
Quizzes platform (Quillz)

Graduation Project, Part-II (SWE 497)

Supervised by Dr. Mohammad Abdullah Alwadud

Submitted by:

Bader Alrashed 439102055

Ibrahim Alnumair 439101936

Mohammed Alkhathlan 439101748

Yousef Alfaifi 437100574

Date of submission: 22/5/2022

ABSTRACT

This document describes the development process of a web application that helps the user to make a quiz by adding engagement activities, that can help for teaching. All this can be done by allowing the user to create a quiz in game-based form or non-game based and allowing participant to engage with the user's quiz.

Contents

1. Introduction	11
3. Domain Analysis.....	12
4. Project Plan	14
4.1 Part one.....	14
4.2 Part two.....	15
5. Quality Assurance Plan	15
5.1 Verification.....	15
5.2 Validation	15
5.3 Reviews	15
5.4 Training Team Members.....	16
5.5 Configuration Management.....	16
5.6 Comments.....	16
6. Requirements.....	16
6.1 Functional Requirements.....	16
6.2 Non-functional Requirements	17
6.2.1 Usability.....	17
6.2.2 Maintainability	17
6.3 Design Constraints	18
7. Problem Complexity.....	18
8. System use-cases	18
8.1 Use Cases descriptions.....	20
8.1.1 Join quiz.....	20
8.1.2 Create quiz	21
8.1.3 View dashboard	22
8.1.4 Save question to question bank.....	23
8.1.5 Edit quiz.....	24
8.1.6 Create question.....	25
8.1.7 Import Question.....	26
8.1.8 Activate quiz.....	27
9. Analysis class.....	28

10. Interaction diagram	29
10.1 Sequence diagrams	29
10.1.1 Join quiz.....	29
10.1.2 Create quiz	30
10.1.3 View dashboard	30
10.1.4 Import question	31
10.1.5 Edit quiz.....	32
10.1.6 Create question.....	32
10.1.7 Save question to question bank.....	33
10.1.8 Active quiz.....	34
10.2 VOPCS.....	34
10.2.1 Create quiz	34
10.2.2 Edit quiz.....	35
10.2.3 Save question to question bank.....	36
10.2.4 Register	37
10.2.5 Create question.....	38
10.2.6 Import question	39
10.2.7 Login.....	40
10.2.8 Join quiz.....	41
10.2.9 Active quiz.....	42
10.2.10 View dashboard	43
11. Design Class.....	44
12. System Architecture.....	46
13. Prototype Description.....	46
13.1 Implementation Platform	46
13.2 Algorithms.....	47
13.2.1 Calculate score in game-based quiz.....	47
13.2.2 Calculate score in non-game-based quiz	47
13.3 Mapping between Requirements and Implemented Functions.....	48
13.4 Implementation Details	51
13.4.1Create a quiz	51

13.4.2 Add MCQ question	53
13.4.3 Active quiz	55
13.5 Database Schema	57
13.6 User Interface	57
13.6.1 Home page	57
13.6.2 View dashboard	58
13.6.3 Create quiz	60
13.6.4 Create question	63
13.6.5 Edit question	65
13.6.6 Edit quiz	67
13.6.7 Delete question	70
13.6.8 Delete quiz	71
13.6.9 Save question to question bank	71
13.6.10 Import question	74
13.6.11 Active quiz	76
13.6.12 Join quiz	77
14. Testing	80
14.1 Test Scenarios	80
14.2 Unit Test	97
14.2.1 'CreateQuiz' function	98
14.2.2 'addMCQQuestion' function	99
14.2.3 'addTFQuestion' function	100
14.2.4 'createCategory' function	101
14.3 Functional Test	101
14.4 Usability Test	101
15. Deployment of the System	104
16. Limitation of the System	105
17. Conclusion and Future Work	105
18. Conclusion	106
19. References	106
20. Appendix	106

20.1 Appendix A. Risks/ Constraints Management History	106
20.2 Appendix B. Functional Requirement Updates.....	108
20.3 Appendix C. Non-Functional Requirement Updates.....	111
20.4Appendix D. Use Case Updates.....	111
20.5 Appendix E. Software Architecture Updates	112
20.6 Appendix F. Design Updates	113
20.7 Appendix G. Algorithm Updates	119
20.8 Appendix H. Database Schema Updates.....	121

Table of Figures

Figure 1: Project plan	14
Figure 2: Project plan 2	15
Figure 3: System use-cases	19
Figure 4: Analysis class	28
Figure 5: Join quiz Sequence diagrams	29
Figure 6: Create quiz Sequence diagram	30
Figure 7: View dashboard Sequence diagram	31
Figure 8: Import question Sequence diagram	31
Figure 9: Edit quiz Sequence diagram.....	32
Figure 10: Create question Sequence diagram.....	33
Figure 11: Save question to question bank Sequence diagram.....	33
Figure 12:Active quiz Sequence diagram.....	34
Figure 13: Create quiz VOPC	35
Figure 14: Edit quiz VOPC.....	36
Figure 15: Save question to question bank VOPC.....	37
Figure 16: Register VOPC	38
Figure 17: Create question VOPC.....	39
Figure 18: Import question VOPC	40
Figure 19: Login VOPC.....	41
Figure 20: Join quiz VOPC	42
Figure 21: Active quiz VOPC.....	43
Figure 22: View dashboard VOPC	44
Figure 23: Design class	45
Figure 24: Calculate score in game-based quiz algorithm	47
Figure 25: Calculate score in non-game-based quiz algorithm.....	48
Figure 26: Create quiz (A).....	51
Figure 27: Create quiz (B).....	51
Figure 28: Create quiz (C).....	52
Figure 29: Create quiz (D)	53
Figure 30: Add MCQ question (A)	54
Figure 31: Add MCQ question (B)	54
Figure 32: Add MCQ question (C)	54
Figure 33: Add MCQ question (D)	54
Figure 34: Active quiz (A)	55
Figure 35: Active quiz (B)	55
Figure 36: Active quiz (C)	55
Figure 37: Active quiz (D)	56
Figure 38: Database schema	57
Figure 39: Home page User Interface	58
Figure 40: View dashboard User Interface	59

Figure 41: View dashboard User Interface 2	60
Figure 42: Create quiz User Interface	61
Figure 43: Create quiz User Interface 2	62
Figure 44: Create quiz User Interface 3	63
Figure 45: Create question User Interface.....	63
Figure 46: Create question User Interface 2.....	64
Figure 47: Create question User Interface 3.....	65
Figure 48: Edit question User Interface	66
Figure 49: Edit question User Interface 2	66
Figure 50: Edit question User Interface 3	67
Figure 51: Edit quiz User Interface.....	68
Figure 52: Edit quiz User Interface 2.....	69
Figure 53: Edit quiz User Interface 3.....	70
Figure 54: Delete question User Interface.....	70
Figure 55: Delete quiz User Interface	71
Figure 56: Save question to question bank User Interface.....	72
Figure 57: Save question to question bank User Interface 2.....	73
Figure 58: Save question to question bank User Interface 3.....	74
Figure 59: Import question User Interface	74
Figure 60: Import question User Interface 2	75
Figure 61: Import question User Interface 3	76
Figure 62: Active quiz User Interface	76
Figure 63: Active quiz User Interface 2	77
Figure 64: Join quiz User Interface.....	78
Figure 65: Join quiz User Interface 2.....	78
Figure 66: Join quiz User Interface 3.....	79
Figure 67: Join quiz User Interface 4.....	80
Figure 68: 'CreateQuiz' function	98
Figure 69: 'addMCQQuestion' function.....	99
Figure 70: 'addTFQuestion' function	100
Figure 71: 'createCategory' function	101
Figure 72: Usability test Answer No.1.....	102
Figure 73: Usability test Answer No.2.....	102
Figure 74: Usability test Answer No.3.....	103
Figure 75: Usability test Answer No.4.....	103
Figure 76: System deployment	104
Figure 77: Old use case	112
Figure 78: Old Join room sequence diagram	113
Figure 79: Old Create session sequence diagram	114
Figure 80: Old View library sequence diagram	115
Figure 81: Old Save session sequence diagram	115
Figure 82: Old Edit session sequence diagram	116

Figure 83: Old Test session sequence diagram.....	117
Figure 84: Old Share session sequence diagram	117
Figure 85: Old Initiate room sequence diagram	118
Figure 86: Old Design class	119
Figure 87: Old calculation of a score in game-based quiz algorithm.....	120
Figure 88: calculate the percentage of each answer in a poll algorithm.....	120
Figure 89: Old Database schema	121

Table of tables

Table 1: Domain analysis of other competitors	13
Table 2: Join quiz Use-case description	20
Table 3: Create quiz Use-case description	22
Table 4: View dashboard Use-case description	22
Table 5: Save question to question bank Use-case description	24
Table 6: Edit quiz Use-case description	25
Table 7: Create question Use-case description	26
Table 8: Import Question Use-case description.....	27
Table 9: Activate quiz Use-case description	28
Table 10: Mapping between Requirements and Implemented Functions	50
Table 11: Test Scenarios	96
Table 12: Unit test.....	97
Table 13: Risk management history.....	107
Table 14: Constraints management history.....	108

1. Introduction

One of the main challenges for instructors in the field of education is to provide a motivating learning environment for their students to effectively achieve important learning outcomes. Lately, the use of games in education has become increasingly popular because of recent technological advancements.

Game-based learning platform merges the learning process along with entertainment which can help to develop an efficient way of teaching.

Since the COVID-19 invaded the world, social distancing being required, the online learning has become an essential part of the teaching mechanism. Therefore, the education field has shifted to online learning platforms which is perfect time to release online learning product.

This report is concerned with developing a game-based learning platform that helps with creating quizzes that will make learning more creative and interactive.

2. Definition & acronyms

1. Non game-based quiz: is a short exam that have multiple answers to choose from, but the correct answer is only one of the choices, or a true/false question

2. Quistion bank: is a page where the user can save written question, so he doesn't need to write it again when he wants to use the same question

3. Category: is a way to organize the already saved question in the question bank under certain topic (Ex. Math, physics, chemistry)

4. Game-based quiz: is a type of question that you have certain amount of time to answer it and the quicker you answer the more point you get

5. Room: is a quiz created by user after being activated

6. Customization page: is a page where you can add question, answer, delete questions

7. Dashboard page: is a page that contain all the options that user can have which is active quiz, edit quiz, delete quiz

3. Domain Analysis

AhaSlide

One the existing game-based learning platforms is AhaSlides, AhaSlides is a website created in 2019 with the idea to help the user create engaging and creative slides with many options to the user's presentation. One of its drawbacks, the user is restricted to create a game-based quiz. In addition, they don't have a full support of Arabic language.

Kahoot

One of the leading platforms in the field is Kahoot! It was founded in March 2017. It's a learning platform that specialized in creating quizzes and it is one of the old websites in this field. Therefore, it's one of largest companies in user count and it has a capital worth billions of dollars. Despite how big the platform is and the company behind it, there is nothing perfect, one of the drawbacks, as AhaSlides the user cannot create non game-based quiz

ClassMarker

It is an online testing system, that allows you to create your own tests, ClassMarker isn't concerned with flashy colors or popping animation; it knows its purpose is to help teachers test students and analyze their performance

Features	<i>ClassMarker</i>	<i>Kahoot!</i>	<i>AhaSlides</i>	<i>Our System</i>
<i>game-based approach</i>	✗	✓	✓	✓
<i>Non game-based approach</i>	✓	✗	✗	✓
<i>Ability to test the quiz after creation</i>	✓	✓	✗	✓

<i>Full Support of Arabic language</i>	✓	✗	✗	✓
<i>Student don't need accounts</i>	✗	✓	✓	✓

Table 1: Domain analysis of other competitors

4. Project Plan

4.1 Part one

In the *figure 1* we will illustrate the project plan 1 and show what we worked on and when.

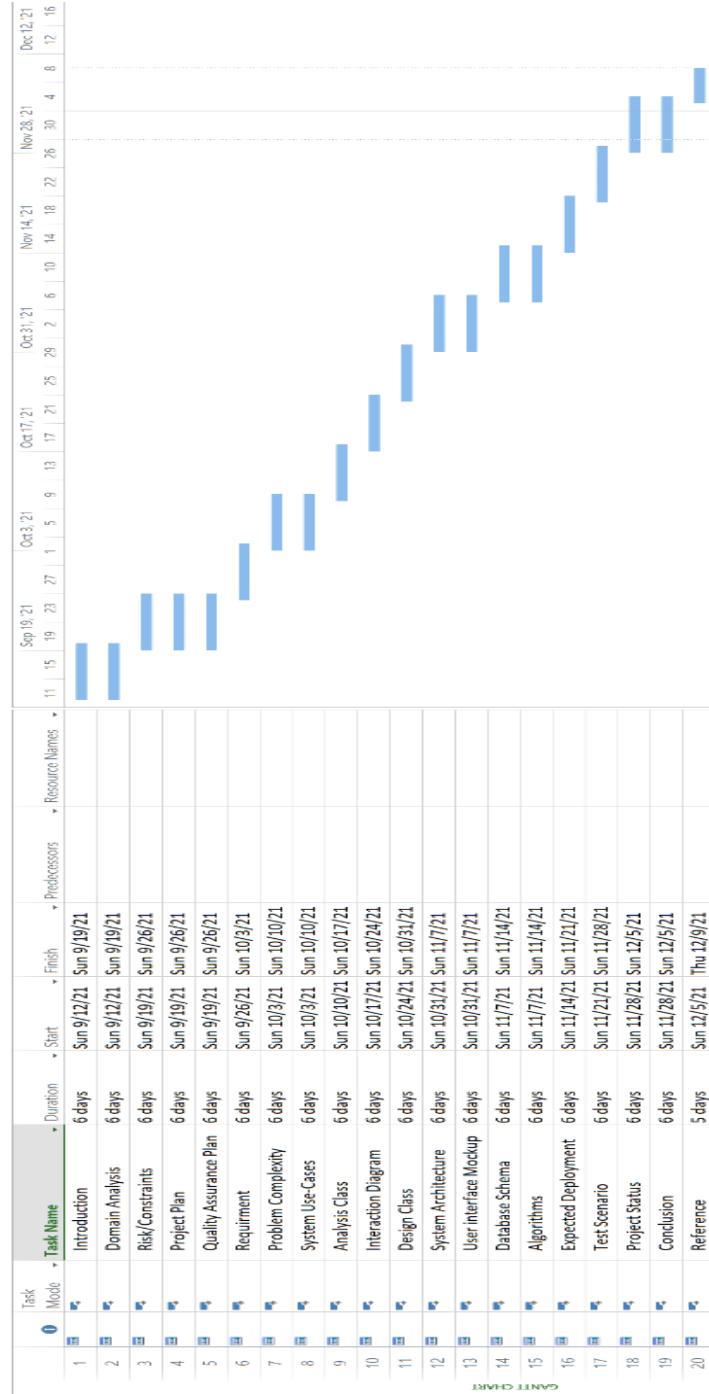


Figure 1: Project plan

4.2 Part two

In the *figure 2* we will illustrate the project plan 2 and show what we worked on and when.

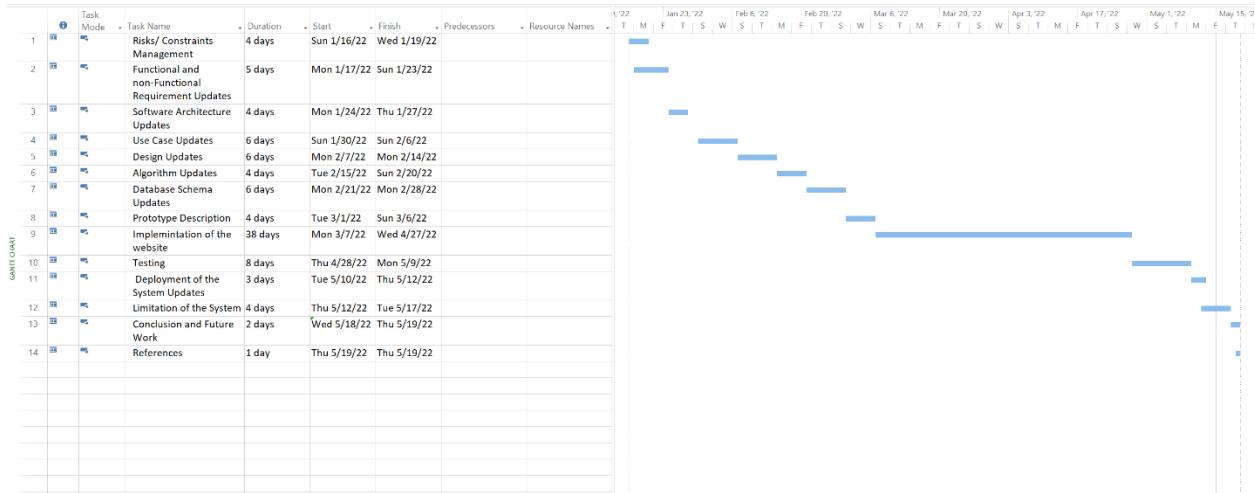


Figure 2: Project plan 2

5. Quality Assurance Plan

5.1 Verification

We will be testing each phase with the previous ones to make sure that our implementation follows our design by using a checklist.

5.2 Validation

We will be looking with the advisor through the system to make sure that we have met all the requirements alongside with security and performance tests.

5.3 Reviews

We will be having weekly reviews after each phase where we will be viewing each other's work to assure that the work meets the expectations and requirements by sharing feedbacks.

5.4 Training Team Members

Every team member will be taking an online course about web development to ensure a complete understanding of the technology before implementing the project.

5.5 Configuration Management

In this project we are going to use GitHub for managing different versions of the system and managing our work as well.

5.6 Comments

Verification: As mentioned, we test every phase with the previous one, also we make sure that the implementation follows the design

Validation: As already mentioned, we tested the software in front of our advisor. Also, in security testing, Laravel blocks SQL injection. It also put a lot of pressure on us because of the situations we faced, such as changing deadlines.

Reviews: Before meeting with an advisor, we review our current state and prepare a list of questions that will be asked. This helps us to prepare for the presentation and also helps us to ask the right questions. We also conducted a verification step to make sure that the implementation is exactly the same.

Training team members: Each team member take some web courses about web development such as Laravel and Bootstrap framework

Configuration management: As mentioned, we used GitHub for controlling different versions in our website

6. Requirements

quillz Here we present functional requirement ,functional requirement and design constraint

We have categorized the functional requirements according to the user roles and when a user have so many functionality, we categorized similar functionality together

6.1 Functional Requirements

1. The system shall allow the user to register.
2. The system shall allow the user to login.
3. The system shall allow the user to reset their password.
4. The system shall allow the user to logout.
5. The system shall allow the user to navigate main pages through a menu.

6. The system shall display sign up and login options.
7. The system shall allow the user to create a quiz with a chosen type.
8. The system shall allow the user to view the quizzes in his dashboard.
9. The system shall allow the user to delete a quiz.
10. The system shall allow the user to edit a quiz.
11. The system shall allow the user to see the result of the quiz.
12. The system shall allow the user to add category from question bank.
13. The system shall allow the user to save question into a specific category from the quiz to question bank.
14. The system shall allow the user to filter category in question bank.
15. The system shall allow the user to import question from question bank to quiz.
16. The system shall allow the user to move question to different category in question bank.
17. The system shall allow the user to create question as multiple-choice question in a quiz.
18. The system shall allow the user to create question as true-false in a quiz.
19. The system shall allow the user to delete a question in a quiz.
20. The system shall allow the user to edit a question in a quiz.
21. The system shall allow the user to add an answer from a question.
22. The system shall allow the user to edit an answer from a question.
23. The system shall allow the user to delete category from question bank.
24. The system shall allow the user to delete question from question bank.
25. The system shall allow the user to determine the time of a quiz.
26. The system shall allow the user to determine the score of a question in a quiz.
27. The system shall allow the user to hide correct answers from the participants in the quiz.
28. The system shall allow the user to activate the quiz by writing the code.
29. The system shall allow the user to deactivate the quiz.
30. The system shall allow the participant to enter username
31. The system shall allow the participant to enter the quiz code.
32. The system shall show the results of a game-based quiz in a leaderboard for all participants based on their scores after they finished it.
33. The system shall display the remaining time of quiz for the participants.

6.2 Non-functional Requirements

6.2.1 Usability

34. Average time to create a quiz shall not exceed 15 minutes if the quiz is consisted of 10 questions and 4 answers for each one and they are ready to be written.
35. Average time for new user to learn how to navigate through all links in the home page shall be no more than 10 minutes.

6.2.2 Maintainability

36. The system shall not be in maintenance more than 12 hours.

6.3 Design Constraints

37. The system shall be developed as web-based service.
38. The front end shall be developed in Bootstrap framework.
39. The back end shall be developed in Laravel framework.
40. The system's database shall be developed in MySQL.

7. Problem Complexity

Many sub-problems: as game-based learning website we have many sub-problems, for instance, our software requires many modules to integrate properly with the system core. In addition, our website uses reusable components commercial off the shelf (COTS) to implement customization module which needs to be integrated properly. Lastly, scores and leaderboard need to be in real-time without any delay is considered as sub-problem.

Diverse groups of stakeholders involved: as website that deals with learning field, many requirements will be changed, and many features will be added. Therefore, this type of system requires a constant communication with diverse stakeholders and endless attempts to grant the satisfaction to the most.

8. System use-cases

In the *Figure 3* we will present the general use case diagram of our system including all the use cases and their relations also we add all the actors.

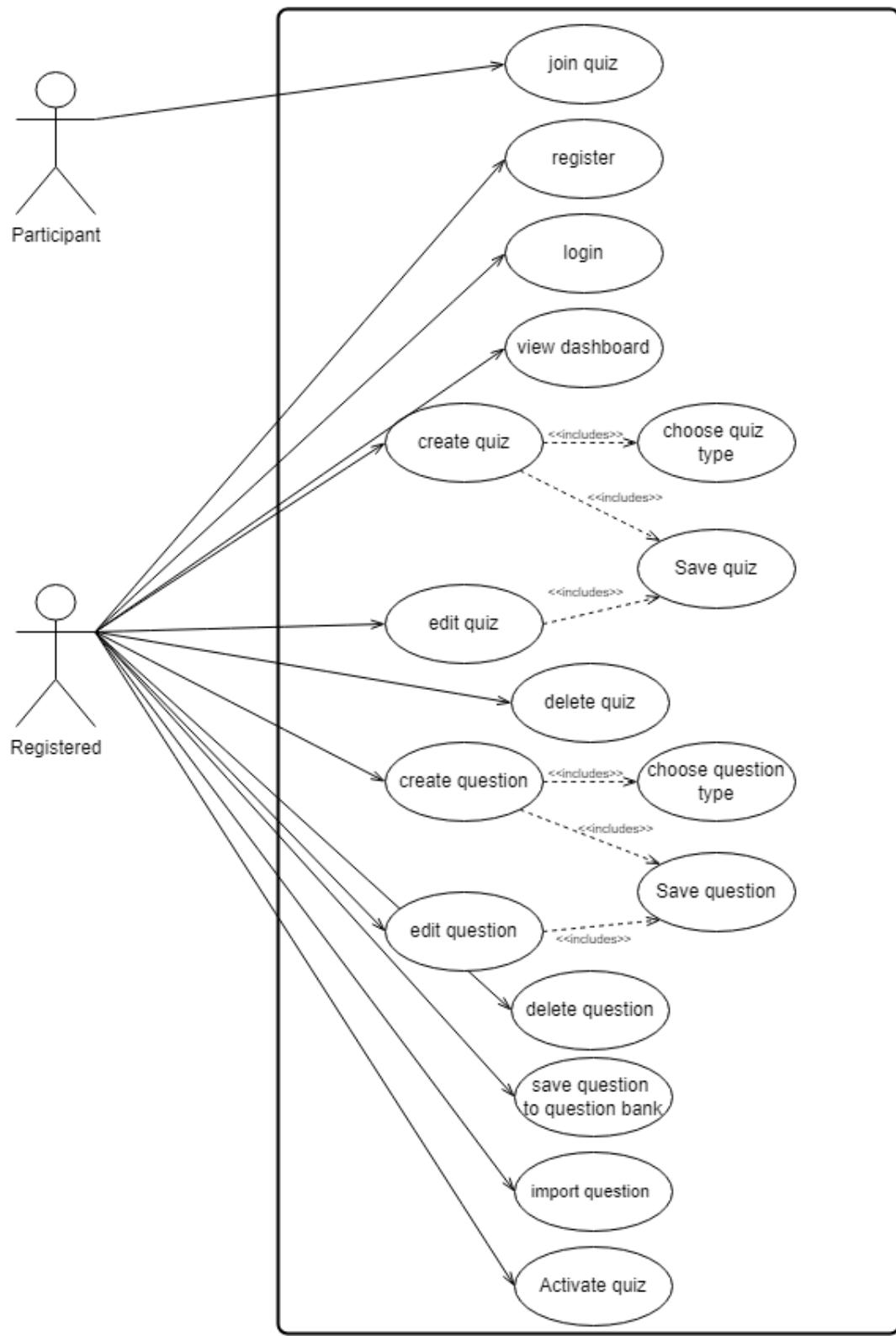


Figure 3: System use-cases

8.1 Use Cases descriptions

8.1.1 Join quiz

The *Table 2* is use case description of when the participant wants to join a quiz

Use Case Description	
Use case ID: 1	
Use Case Name: join quiz	
Primary actor: participant	Other actors: NA
Stakeholders: NA	
Description: this use case describes how the participant joins a quiz.	
Relationships:	
<ul style="list-style-type: none">● Include: NA● Extends: NA	
Input: quiz code	
Pre-conditions: NA	
Steps:	
participant	System
1- The participant enters the quiz code. 2- The participant asks to join a quiz. 4-The participant enters his name.	3-The system asks the participant for his name. 5-The system redirects the participant to the quiz page.
Alternative and exceptional flows: 1a- The participant enters a non-existent room code. 1a1- The system displays an error message.	
Post-conditions: The participant entered the quiz.	

Table 2: Join quiz Use-case description

8.1.2 Create quiz

The *Table 3* is use case description for when the registered user wants to create a quiz.

Use Case Description	
Use case ID: 2	
Use Case Name: Create quiz	
Primary actor: Registered user	Other actors:
Stakeholders: NA	
Description: This use case describes how the registered user creates a new quiz	
Relationships:	
<ul style="list-style-type: none"> • Include: choose quiz type, save quiz • Extends: NA 	
Input: quiz name, quiz category, duration.	
Pre-conditions: registered user must be logged in	
Steps:	
Registered user	System
1- The registered user asks to create a quiz. 3- The registered user chooses the desired quiz type. 5- The registered user enters the quiz information	2- The system asks to choose quiz type. 4- The system asks for quiz information. 6- The system redirects the registered user to the quiz customization page.
Alternative and exceptional flows: NA	

Post-conditions: a new quiz is created.

Table 3: Create quiz Use-case description

8.1.3 View dashboard

The *Table 4* is use case description for when the registered user wants to view his dashboard.

Use Case Description	
Use case ID: 3	
Use Case Name: view dashboard	
Primary actor: registered user	Other actors: NA
Stakeholders: NA	
Description: this use case describes how the user view dashboard.	
Relationships:	
<ul style="list-style-type: none"> ● Include: NA ● Extends: NA 	
Input: NA	
Pre-conditions: logged in user	
Steps:	
Registered user	System
1- The registered user asks to go to the dashboard.	2- The system displays the dashboard page.
Alternative and exceptional flows: NA	
Post-conditions: a dashboard page is displayed.	

Table 4: View dashboard Use-case description

8.1.4 Save question to question bank

The *Table 5* is use case description for when the registered user wants to save a question to the question bank.

Use Case Description	
Use case ID: 4	
Use Case Name: Save question to question bank.	
Primary actor: Registered user	Other actors: NA
Stakeholders: NA	
Description: this use case describes how the registered user can save the question that they are working on directly to the question bank from the customization page.	
Relationships:	
<ul style="list-style-type: none"> ● Include: NA ● Extends: NA 	
Input: selected question, category.	
Pre-conditions: created question.	
Steps:	
Registered user	System
1-The registered user asks to save question to the question bank. 3- The registered user chooses the desired category.	2- The system asks to choose category. 4- The system saves the question to the question bank with chosen category.
Alternative and exceptional flows:	
3a- The registered user didn't choose category. 3a1- The system saves the question to "Without Category" category in question bank.	

