Fundamentals of Software Engineering
CSE 3310-002
Fall 2019

Group X5

Group Members:

Jared Gatlin

Pravash KC

Sudeep Bhadel

Tommy Philip

Diwakar Parajuli

Boost

1. The app we are going to be making over this project is titled, Boost. This app will
1.1. be a quiz app that will give the user the opportunity to strengthen their knowledge
1.2. specifically, in math. The goal with the app, Boost, is to give everyone
1.3. a fun and rewarding experience as well as teaching them essential math
1.4. skills. The way the app will work is that it will allow the user to start a quiz,
1.5. allowing them to test their skills at whatever level they decide to. For this
1.6. app to be developed efficiently, we have some requirements that we are to meet
1.7. throughout the development. Some of these are: being able to support Android 4
1.8. and higher, keep track of user data, as well as keeping track of individual progress
1.9. throughout each session in the app.

2. Interface Requirements:

2.1. The app will keep simplicity in mind while developing the app.
2.2. The app will be able to adapt to any size screen for the benefit of the user.
2.3. The app will show the users score as they take that quiz.

3. Functions:

3.1. **Login**- a function that allows the user to enter the name and start the quiz.
3.2. **Difficulty Level –** a function that gives user ability to choose the difficulty level.
3.3. **Timer –** a function that gives user certain time frame to complete the quiz.
3.4. **Skip**- function giving the user the ability to skip a question if they feel it is above their skill level.
3.5. **Score -** a function that allows the user to see their score after the quiz.
3.6. **Add Question** - function that gives the user the ability to add their own questions into the pool of questions
3.7. **Exit –** a function that gives user ability to exit from the app.
3.8. **Next –** a function that gives user to move onto next question after answering one.

4. Constraints:

4.1. The app will be developed entirely in Android Studio.

Project Biographies

Jared Gatlin- As a part of this team I am able to bring what experience I have used Java, C, as well as C++. I have participated in some other projects, one of them being a project where we developed a game where we had let our players experience a virtual tour on our space station and play a bunch of different fun games. I feel that I am able to bring new ideas and the ability to always try and keep things moving forward.

Pravash KC- My contribution to my team will be based on my overall experiences, I've had till today. I've had courses with JAVA, C, C++. My recent project was based on using C++ and gtkmm. I and my team had created a banking application in our CSE1325 class. This application helped the user in creating, signing up bank accounts and making transactions. Moreover, I feel I can help my teammates in coming up with innovative ideas and assist them throughout the semester.

Sudeep Bhadel- I have gained experience in programming using Java, C, and C++. I have done projects in the team before, one of my recent projects was that I with my team developed a GUI application using C++ and gtkmm. It was a banking application, which could be used for signing up for online banking and other banking features. I also have made a simple calculator app for android phones using Java to have a basic idea of android app development. I will give my full contribution to solving problems, sharing ideas and work with my teammates in this project.

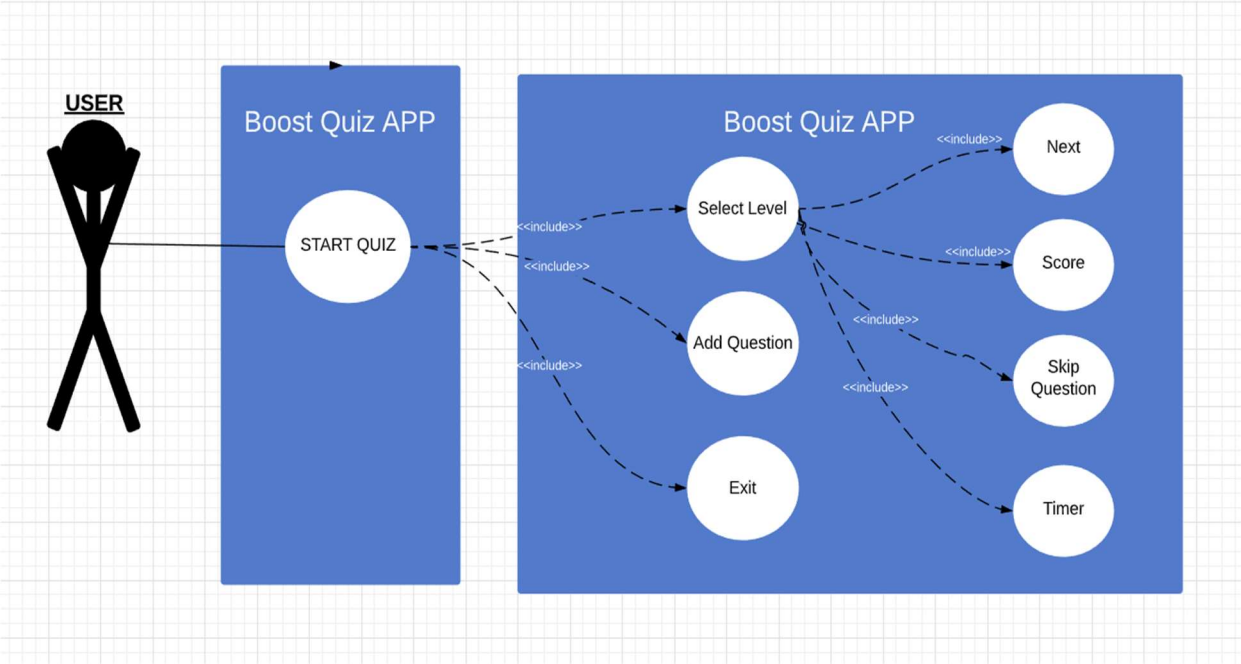
Diwakar Parajuli: The learning process never ends and till today I have learned three popular programming languages till today and they are C, C++, and java. I have worked on some small projects with java and last semester I have worked on C++ to create a GUI project, where our team created a banking application. Besides this, I have some experience in CMS(Content Management System) where I have worked on WordPress and Open cart. I am so excited that I have got a chance to work such wonderful team members this semester and I believe that working together we will achieve our goal.

Tommy Philip: Like my teammates, I will also be able to bring knowledge of the programming languages Java, C, and C++. My most recent project was with C++ in the class CSE 1325 in which I worked on creating a MavMart store application that was heavily revolved around GUI. Along with this, I also have some experience with the UML diagram tool, Umbrello UML Modeller, as we used this for my C++ class as well. I am very excited to finally be able to put our learned knowledge into real-world applications and I believe that with this great team we will be able to achieve anything we set our minds to.

