Second Prototype Report

of the

Battle of Minds

Product Manager:

Yağız Hakkı Aydın

Product Owner:

Mustafa Arda Biçen

Developers:

Sena Özbelen

Muhammet Furkan Yıldız

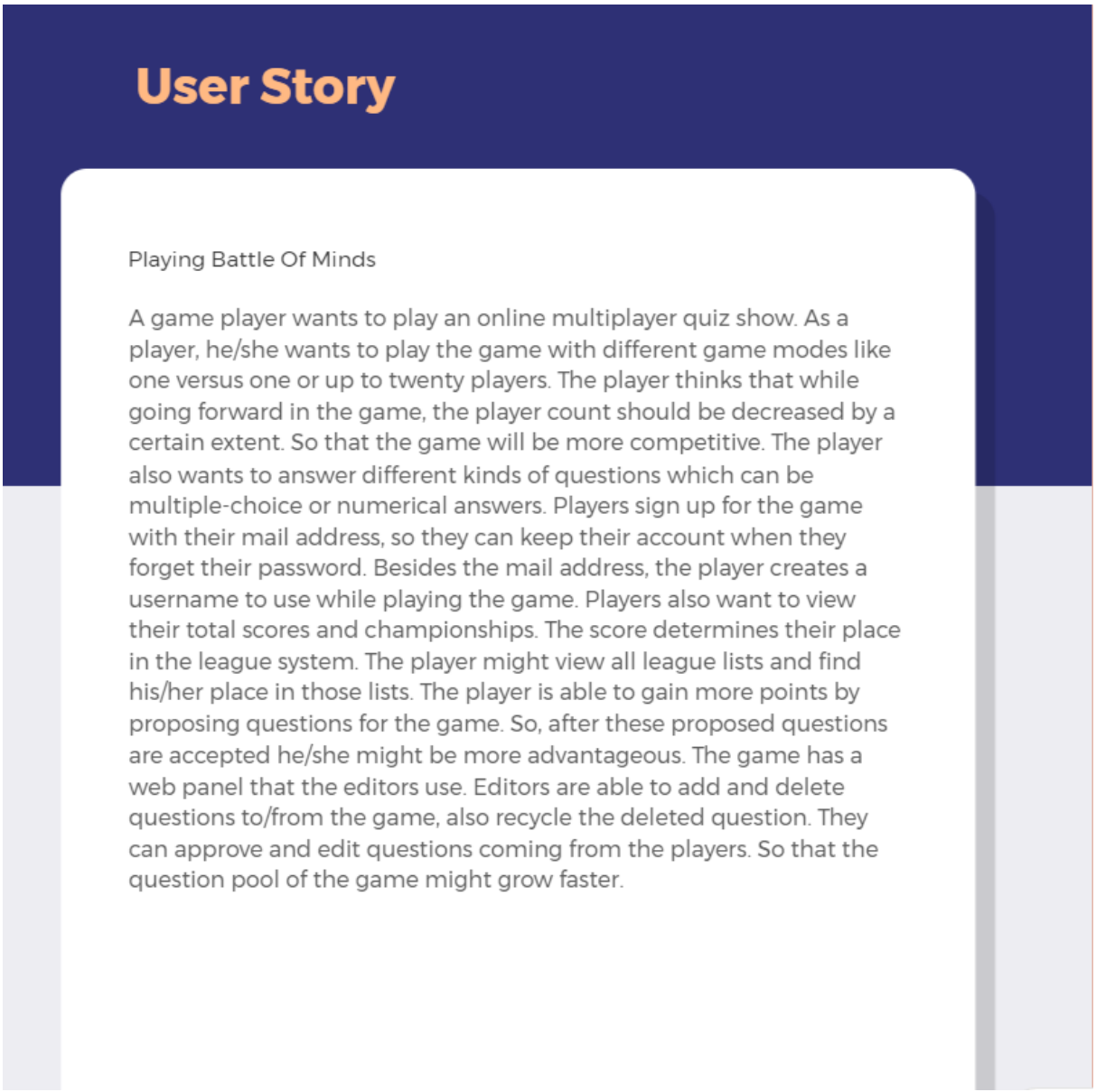
Erva Aksu

Berry Hibiscus team

03/12/2022

Table of Contents

1.User Story and Related Scenarios



User Scenarios

Signing-up for Battle Of Minds

Initial Assumption :
A player or group of players have the latest version of the game Battle of Minds downloaded on their mobile phones. They have a qualified internet connection.