Post-conditions: the question is saved to the question bank.

Table 5: Save question to question bank Use-case description

8.1.5 Edit quiz

The *Table 6* is use case description for when the registered user wants to edit a quiz.

Use Case Description	
Use case ID: 5	
Use Case Name: edit quiz	
Primary actor: registered user	Other actors: NA
Stakeholders: NA	
Description: this use case describes how the user edit a quiz.	
Relationships:	
<ul style="list-style-type: none"> ● Include: save quiz ● Extends: NA 	
Input: selected quiz	
Pre-conditions: logged in user and created quiz	
Steps:	
Registered user	System
1- The registered user selects edit quiz. 3- The registered user selects from edit options; the main functions are: (add/delete question, edit answers, save question to question bank, edit quiz name, edit quiz duration, edit quiz category)	2- The system displays the customization page. 4-The system applies the quiz changes.
Alternative and exceptional flows: NA	

Post-conditions: an edited quiz is saved.

Table 6: Edit quiz Use-case description

8.1.6 Create question

The *Table 7* is use case description for when the registered user wants to create a question.

Use Case Description	
Use case ID: 6	
Use Case Name: Create question	
Primary actor: Registered user	Other actors:
Stakeholders: NA	
Description: This use case describes how the registered user creates a new question.	
Relationships:	
<ul style="list-style-type: none"> ● Include: choose question type, save question. ● Extends: NA 	
Input: head, answers, score, and duration.	
Pre-conditions: registered user must be logged in	
Steps:	
Registered user	System
1- The registered user asks to create a question.	2- The system asks to choose question type.
3- The registered user chooses the desired question type.	4- The system asks for question information.
5- The registered user enters the question information	6- The system saves the question.

Alternative and exceptional flows: NA

Post-conditions: a new question is created.

Table 7: Create question Use-case description

8.1.7 Import Question

The *Table 8* is use case description for when the registered user wants to import a question.

Use Case Description	
Use case ID: 7	
Use Case Name: import question	
Primary actor: Registered user	Other actors: NA
Stakeholders: NA	
Description: This use case describes how the registered user import a question	
Relationships: <ul style="list-style-type: none">● Include: NA● Extends: NA	
Input: Selected question from question bank.	
Pre-conditions: logged in registered user, created quiz.	
Steps:	
Registered user	System
1- The registered user asks to import the selected question.	2- The system asks to choose a quiz.
3- The registered user chooses the desired quiz to import the question to.	

	3- The system imports the question to the chosen quiz.
Alternative and exceptional flows:	
2a- The quiz list is empty.	2a1- The system displays an error message.
Post-conditions: The selected question is imported to the chosen quiz.	

Table 8: Import Question Use-case description

8.1.8 Activate quiz

The *Table 9* is use case description for when the registered user wants to activate quiz

Use Case Description	
Use case ID: 8	
Use Case Name: activate quiz.	
Primary actor: Registered user	Other actors: NA
Stakeholders: NA	
Description: This use case describes how the registered user activate quiz.	
Relationships:	
<ul style="list-style-type: none"> ● Include: NA ● Extends: NA 	
Input: Selected quiz, quiz code.	
Pre-conditions: Logged in registered user, created quiz.	
Steps:	
Registered user	System
1- The registered user asks to activate quiz	2- The system validates if the quiz isn't empty. 3- The system asks for a quiz code.
4- The registered user enters the code	5- The system validates the entered code

	6- The system activates the quiz.
Alternative and exceptional flows:	
2a- The selected quiz is empty. 2a1- The system displays error message	
4a- The entered code is already used 4a1- The system displays error message 4a2- The system asks for new quiz code	
Post-conditions: The quiz is activated.	

Table 9: Activate quiz Use-case description

9. Analysis class

The figure 4 illustrate the analysis class where all entities and their relationships are shown

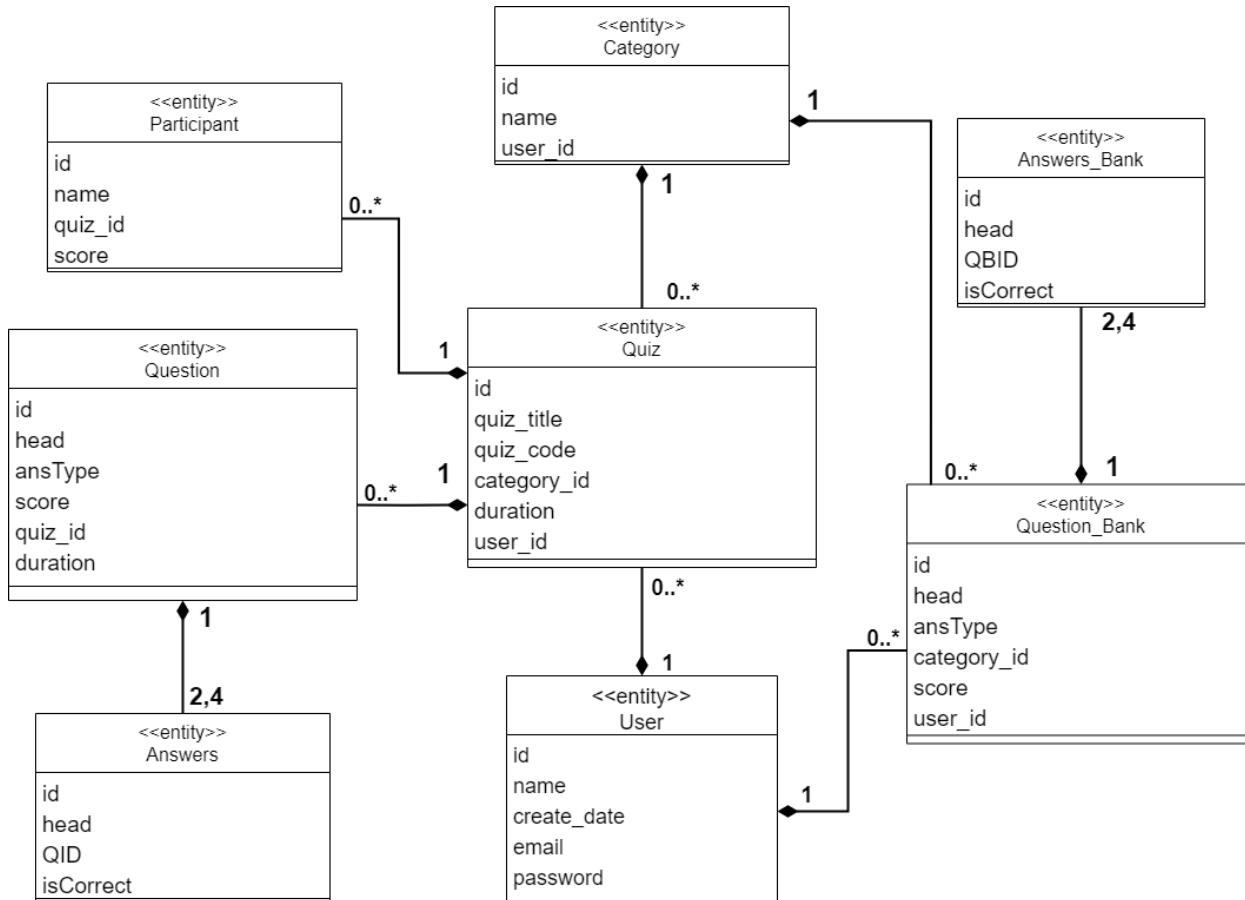


Figure 4: Analysis class

10. Interaction diagram

in the interaction diagram section, we made a sequence diagram and VOPC diagram for all the major functions in the system.

10.1 Sequence diagrams

10.1.1 Join quiz

The *Figure 5* is the sequence diagram of 'Join quiz' use case when the participant wants to join a quiz.

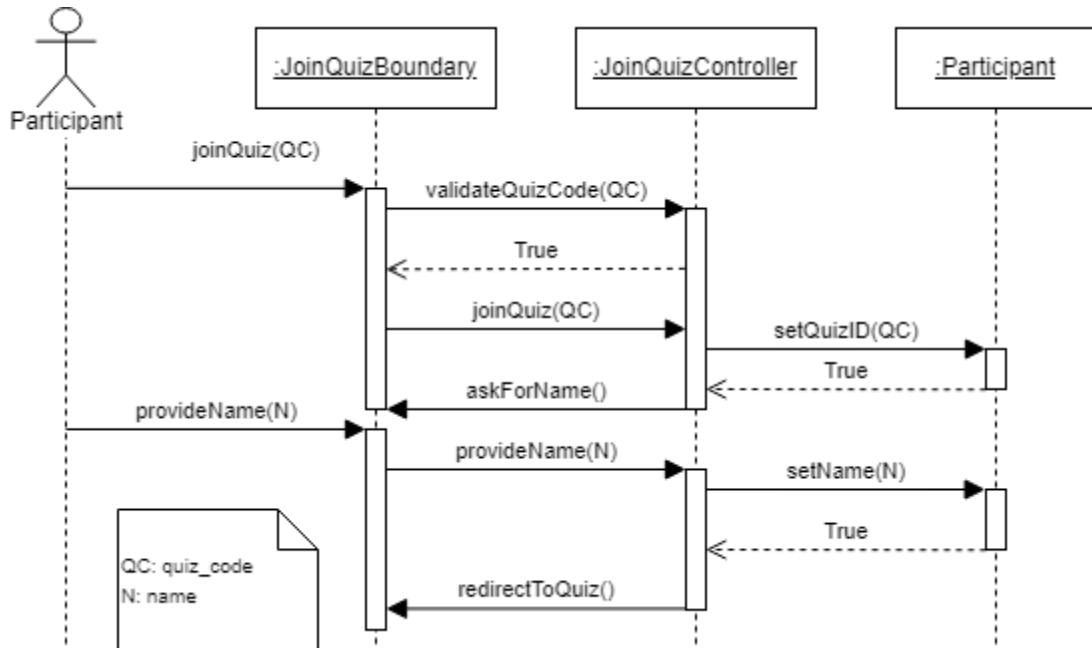


Figure 5: Join quiz Sequence diagrams

10.1.2 Create quiz

The *Figure 6* is the sequence diagram of 'Create quiz' use case when the registered user wants to create a quiz.

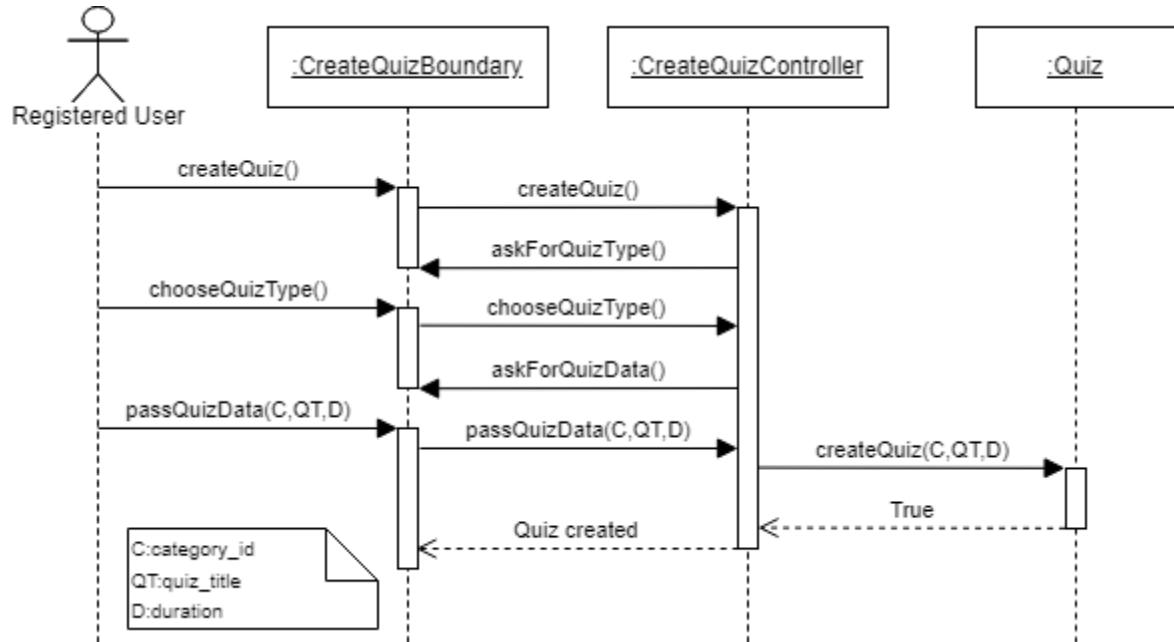


Figure 6: Create quiz Sequence diagram

10.1.3 View dashboard

The *Figure 7* is the sequence diagram of 'View dashboard' use case when the registered user wants to view his Dashboard.

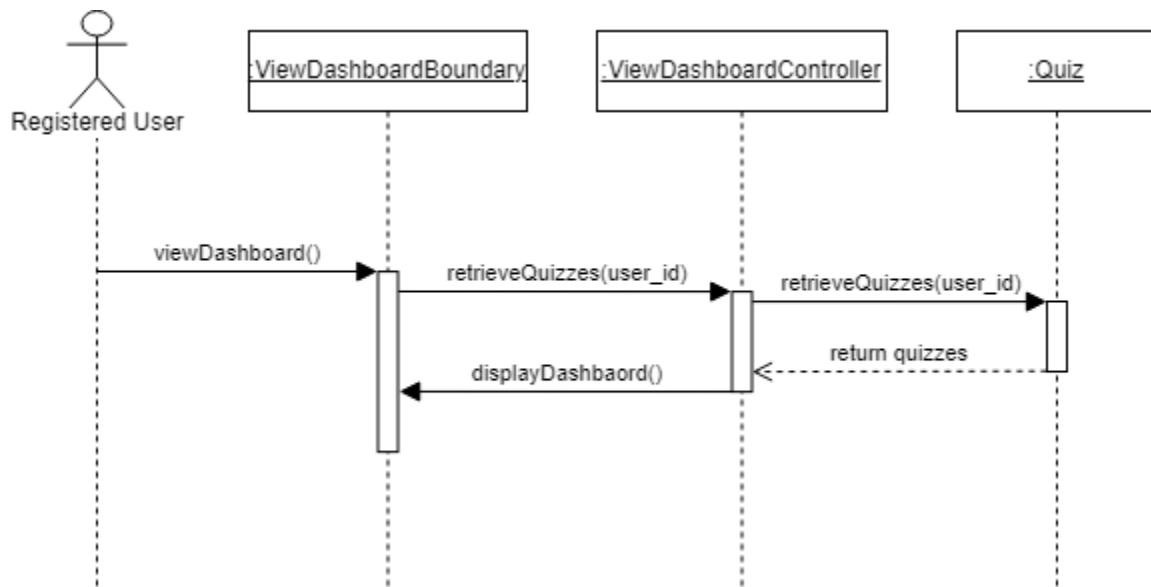


Figure 7: View dashboard Sequence diagram

10.1.4 Import question

The Figure 8 is the sequence diagram of 'Import question' use case when the registered user wants to import question to specific quiz.

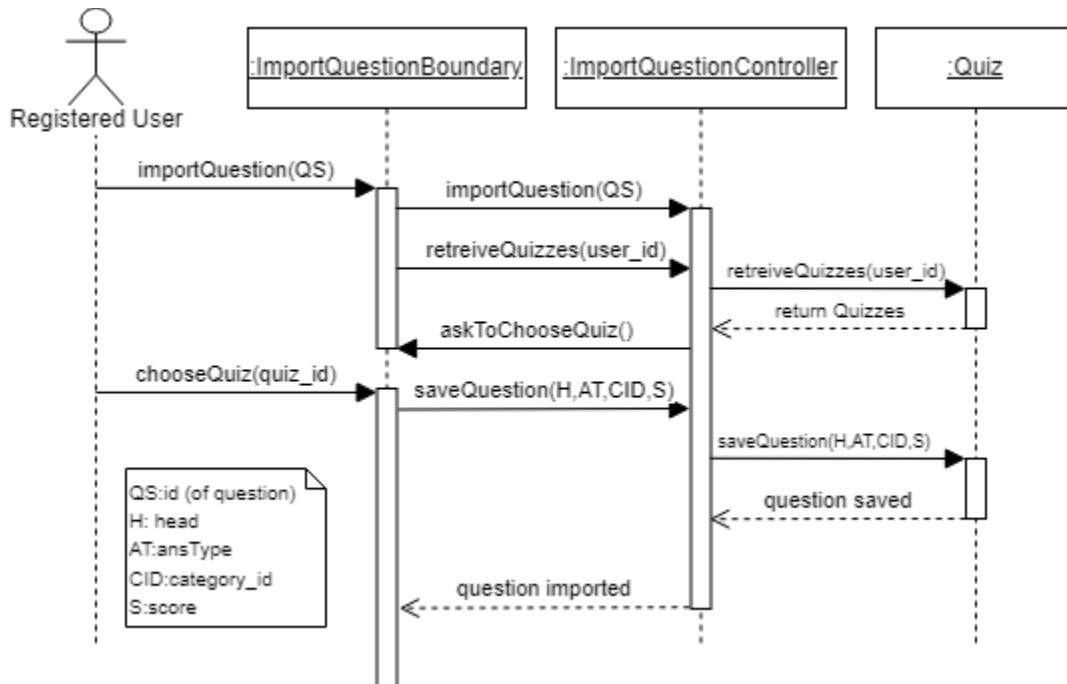


Figure 8: Import question Sequence diagram

10.1.5 Edit quiz

The *Figure 9* is the sequence diagram of 'Edit quiz' use case when the registered user wants to edit a quiz.

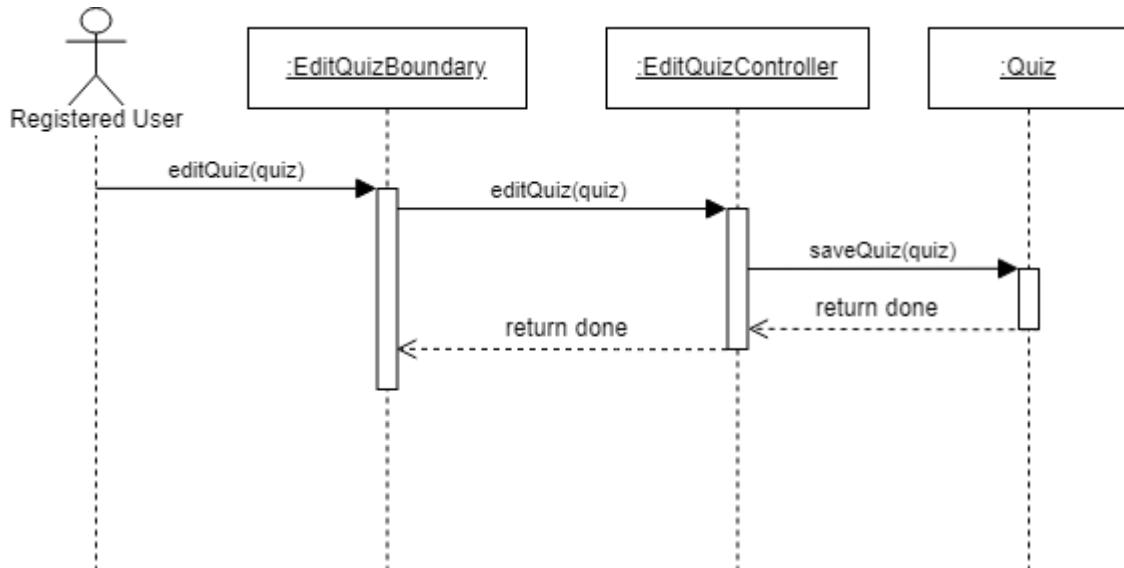


Figure 9: Edit quiz Sequence diagram

10.1.6 Create question

The *Figure 10* is the sequence diagram of 'Create question' use case when the registered user wants to create a question.

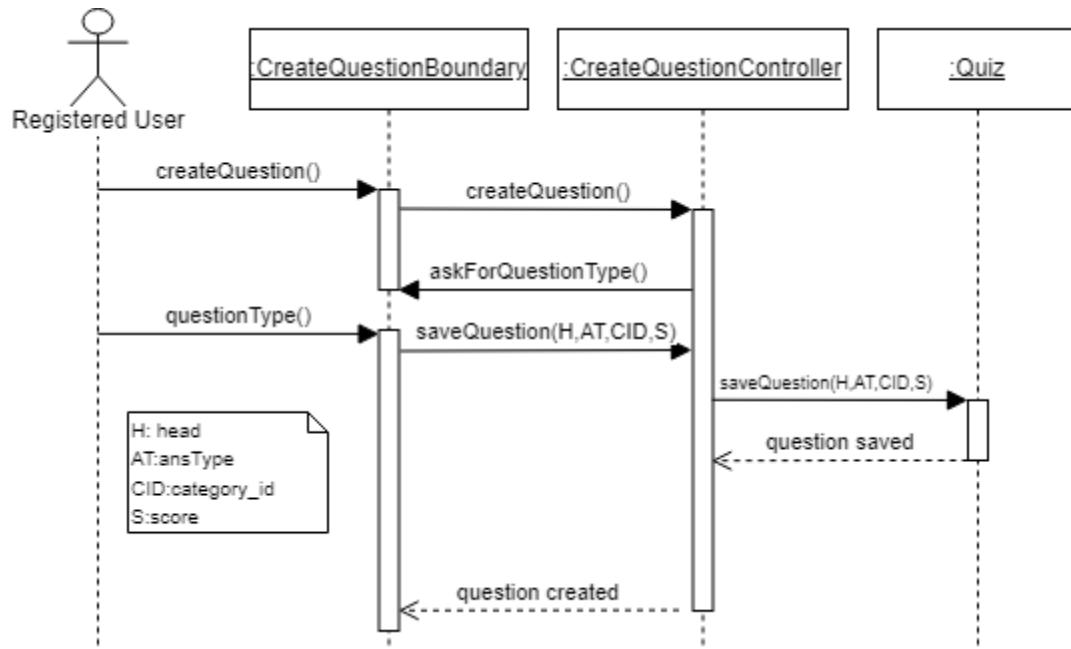


Figure 10: Create question Sequence diagram

10.1.7 Save question to question bank

The *Figure 11* is the sequence diagram of 'Save question to question bank' use case when the registered user wants to save question into question bank.

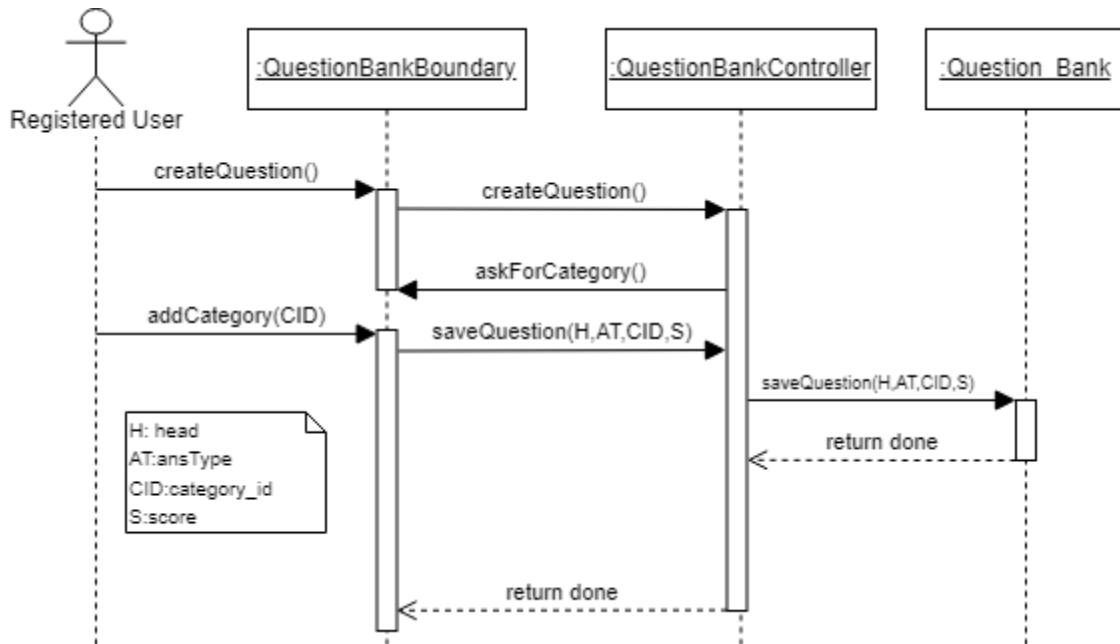


Figure 11: Save question to question bank Sequence diagram

10.1.8 Active quiz

The *Figure 12* is the sequence diagram of 'Active quiz' use case when the registered user wants to Active a quiz.

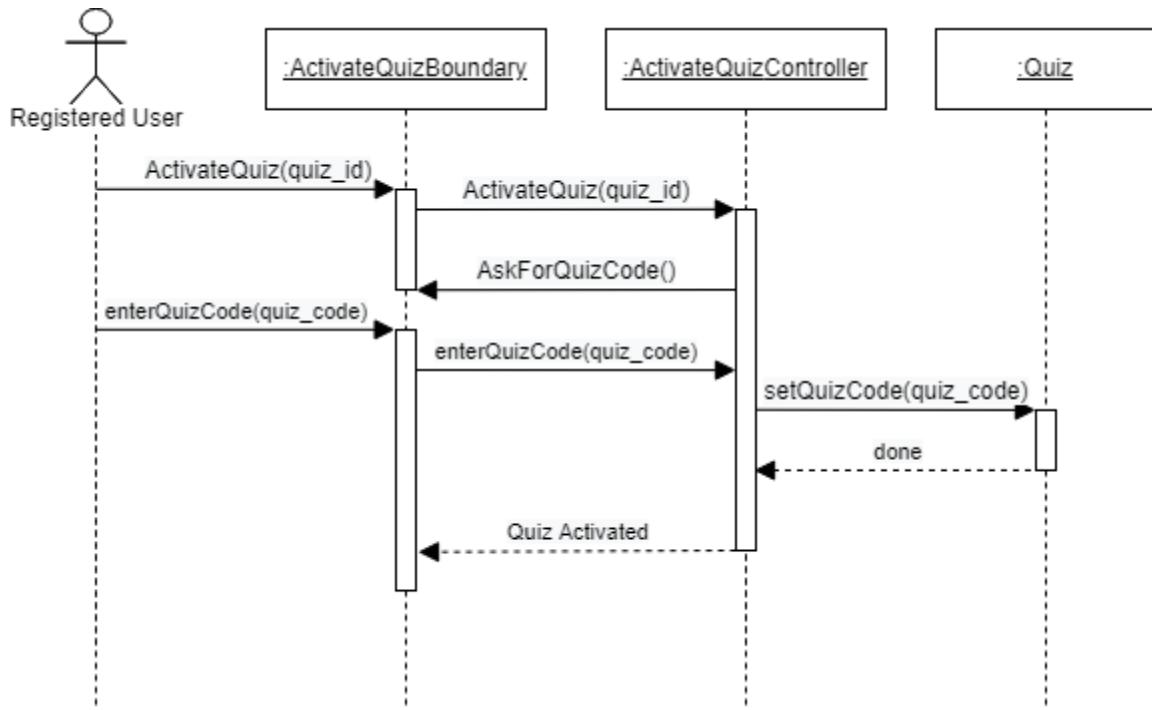


Figure 12:Active quiz Sequence diagram

10.2 VOPCS

10.2.1 Create quiz

The *Figure 13* illustrates the VOPC of the 'Create quiz' use case that enable the user to create quiz.