Requirements

#	Functional Requirements
R1	The app shall give the user ability to enter the name. (3.1)
R2	The app shall have different difficulty levels (3.2)
R2.0	The app shall allow the user to go to next question after answering previous one. (3.8)
R2.1	The app shall give the user ability to see their quiz score. (3.5)
R2.2	The app shall give the user ability to skip question. (3.4)
R2.3	The app shall give user certain time frame to complete the quiz. (3.3)
R3	The app shall give the user ability to add their own question (3.6)
R4	The app shall give the user ability to exit the app. (3.7)
	Non-Functional Requirements
R7	The app shall support Android 4.x & higher (1.7)
R8	The app shall store user data to allow the continuous advancement in knowledge (1.8)
R9	The app shall keep track of user progress throughout the app (1.8)
	Interface Requirements
R10	The app shall keep simplicity in mine in regard to the interface (2.1)
R11	The app shall adapt to any size mobile screen (2.2)
R12	The app shall show the users score throughout their time taking the quiz (2.3)
	Constraints
CR1	The app will be developed in Android Studio (4.1)

Use Case Diagram



High-Level Use Cases

- **UC 1: Login to the quiz program**
  - TUCBW the user clicks on “Start Quiz” and enters his/her name/username.
  - TUCEW the user gets logged in and the quiz home screen is displayed.
- **UC 1.1: Select level**
  - TUCBW the user clicks on the difficulty level from the quiz home screen.
  - TUCEW the user clicks on “GO” button and the quiz game page is displayed.
- **UC 1.1.1: Next**
  - TUCBW the user clicks on one of the options from the given question.
  - TUCEW the user clicks on the “NEXT” button from the quiz game page.
- **UC 1.1.2: Score**
  - TUCBW the user clicks on “SHOW RESULT” from the quiz home screen.
  - TUCEW the score details are displayed on the screen and the user selects “NEW MODE” button.
- **UC 1.1.3: Skip Question**
  - TUCBW the user clicks on “NEXT/SKIP” button from the quiz game page.
  - TUCEW question being skipped, and a new question is displayed on the screen.
- **UC 1.1.4: Timer**
  - TUCBW the default time set for that particular level quiz.
  - TUCEW the prompt displaying the quiz is over.
- **UC 1.2: Add Question**
  - TUCBW the user clicks on “Add Question” from the displayed screen.
  - TUCEB user types in the question and answer on a topic and submits it.
- **UC 1.3: Exit**
  - TUCBW the user clicks on “Exit” from the quiz home screen.
  - TUCEW the program getting terminated.

Requirements-Use case Traceability Matrix

	Priority Weight	UC1	UC1.1	UC1.1.1	UC1.1.2	UC1.1.3	UC1.1.4	UC1.2	UC1.3
R1	2	X							
R2	4		X						
R2.0	4			X					
R2.1	4				x				
R2.2	4					x			
R2.3	4						x		
R3	1							x	
R4	3								x
Score		2	4	4	4	4	4	1	3

Legend**:**

Priority	**Weight**
High	4
Low	1

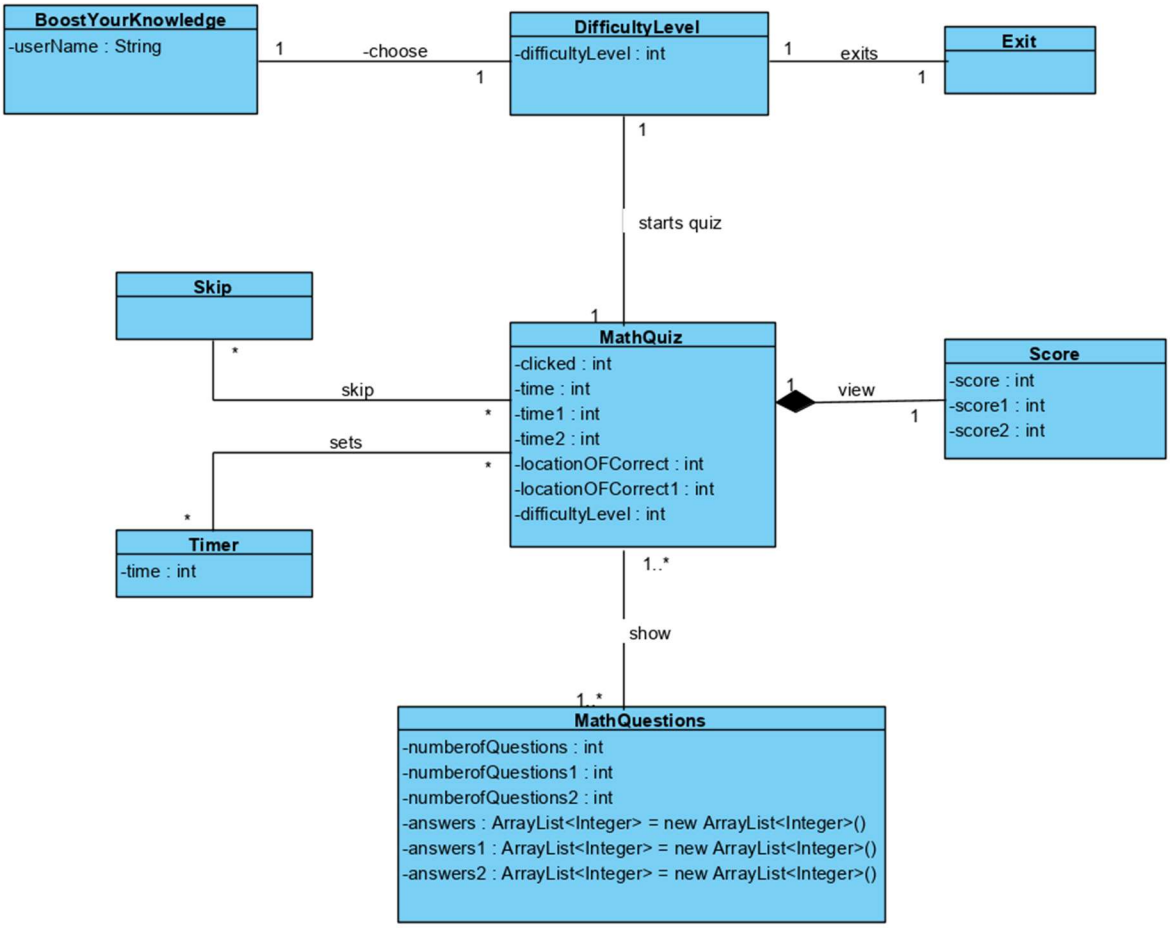
Increment Matrix

Use Case	Priority	Effort (person - week)	Depends On	Iteration 1 (due date) 09/27/2019	Iteration 2 (due date) 10/25/2019	Iteration 3 (due date) 12/06/2019
UC1	2	2	None		1	1
UC1.1	4	4	UC1	2	1	1
UC1.1.1	4	2	UC1.1	-	1	1
UC1.1.2	4	1	UC1.1.1	-	1	-
UC1.1.3	4	2	None	-	2	-
UC1.1.4	4	3	UC1.1	-	2	1
UC1.2	1	3	None	-	-	-
UC1.3	3	1	UC1	-	1	-
Total Effort		18		2	9	4