Normal :
The player enters a user name, a mail address and creates a password from the sign-up screen. The system automatically sends an email to the player's mail address. This email contains a validation code. The player enters the code into the specific field on the screen.

What Can Go Wrong :
The mail address or password entered is not valid. Players should be informed about possible errors.
The validation code is not correct. Players should be informed about the error and directed to the sign-up screen again.

Other Activities :
Multiple players may sign-up for the game at the same time.
An editor may be added to the system manually by the admin. Since editors do not sign-up for the system but are defined on the system for operating from the web panel.

System State On Completion :
The player is signed-up. The player is approved and added to the system.

Logging-in to Battle Of Minds for Players

Initial Assumption :
A player or group of players have been signed up for the game Battle of Minds successfully. They have a qualified internet connection.

Normal :
The player enters his/her mail address and password to the log-in screen. The player presses the log-in button.

What Can Go Wrong :
The mail address or password is not correct. Players should be informed about possible errors.
The player forgets his/her password. The player should be directed to another screen, an email should be sent to the player's mail address. This mail contains a new password for the player to log in to the system with it.

Other Activities :
Multiple players may log in to the system at the same time.

System State On Completion :
The player is logged in successfully. A new screen that demonstrates the player panel is displayed.

Logging-in to Battle Of Minds for Editors

Initial Assumption :
An editor or group of editors have been defined in the system. They have a qualified internet connection.

Normal :
The editor enters his/her mail address and password to the log-in screen on the web panel. The editor presses the log-in button.

What Can Go Wrong :
The mail address or password is not correct. Editors should be informed about possible errors.
The editor forgets his/her password. The editor should be directed to another screen, an email should be sent to the editor's mail address. This mail contains a new password for the editor to log in to the system with it.

Other Activities :
Multiple editors may log in to the system at the same time.

System State On Completion :
The editor is logged in successfully. A new screen that demonstrates the editor panel is displayed on the web panel.

Going into a game in Battle Of Minds

Initial Assumption :
A player or group of players have been logged-on to the game. They have a qualified internet connection.

Normal :
The player presses the button to start the game. A new screen that demonstrates game modes is displayed. The player chooses the game mode which determines the player count on the game. A loading screen is displayed until the game starts.

What Can Go Wrong :
The Player's internet connection has been interrupted. Player should be informed about the error and try to connect to the system again.

Other Activities :
Other players may go into the game at the same time. There might be a waiting time process until the player capacity of the game is filled.

System State On Completion :
The player has entered the game with the specific game mode which has been chosen.

Playing the game in Battle Of Minds

Initial Assumption :
A player or group of players have been entered into a game. They have a qualified internet connection.

Normal :
Questions are displayed on the player's screen with a counter which shows the time that remains. There are two types of questions. Multiple-choice questions allow players to choose an answer between multiple options. Classical questions allow players to enter a specific number for answering. All questions have a limit of players to accept for the next question. After answering the current question if the player's answer is correct and the player count for the next question is not out of the limit, then the player gains a specific point and passes to the next question. Otherwise, the game is over for the player. Eventually player wins the game if he/she is the one left in the game and gains extra points.

What Can Go Wrong :
The Player's internet connection has been interrupted. Players should be informed about the error. The game should be ended for the specific player.
The player should be informed of a possible delay due to the connection.

Other Activities :
Other players also play the game at the same time. There might be a slight delay while playing for different players.

System State On Completion :
The player plays the game according to the game mode successfully.

Adding Questions to Battle Of Minds for Players

Initial Assumption :
A player or group of players has been logged-on to the system. They have a qualified internet connection.

Normal :
The player chooses the add question option from the menu. The player enters the question details such as the type of the question, options, and answer. The player sends the proposed question. An on-screen message is generated to the player that the question is uploaded to the system.

What Can Go Wrong :
The Player's internet connection has been interrupted. Players should be informed about the error.
Some fields about the question are not filled. Players should be informed about the error.

Other Activities :
The editor may be logged-on the system and may approve and edit the questions as they are uploaded.

System State On Completion :
Questions that are proposed by the player are added to the question pool with the validation of the editor.

Adding Questions to Battle Of Minds for Editors

Initial Assumption :
An editor or group of editors have been logged-on to the system. They have a qualified internet connection.

Normal :
The editor chooses the add question option from the menu on the web panel. The editor enters the question details such as the type of the question, options, and answer. The editor sends the proposed question. An on-screen message is generated to the editor that the question is added to the system.