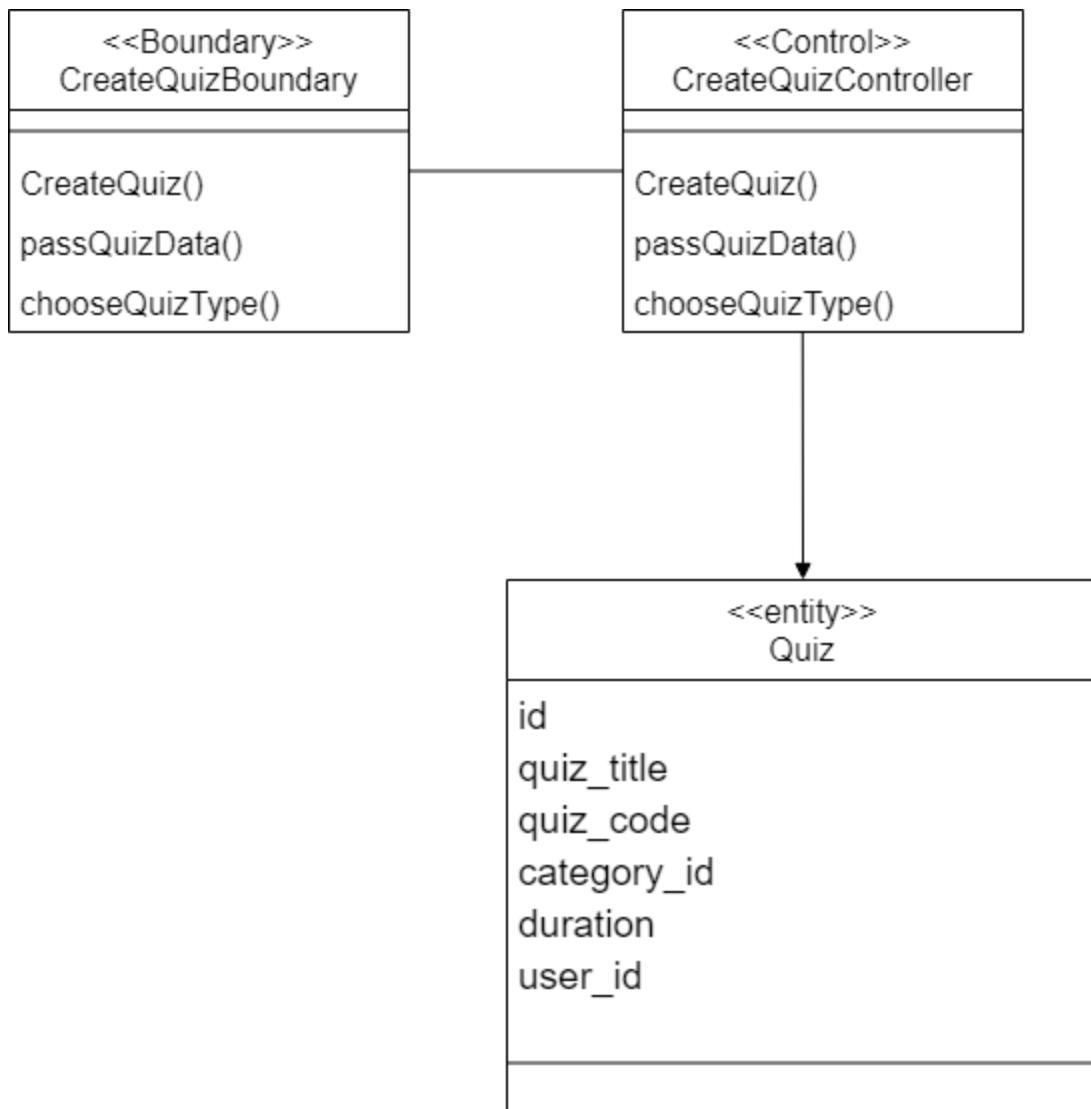


Figure 13: Create quiz VOPC

10.2.2 Edit quiz

The *Figure 14* illustrates VOPC of ‘edit quiz’ use case, the registered user edit the selected question.

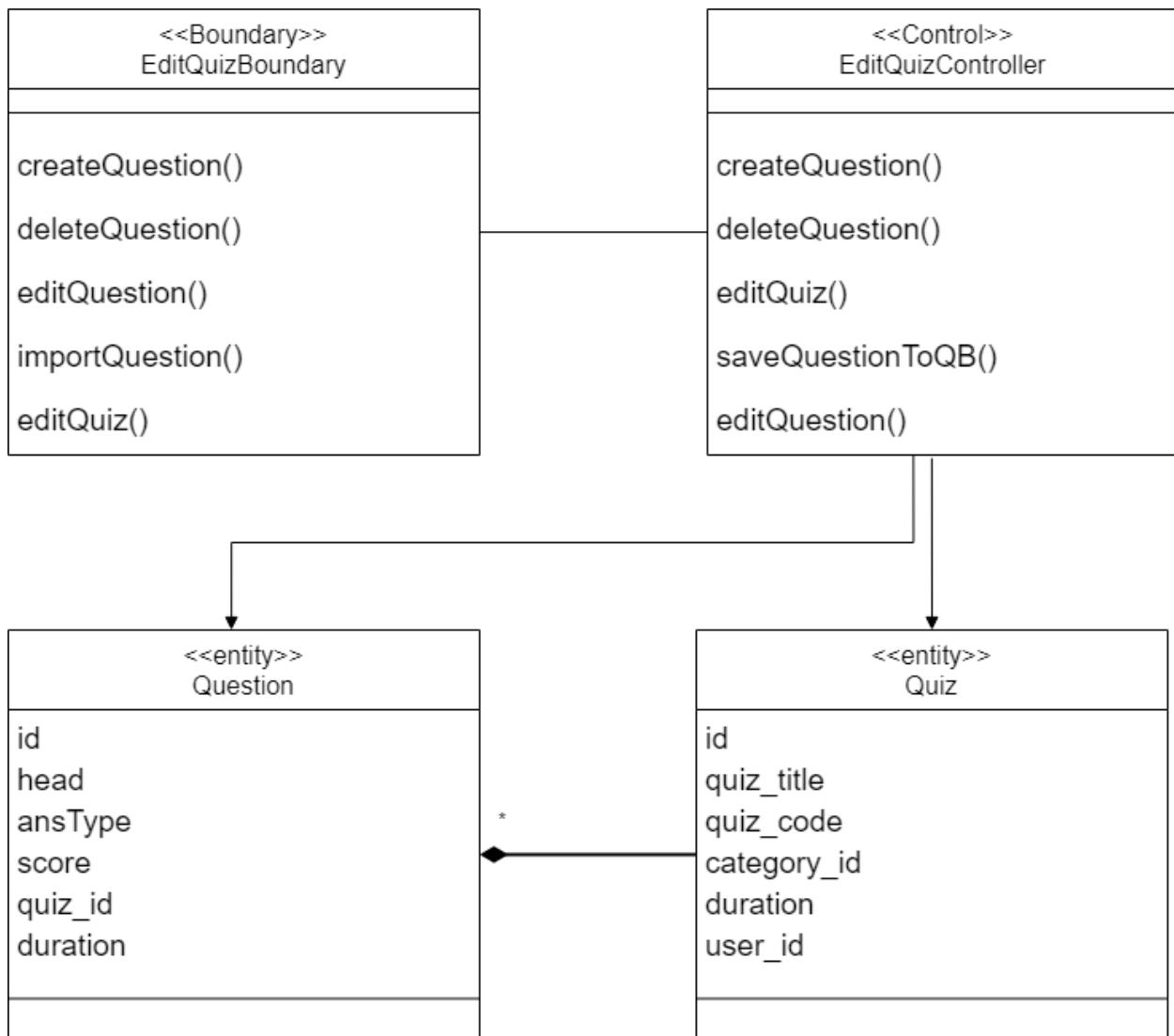


Figure 14: Edit quiz VOPC

10.2.3 Save question to question bank

Figure 15 shows the ‘save question to question bank’ use case that allow the registered user to save question into question bank.

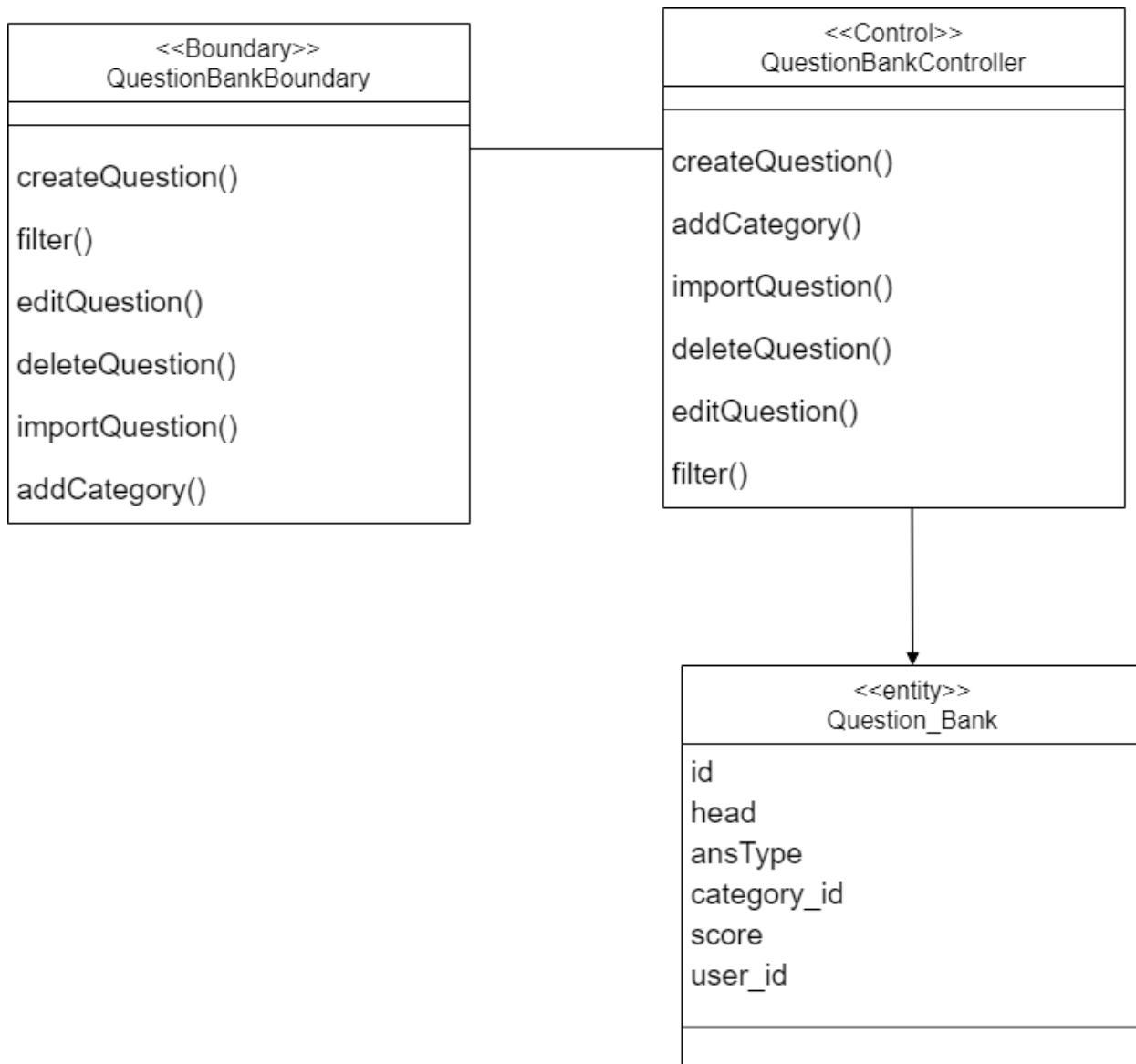


Figure 15: Save question to question bank VOPC

10.2.4 Register

The *Figure 16* illustrates the VOPC for a user to be able to register in the system.

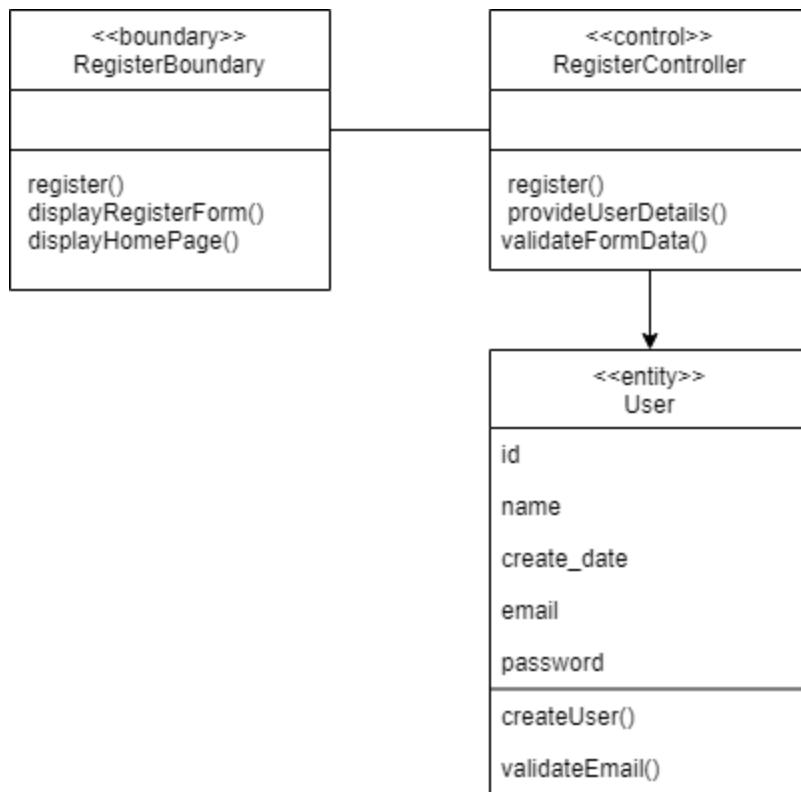


Figure 16: Register VOPC

10.2.5 Create question

The *Figure 17* illustrates VOPC of ‘Create question’ use case that allow the user to create a question.

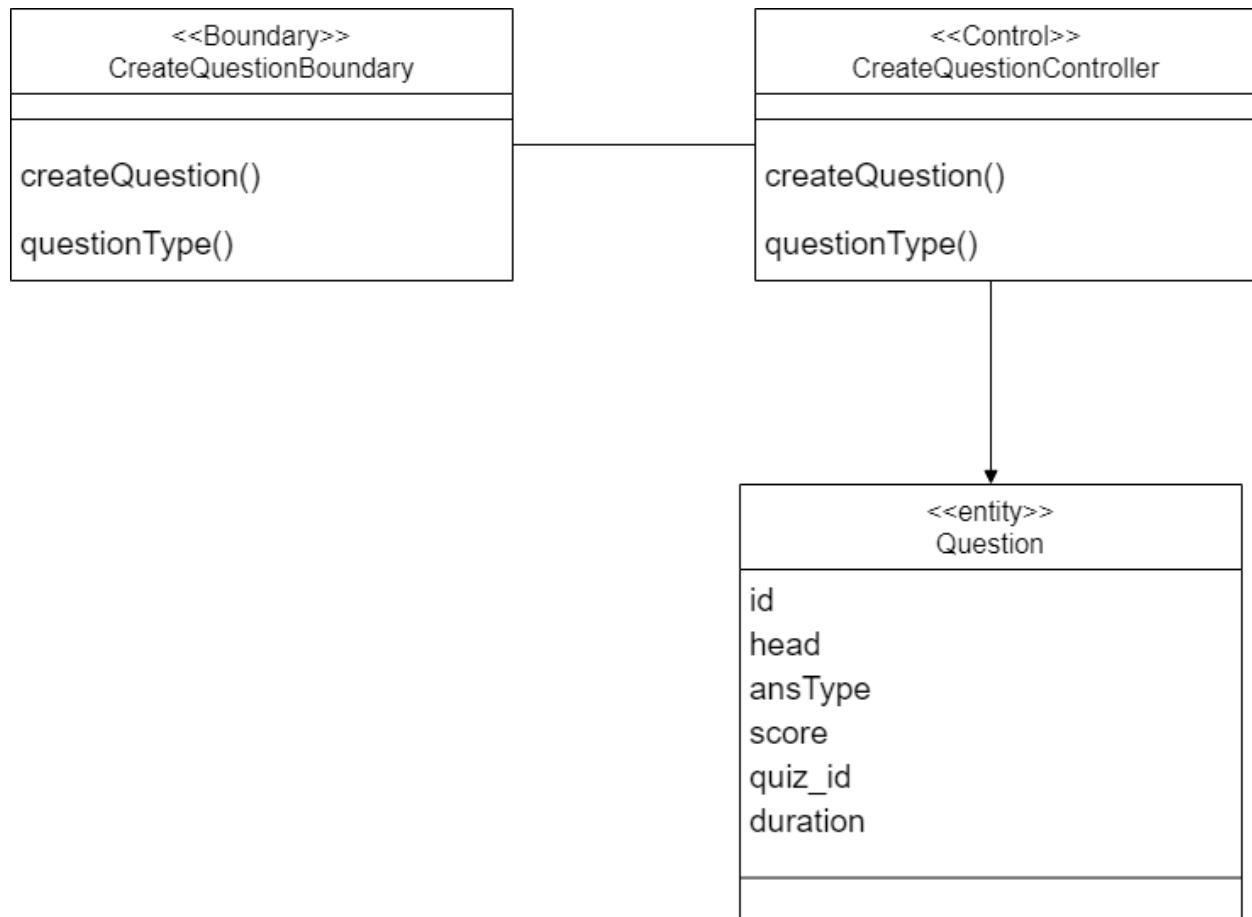


Figure 17: Create question VOPC

10.2.6 Import question

The *Figure 18* illustrates VOPC of “Import question” use case that allow the user to import question to specific quiz.

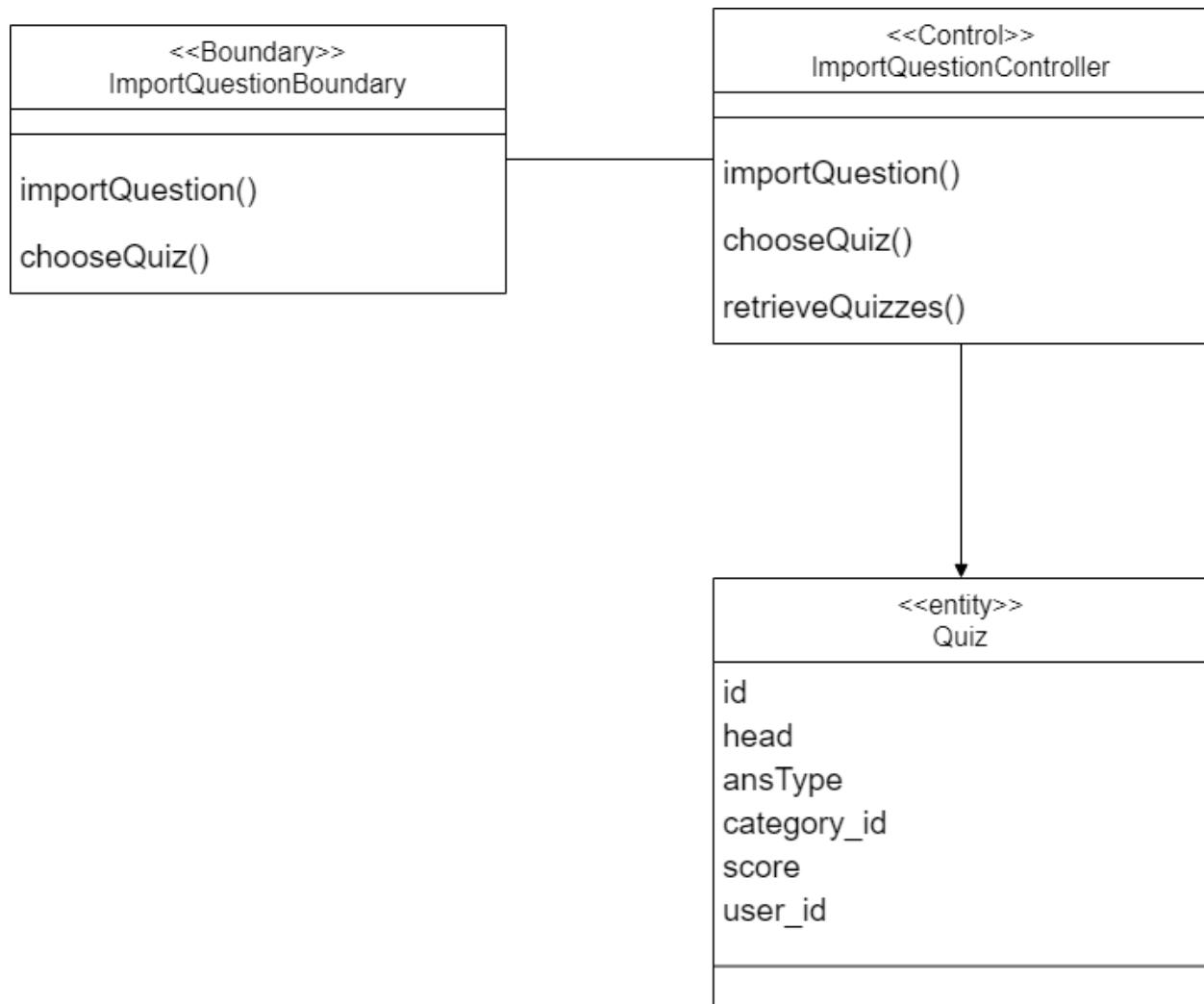


Figure 18: Import question VOPC

10.2.7 Login

The *Figure 19* illustrates the VOPC for the ‘login’ use case, it allows the registered user to login into his account.

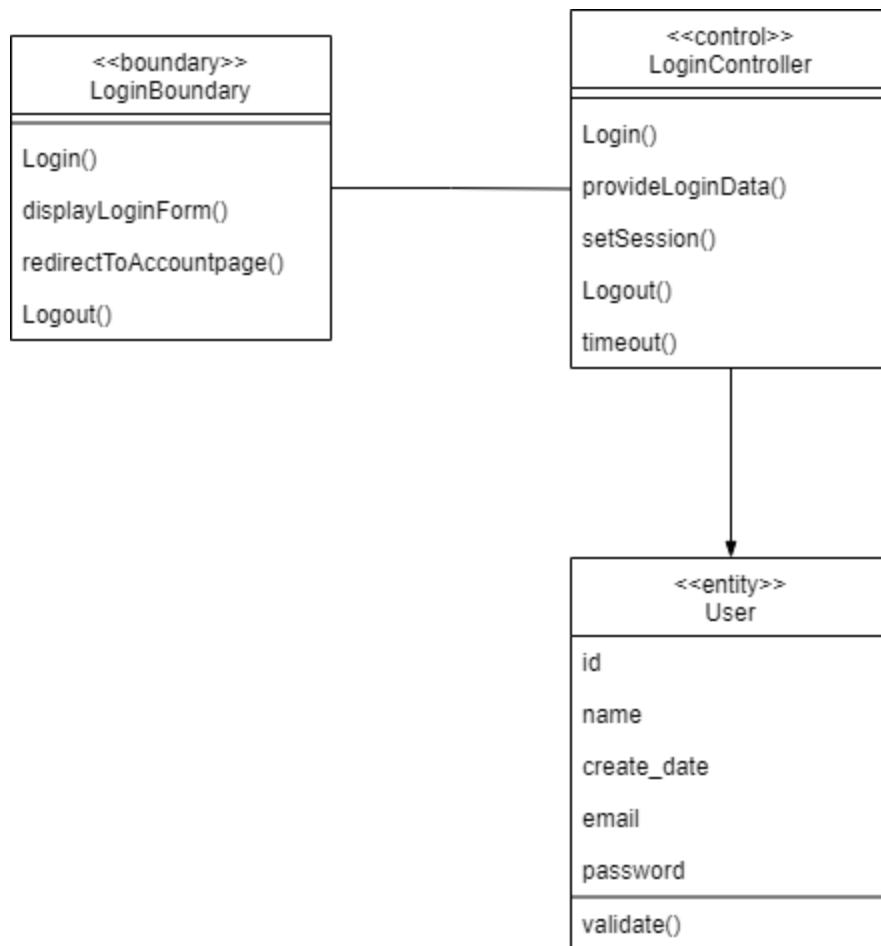


Figure 19: Login VOPC

10.2.8 Join quiz

Figure 20 illustrates the VOPC for the ‘Join quiz’ use case, it allows the participant to join the quiz after its been activated by the user.

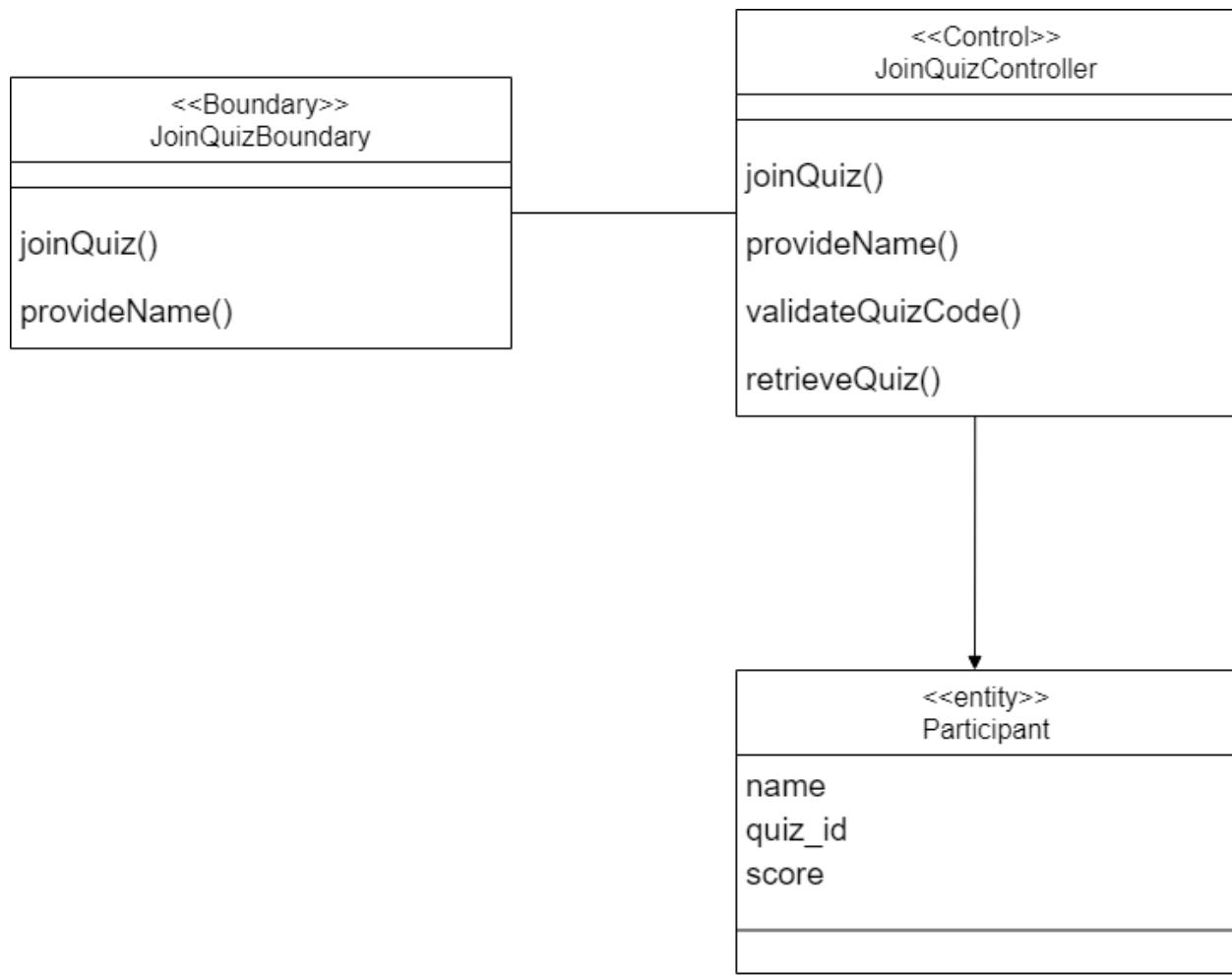


Figure 20: Join quiz VOPC

10.2.9 Active quiz

As the *Figure 21* shown, the user can active quiz to allow the participant to enter the quiz.