Legend**:**

Priority	**Weight**
High	4
Low	1

Domain Model Diagram

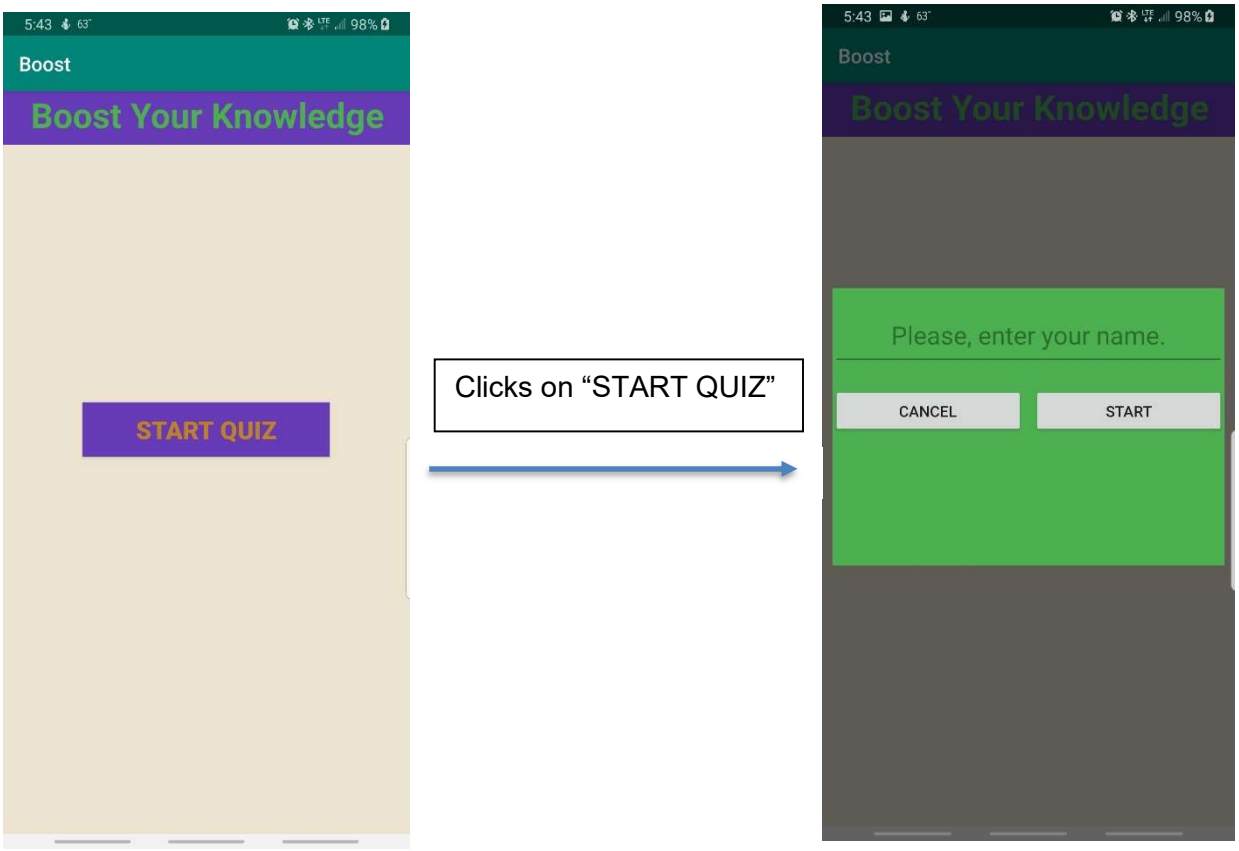


Expanded Use Case

EUC 1:

Precondition: *This use case assumes that the user just opened the app.*	
Actor: User	System: Boost
1. **TUCBW** the user clicks the "Start Quiz" button. 3. The user enters his/her name in the box and clicks "START" button. 5. **TUCEW** the user gets logged in and the quiz home screen is displayed.	0. System displays the app main page. 2. System displays the add player's name box. 4. *System checks if the there is a name entered in the player's name box. If no name is entered, error message is displayed.
Postcondition: *The user is now ready to select the type of quiz to take.*	

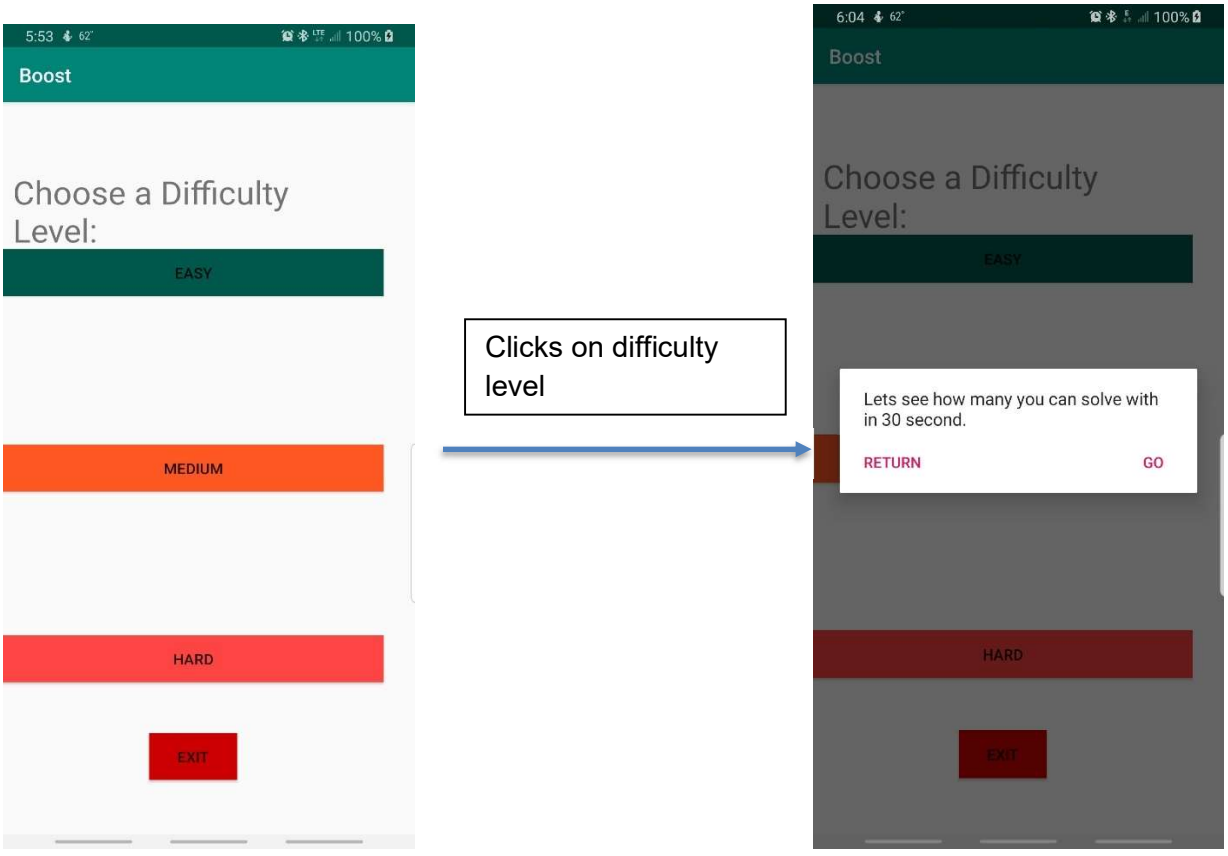
USER INTERFACE PROTOTYPE:



