SDLC. We basically developed a Work Breakdown Structure(WBS) for our project. Each phases of the project development were assigned a duration.

Connecting theoretical knowledge to practical implementation is always a bit tedious part. We were actually naïve to the web technologies. We were able to firm our grip on technologies along with problem translation into practical implementation.

We were successful in practical implementation of work. Assessing the project outcome, our development approach was very good. We make sure that the project requirements and objectives are met.

7.3 Future Work

The system has room for improvements in future. The system was developed on MVC architecture pattern for a reason that the system should be easily scalable in future. The system has improvement room in Rubric implementation. Different NLP techniques can be applied in future for sematic analysis of the free text question. The system has also room for adding any other type of question.

REFERENCES

- David, Jean-Pierre, et al. "Design patterns for recording and analysing usage of learning systems: State of art of tracking and analysing usage." (2005).
- Kean, Jacob, and Jamie Reilly. "Item response theory." *Handbook for clinical research: Design, statistics and implementation* (2014): 195-198.
- Drasgow, Fritz, and Charles L. Hulin. "Item response theory." (1990).
- https://ironpython.net
- https://www.pythonanywhere.com
- https://flask.palletsprojects.com/en/2.0.x/