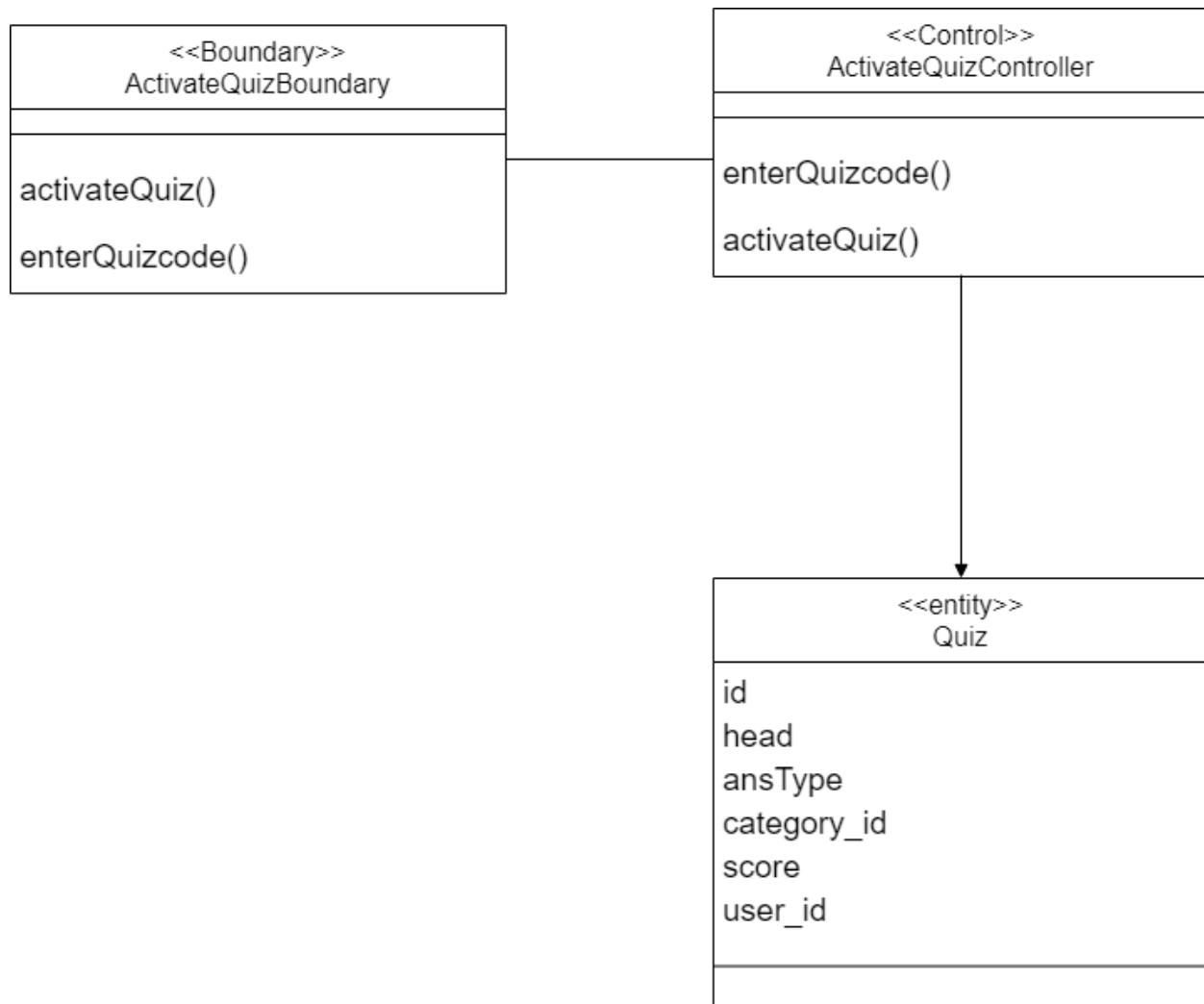


Figure 21: Active quiz VOPC

10.2.10 View dashboard

The *Figure 22* shows the VOPC of the “view dashboard” use case that allows the user to view the dashboard that contains quizzes.

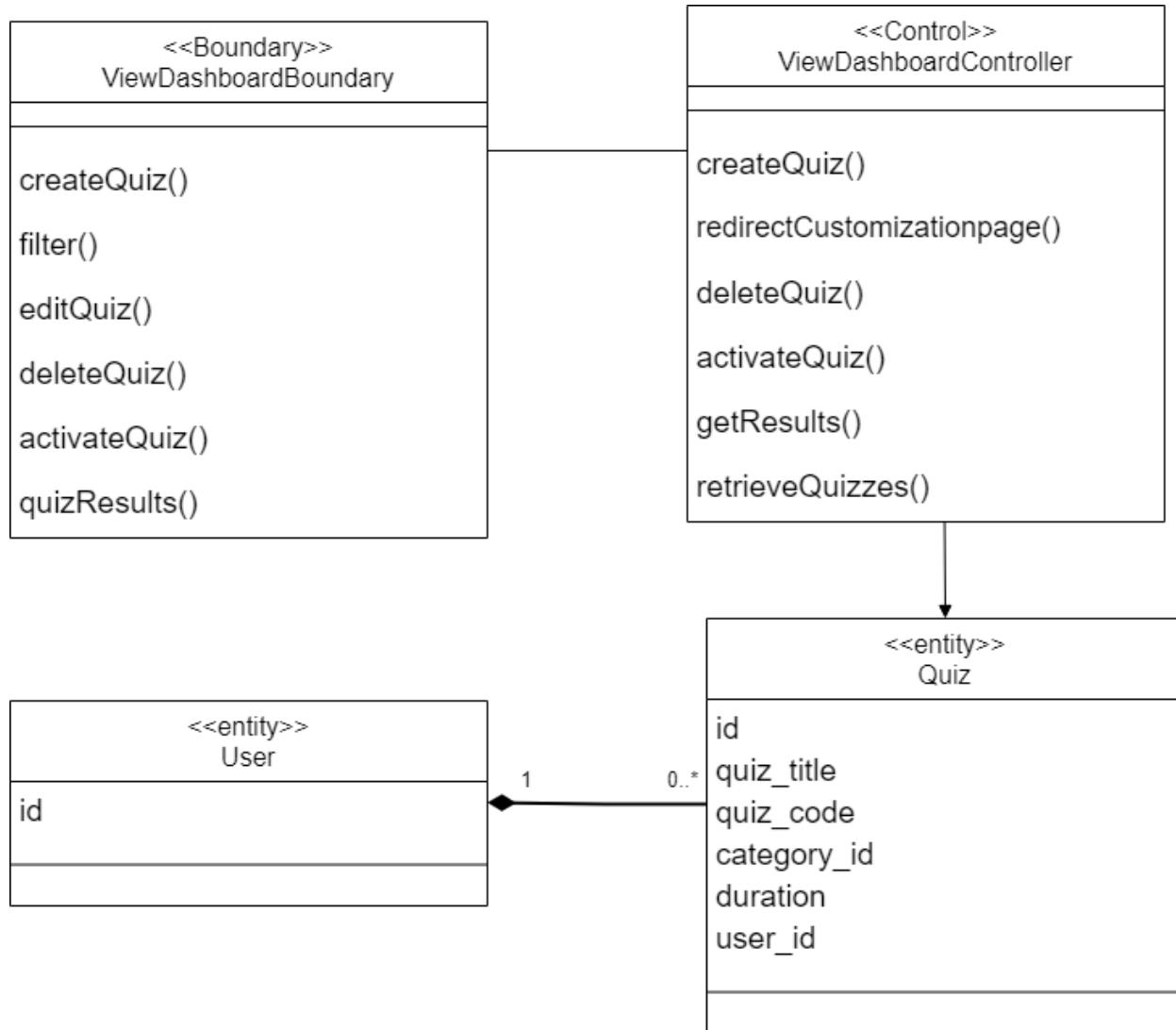


Figure 22: View dashboard VOPC

11. Design Class

The *Figure 23* is composition of all the VOPCs with all attributes, parameters and multiplicities.

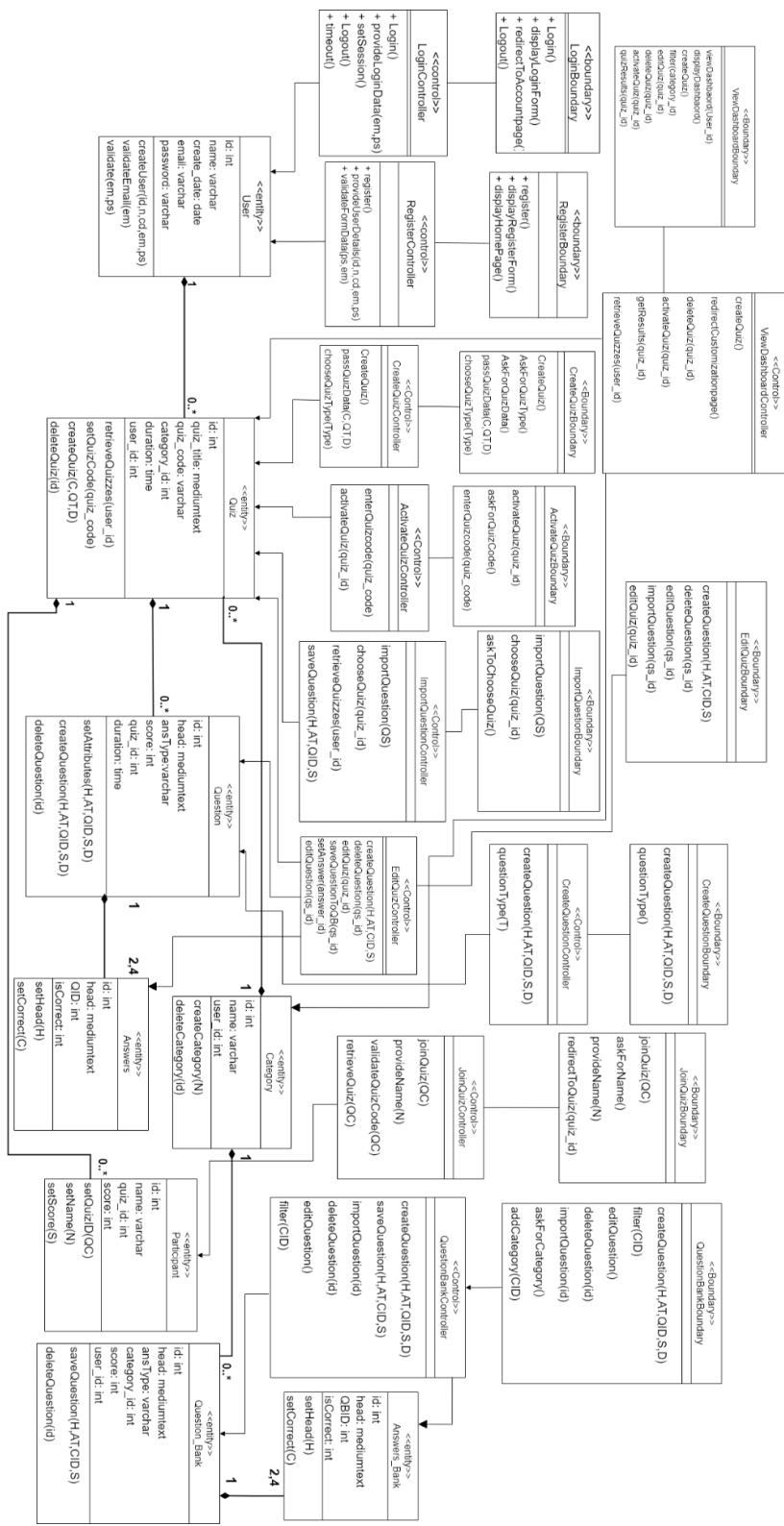


Figure 23: Design class

12. System Architecture

In our system we will use a Client/Server architecture. After researching many architectures, we concluded that Client/Server is the most appropriate and applicable architecture for our system.

Client/server

Client/Server model is the most commonly distributed system architecture, one of the main reasons that we use this model due its responsibility separation such as user interface presentation and business logic processing.

We considered these architectures, but we did not choose:-

- **Presentation-Abstraction-Control (PAC) because:**

1- development complexity.
2- The PAC is suitable for an interactive system where the system can be divided into many cooperating agents in a hierarchical manner, but it doesn't fit with our system that's need multiple views of interfaces, also we want adding or changing interfaces to be easy.

- **multi-tier architecture because:**

1- it's difficult to test the system due the lack of testing tools.
2- Adding a lot of servers can affect the system availability and reliability which is critical in our system.

13. Prototype Description

This section is a description for our software, you can access our software from '<http://quillz.herokuapp.com/en>'.

13.1 Implementation Platform

In developing this website, we've used the following:

In Frontend:

- o **HTML & CSS:** HTML is the most basic building block of the website, and CSS is used to enhance the appearance of the website.

- o **Bootstrap:** A CSS framework that enables developing a responsive and elegant interface.

In **Backend:**

- o **Ajax:** A set of techniques used to create asynchronous websites. With AJAX we were able to process backend requests without refreshing the webpage.
- o **Laravel:** A PHP MVC framework. With this framework we were able to get rid of MySQL queries since this framework uses ‘Eloquent’ to communicate with the database. This framework is also well known for its security and high performance.

13.2 Algorithms

13.2.1 Calculate score in game-based quiz

The *Figure 24* illustrates the algorithm used to calculate the score in a game-based quiz in pseudo code.

```

1 Function GB_TrueAns(answerTime,totalTime)
2   #this function will calculate how much the user will get if the answer is correct
3   #in a game based quiz
4
5   #answerTime is time remaining
6   #when the user answered the question
7
8   #totalTime is the time assigned to the question
9
10  percentage=(answerTime/totalTime) * 50
11  #percentage calculate the percentage between totalTime and answerTime
12
13  score=50 + percentage
14  #score is the variable the will be added to the user final score
15
16  return score
17 end function

```

Figure 24: Calculate score in game-based quiz algorithm

13.2.2 Calculate score in non-game-based quiz

The *Figure 25* illustrates the algorithm used to calculate the score in a non-game-based quiz in pseudo code.

```

1 #this function will calculate the score of non-game based quiz
2 Function nonGB(userAnswer,correctAnswer, score)
3     #userAnswer is user's answer to the question
4     #correctAnswer is question's correct answer
5     #score is a variable that contains the score of an individual question
6
7
8     #if the user answer is correct
9     IF userAnswer == correctAnswer Then
10         SET score = score + 1
11         return score
12
13     #if the user answer is incorrect
14     Else
15         return score
16
17     #the end of if-statement
18     Endif
19
20     #the end of the function
21 End Function

```

Figure 25: Calculate score in non-game-based quiz algorithm

13.3 Mapping between Requirements and Implemented Functions

The table 10 shows a mapping between the projected functionalities of the system and the corresponding implemented modules.

<i>Functional requirement</i>	<i>Functions/Classes that implemented features</i>
SR1	RegisterController.php
SR2	LoginController.php
SR3	ResetPasswordController.php
SR4	LoginController.php
SR5	PagesController.php

SR6	navbar.blade.php
SR7	QuizzesController.create()
	QuizzesController.store()
SR8	QuizzesController.index()
SR9	QuizzesController.destroy(\$id)
SR10	QuizzesController.update(Request \$request, \$id)
SR11	QuizResult.blade.php
SR12	CategoryController.createCategory()
	CategoryController.store(Request \$request)
SR13	QuestionBankController.createMCQ()
	QuestionBankController.storeMCQ(Request \$request)
	QuestionBankController.createTF()
	QuestionBankController.storeTF(Request \$request)
SR14	QuestionBankController.filter(Request \$request)
SR15	QuestionBankController.import(\$id)
	QuestionBankController.importQuestionMCQ(Request \$request)
	QuestionBankController.importQuestionTF(Request \$request)
SR16	QuestionBankController.edit(\$id)
SR17	QuestionController.createMCQ(\$id)
	QuestionController.storeMCQ(Request \$request, \$id)
SR18	QuestionController.createTF(\$id)
	QuestionController.storeTF(Request \$request, \$id)
SR19	QuestionController.destroyMCQ(\$id)
	QuestionController.destroyTF(\$id)
SR20	QuestionController.updateMCQ(Request \$request, \$id)
	QuestionController.updateTF(Request \$request, \$id)
SR21	AnswersController.create(\$id)

	AnswersController.store(Request \$request,\$id)
SR22	QuestionController.updateMCQ(Request \$request, \$id)
	QuestionController.updateTF(Request \$request, \$id)
SR23	CategoryController.destroy(\$id)
SR24	QuestionBankController.destroyMCQ(\$id)
	QuestionBankController.destroyTF(\$id)
SR25	startTimer(time)
SR26	QuizzesController.store()
SR27	takeQuiz.blade.php
SR28	QuizzesController.saveCode(Request \$request, \$id)
SR29	QuizzesController.saveCode(Request \$request, \$id)
	Home.blade.php
SR30	QuizzesController.saveCode(Request \$request, \$id)
	Home.blade.php
SR31	navbar.blade.php
	ActiveQuizController.takeQuiz(\$id)
SR32	PrticipantsController.php
SR33	PrticipantsController.php
	navbar.blade.php

Table 10: Mapping between Requirements and Implemented Functions

13.4 Implementation Details

In this section, we will show the main logic of the core use cases in our system.

Note: There might be slight differences between the figures and the actual code since we are continually enhancing the code.

13.4.1 Create a quiz

In figure 26, the system shows a modal that contains two options. It redirects the user to create quiz page.

```
13      <!-- Button trigger modal -->
14      <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#exampleModal">
15          Create Quiz
16      </button>
17
18      <!-- Modal -->
19      <div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">
20          <div class="modal-dialog modal-dialog-centered">
21              <div class="modal-content">
22                  <div class="modal-header">
23                      <h5 class="modal-title" id="exampleModalLabel">Quiz type</h5>
24                      <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
25                  </div>
26                  <div class="modal-body mx-auto">
27                      <a href="/quizzes/create" class="btn btn-success btn-lg">Game-based quiz</a>
28                      <a href="/quizzes/create" class="btn btn-primary btn-lg">Traditional quiz</a>
29                  </div>
30              </div>
31          </div>
32      </div>
```

Figure 26: Create quiz (A)

In figure 27, the system generates the form for creating a new resource.

```
1  */
2  public function create()
3  {
4      $user_id = auth()->user()->id;
5      $courses=DB::table('courses')
6          ->where('user_id', '=', $user_id)
7          ->pluck('name', 'name');
8      $coursesName = array_merge(['Without course' => 'Without Course'], $courses->all());
9      $quizType = array(['game' => 'Game Based Quiz', 'formal' => 'Formal Quiz']);
10     return view('quizzes.create')->with('coursesName', $coursesName);
11 }
```

Figure 27: Create quiz (B)

In figure 28, the system shows the form which contains the data required to save the quiz in database. When the user clicks ‘Submit’ the system will request to store () function.

```
3  @section('content')
4      <h1>{{__(‘text.create_quiz’)}}</h1>
5      {!! Form::open([‘action’=> ‘App\Http\Controllers\QuizzesController@store’, ‘method’ => ‘POST’]) !!}
6
7      <div class="mb-3">
8          {{Form::label(‘head’, __(‘text.quiz_name’))}}
9          {{Form::text(‘head’, ‘’, [‘class’ => ‘form-control’ , ‘placeholder’ => ‘e.g, Quiz 2’])}}
10     </div>
11
12     <div class="mb-3">
13         {{Form::label(‘course’, __(‘text.quiz_course’))}}
14         {{Form::select(‘course’, $coursesName, “”, [‘class’ => ‘form-control’])}}
15     </div>
16
17     <div class="mb-3">
18         {{Form::label(‘duration’, __(‘text.quiz_duration’))}}
19         {{Form::number(‘duration’, ‘’, [‘class’ => ‘form-control ckeditor’ , ‘min’ => ‘1’])}}
20     </div>
21
22     {{Form::submit(__('button.submit'), [‘class’=>‘btn btn-primary’])}}
23     {!! Form::close() !!}
24
25 @endsection
```

Figure 28: Create quiz (C)

In figure 29, the function shows the form which contains the data required to save the quiz in database.

```

59     public function store(Request $request)
60     {
61         $user_id = auth()->user()->id;
62
63         $this->validate($request, [
64             'head' => 'required',
65             'course' => 'required',
66             'duration' => 'required',
67             // 'quizType'=>'required'
68         ]);
69
70         if($request->input('course') == 'Without course'){
71             $quizId= DB::table('quizzes')
72                 ->insertGetId([
73                     'head' => $request->input('head') ,
74                     'quiz_type'=>$request->select('quiz_type'),
75                     'user_id'=> auth()->user()->id,
76                     'duration' => $request->input('duration')
77                 ]);
78         }
79         else{
80             $quizId= DB::table('quizzes')
81                 ->insertGetId([
82                     'head' => $request->input('head') ,
83                     'quiz_type'=>$request->input('quiz_type'),
84                     'CID' => DB::table('courses')
85                         ->where('name', '=', $request->input('course'))
86                         ->where('user_id', '=', $user_id)
87                         ->value('id') ,
88                     'user_id'=> auth()->user()->id,
89                     'duration' => $request->input('duration')
90                 ]);
91         }
92     }
93

```

Figure 29: Create quiz (D)

13.4.2 Add MCQ question

When clicking ‘Add multiple choice Question’ button it will go to the route ‘/questions/createMCQ as seen in Figure 30

```
<a href="/questions/createMCQ/{$quiz->id}" class="btn btn-primary">{$__('button.addmcq')}</a>
```

Figure 30: Add MCQ question (A)

Then the system will show the page After filling the information and clicking ‘submit’ button the software will go to the route ‘/create_project’as seen in Figure 31.

```
{!! Form::open(['action'=> ['App\Http\Controllers\QuestionController@storeMCQ', $quiz->id], 'method' => 'POST']) !!}
```

Figure 31: Add MCQ question (B)

Then the system will call the function ‘storeMCQ’ in the ‘questionController’ class with the question information from the form as seen in Figure 32

```
Route::post('/questionBank/saveMCQ', 'App\Http\Controllers\QuestionBankController@storeMCQ');
```

Figure 32: Add MCQ question (C)

Figure 33 shows the code for the main logic of the function ‘storeMCQ’.

```
public function storeMCQ(Request $request, $id)
{
    $questionId = DB::table('question')
        ->insertGetId([
            'head' => $request->input('head'),
            'score' => $request->input('score'),
            'quiz_id' => $id,
            'ansType' => "MCQ"
        ]);

    for ($i = 1; $i < 5; $i++) {
        $answer = new answers;
        $answer->answer = $request->input('answer' . (string)$i);
        $answer->QID = $questionId;
        if ($request->input('correct') == $i)
            $answer->isCorrect = 1;
        else
            $answer->isCorrect = 0;

        $answer->save();
    }

    $questions = DB::table('question')
        ->where('quiz_id', '=', $id)->get();
    $quiz = Quiz::find($id);

    return redirect()->action(
        [QuestionController::class, 'index'], ['id' => $id]
    );
}
```

Figure 33: Add MCQ question (D)

13.4.3 Active quiz

When clicking ‘Activate quiz’ button the system will trigger a modal with id ‘disModal’ as shown in Figure 34

```
<button class="btn btn-info" onclick="disModal('{{ $countDis++ }}')>{{__("button.activate")}}</button>
```

Figure 34: Active quiz (A)

After filling the information and clicking on ‘submit’ the system will go to the route ‘/quiz/saveCode’ as shown in Figure 35.

```
{!! Form::open(['action' => ['App\Http\Controllers\QuizzesController@saveCode', $quiz->id], 'method' => 'POST', 'class' => 'modal']) !!}
```

Figure 35: Active quiz (B)

The route will call a function called ‘saveCode’ in the ‘QuizzesController’ class with the information of the task as shown in Figure 36.

```
Route::post('/quiz/saveCode/{id}', 'App\Http\Controllers\QuizzesController@saveCode');
```

Figure 36: Active quiz (C)

Figure 37 shows the code for the main logic of the function ‘saveCode’.

```
public function saveCode(Request $request, $id)
{
    if($Codes = DB::table('quizzes')->where('code', $request->input('code'))->exists())
    {
        $user_id = auth()->user()->id;
        $user = User::find($user_id);
        $courses = courses::orderBy('created_at', 'desc')->paginate(10);
        $coursesName= DB::table('courses')
                    ->where('user_id', '=', $user_id)
                    ->pluck('name', 'name');
        $coursesName= array_merge(['all' => 'All courses'], $coursesName->all());

        return redirect('/home')->with(['quizzes', $user->quizzes]
            ->with('courses', $courses)
            ->with('coursesName', $coursesName)
            ->with('danger', 'Code Taken'));
    }
    $questions = DB::table('question')->where('quiz_id', '=', $id);

    if (!$questions->exists()) {
        $user_id = auth()->user()->id;
        $user = User::find($user_id);
        $courses = courses::orderBy('created_at', 'desc')->paginate(10);
        $coursesName= DB::table('courses')
                    ->where('user_id', '=', $user_id)
                    ->pluck('name', 'name');
        $coursesName= array_merge(['all' => 'All courses'], $coursesName->all());

        return redirect('/home')->with(['quizzes', $user->quizzes]
            ->with('courses', $courses)
            ->with('coursesName', $coursesName)
            ->with('danger', 'Quiz Empty'));
    }

    $quiz = Quiz::find($id);
    $quiz -> code = $request->input('code');
    $quiz->save();

    $user_id = auth()->user()->id;
    $user = User::find($user_id);
    $courses = courses::orderBy('created_at', 'desc')->paginate(10);
    $coursesName= DB::table('courses')
                    ->where('user_id', '=', $user_id)
                    ->pluck('name', 'name');
    $coursesName= array_merge(['all' => 'All courses'], $coursesName->all());

    return view('home')->with(['quizzes', $user->quizzes]
        ->with('courses', $courses)
        ->with('coursesName', $coursesName)
        ->with('success', 'Code saved'));
}
```

Figure 37: Active quiz (D)

13.5 Database Schema

The *Figure 38* illustrates the database schema of the system.

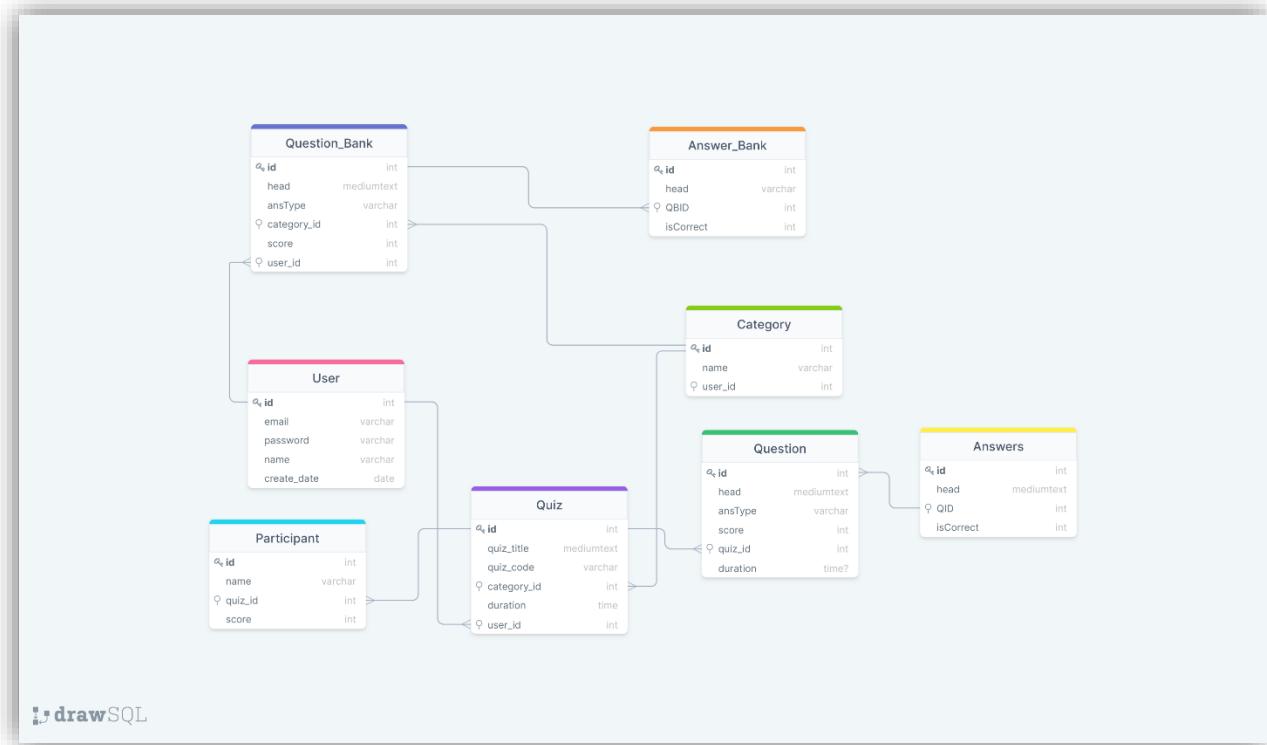


Figure 38: Database schema

13.6 User Interface

13.6.1 Home page

The *Figure 39* illustrates the User interface of the “Home Page” where the user can register or login or enter a room.