EUC 1.1:

Precondition: *This use case assumes that the quiz home screen is displayed.*	
Actor: User	System: Boost
1. **TUCBW** the user clicks on the difficulty level from the "Quiz Home" screen. 3. **TUCEW** the user clicks on "GO" button and the quiz game page is displayed.	0. System displays the quiz home screen. 2. System displays the amount of time given for the particular level of quiz.
Postcondition: *The program is then ready to start quiz with the given difficulty level*	

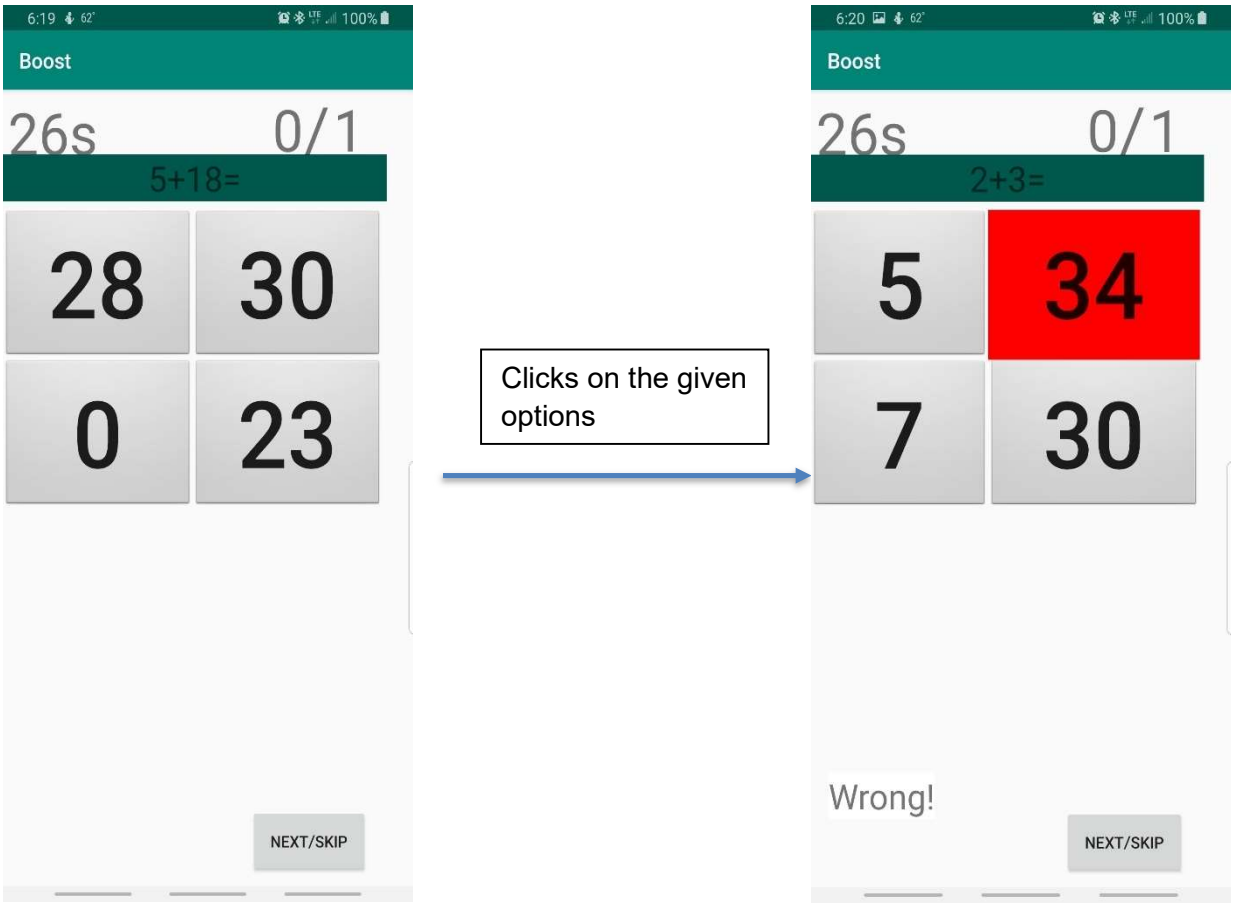
USER INTERFACE PROTOTYPE:



EUC 1.1.1:

Precondition: *This use case assumes that the quiz game page is displayed.*	
Actor: User	System: Boost
1. **TUCBW** the user clicks on one of the options from the given question. 4. **TUCEW** the user clicks on the "NEXT" button from the quiz game page.	0. System displays the quiz game page. 3. *System checks if the answer is correct and displays the result.
Postcondition: *The user is now ready to answer the next question on the quiz game page.*	

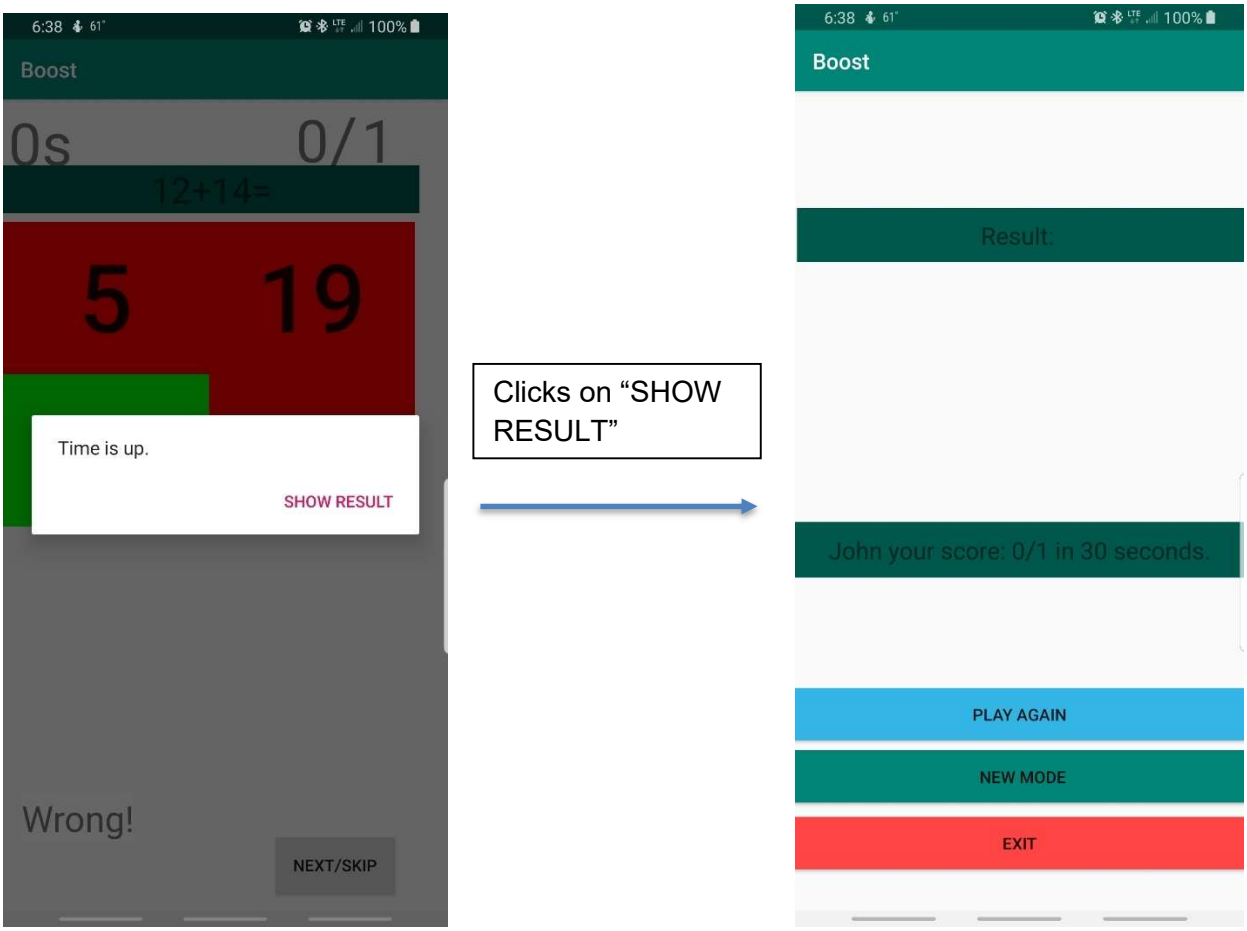
USER INTERFACE PROTOTYPE:



EUC 1.1.2:

Precondition: *This use case assumes that the quiz game has been completed*	
Actor: User	System: Boost
1. **TUCBW** the user clicks on "SHOW RESULT" from the quiz home screen. 3. **TUCEW** the score details displayed on the screen and the user selects "NEW MODE" button.	0. System displays the quiz completion dialog box. 2. *System checks the number of answers done correct in the given time.
Postcondition: *The program is then brought back to the quiz home screen and the user is now ready to take another level of difficulty quiz.*	

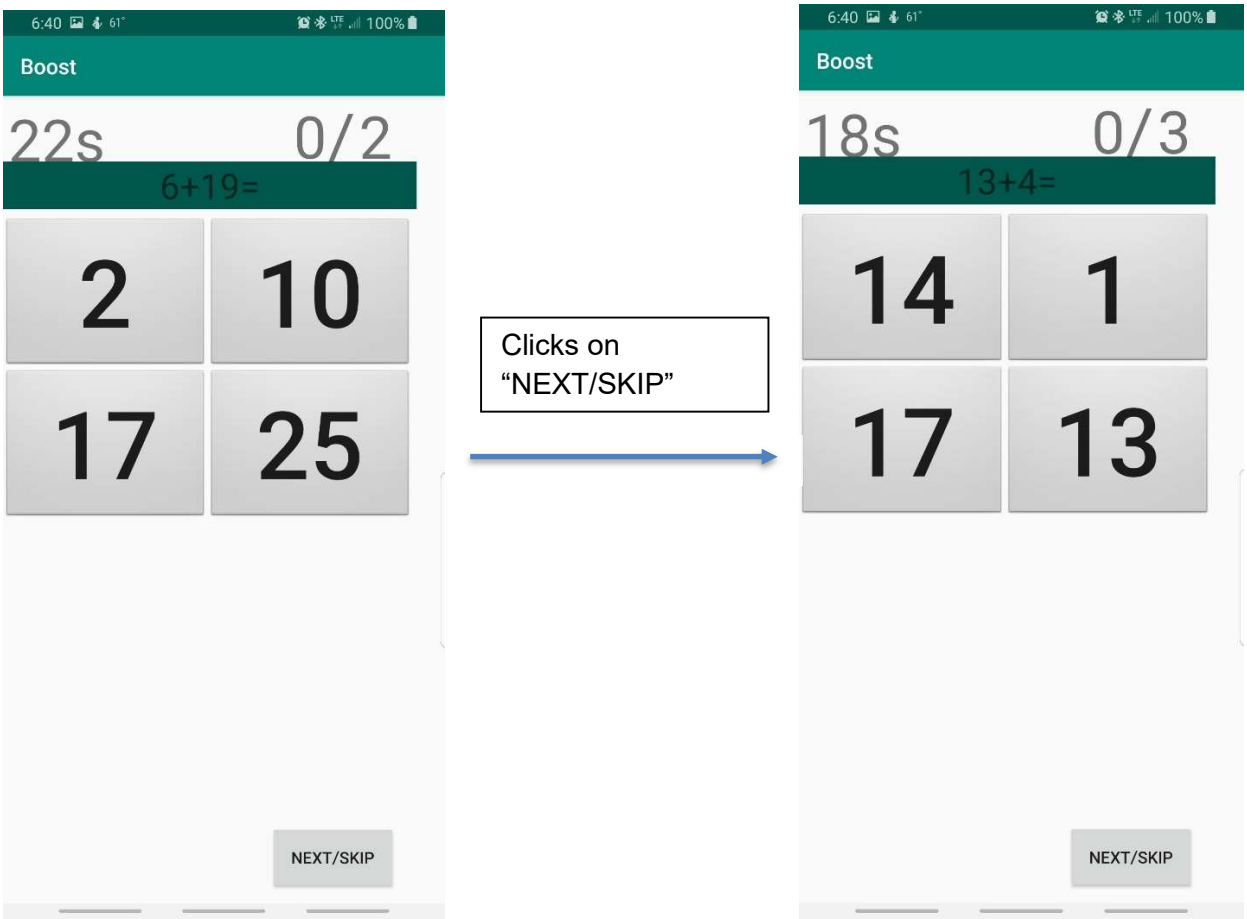
USER INTERFACE PROTOTYPE:



EUC 1.1.3:

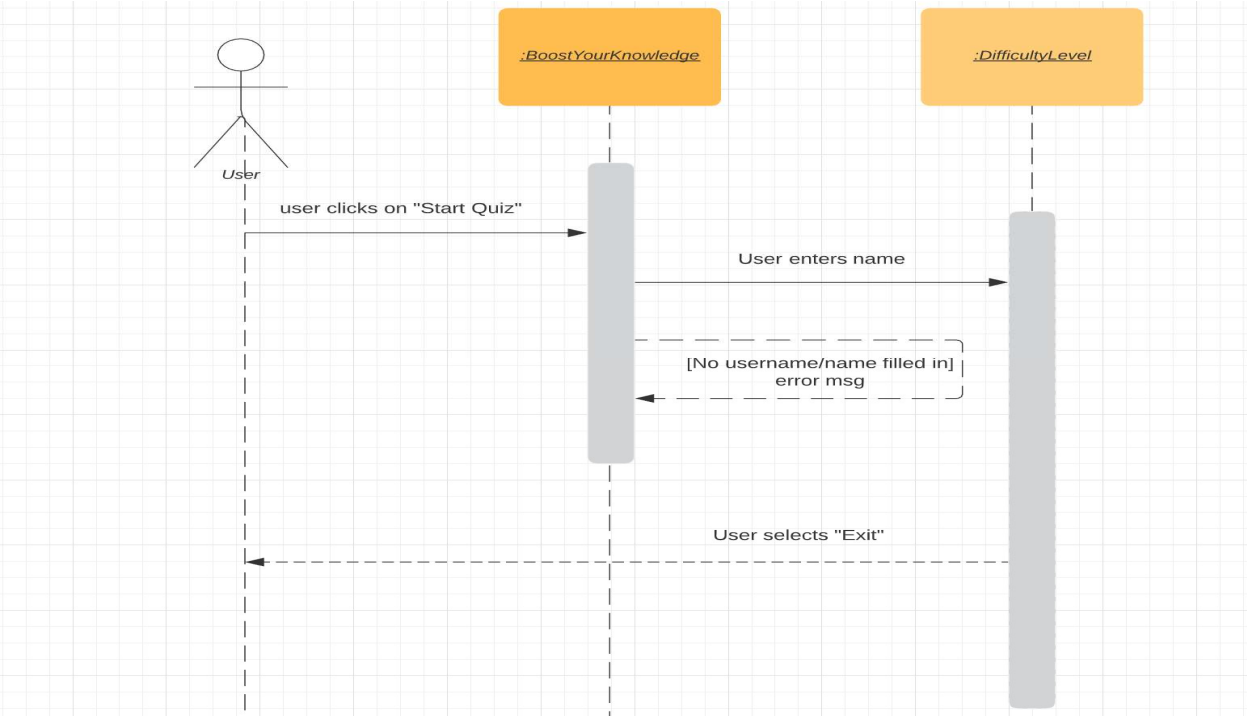
Precondition: *This use case assumes that the quiz game page is being ran*	
Actor: User	System: Boost
1. **TUCBW** the user clicks on "NEXT/SKIP" button from the quiz game page. 3. **TUCEW** the question being skipped, and a new question is displayed on the screen	0. System displays the quiz game page screen. 2. System skips the given question and counts the answer as incorrect.
Postcondition: *The user is now ready to answer the next question on the quiz game page.*	

USER INTERFACE PROTOTYPE:

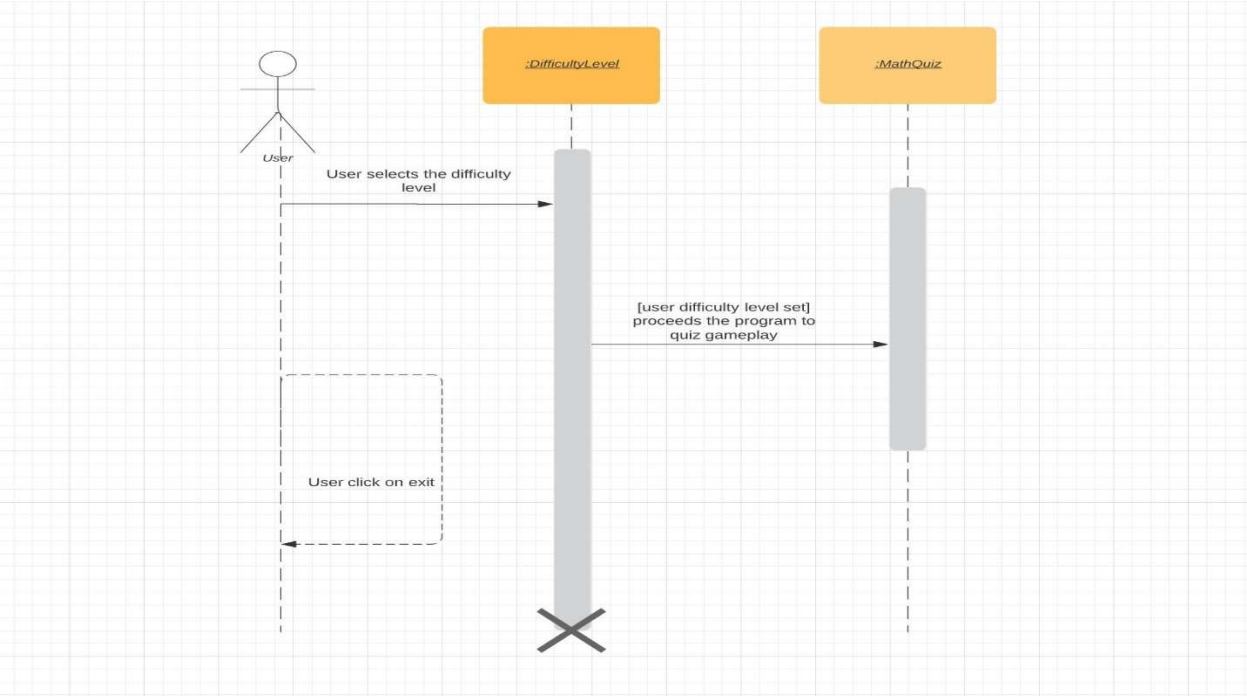


Analysis Sequence Diagrams

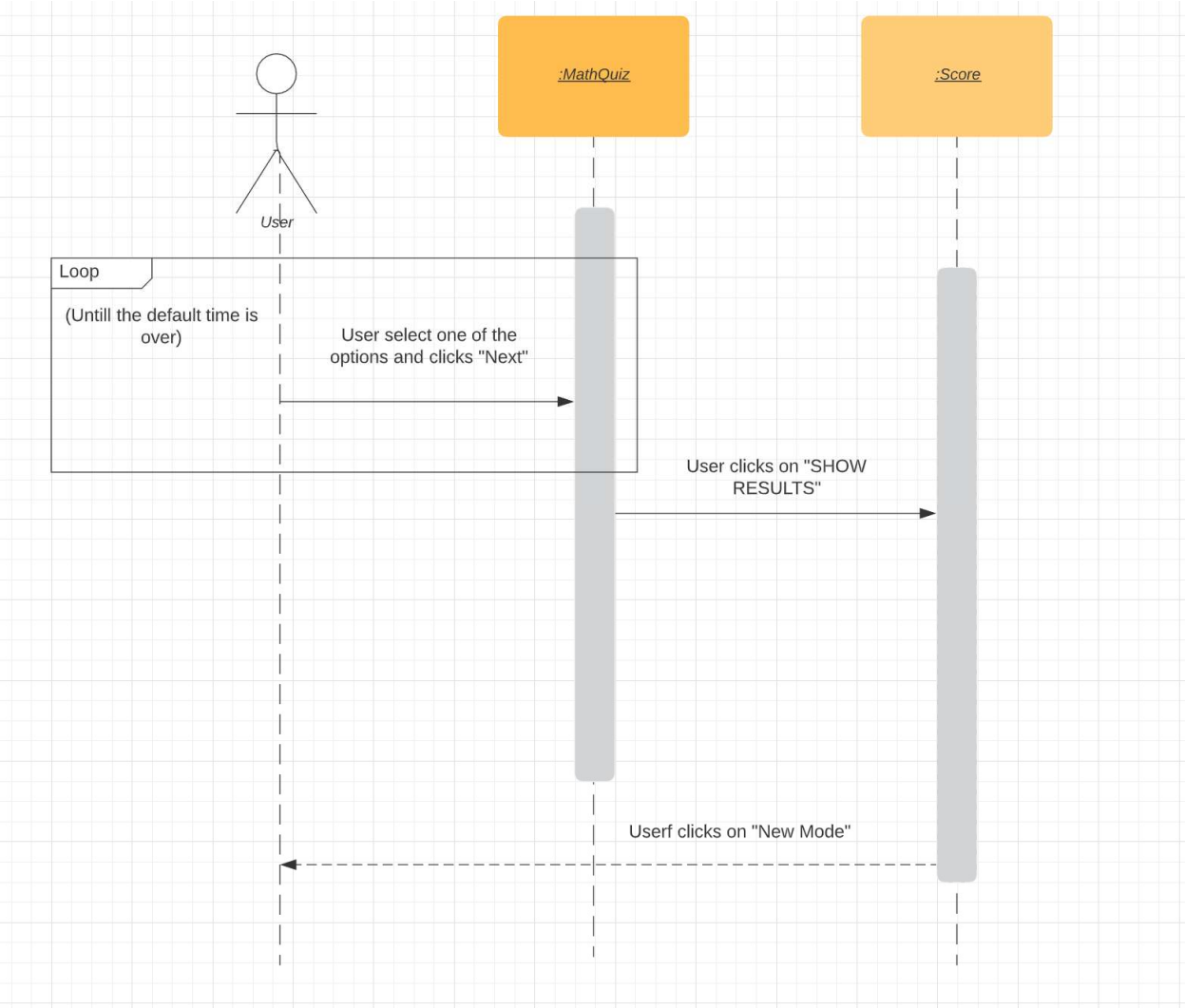
EUC 1:



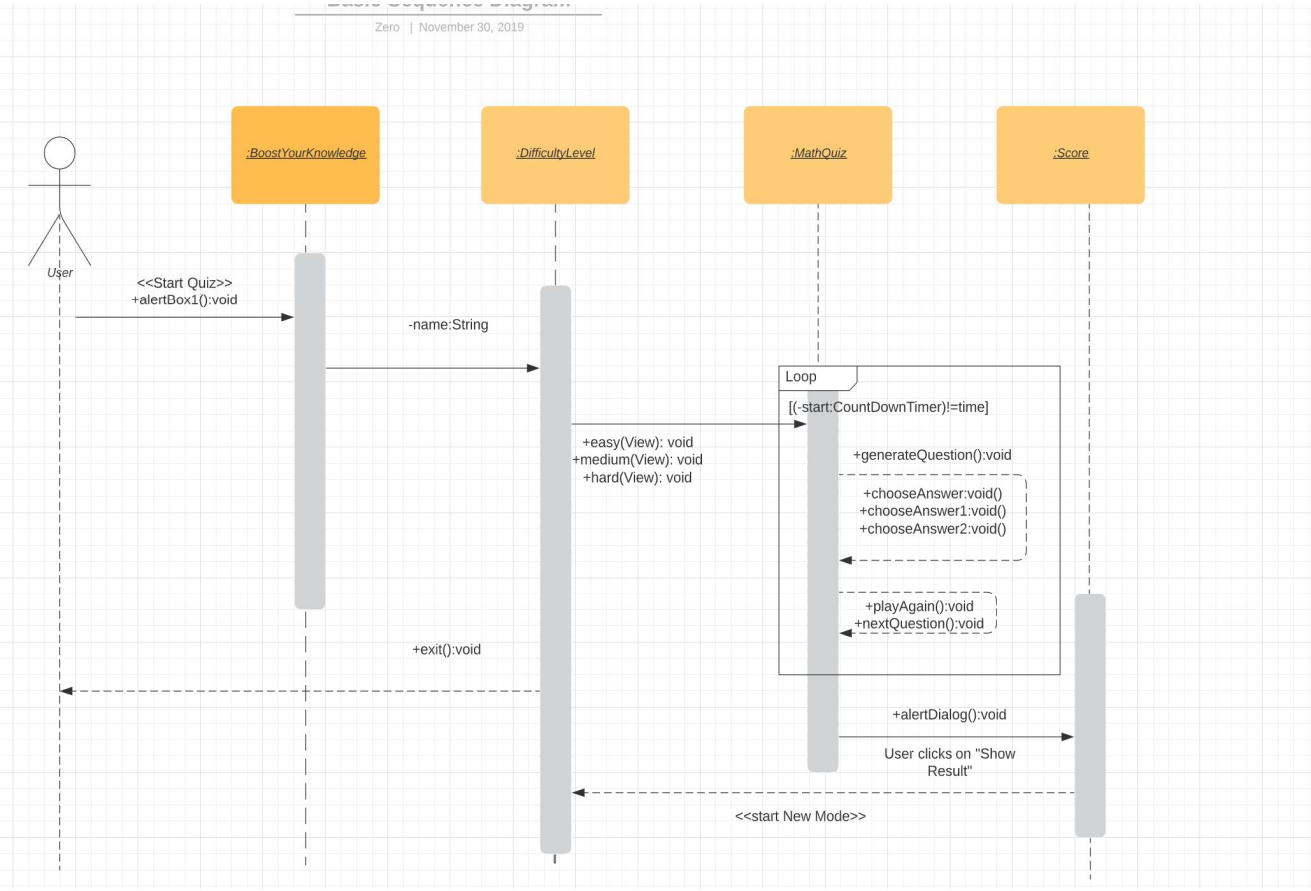
EUC 1.1:



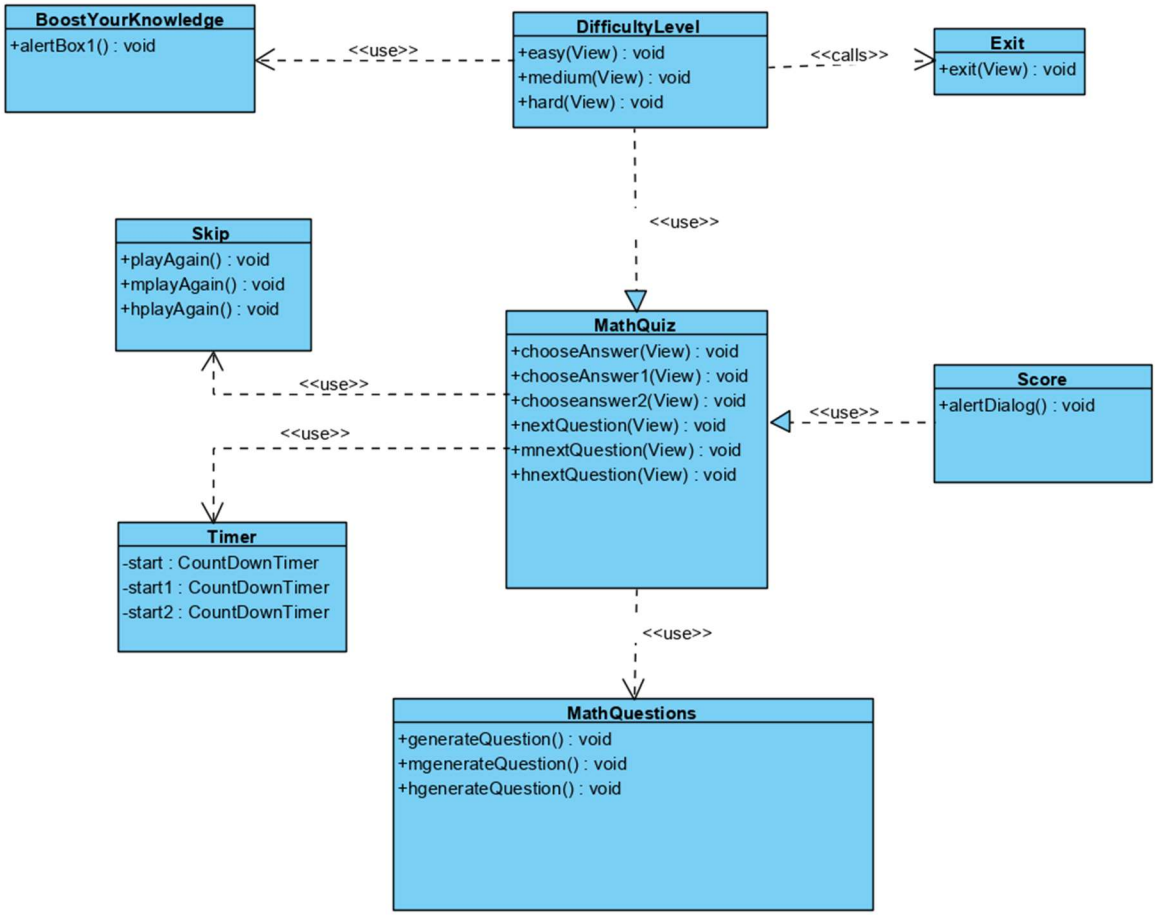
EUC 1.1.2:


:MathQuiz
:Score
User
Loop
(Untill the default time is over)
User select one of the options and clicks "Next"
User clicks on "SHOW RESULTS"
Userf clicks on "New Mode"

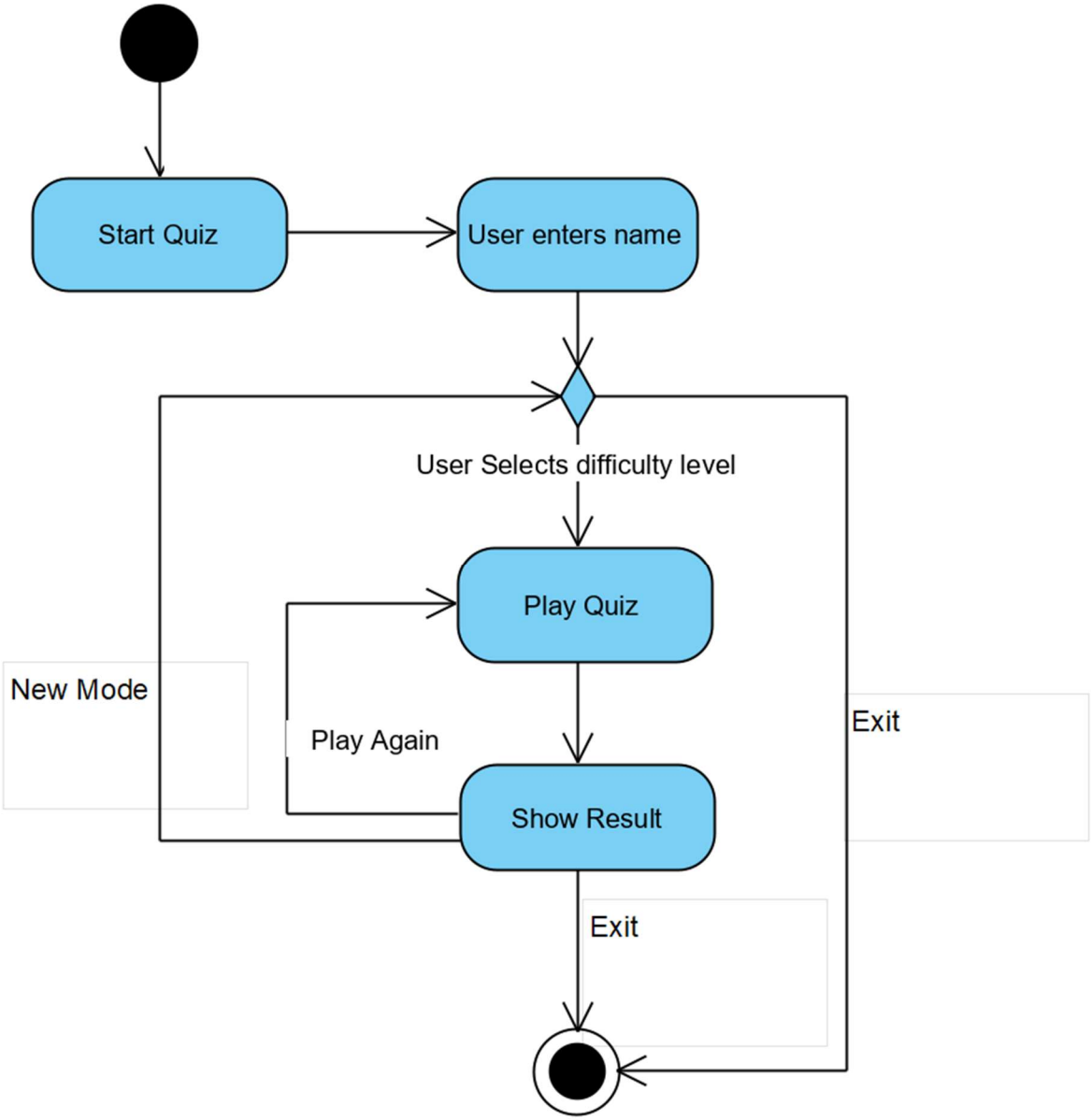
Design Sequence Diagram



Design Class Diagram



Activity Diagram



App Demo

YouTube Link:
https://www.youtube.com/watch?time_continue=23&v=DKafrzUgJsA&feature=emb_logo