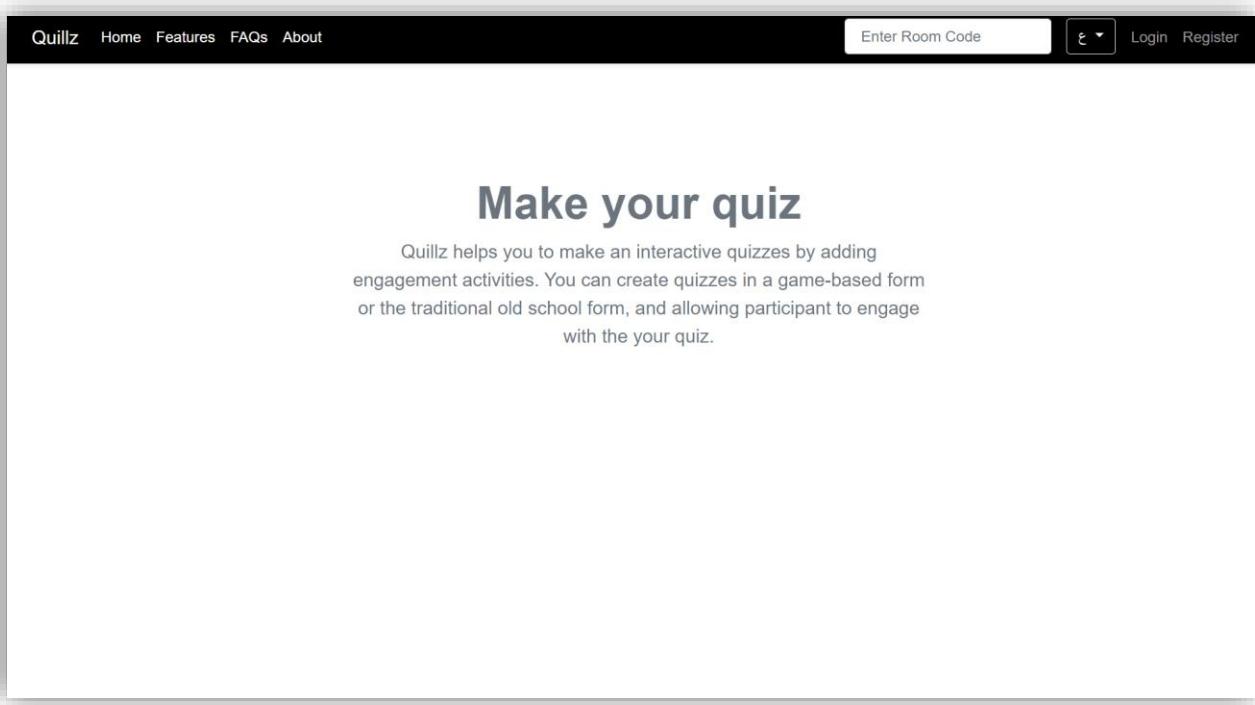


Figure 39: Home page User Interface

13.6.2 View dashboard

The *Figure 40* illustrates the User interface of the “View dashboard” use case that allows the user to access the dashboard.

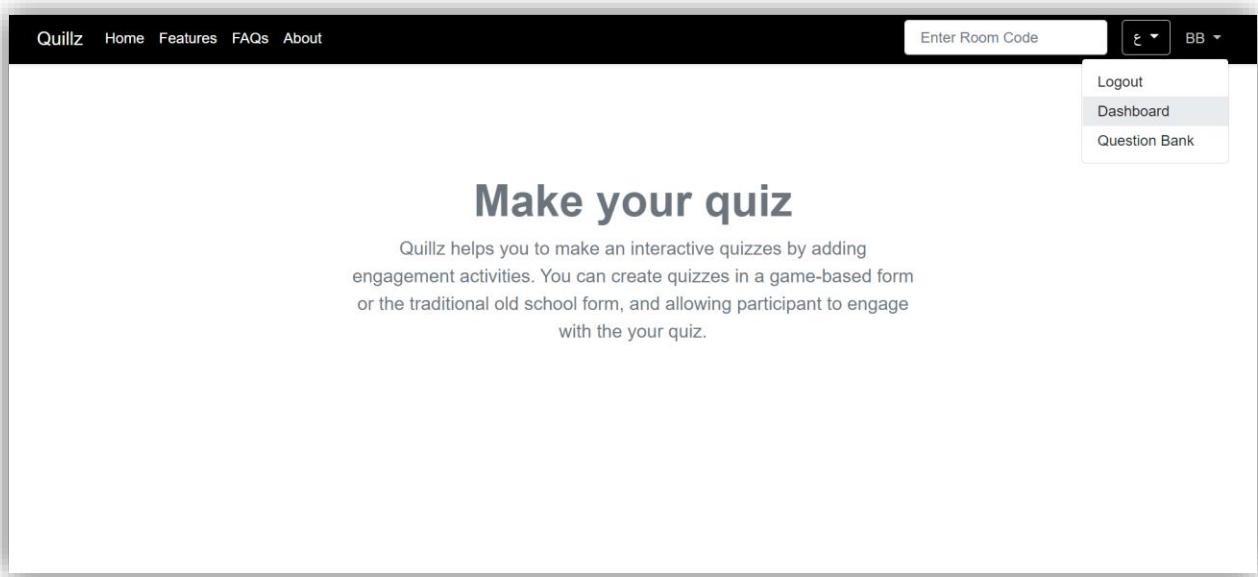


Figure 40: View dashboard User Interface

The *Figure 41* illustrates the User interface of the “View dashboard” use case that allows the user to view the dashboard.

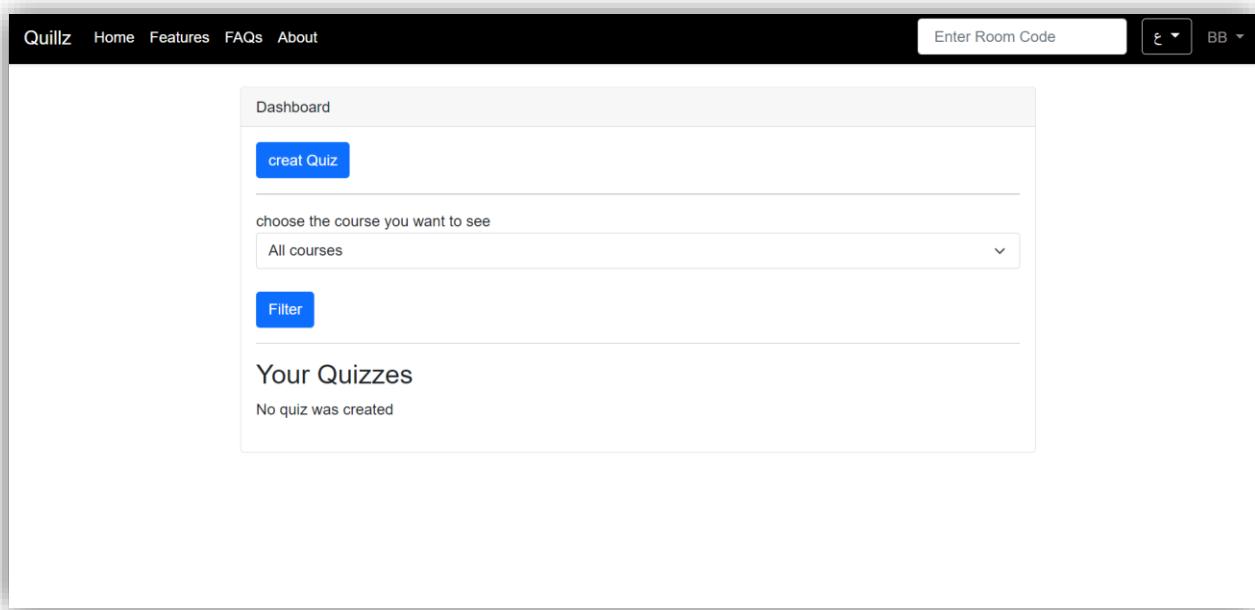


Figure 41: View dashboard User Interface 2

13.6.3 Create quiz

The *Figure 42* illustrates the User interface of the “View dashboard” use case and how to access create quiz.

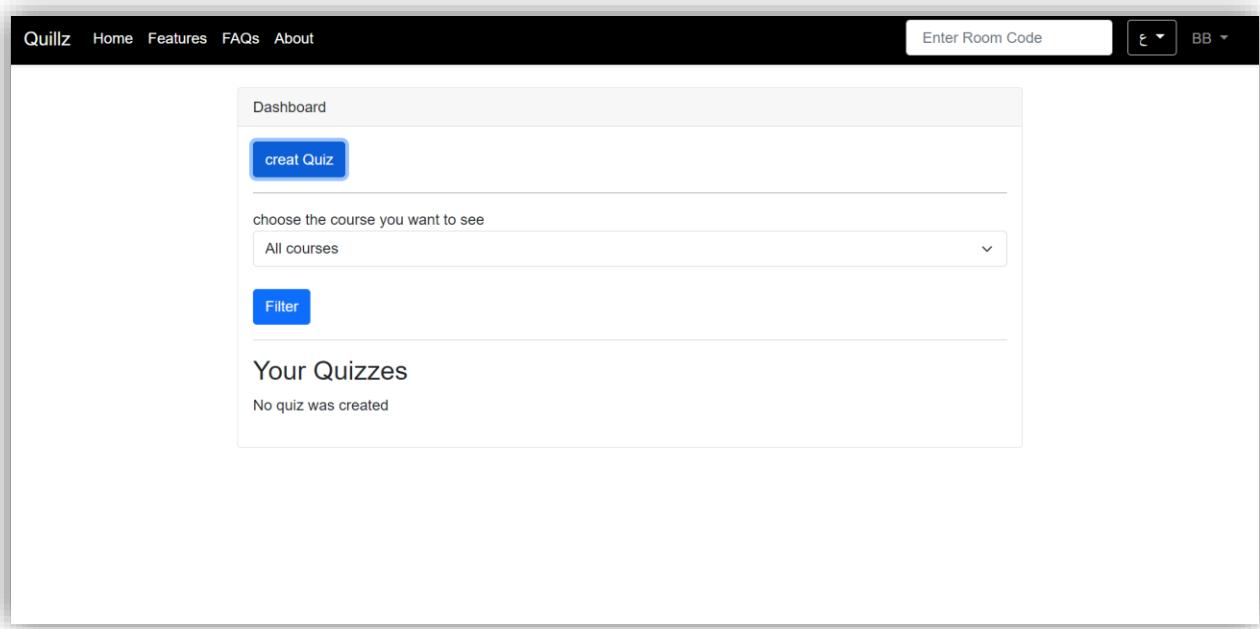


Figure 42: Create quiz User Interface

The *Figure 43* illustrates the User interface of the “Choose quiz type” use case and how to choose between game or non-game based quiz.

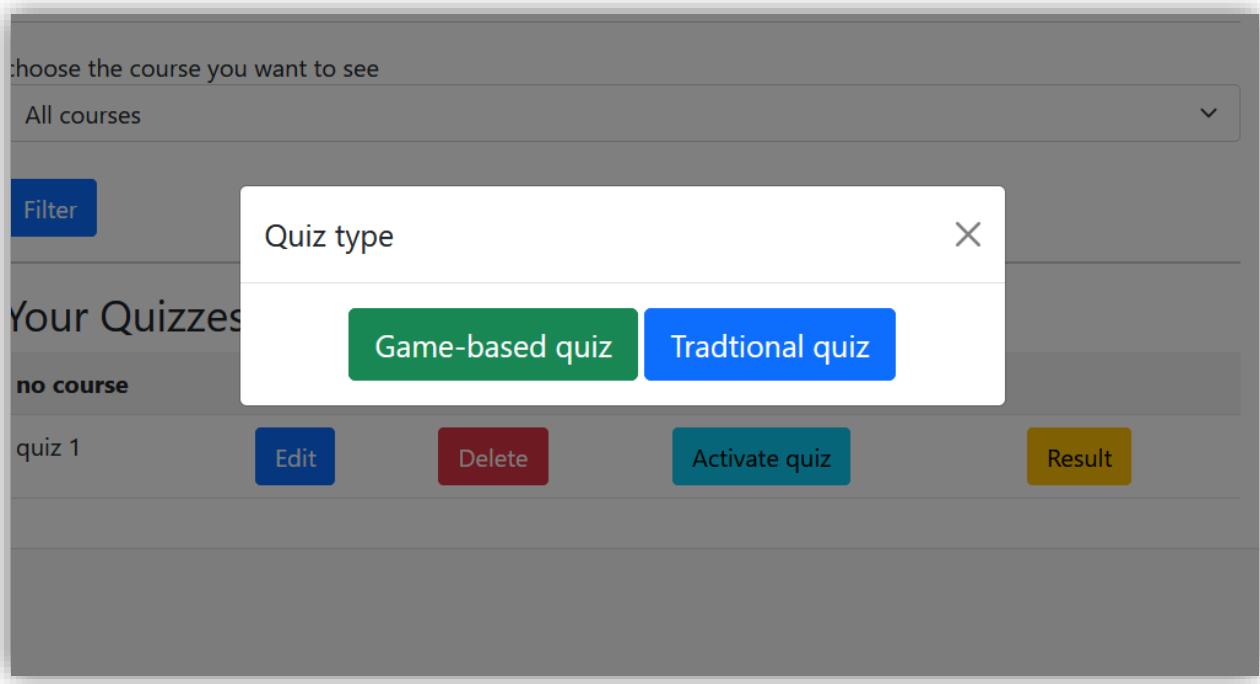


Figure 43: Create quiz User Interface 2

The *Figure 44* illustrates the User interface of the “Create quiz” use case and how create quiz by filling required information.

Quillz Home Features FAQs About

Enter Room Code

Create quiz

Quiz name
Math quiz1

choose the course
Without course

duration of the quiz: (in minutes)
10

Figure 44: Create quiz User Interface 3

13.6.4 Create question

The *Figure 45* illustrates the User interface of the “customize your quiz” use case where you can choose between true/false or multiple choice answer.

Quillz Home Features FAQs About

Enter Room Code

Customize your quiz

Your Questions

No question was added

Figure 45: Create question User Interface

The *Figure 46* illustrates the User interface of the “create question” use case of the true/false question and its required information.

Create question

What is the question ?

Write your question here

Question score:

25

Is the question

- True
- False

Submit

Figure 46: Create question User Interface 2

The *Figure 47* illustrates the User interface of the “create question” use case of the multiple choice question and its required information.

Create question

What is the question ?

Question score:

Write your question's answers

Answer (A)

Answer (B)

Answer (C)

Answer (D)

select the correct answer:

Figure 47: Create question User Interface 3

13.6.5 Edit question

The *Figure 48* illustrates the User interface of the “customize your quiz” and you can access edit question by click at desired question

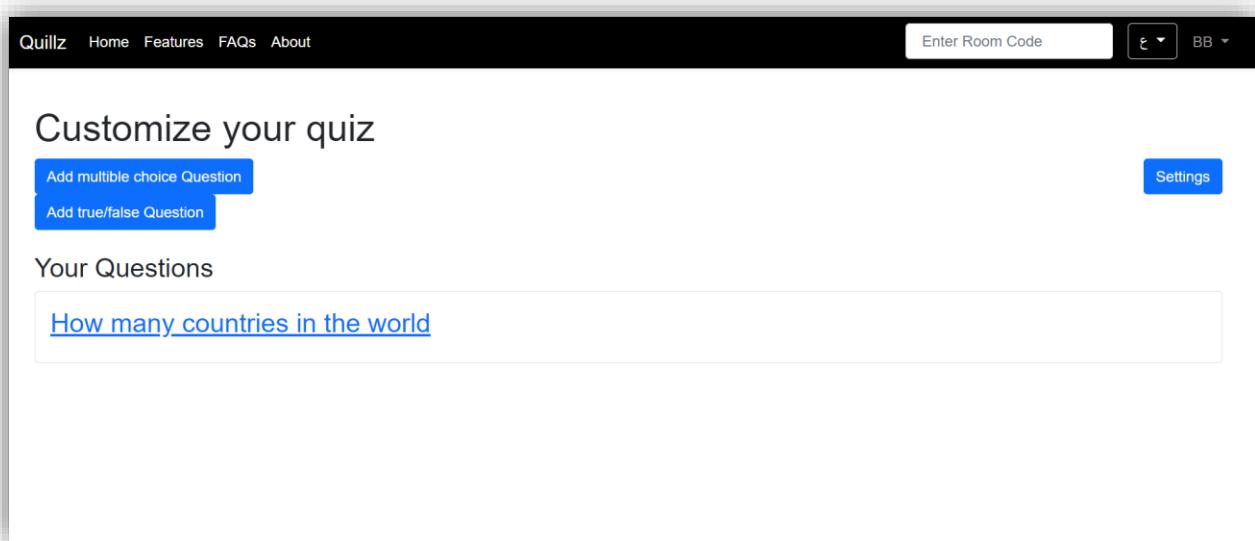


Figure 48: Edit question User Interface

The *Figure 49* illustrates the User interface of the “customize your quiz” and you can access edit question by click at Edit

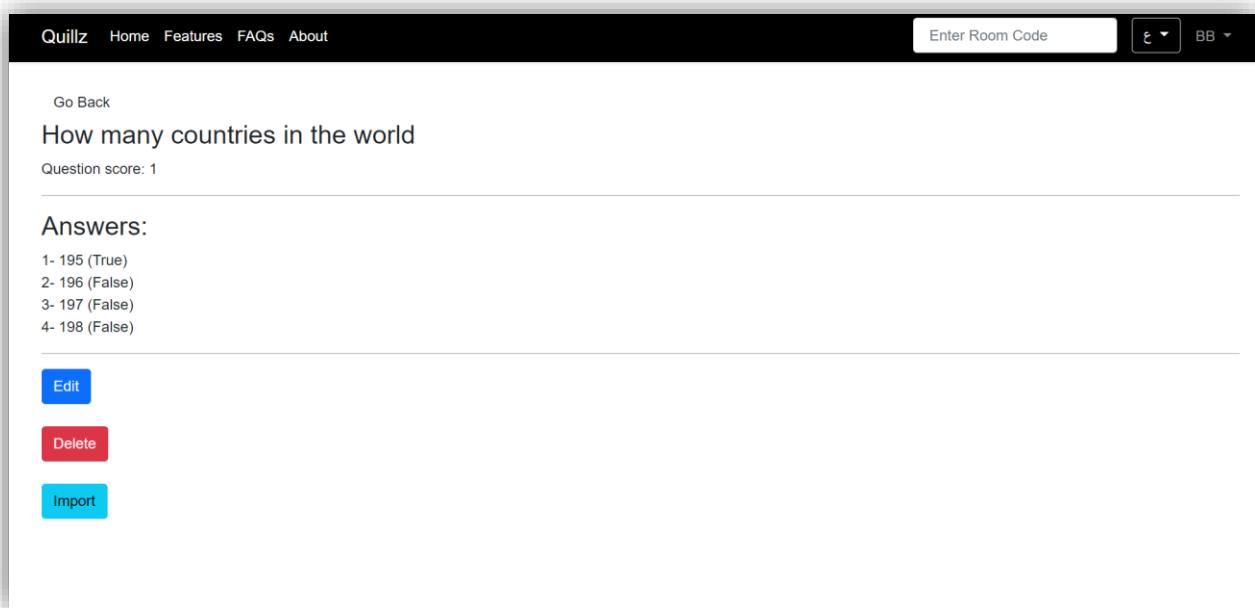


Figure 49: Edit question User Interface 2

The *Figure 50* illustrates the User interface of the “Edit question” and you can edit your question then click submit when finished

The screenshot shows a web-based form titled "Create question". At the top, there is a field labeled "What is the question ?" containing the text "How many countries in the world". Below it is a field labeled "Question score:" containing the value "1". A horizontal line separates this from the next section. The next section is titled "Write your question's answers". It contains four answer fields: "Answer (A)" with value "195", "Answer (B)" with value "196", "Answer (C)" with value "197", and "Answer (D)" with value "198". Below these is a field labeled "select the correct answer:" with a dropdown menu showing "Answer (A)". At the bottom left is a blue "Submit" button.

Figure 50: Edit question User Interface 3

13.6.6 Edit quiz

The *Figure 51* illustrates the User interface of the “View dashboard” and how to edit an quiz by clicking edit on selected quiz

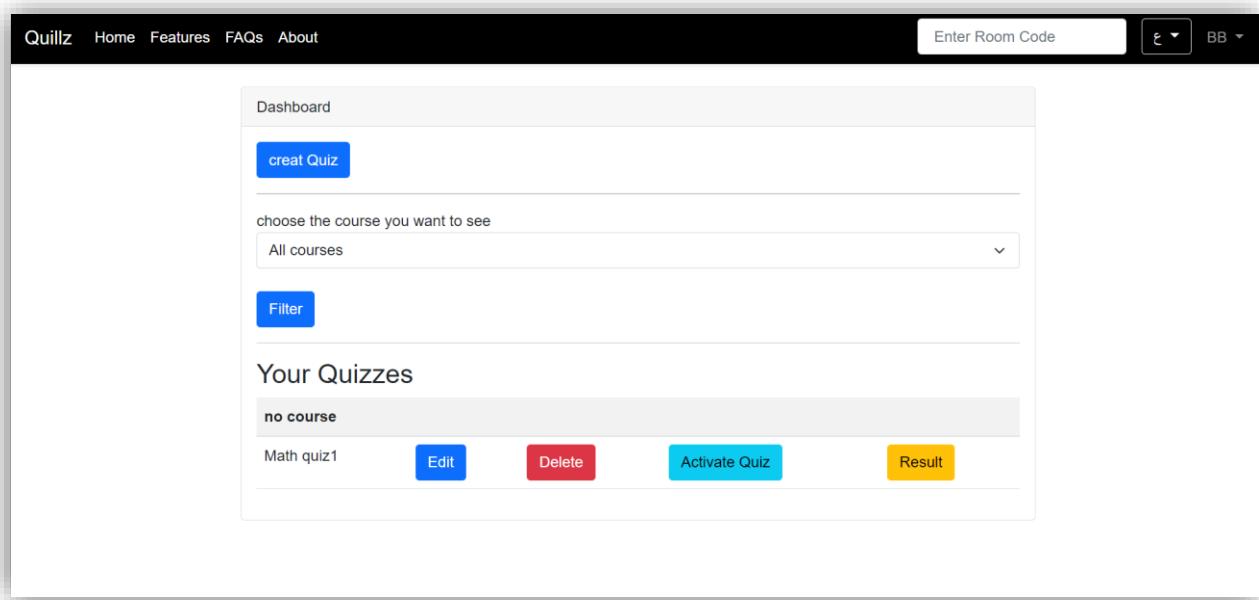


Figure 51: Edit quiz User Interface

The *Figure 52* illustrates the User interface of the “customize your quiz” and by clicking settings you can edit your quiz.

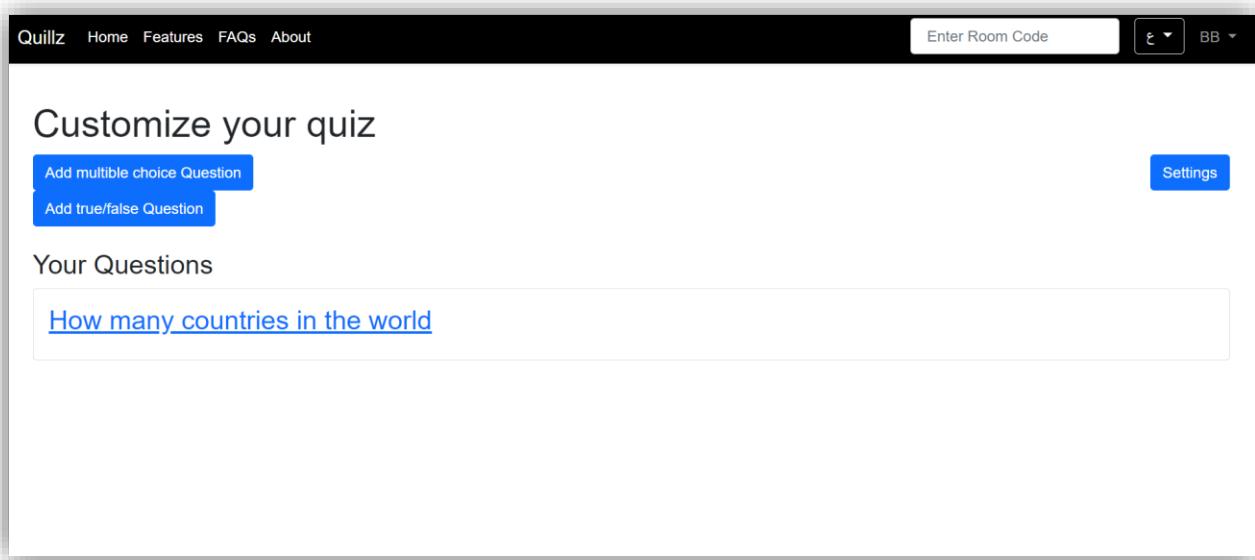


Figure 52: Edit quiz User Interface 2

The *Figure 53* illustrates the User interface of the “edit quiz” use case and the information required to edit the quiz then after finished press submit

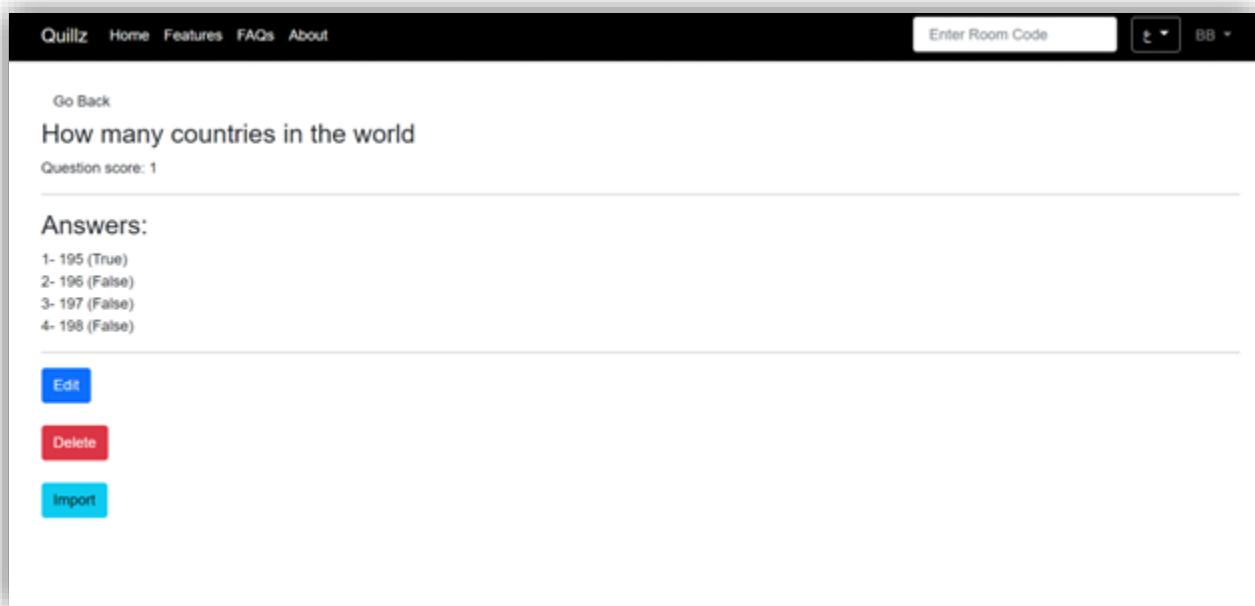


The screenshot shows the 'Edit quiz' page of the Quillz application. At the top, there is a navigation bar with links for 'Quillz', 'Home', 'Features', 'FAQs', and 'About'. To the right of the navigation bar are three input fields: 'Enter Room Code' with a dropdown arrow, a small 'e' icon, and a 'BB' dropdown menu. Below the navigation bar, the main content area has a title 'Edit quiz'. It contains several form fields: 'Quiz name' with the value 'Math quiz1', 'choose the course' (a dropdown menu), 'duration of the quiz: (in minutes)' with the value '10', and a blue 'Submit' button.

Figure 53: Edit quiz User Interface 3

13.6.7 Delete question

The *Figure 54* illustrates the User interface of the “customize your quiz” and you can delete the question by clicking delete



The screenshot shows the 'Delete question' page of the Quillz application. At the top, there is a navigation bar with links for 'Quillz', 'Home', 'Features', 'FAQs', and 'About'. To the right of the navigation bar are three input fields: 'Enter Room Code' with a dropdown arrow, a small 'e' icon, and a 'BB' dropdown menu. Below the navigation bar, the main content area shows a question: 'How many countries in the world' with a score of 'Question score: 1'. Above the question is a 'Go Back' link. Below the question, under 'Answers:', there is a list of four options: '1- 195 (True)', '2- 196 (False)', '3- 197 (False)', and '4- 198 (False)'. At the bottom of the page are three buttons: 'Edit' (blue), 'Delete' (red), and 'Import' (light blue).

Figure 54: Delete question User Interface

13.6.8 Delete quiz

The *Figure 55* illustrates the User interface of the “View dashboard” and you can delete the quiz by clicking delete button

The screenshot shows the Quillz dashboard. At the top, there is a navigation bar with links for Quillz, Home, Features, FAQs, and About. To the right of the navigation bar are fields for 'Enter Room Code' and 'BB' with a dropdown arrow. Below the navigation bar is a 'Dashboard' section with a 'creat Quiz' button. Underneath it is a dropdown menu labeled 'choose the course you want to see' with 'All courses' selected. A 'Filter' button is also present. The main area is titled 'Your Quizzes' and shows a list with one item: 'Math quiz1'. Below this list are four buttons: 'Edit' (blue), 'Delete' (red), 'Activate Quiz' (cyan), and 'Result' (yellow).

Figure 55: Delete quiz User Interface

13.6.9 Save question to question bank

The *Figure 56* illustrates the User interface of the “customize your quiz” and you can save question to question bank by clicking import

The screenshot shows a question card on the Quillz platform. At the top, there's a navigation bar with links for Quillz, Home, Features, FAQs, and About. To the right of the navigation is a search bar labeled "Enter Room Code" and a dropdown menu with "E" and "BB". Below the navigation, there's a "Go Back" link. The main content area displays a question: "How many countries in the world" with a question score of 1. Underneath the question, it says "Answers:" followed by a list of four options: 1- 195 (True), 2- 196 (False), 3- 197 (False), and 4- 198 (False). At the bottom of the card are three buttons: "Edit" (blue), "Delete" (red), and "Import" (teal).

Figure 56: Save question to question bank User Interface

The *Figure 57* illustrates the User interface of the “Edit question” and you can choose the category to import the selected question then click submit

What is the question ?
How many countries in the world

Question score:
1

choose the course
Math

Write your answer here

Answer (A)
195

Answer (B)
196

Answer (C)
197

Answer (D)
198

select the correct answer:
Answer (A)

Submit

The screenshot shows a web-based form for saving a question. At the top, there are fields for 'What is the question?' containing 'How many countries in the world', 'Question score:' with a value of '1', and 'choose the course' with a dropdown menu showing 'Math'. Below this, a large bold heading says 'Write your answer here'. Underneath are four answer options labeled (A) through (D), each with a numerical value: 195, 196, 197, and 198 respectively. Further down, a 'select the correct answer:' label is followed by a dropdown menu containing 'Answer (A)'. At the bottom right is a blue 'Submit' button.

Figure 57: Save question to question bank User Interface 2

The *Figure 58* illustrates the User interface of the “Question bank” you can see the save question in this page

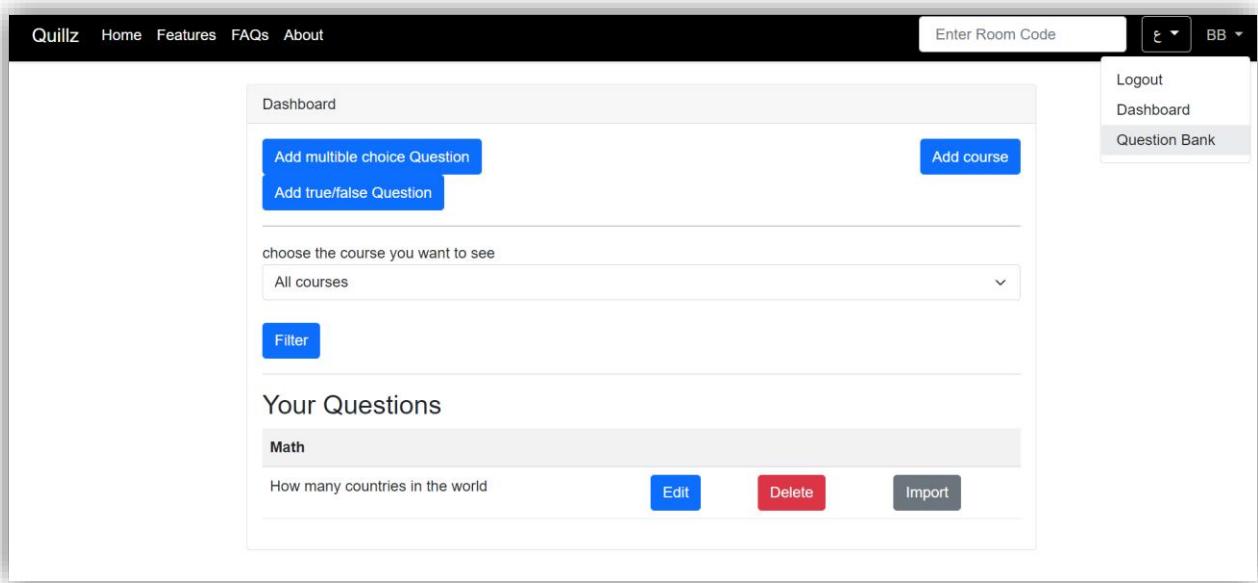


Figure 58: Save question to question bank User Interface 3

13.6.10 Import question

The *Figure 59* illustrates the User interface of the “Question bank” you can click import to import the question to selected quiz

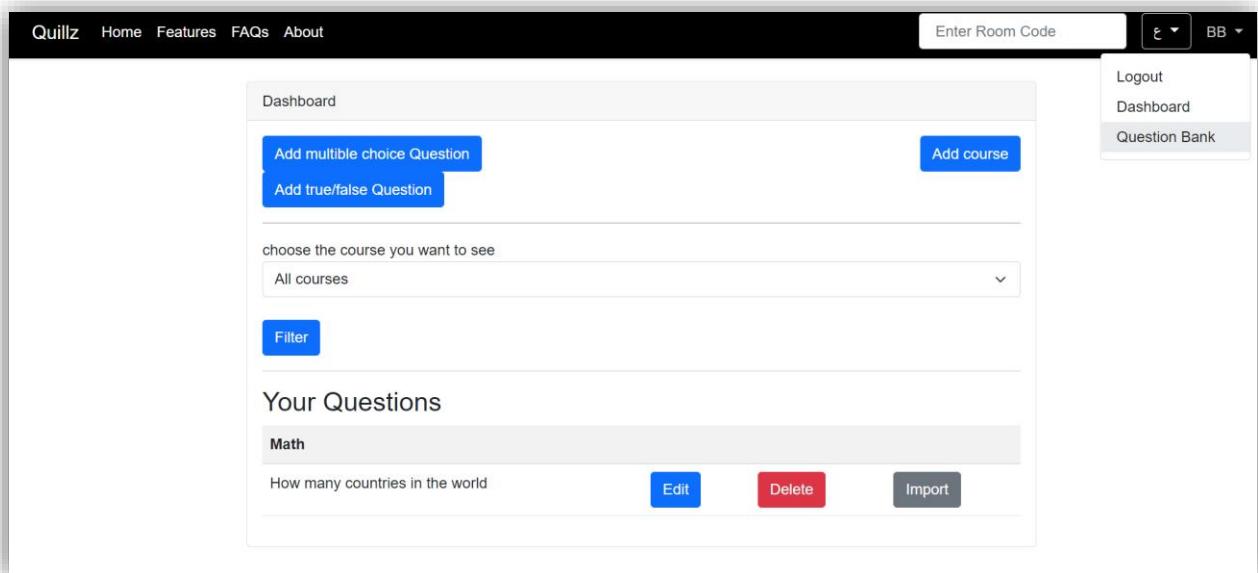


Figure 59: Import question User Interface

The *Figure 60* illustrates the User interface of the “Edit question” you can choose the quiz to import the selected question to it

Question score:
1

Write your question's answers

Answer (A)
195

Answer (B)
196

Answer (C)
197

Answer (D)
198

select the correct answer:
Answer (A)

text.select quiz
Math quiz1

Submit

Figure 60: Import question User Interface 2

The *Figure 61* illustrates the User interface of the “customize your quiz” and we can see that the selected question is successfully imported

The screenshot shows a web-based quiz creation tool. At the top, there's a navigation bar with links for Quillz, Home, Features, FAQs, and About. To the right of the navigation is a search bar labeled "Enter Room Code" and some user settings. Below the navigation, the main title is "Customize your quiz". There are two blue buttons: "Add multiple choice Question" and "Add true/false Question". On the far right, there's a "Settings" button. The section titled "Your Questions" contains two identical entries, each with a blue link: "How many countries in the world".

Figure 61: Import question User Interface 3

13.6.11 Active quiz

The *Figure 62* illustrates the User interface of the “View dashboard” and you can Active quiz by clicking active quiz and write the quiz code

The screenshot shows a "View dashboard" interface. At the top, there's a navigation bar with links for Quillz, Home, Features, FAQs, and About. To the right is a search bar labeled "Enter Room Code" and user settings. A modal window is open, prompting the user to "Enter the Quizzes code here (without spaces)" with the input field containing "Math101". Below the modal is a "Submit" button. The main content area is titled "Your Quizzes" and shows a list with one item: "no course" followed by "Math quiz1". For this quiz, there are four buttons: "Edit", "Delete", "Activate Quiz" (which is highlighted in blue), and "Result".

Figure 62: Active quiz User Interface

The *Figure 63* illustrates the User interface of the “View dashboard” and you see that the quiz is now activated

The screenshot shows the Quillz application's dashboard. At the top, there is a navigation bar with links for Quillz, Home, Features, FAQs, and About. On the right side of the navigation bar are fields for "Enter Room Code" and user settings. Below the navigation bar, the main area is titled "Dashboard". It features a "creat Quiz" button. A dropdown menu labeled "choose the course you want to see" contains the option "All courses", with a "Filter" button below it. The main content area is titled "Your Quizzes" and displays a list of quizzes. The first quiz listed is "Math quiz1" with a "Quiz code: Math101". To the right of this entry are two buttons: "Deactivate Quiz" (blue) and "Result" (yellow). Below this entry, the word "Math" is displayed, likely indicating the course category. The overall interface is clean and modern, using a light gray background with blue and yellow accents for buttons.

Figure 63: Active quiz User Interface 2

13.6.12 Join quiz

The *Figure 64* illustrates the User interface of the “Home Page” where you can put the quiz code in the field above

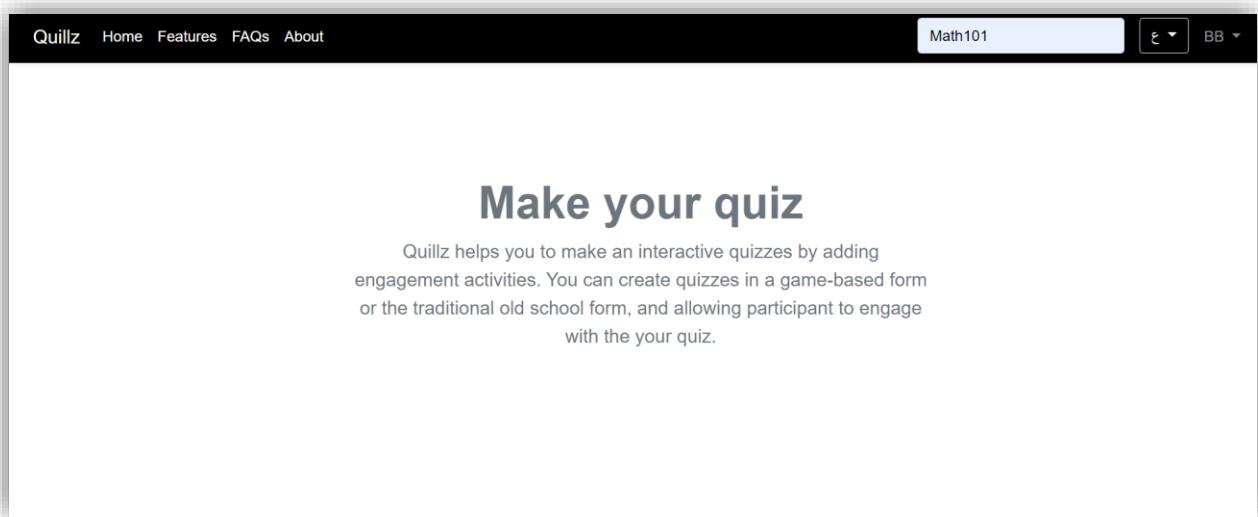


Figure 64: Join quiz User Interface

The *Figure 65* illustrates the User interface of the “Start quiz” where the participant enters his name then click submit.

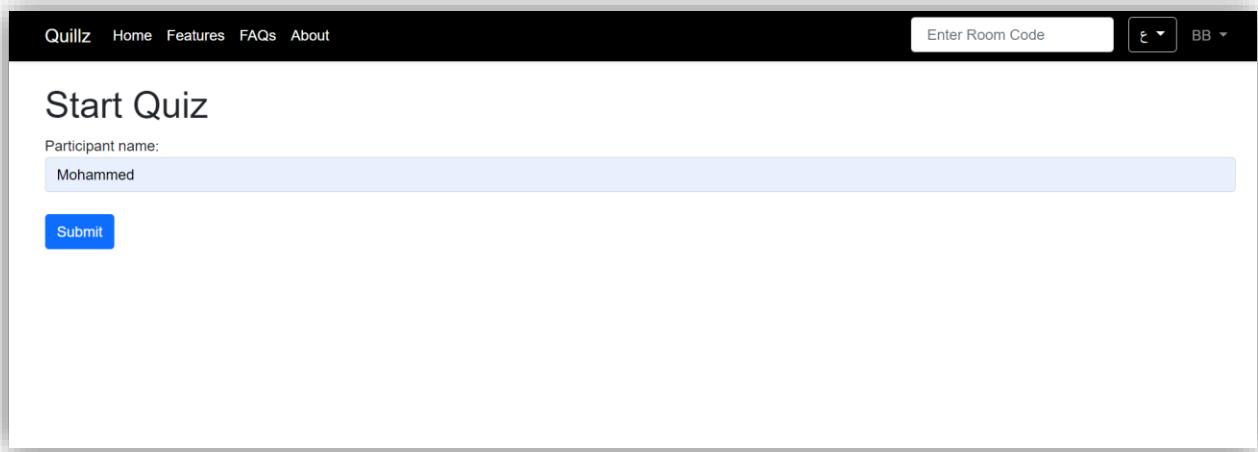


Figure 65: Join quiz User Interface 2

The *Figure 66* illustrates the User interface of the “Start quiz” where the participant click start quiz to begin the quiz.



Figure 66: Join quiz User Interface 3

The *Figure 67* illustrates the User interface of the “quiz page” where the participant start taking the quiz.

The screenshot shows a web-based quiz interface. At the top, there's a navigation bar with links for Quillz, Home, Features, FAQs, and About. To the right of the navigation is a search bar labeled "Enter Room Code" with a dropdown arrow, and a "BB" button. Below the navigation, the title "Quiz: Math quiz1" is displayed, along with the time "09:54".

Question 1: How many countries in the world

Four radio buttons are shown: 195, 196, 197, and 198.

Question 2: How many countries in the world

Four radio buttons are shown: 195, 196, 197, and 198.

A blue "Submit" button is located at the bottom left of the quiz area.

Figure 67: Join quiz User Interface 4

14. Testing

Testing is one of the important factors in the project because you could see if you met every requirement and to test if the system work with no problems. For test scenarios and functional testing, we perform the testing by creating a template to test the system manually. For unit testing we used Php unit. And for the usability testing we make some users to use and test our system and then we take their answers by creating form.

14.1 Test Scenarios

In Table 11 we will list all the test scenarios

Related Function	TEST CASE ID	TEST SCENARIO	TEST Steps and data	EXPECTED RESULT	ACTUAL RESULT	POST CONDITION
Register	TC1	Verify the register	The user register with Name: Mohammed Email Address: B@gmail.com	The user is registered	The user is registered	The user is redirected to

			Password: qwerty12345 Confirm Password: qwerty12345			dashboard page
Register	TC2	Verify the register	The user register with Name: Mohammed Email Address: <u>B</u> Password: qwerty12345 Confirm Password: qwerty12345	Display an error message	Display an error message	The user is redirected to register page
Register	TC3	Verify the register	The user register with Name: Mohammed Email Address: <u>B@gmail.com</u> Password: qwerty12345 Confirm Password: qwerty12345	Display an error message	Display an error message	The user is redirected to register page
Register	TC4	Verify the register	The user register with Name: “Blank” Email Address: <u>B@gmail.com</u> Password: qwerty12345 Confirm Password: qwerty12345	Display an error message	Display an error message	The user is redirected to register page
Register	TC5	Verify the register	The user register with Name: Mohammed Email Address: <u>B</u> Password: qwerty Confirm Password: qwerty	Display an error message	Display an error message	The user is redirected to register page
Register	TC5	Verify the register	The user register with Name: Mohammed Email Address: <u>B@gmail.com</u> Password: qwert12345 Confirm Password: qwerty12345	Display an error message	Display an error message	The user is redirected to register page

Register	TC6	Verify the register	The user register with Name: Mohammed Email Address: <u>B@gmail.com</u> Password: qwerty12345 Confirm Password: qwert12345	Display an error message	Display an error message	The user is redirected to register page
Register	TC7	Verify the register	The user register with Name: Mohammed Email Address: <u>B@gmail.com</u> Password: qwerty12345 Confirm Password: “Blank”	Display an error message	Display an error message	The user is redirected to register page
Register	TC8	Verify the register	The user register with Name: Mohammed Email Address: <u>B@gmail.com</u> Password: “Blank” Confirm Password: qwerty12345	Display an error message	Display an error message	The user is redirected to register page
Register	TC9	Verify the register	The user register with Name: Mohammed Email Address: “Blank” Password: qwerty12345 Confirm Password: qwerty12345	Display an error message	Display an error message	The user is redirected to register page
Login	TC10	Verify the Login	The user Login with Email Address: <u>B@gmail.com</u> Password: qwerty12345	The user Logged in	The user Logged in	The user is redirected to dashboard page
Login	TC11	Verify the Login	The user Login with Email Address: <u>B@gmail.com</u> Password: qwerty1234	Display an error message	Display an error message	The user is redirected to Login page

Login	TC12	Verify the Login	The user Login with Email Address: <u>B</u> @ Password: qwerty12345	Display an error message	Display an error message	The user is redirected to Login page
Login	TC13	Verify the Login	The user Login with Email Address: <u>B</u> @gmail.com Password: “Blank”	Display an error message	Display an error message	The user is redirected to Login page
Login	TC14	Verify the Login	The user Login with Email Address: “Blank” Password: qwerty12345	Display an error message	Display an error message	The user is redirected to Login page
View Dashboard	TC15	Verify the Dashboard	1-The user click on his name at top right 2-Then click on Dashboard	Dashboard page is displayed	Dashboard page is displayed	The user is redirected to dashboard page
Choose quiz type	TC16	Verify the ability to choose game-based quiz	1-the user click on create a quiz in the dashboard 2-the user click on game-based quiz	The user chooses the game-based quiz	The user chooses the game-based quiz	The user is redirected to create quiz page
Choose quiz type	TC17	Verify the ability to choose non game-based quiz	1-the user click on create a quiz in the dashboard 2-the user click on non game-based quiz	The user chooses the non game-based quiz	The user chooses the non game-based quiz	The user is redirected to create quiz page
Create quiz	TC18	Verify the Create quiz	In the create quiz page the user fills the information with Quiz name: Quiz1 Choose the course: Without course Duration of the quiz(in minutes): 10	The quiz is created	The quiz is created	The user is redirected to quiz customization page

Create quiz	TC19	Verify the Create quiz	In the create quiz page the user fills the information with Quiz name: "Blank" Choose the course: Without course Duration of the quiz(in minutes): 10	Display an error message	Display an error message	The user is redirected to create quiz page
Create quiz	TC20	Verify the Create quiz	In the create quiz page the user fills the information with Quiz name: Quiz1 Choose the course: Without course Duration of the quiz(in minutes): "Blank"	Display an error message	Display an error message	The user is redirected to create quiz page
Choose quiz type	TC21	Verify the ability to choose multiple choice question	1-In the customization page the user chooses Add multiple choice question	create multiple choice question page is displayed	create multiple choice question page is displayed	The user is redirected to create multiple choice question page
Choose quiz type	TC22	Verify the ability to choose True/False question	1-In the customization page the user chooses Add True/False question	create True/False question page is displayed	create True/False question page is displayed	The user is redirected to create True/False question page
Create question	TC23	Verify the ability to create multiple choice question	1-In the customization page the user chooses Add multiple choice question 2- The create question page now is shown and the user fills the information with What is the question: How many countries in the world? Question score: 1	The multiple choice question is been created	The multiple choice question is been created	The user is redirected to quiz customization page

			<p>Answer(A): 195 Answer(B): 196 Answer(C): 197 Answer(D): 198</p> <p>Select the correct answer: Answer(A)</p>			
Create question	TC24	Verify the ability to create multiple choice question	<p>1-In the customization page the user chooses Add multiple choice question</p> <p>2- The create question page now is shown and the user fills the information with</p> <p>What is the question: “Blank”</p> <p>Question score: 1</p> <p>Answer(A): 195 Answer(B): 196 Answer(C): 197 Answer(D): 198</p> <p>Select the correct answer: Answer(A)</p>	Display an error message	Display an error message	The user is redirected to create question page
Create question	TC25	Verify the ability to create multiple choice question	<p>1-In the customization page the user chooses Add multiple choice question</p> <p>2- The create question page now is shown and the user fills the information with</p> <p>What is the question: How many countries in the world?</p> <p>Question score: “Blank”</p> <p>Answer(A): 195 Answer(B): 196 Answer(C): 197</p>	Display an error message	Display an error message	The user is redirected to create question page

			Answer(D): 198 Select the correct answer: Answer(A)			
Create question	TC26	Verify the ability to create multiple choice question	1-In the customization page the user chooses Add multiple choice question 2- The create question page now is shown and the user fills the information with What is the question: How many countries in the world? Question score: 1 Answer(A): "Blank" Answer(B): 196 Answer(C): 197 Answer(D): 198 Select the correct answer: Answer(A)	Display an error message	Display an error message	The user is redirected to create question page
Create question	TC27	Verify the ability to create multiple choice question	1-In the customization page the user chooses Add multiple choice question 2- The create question page now is shown and the user fills the information with What is the question: How many countries in the world? Question score: 1 Answer(A): 195 Answer(B): "Blank" Answer(C): 197 Answer(D): 198	Display an error message	Display an error message	The user is redirected to create question page

			Select the correct answer: Answer(A)			
Create question	TC28	Verify the ability to create multiple choice question	<p>1-In the customization page the user chooses Add multiple choice question</p> <p>2- The create question page now is shown and the user fills the information with</p> <p>What is the question: How many countries in the world?</p> <p>Question score: 1</p> <p>Answer(A): 195</p> <p>Answer(B): 196</p> <p>Answer(C): “Blank”</p> <p>Answer(D): 198</p> <p>Select the correct answer: Answer(A)</p>	Display an error message	Display an error message	The user is redirected to create question page
Create question	TC29	Verify the ability to create multiple choice question	<p>1-In the customization page the user chooses Add multiple choice question</p> <p>2- The create question page now is shown and the user fills the information with</p> <p>What is the question: How many countries in the world?</p> <p>Question score: 1</p> <p>Answer(A): 195</p> <p>Answer(B): 196</p> <p>Answer(C): 197</p> <p>Answer(D): “Blank”</p> <p>Select the correct answer: Answer(A)</p>	Display an error message	Display an error message	The user is redirected to create question page

Create question	TC30	Verify the ability to create True/False question	<p>1-In the customization page the user chooses Add True/False question</p> <p>2- The create question page now is shown and the user fills the information with</p> <p>What is the question: The heaviest animal in the world is blue whale?</p> <p>Question score: 2</p> <p>Is the question True/False: True</p>	The True/False question has been created	The True/False question has been created	The user is redirected to quiz customization page
Create question	TC31	Verify the ability to create True/False question	<p>1-In the customization page the user chooses Add True/False question</p> <p>2- The create question page now is shown and the user fills the information with</p> <p>What is the question: The heaviest animal in the world is blue shark?</p> <p>Question score: 2</p> <p>Is the question True/False: false</p>	The True/False question has been created	The True/False question has been created	The user is redirected to quiz customization page
Create question	TC32	Verify the ability to create True/False question	<p>1-In the customization page the user chooses Add True/False question</p> <p>2- The create question page now is shown and the user fills the information with</p> <p>What is the question: “Blank”</p> <p>Question score: 2</p> <p>Is the question True/False: false</p>	Display an error message	Display an error message	The user is redirected to create question page
Create question	TC33	Verify the ability to create True/False question	<p>1-In the customization page the user chooses Add True/False question</p> <p>2- The create question page now is shown and the user fills the information with</p> <p>What is the question: The heaviest animal in the world is blue shark?</p>	Display an error message	Display an error message	The user is redirected to create question page

			Question score: “Blank” Is the question True/False: false			
Edit quiz	TC34	Verify Edit quiz	1-in the dashboard page press edit to the chosen quiz 2-in the customization page press settings in the top right 3- The edit page now is shown and the user fills the information with Quiz name: Quiz1 changed to Quiz2 Choose the course: Without course Duration of the quiz(in minutes): 10 changed to 15	Edit quiz is complete	Edit quiz is complete	The user is redirected to quiz customization page
Edit quiz	TC35	Verify Edit quiz	1-in the dashboard page press edit to the chosen quiz 2-in the customization page press settings in the top right 3- The edit page now is shown and the user fills the information with Quiz name: Quiz1 changed to “Blank” Choose the course: Without course Duration of the quiz(in minutes): 10 changed to 15	Display an error message	Display an error message	The user is redirected to edit quiz page
Edit quiz	TC36	Verify Edit quiz	1-in the dashboard page press edit to the chosen quiz 2-in the customization page press settings in the top right 3- The edit page now is shown and the user fills the information with Quiz name: Quiz1 changed to Quiz2 Choose the course: Without course	Display an error message	Display an error message	The user is redirected to edit quiz page

			Duration of the quiz(in minutes): 10 changed to “Blank”			
Edit question	TC37	Verify Edit question	<p>1-in the customization page choice the desired question we will choose “How many countries in the world?” question</p> <p>2-then we will press edit</p> <p>3- The create question page of the selected question now is shown and the user fills the information with</p> <p>What is the question: How many countries in the world?</p> <p>Question score: 1</p> <p>Answer(A): 195</p> <p>Answer(B): 196</p> <p>Answer(C): 197 changed to 199</p> <p>Answer(D): 198</p> <p>Select the correct answer:</p> <p>Answer(A)</p>	Edit question is complete	Edit question is complete	The user is redirected to quiz customization page
Edit question	TC38	Verify Edit question	<p>1-in the customization page choice the desired question we will choose “How many countries in the world?” question</p> <p>2-then we will press edit</p> <p>3- The create question page of the selected question now is shown and the user fills the information with</p> <p>What is the question: How many countries in the world? Changed to How many countries in the world?!?</p> <p>Question score: 1</p> <p>Answer(A): 195</p> <p>Answer(B): 196</p>	Edit question is complete	Edit question is complete	The user is redirected to quiz customization page

			Answer(C): 199 Answer(D): 198 Select the correct answer: Answer(A)			
Edit question	TC39	Verify Edit question	1-in the customization page choice the desired question we will choose “How many countries in the world?” question 2-then we will press edit 3- The create question page of the selected question now is shown and the user fills the information with What is the question: How many countries in the world?!? Question score: 1 changed to 2 Answer(A): 195 Answer(B): 196 Answer(C): 199 Answer(D): 198 Select the correct answer: Answer(A)	Edit question is complete	Edit question is complete	The user is redirected to quiz customization page
Edit question	TC40	Verify Edit question	1-in the customization page choice the desired question we will choose “How many countries in the world?” question 2-then we will press edit 3- The create question page of the selected question now is shown and the user fills the information with What is the question: How many countries in the world?!? Question score: 2 Answer(A): 195 changed to 196	Edit question is complete	Edit question is complete	The user is redirected to quiz customization page

			<p>Answer(B): 196 changed to 195 Answer(C): 199 Answer(D): 198 Select the correct answer: Answer(A) changed to Answer(B)</p>			
Edit question	TC41	Verify Edit question	<p>1-in the customization page choice the desired question we will choose “How many countries in the world?” question 2-then we will press edit 3- The create question page of the selected question now is shown and the user fills the information with What is the question: How many countries in the world?!? Changed to “Blank” Question score: 2 Answer(A): 196 Answer(B): 195 Answer(C): 199 Answer(D): 198 Select the correct answer: Answer(A)</p>	Display an error message	Display an error message	The user is redirected to edit question page
Edit question	TC42	Verify Edit question	<p>1-in the customization page choice the desired question we will choose “How many countries in the world?” question 2-then we will press edit 3- The create question page of the selected question now is shown and the user fills the information with What is the question: How many countries in the world?!?</p>	Display an error message	Display an error message	The user is redirected to edit question page

			<p>Question score: 2 changed to “Blank”</p> <p>Answer(A): 196</p> <p>Answer(B): 195</p> <p>Answer(C): 199</p> <p>Answer(D): 198</p> <p>Select the correct answer:</p> <p>Answer(A)</p>			
Edit question	TC43	Verify Edit question	<p>1-in the customization page choice the desired question we will choose “How many countries in the world?” question</p> <p>2-then we will press edit</p> <p>3- The create question page of the selected question now is shown and the user fills the information with</p> <p>What is the question: How many countries in the world?!?</p> <p>Question score: 2 changed to “Blank”</p> <p>Answer(A): 196</p> <p>Answer(B): 195</p> <p>Answer(C): 199 changed to “Blank”</p> <p>Answer(D): 198</p> <p>Select the correct answer:</p> <p>Answer(A)</p>	Display an error message	Display an error message	The user is redirected to edit question page
Delete question	TC44	Verify delete question	<p>1-in the customization page choose the desired question</p> <p>2-click on Delete button</p>	The chosen question is deleted	The chosen question is deleted	The user is redirected to quiz customization page

Delete quiz	TC45	Verify delete quiz	1-in the dashboard page press delete button on selected quiz	The chosen quiz is deleted	The chosen quiz is deleted	The user is redirected to dashboard page
Save question to question bank	TC46	Verify saving question to question bank	1-from question bank page we click on add new category 2- we create new category named Math 3-from quiz customization page we select the desired question then click on import 4- The create question page of the selected question now is shown and we need to choose Math on choose the course	The chosen question is imported to question bank	The chosen question is imported to question bank	The user is redirected to question bank page
Save question to question bank	TC47	Verify saving question to question bank	1-from quiz customization page we select the desired question then click on import 2- The create question page of the selected question now is shown and we don't have any saved category	Display an error message	Display an error message	The user is redirected to edit question page
Import question	TC48	Verify importing question to quiz	1-from question bank page choose the desired question to import by pressing import button 2-then select the quiz you want to import the question to, we will choose Quiz1	The question is imported to selected quiz	The question is imported to selected quiz	The user is redirected to question bank page
Import question	TC49	Verify importing question to quiz	1-from question bank page choose the desired question to import by pressing import button	Display an error message	Display an error message	The user is redirected to edit question page

			2-then select the quiz you want to import the question to, but we don't have any created quizzes			
Active quiz	TC50	Verify active quiz	1-from dashboard page by choosing desired quiz click Active quiz 2-write quiz code without spacing in Enter the Quizzes code here (without spaces): 123	The chosen quiz is activated	The chosen quiz is activated	The user is redirected to dashboard page
Active quiz	TC51	Verify active quiz	1-from dashboard page by choosing desired quiz click Active quiz 2-write quiz code without spacing in Enter the Quizzes code here (without spaces): 123	Display an error message	Display an error message	The user is redirected to dashboard page
Active quiz	TC52	Verify active quiz	1-from dashboard page by choosing desired quiz click Active quiz (the selected quiz doesn't have any questions) 2-write quiz code without spacing in Enter the Quizzes code here (without spaces): 12345	Display an error message	Display an error message	The user is redirected to dashboard page
Join quiz	TC53	Verify join quiz	1-from the home page the user enters room code at the textbox on the top right 2-the participant enter 123 as the room code 3- the participant enters his name as Khalid	The participant enters the quiz	The participant enters the quiz	The participant redirected to start quiz page
Join quiz	TC54	Verify join quiz	1-from the home page the user enters room code at the textbox on the top right 2-the participant enter 123 as the room code 3- the participant enters his name as "Blank"	Display an error message	Display an error message	The participant redirected to enter participant name page

Join quiz	TC55	Verify join quiz	1-from the home page the user enters room code at the textbox on the top right 2-the participant enter 123123 as the room code	Display an error message	Display an error message	The participant redirected to home page
-----------	------	------------------	---	--------------------------	--------------------------	---

Table 11: Test Scenarios

14.2 Unit Test

In Table 12 we will test some of main function in our system

#	Function name	Input	Expected output	Resulting output	Pass/ Fail
1	CreateQuiz	Quiz name,Course, duration	True if the quiz is created, then gets redirected to the customization page.	True	Pass
2	AddMCQQuestion	Question head, Score, Answer A, Answer B, Answer C, Answer D, Correct answer	True if the question is created, then gets redirected to the customization page.	True	Pass
3	addTFQuestion	Question head, Score, Answer	True if the question is created, then gets redirected to the customization page	True	Pass
4	CreateCategory	Category name	True if the category is created, then gets redirected to the question bank page	True	Pass

Table 12: Unit test

14.2.1 ‘CreateQuiz’ function

Figure 68 illustrates the code for the ‘createQuiz’ test case and the results.

```
21  /**@test */
22  public function test_create_quiz()
23  {
24
25      $user=User::find(1);
26
27      $response=$this->actingAs($user)->call('POST','/quizzes', [
28          'head'=>Str::random(15),
29          'course'=>'swe481',
30          'duration'=>rand(0,50),
31      ]);
32
33      $response->assertRedirectContains('questions');
34
35
36  }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

```
PASS Tests\Unit\ProjectTest
✓ create quiz
✓ add mcq question
✓ add tf question
✓ create course

Tests: 4 passed
Time: 0.49s
```

Figure 68: ‘CreateQuiz’ function

14.2.2 ‘addMCQQuestion’ function

Figure 69 illustrates the code for ‘addMCQQuestion’ test case and the results.

```
38     public function test_add_mcq_question()
39     {
40
41         $user=User::find(1);
42
43         $response=$this->actingAs($user)->call('POST', 'questions/saveMCQ/1', [
44             'head'=>Str::random(15),
45             'score'=>rand(0,100),
46             'answer1'=>Str::random(15),
47             'answer2'=>Str::random(15),
48             'answer3'=>Str::random(15),
49             'answer4'=>Str::random(15),
50             'correct'=>1,
51         ]);
52
53         $response->assertRedirectContains('questions');
54
55
56     }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

```
PASS Tests\Unit\ProjectTest
✓ create quiz
✓ add mcq question
✓ add tf question
✓ create course

Tests: 4 passed
Time: 0.49s
```

Figure 69: ‘addMCQQuestion’ function

14.2.3 ‘addTFQuestion’ function

Figure 70 illustrates the code for ‘addTFQuestion’ test case and result.

```
58     public function test_add_tf_question()
59     {
60
61         $user=User::find(1);
62
63         $response=$this->actingAs($user)->call('POST','questions/saveTF/1', [
64             'head'=>Str::random(15),
65             'score'=>rand(0,100),
66             'answer'=>'True',
67         ]);
68
69         $response->assertRedirectContains('questions');
70
71     }
72
73 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

PASS Tests\Unit\ProjectTest
✓ create quiz
✓ add mcq question
✓ add tf question
✓ create course

Tests: 4 passed
Time: 0.49s

Figure 70: ‘addTFQuestion’ function

14.2.4 ‘createCategory’ function

Figure 71 illustrates the code for ‘createCategory’ test case and result.

The screenshot shows a terminal window with the following content:

```
74     public function test_create_category()
75     {
76
77         $user=User::find(1);
78
79         $response=$this->actingAs($user)->call('POST','/questionBank/store', [
80             'name'=>Str::random(15),
81
82         ]);
83
84         $response->assertRedirectContains('questionBank');
85
86
87     }
88
89 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

```
PASS Tests\Unit\ProjectTest
✓ create quiz
✓ add mcq question
✓ add tf question
✓ create course

Tests: 4 passed
Time: 0.49s
```

Figure 71: ‘createCategory’ function

14.3 Functional Test

By performing test scenarios in section 14.1 we also completed functional testing.

14.4 Usability Test

We have made surveys with open and close ended question about the useability of our system:

Q1: How would you describe your overall experience with the website?

Figure 72 show the answer results

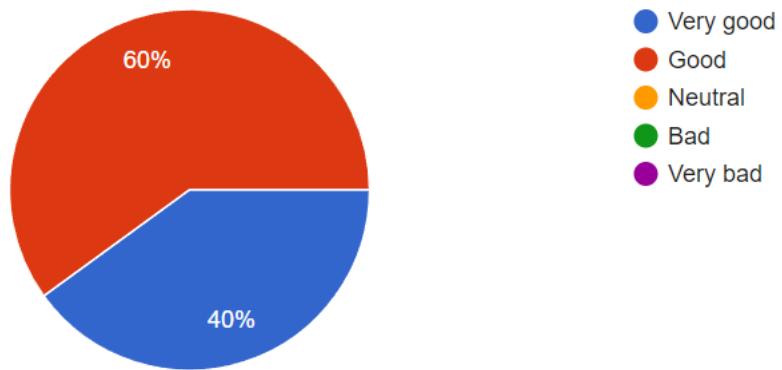


Figure 72: Usability test Answer No.1

Q2: I think that I would use this system frequently

Figure 73 show the answer results

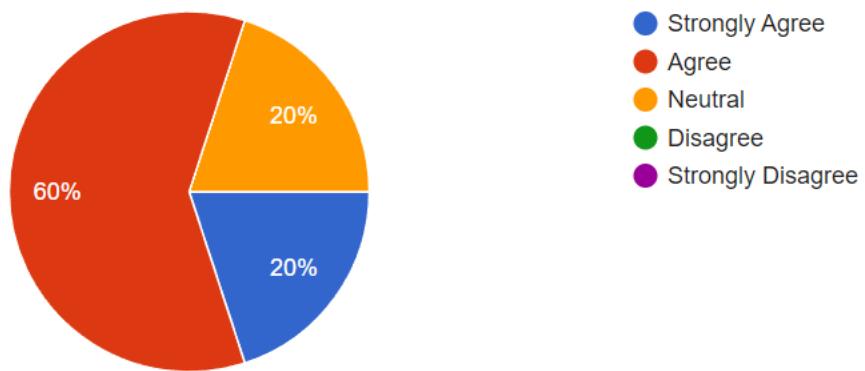


Figure 73: Usability test Answer No.2

Q3: I found the system unnecessarily complex

Figure 74 show the answer results



Figure 74: Usability test Answer No.3

Q4: I think the system is easy to use

Figure 75 show the answer results

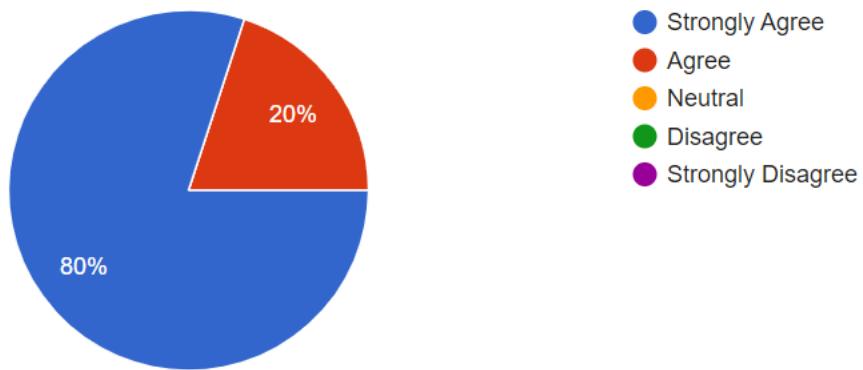


Figure 75: Usability test Answer No.4

Q5: Anything you want to add to the system?

The users suggest:

- 1- Another type of question like: multiple answer questions
- 2- Ability to add images and audio to the quiz

15. Deployment of the System

The *Figure 76* illustrates the system deployment where we have the web browser which displays web pages that the user interacts with. The web server and application server stores business logic. Database where the data is stored and managed.

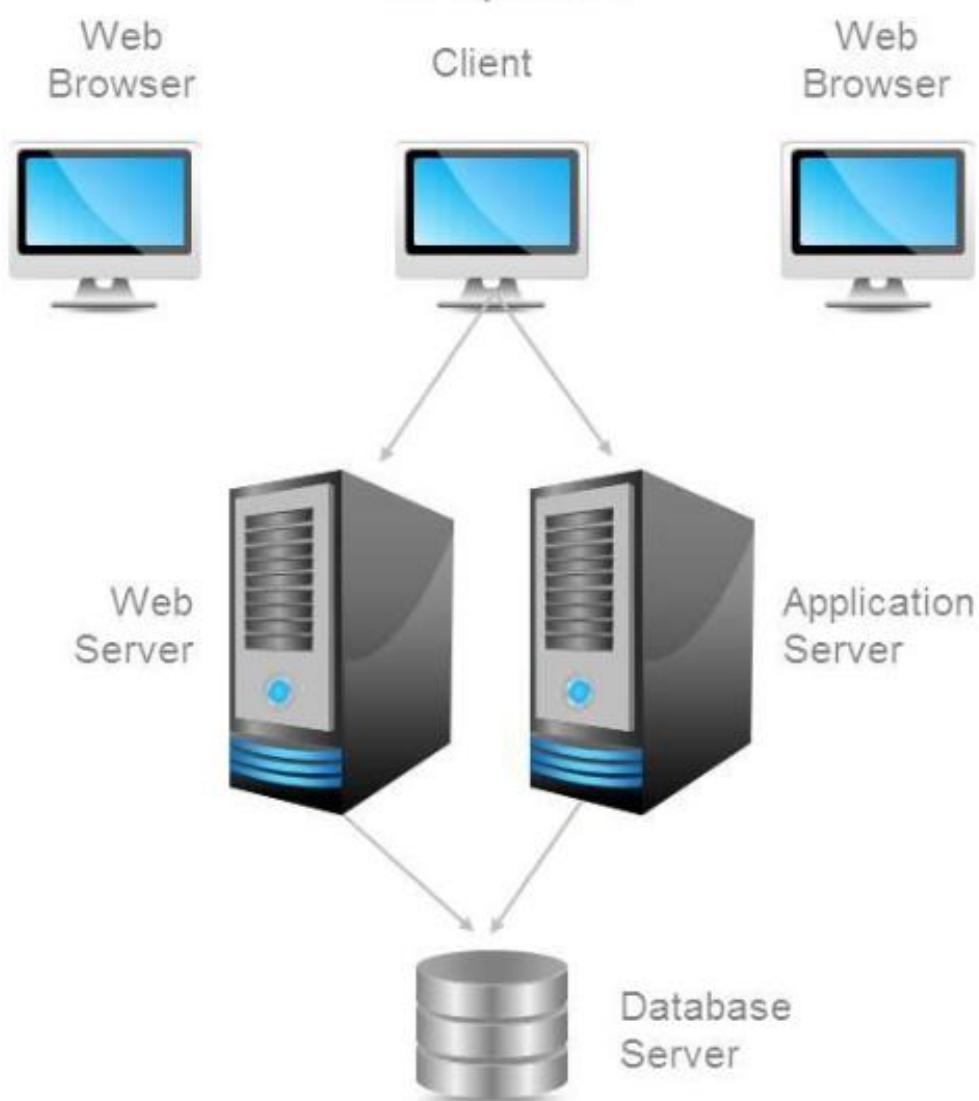


Figure 76: System deployment

16. Limitation of the System

There is a lot of futures and improvement we want to add to our system, but due the change of deadline of the project and the pressure of the other courses, we could not implement them

1-Ability to add an Audio to the quiz

2-Ability to add an Image to the quiz

3-Improvement in the User Interface

17. Conclusion and Future Work

in this document we were able to refine it using the appendix where we write the changes and everything new we added, which help us at the end to develop our website as intended.

For the future, we want to keep going on developing our website where we want to add the deleted function that we were not able to implement due the lack of time and experience also we want to develop more enhanced UI and we want to move our website to better hosting that can handle more users.

18. Conclusion

Throughout this report, we have gradually refined the document and encounter some the difficulties that project managers might face during the planning and designing phases we have acquired the required knowledge on the various aspect of the project management and how to organize the development process. although we have found quite the competitors (Kahoot, AhaSlides, ClassMarker) but that did not stop us from trying to build an even better and special software. In this report we defined a project plan that shows what will be done each week with the risks we might face. We also defined the functionalities of the system, use cases, and illustrated those use cases to give a better detailed understanding. Furthermore, we have shown different components of our system and how it will be deployed. To conclude, we hope the future of this project will be a highly rewording experience for us all and we aspire to raise our knowledge in different development aspects.

19. References

1. Zoom. 2021. *Polling for meetings*. [online] Available at: <https://support.zoom.us/hc/en-us/articles/213756303-Polling-for-meetings>
2. Kahoot. 2021. *About us*. [online] Available at: <https://kahoot.com/company>
3. AhaSlides. 2021. *About Us - AhaSlides*. [online] Available at: <<https://ahaslides.com/about/>>
4. Rawsthorne, P., 2011. *MVC in a three-tier architecture - TRANSLATED*. [online] Criticaltechnology.blogspot.com. Available at: <http://criticaltechnology.blogspot.com/2011/10/mvc-in-three-tier-architecture.html>
5. Ibm.com. 2020. *What is Three-Tier Architecture*. [online] Available at: <https://www.ibm.com/sa-en/cloud/learn/three-tier-architecture>
6. Quian, K., 2010. *Software architecture and design illuminated*. 1st ed. Sudbury: Jones and Bartlett Publishers.
7. Luchaninov, Y., 2021. *Web Application Architecture: Choosing the Right Type in 2021*. [online] MobiDev. Available at: <<https://mobidev.biz/blog/web-application-architecture-types>>

20. Appendix

20.1 Appendix A. Risks/ Constraints Management History

In the *Table 13* we will describe the risks and how do we manage it.

No.	Risk	How we managed it?
1	A misunderstanding of the requirements	We had weekly meetings with the advisor to review our work and discuss what should be changed and what should remain the same
2	Conflicts between projects deliverables and other exams/assignments.	We divide the work between team members and make a good plan to ensure no conflict between this project and the other project
3	Since the project is web-based, you need to ensure high connectivity and reduced downtime.	We purchase a good hosting to ensure high connectivity and reduce downtime

Table 13: Risk management history

Also, in the *Table 14* we will describe the constraints on the project and how do we manage it.

Project constraints		
No.	Constraint	How we managed it?
1	The system shall be a web application.	We use the best tools and framework of web application development
2	The project will need to be developed and deployed within this semester	We organized our time and work so we can deliver the project on the deadline
3	The system shall support multiple browsers	After the implementation of the website, we test multiple browser to

		make sure everything work as intended
--	--	---------------------------------------

Table 14: Constraints management history

20.2 Appendix B. Functional Requirement Updates

First, deleted requirement (SR No. based on SWE496 document)

Delete SR2 which was ‘The system shall send a confirmation email to the user to activate his account after sign up process is complete successfully.’

Delete SR5 which was ‘The system shall allow the user to change his email.’

Delete SR7 which was ‘The system shall allow the admin to add an account.’

Delete SR8 which was ‘The system shall allow the admin to delete an account.’

Delete SR9 which was ‘The system shall allow the admin to edit account information.’

Delete SR10 which was ‘The system shall allow the admin terminate room.’

Delete SR13 which was ‘The system shall allow the user to search for a certain page and FAQs.’

Delete SR19 which was ‘The system shall allow the user to create a poll, the poll shall be saved in the ‘Library’.

Delete SR20 which was ‘The system shall allow the user to delete a poll.’

Delete SR21 which was ‘The system shall allow the user to edit a poll.’

Delete SR22 which was ‘The system shall allow the user to see the statistics of the poll.’

Delete SR23 which was ‘The system shall allow the user to share a quiz or a poll to the public by creating a code.’

Delete SR24 which was ‘The system shall allow the user to create questions in a quiz or poll.’

Delete SR26 which was ‘the system shall allow the user to save question from session to question bank.’

Delete SR34 which was ‘The system shall allow the user to delete an answer from a question or poll.’

Delete SR40 which was ‘The system shall allow the user to import an image in a quiz or poll as a question or answer.’

Delete SR41 which was ‘The system shall allow the user to import an audio in a quiz or poll as a question.’

Delete SR42 which was ‘The system shall allow the user to choose an answer or more to be the correct answer based on the question type.’

Delete SR44 which was ‘The system shall allow the user to try his own quiz.’

Delete SR45 which was ‘The system shall allow the user to delete an image or audio from a question or poll.’

Delete SR46 which was ‘The system shall allow the user to change an image or audio in a question or poll.’

Delete SR47 which was ‘The system shall allow the user to crop an image in a question or poll.’

Delete SR48 which was ‘The system shall allow the user to hide participants names from the poll.’

Delete SR49 which was ‘The system shall allow the user to view their poll statistics in a pie, bar or a donut chart.’

Delete SR50 which was ‘The system shall allow the user to hide the poll results from participants.’

Delete SR53 which was ‘The system shall allow the user to pause at any time during the quiz.’

Delete SR54 which was ‘The system shall allow the user to resume the quiz whenever paused.’

Delete SR58 which was ‘The system shall keep track of users scores’

Delete SR59 which was ‘The system shall allow the user to choose one answer or more based on the quiz type.’

Delete SR61 which was ‘The system shall display the top 5 scores with their names at the end of the game-based quiz.’

Second, modified requirement

Change SR7 from ‘The system shall allow the user to create a quiz with a chosen type, the quiz shall be saved in the ‘Library’.’ **to** ‘The system shall allow the user to create a quiz with a chosen type’.

Change SR8 from ‘The system shall allow the user to view the library in his dashboard.’ **to** ‘The system shall allow the user to view the quizzes in his dashboard.’

Change SR15 from ‘The system shall allow the user to import question from question bank to session.’ **to** ‘The system shall allow the user to import question from question bank to quiz.’

Change SR17 from ‘The system shall allow the user to create question as multiple-choice question or true-false in a quiz.’ **to** ‘The system shall allow the user to create question as multiple-choice question in a quiz.’

Change SR20 from ‘The system shall allow the user to edit a question in a quiz or poll.’ **to** ‘The system shall allow the user to edit a question in a quiz.’

Change SR25 from ‘The system shall allow the user to determine the time of each question in quiz.’ **to** ‘The system shall allow the user to determine the time of quiz.’

Change SR26 from ‘The system shall allow the user to determine the maximum and minimum score of a question if the quiz type is game-based quiz.’ **to** ‘The system shall allow the user to determine the score of a question in a quiz’

Change SR28 from ‘The system shall allow the user to start the quiz’ **to** ‘The system shall allow the user to activate the quiz by writing the code.’

Change SR29 from ‘The system shall allow the user to end the quiz at any time’ **to** ‘The system shall allow the user to deactivate the quiz.’

Change SR30 from ‘The system shall allow the user to enter username’ **to** ‘The system shall allow the participant to enter username to take the quiz.’

Change SR31 from ‘The system shall allow the user to enter quiz code.’ **to** ‘The system shall allow the participant to enter the quiz code’

Change SR32 from ‘The system shall view results after each question in ordered leaderboard based on their scores.’ **to** ‘The system shall show the results of a game-based quiz in a leaderboard for all participants based on their scores after they finished it.’

Third, new requirement

SR18.The system shall allow the user to create question as true-false in a quiz.

SR22.The system shall allow the user to edit an answer from a question

SR33.The system shall display the remaining time of quiz for the participants.

20.3 Appendix C. Non-Functional Requirement Updates

First, deleted non-functional requirement (SR No. based on SWE496 document)

Delete SR64 which was the system shall have the ability to add module by developing and “plugging in” the software necessary to support the new module. A new module shall not require changes to the core software or original source code.

Second, modified requirement

Change SR34 from Average time to create a quiz or a poll shall not exceed 15 minutes if the quiz or poll is consisted of 10 questions and 4 answers for each one and they are ready to be written. **to** Average time to create a quiz shall not exceed 15 minutes if the quiz is consisted of 10 questions and 4 answers for each one and they are ready to be written.

20.4 Appendix D. Use Case Updates

Changes on use case:

- Change the name of join room to join quiz
- Change the name of view library to view dashboard
- Change the name of create session to create quiz
- Change the name of choose session type to choose quiz type
- Change the name of edit session to edit quiz
- Change the name of save session to save quiz
- Change the name of delete session to delete quiz
- Add create question
- Add choose question type
- Add save question
- Add save question to question bank
- Add import question
- Delete admin actor
- Delete manage platform users
- Delete manage platforms room

In figure 77 in shown the old use case

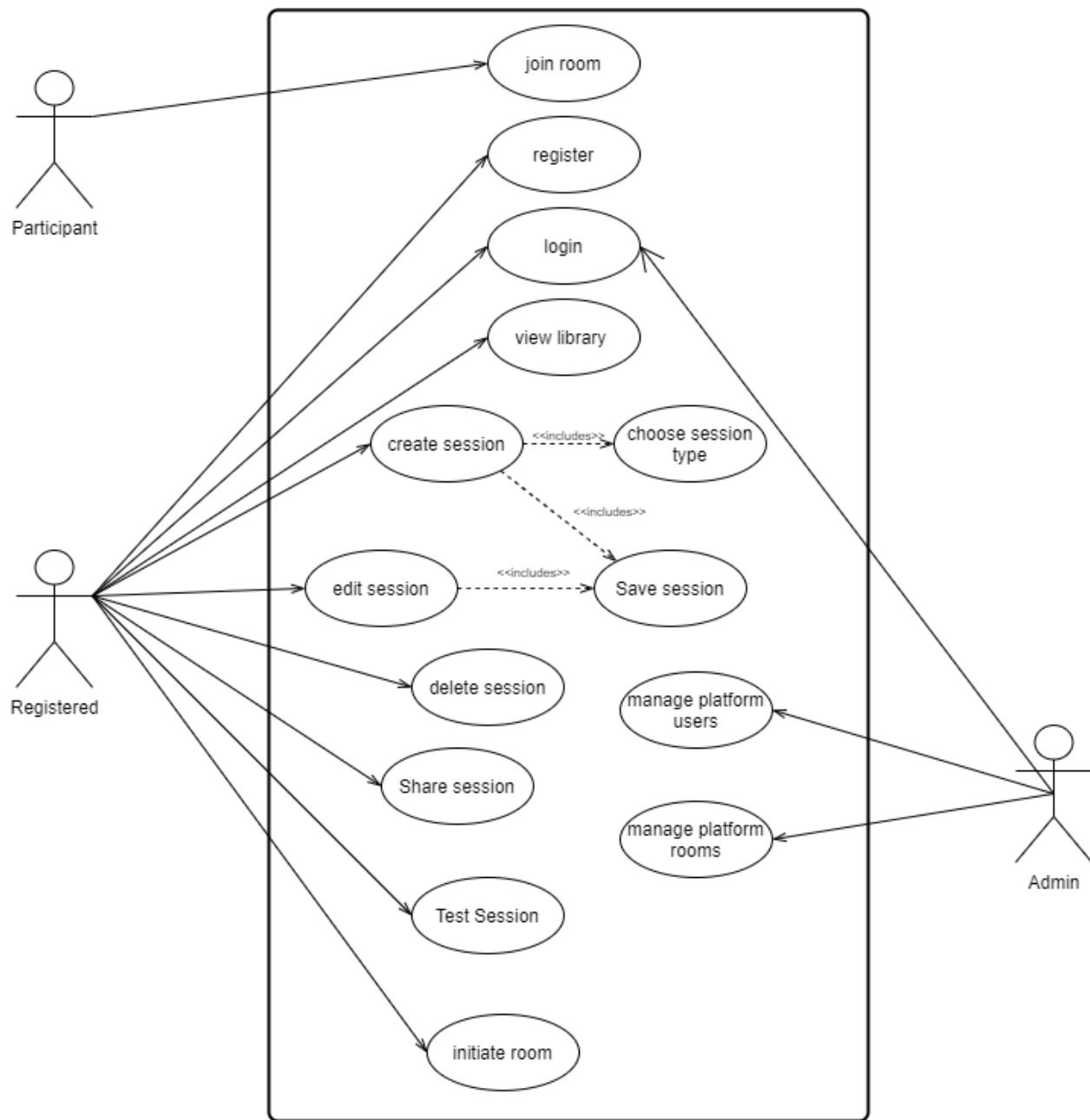


Figure 77: Old use case

20.5 Appendix E. Software Architecture Updates

We have changed the architecture of our system from a combination of MVC and multi-tier architecture to client/server architecture

20.6 Appendix F. Design Updates

Figure 78 illustrate old Join room sequence diagram and the updated one shown in section 10.1 figure 5

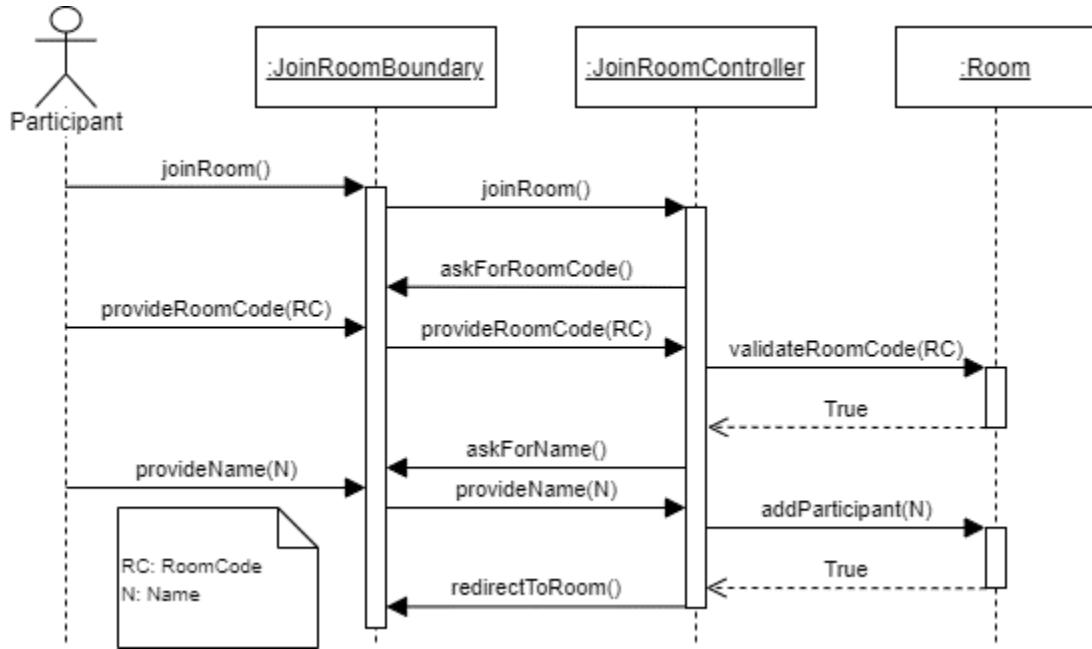


Figure 78: Old Join room sequence diagram

Figure 79 illustrate old Create session sequence diagram and the updated one shown in section 10.1 figure 6

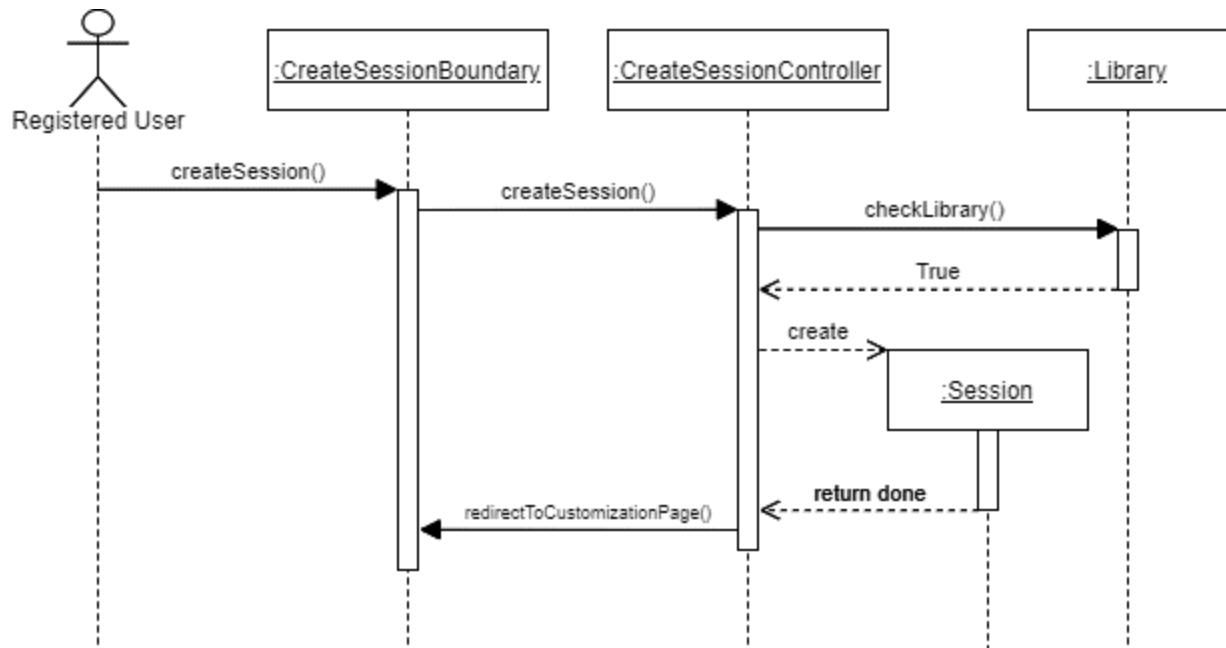


Figure 79: Old Create session sequence diagram

Figure 80 illustrate old View library sequence diagram and the updated one shown in section 10.1 figure 7

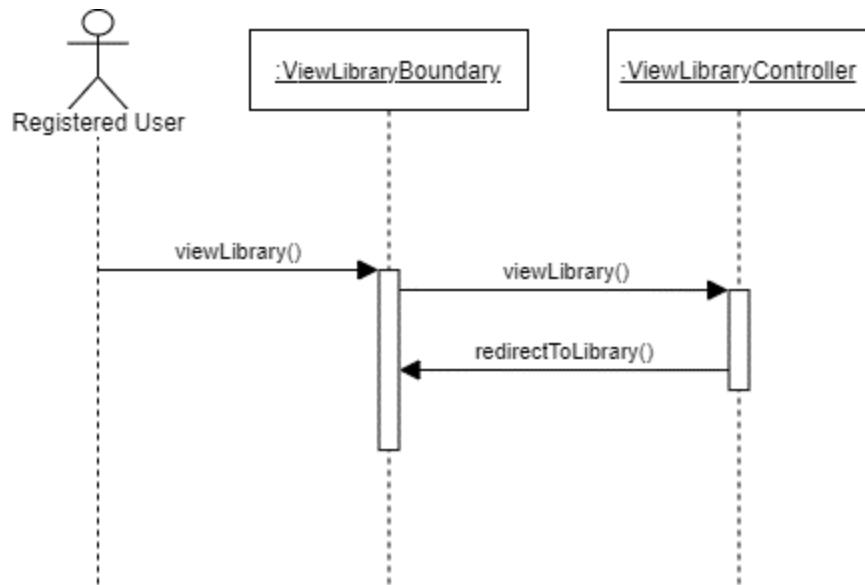


Figure 80: Old View library sequence diagram

Figure 81 illustrate old Save session sequence diagram

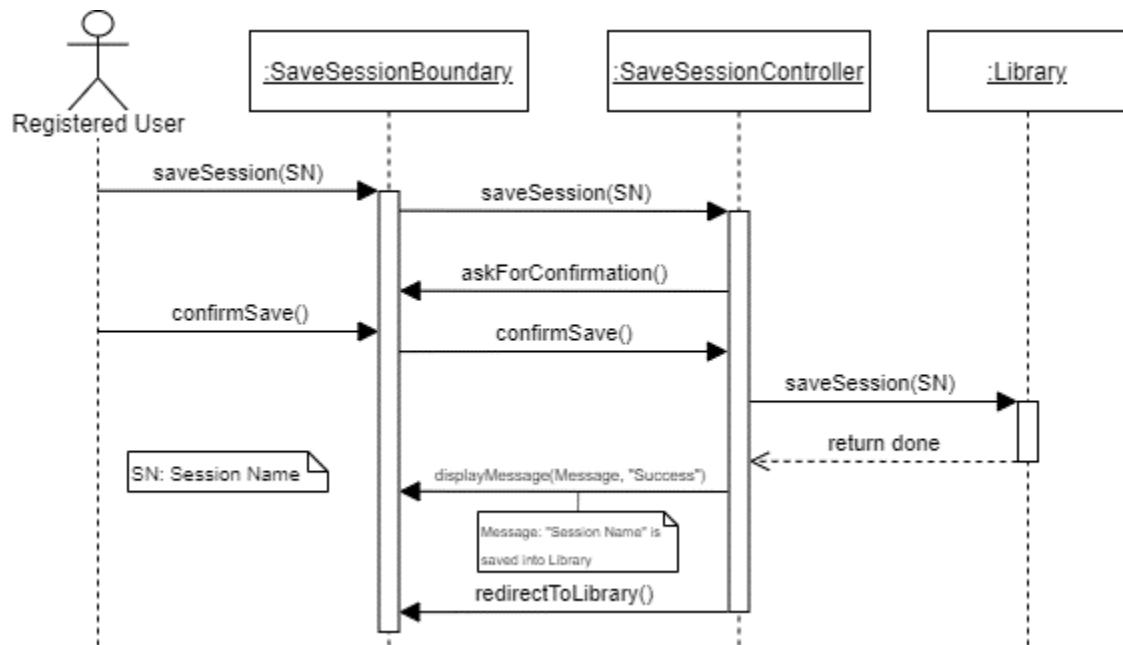


Figure 81: Old Save session sequence diagram

Figure 82 illustrate old Edit session sequence diagram and the updated one shown in section 10.1 figure 9

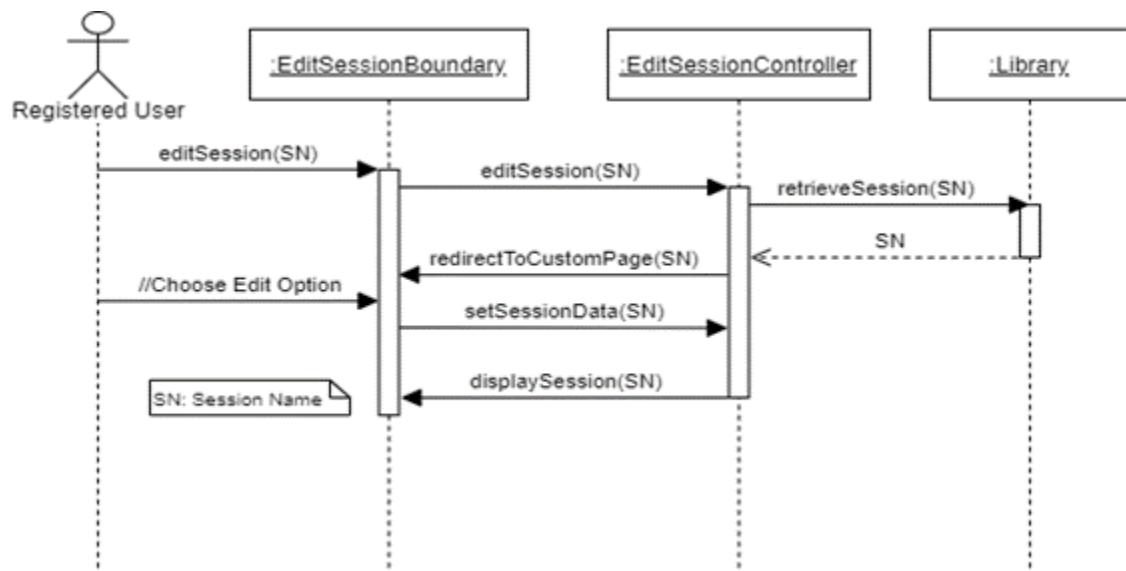


Figure 82: Old Edit session sequence diagram

Figure 83 illustrate old Test session sequence diagram

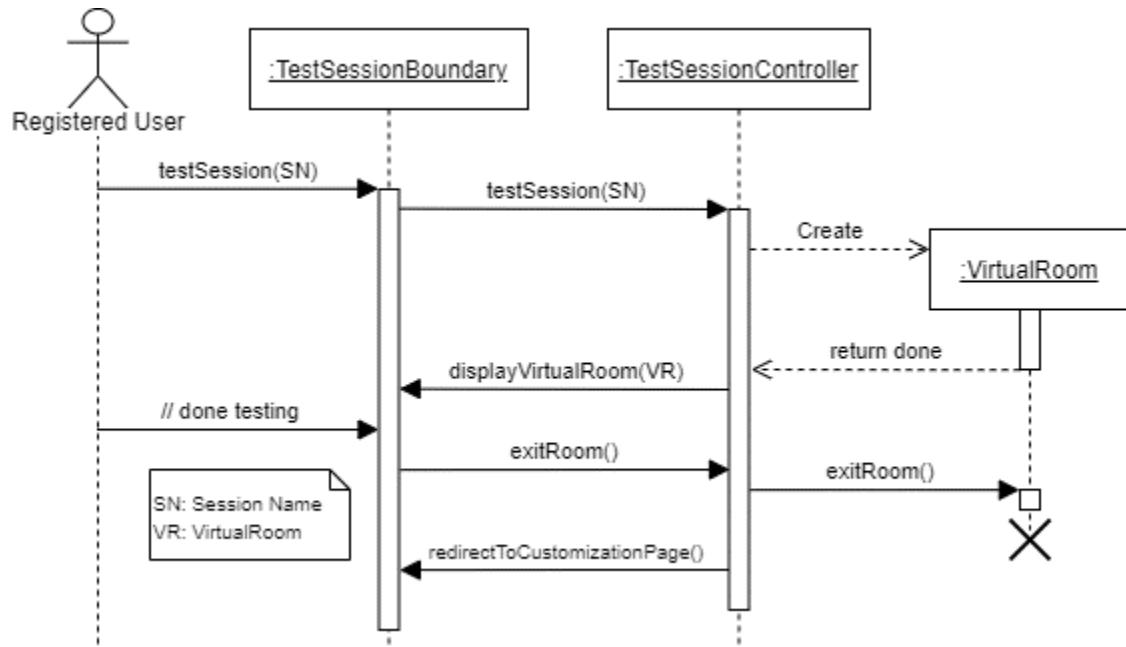


Figure 83: Old Test session sequence diagram

Figure 84 illustrate old Share session sequence diagram

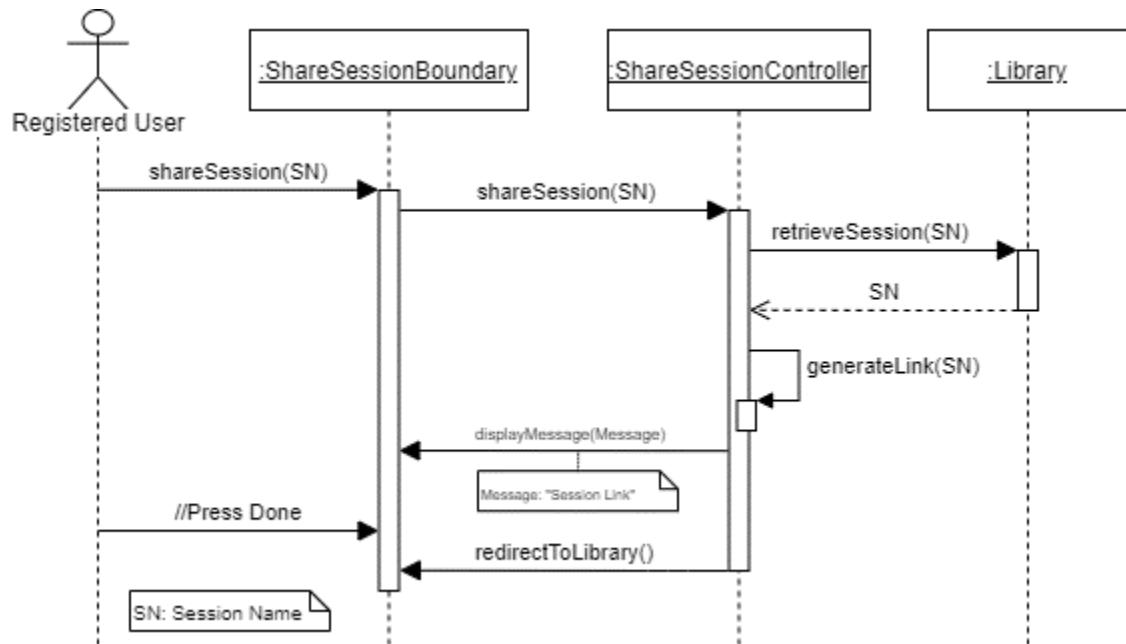


Figure 84: Old Share session sequence diagram

Figure 85 illustrate old Initiate room sequence diagram and the updated one shown in section 10.1 figure 12

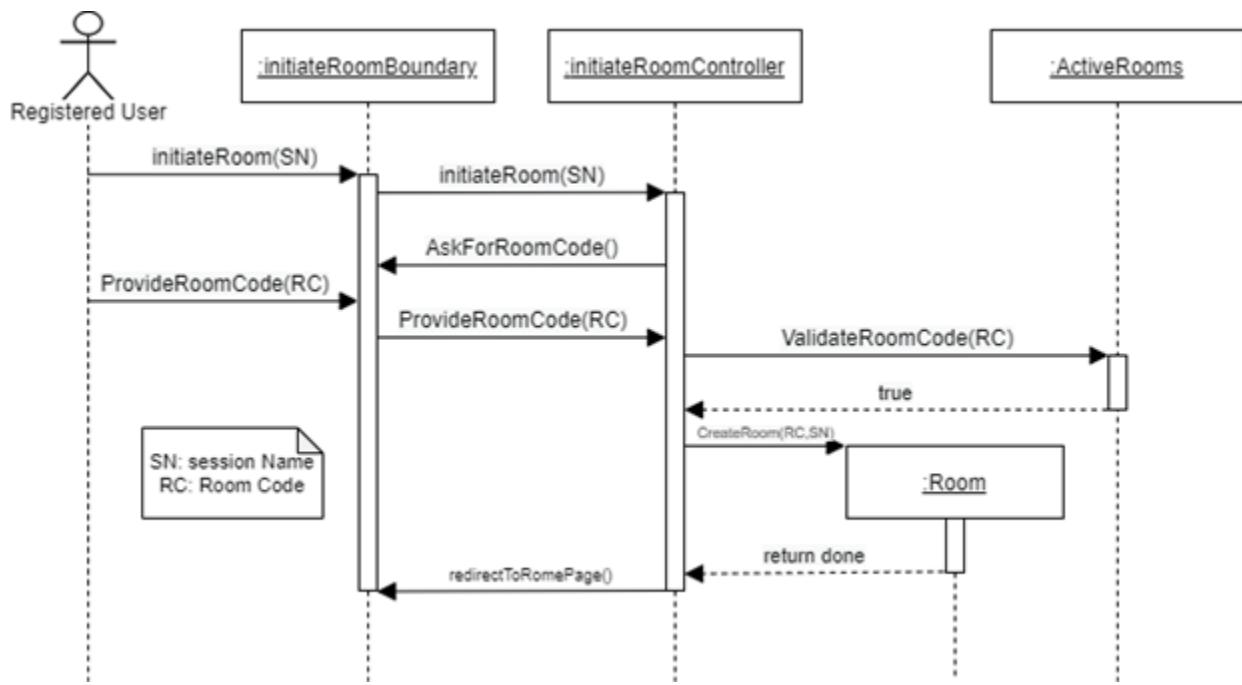


Figure 85: Old Initiate room sequence diagram

Figure 86 illustrate old Design class and the updated one shown in section 11 figure 23

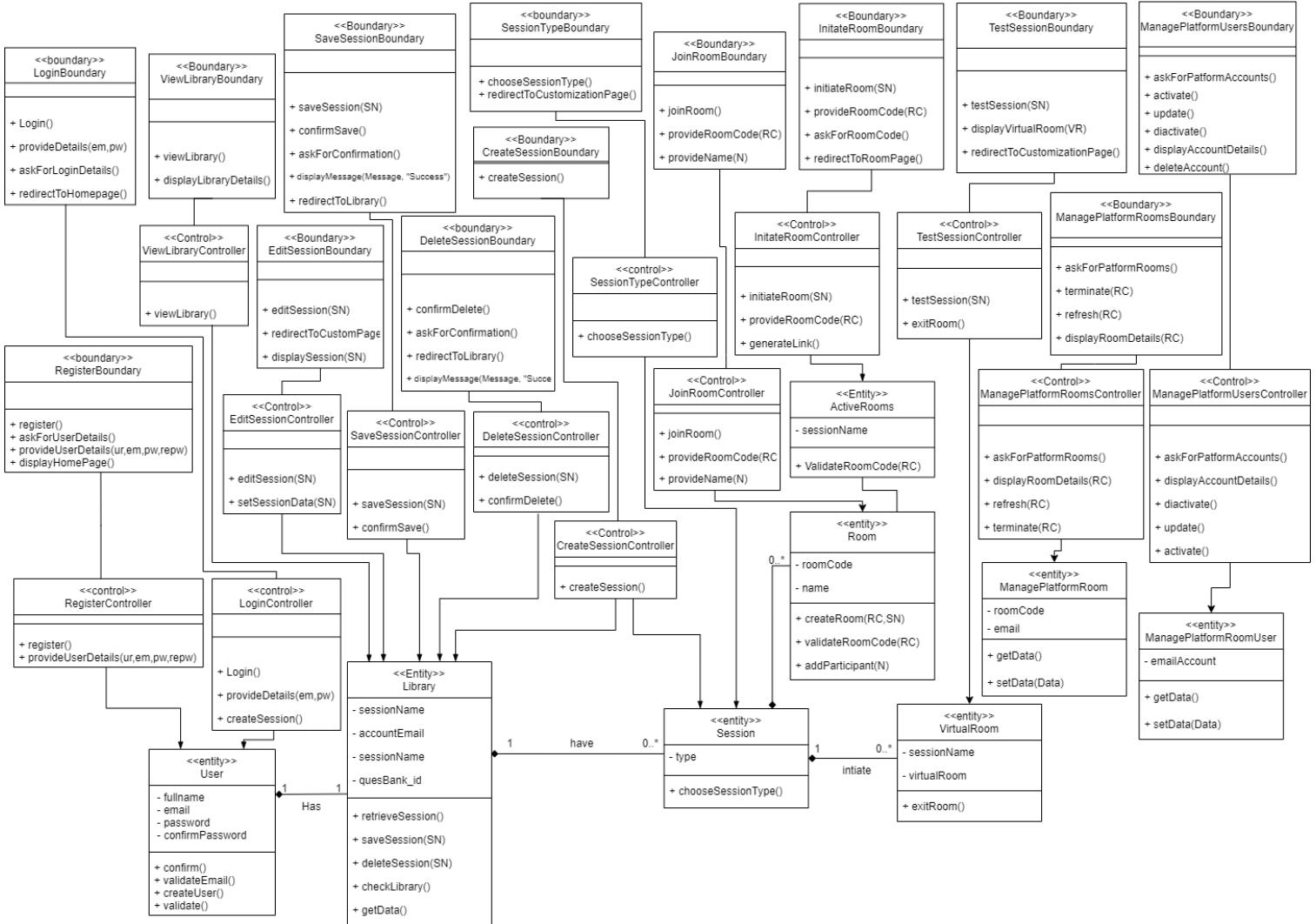


Figure 86: Old Design class

20.7 Appendix G. Algorithm Updates

First, modified algorithm

Figure 87 show old calculation of a score in game-based quiz, and the updated one shown in section 13.2.1 figure 24

```

1 Function GB_TrueAns(answerTime,totalTime)
2     #this function will calculate how much the user will get if the answer is correct
3     #in a game based quiz
4
5     #answerTime is time remaining
6     #when the user answered the question
7
8     #totalTime is the time assigned to the question
9
10    percentage=(answerTime/totalTime) * 50
11    #percentage calculate the percentage between totalTime and answerTime
12
13    score=50 + percentage
14    #score is the variable the will be added to the user final score
15
16    return score
17 end function

```

Figure 87: Old calculation of a score in game-based quiz algorithm

Second, deleted algorithm

The *Figure 88* illustrates the algorithm used to calculate the percentage of each answer in a poll in pseudo code.

```

1 #this function will calculate the precantage of each answer in a poll
2 Function polls(answers,totalAnswers)
3
4     #answers is an array that contains all the question's answers
5     #totalAnswers is number of all users that answered the question
6
7     #this loop will claculate each answer precantage
8     #answers[x].precantage is the answer precantage
9     #answers[x].numOfusers is the number of users that chose this answer
10    For x=0 to numOfAnswers
11        set answers[x].precantage = (answers[x].numOfusers / totalAnswers) *100
12    Endfor
13
14 end function

```

Figure 88: calculate the percentage of each answer in a poll algorithm

20.8 Appendix H. Database Schema Updates

Figure 89 illustrate old Database schema and the updated one shown in section 13.5 figure 38

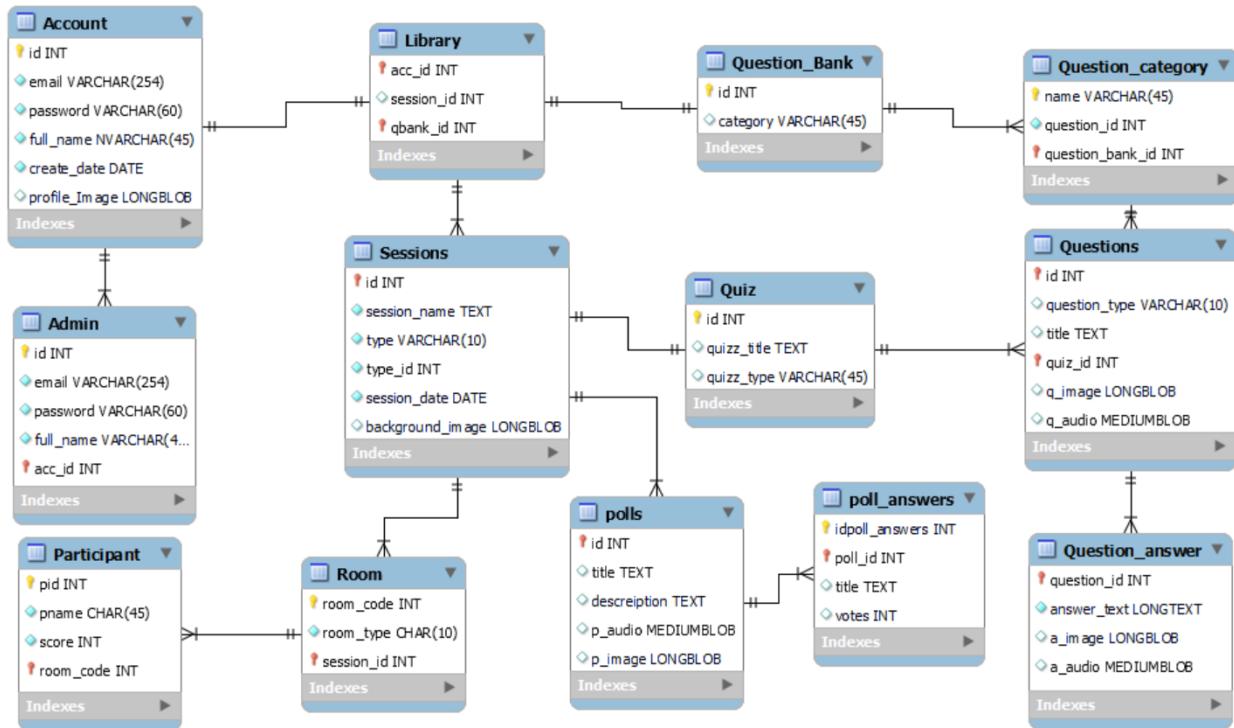


Figure 89: Old Database schema