What Can Go Wrong :
The editor's internet connection has been interrupted. The editor should be informed about the error.
Some fields about the question are not filled. The editor should be informed about the error.

Other Activities :
Other editors may add questions to the system or approve questions at the same time.
The game could be played at the same time and the question added may be encountered afterward.

System State On Completion :
Questions that are written by the editor are directly added to the question pool.

Deleting Questions from Battle Of Minds

Initial Assumption :
An editor or group of editors have been logged-on to the system. They have a qualified internet connection.

Normal :
The editor chooses the delete question option from the menu on the web panel. The editor displays all the questions on the system. The editor chooses a question to delete and an on-screen message is generated to the editor that the question is deleted from the system.

What Can Go Wrong :
The editor's internet connection has been interrupted. The editor should be informed about the error.

Other Activities :
Other editors may delete questions from the system at the same time.
The game could be played at the same time and the question deleted is not encountered afterward.

System State On Completion :
Questions that are deleted by the editor are directly deleted from the question pool.

Recycling the Deleted Questions

Initial Assumption :
An editor or group of editors have been logged-on to the system. They have a qualified internet connection.

Normal :
The editor chooses the recycle bin option from the menu on the web panel. The editor displays all the deleted questions on the system. The editor chooses a question to recycle and an on-screen message is generated to the editor that the question is added to the system again.

What Can Go Wrong :
The editor's internet connection has been interrupted. The editor should be informed about the error.

Other Activities :
Other editors may recycle questions at the same time. The game could be played at the same time and the question recycled may be encountered afterward.

System State On Completion :
Questions that are recycled by the editor are added to the question pool again.

Updating the Password in Battle Of Minds

Initial Assumption :
An editor, player, or group of editors, or players have been logged-on to the system. They have a qualified internet connection.

Normal :
The user chooses the update password option from the menu. The user enters his/her current password and a new password. An on-screen message is generated to the user that his/her password is updated.

What Can Go Wrong :
The user's current password is wrong. The user should be informed about the error.
The user's new password is not valid. The user should be informed about the error.

Other Activities :
Multiple users may update their password at the same time.

System State On Completion :
The user's password is updated. The user is able to log-in to the system using the updated password.

Displaying Scores, Championships and League in Battle Of Minds

Initial Assumption :
A player or group of players has been logged on to the system. They have a qualified internet connection.

Normal :
The player chooses the display scores, championships, and league option from the menu. The scores section displays the total score that the player gain thus far. The Championships section displays how many times the player ranked first, second, or third. The league displays the player's current league.

What Can Go Wrong :
The Player's internet connection has been interrupted. Players should be informed about the error.

Other Activities :
Multiple players may display their scores, championships, and league at the same time.

System State On Completion :
The score, championships, and league of the corresponding player are displayed.

Displaying League Lists in Battle Of Minds

Initial Assumption :
A player or group of players has been logged on to the system. They have a qualified internet connection.

Normal :
The player chooses the display league lists option from the menu. The player chooses different kinds of league types to view the players in this league list.

What Can Go Wrong :
The Player's internet connection has been interrupted. Players should be informed about the error.

Other Activities :
Multiple players may display league lists at the same time.

System State On Completion :
The league lists are displayed to the player.

2.Project Backlog Documents

2.1 Tasks for the web side

ID	Task	Priority	Estimated Efford
W1	login screen designing and switching to main menu screen	High	5
W2	designing main menu screen and adding functionality to main menu	High	9
W3	designing question reject / edit and approve screen	Medium	8
W4	adding functionality to reject / edit and approve screen	Medium	9
W5	designing edit existing questions screen and adding functionallity	Medium	7
W6	designing add questions screen and adding functionallity	Medium	9
W7	designing recycle screen and adding functionallity	Medium	8

2.2 Tasks for the server side

ID	Task	Priority	Estimated Efford
S1	basic database designing	High	4
S2	mvc creating	High	2
S3	web api creating	High	2
S4	web api main methods creating and designing	High	5
S5	adding sign in and sign up functionallity	Medium	8
S6	database designing for competetions	High	10
S7	connection for interaction during game	High	10
S8	applying league system	Low	5
S9	designing user question add method for web side and interacting with web editor	Medium	4
S10	designing user question add method for game side and interacting with web editor	Low	4
S11	designing edit and delete questions methods	Low	4
S12	designing recycle methods for deleted questions	Low	5

2.3 Tasks for the game side

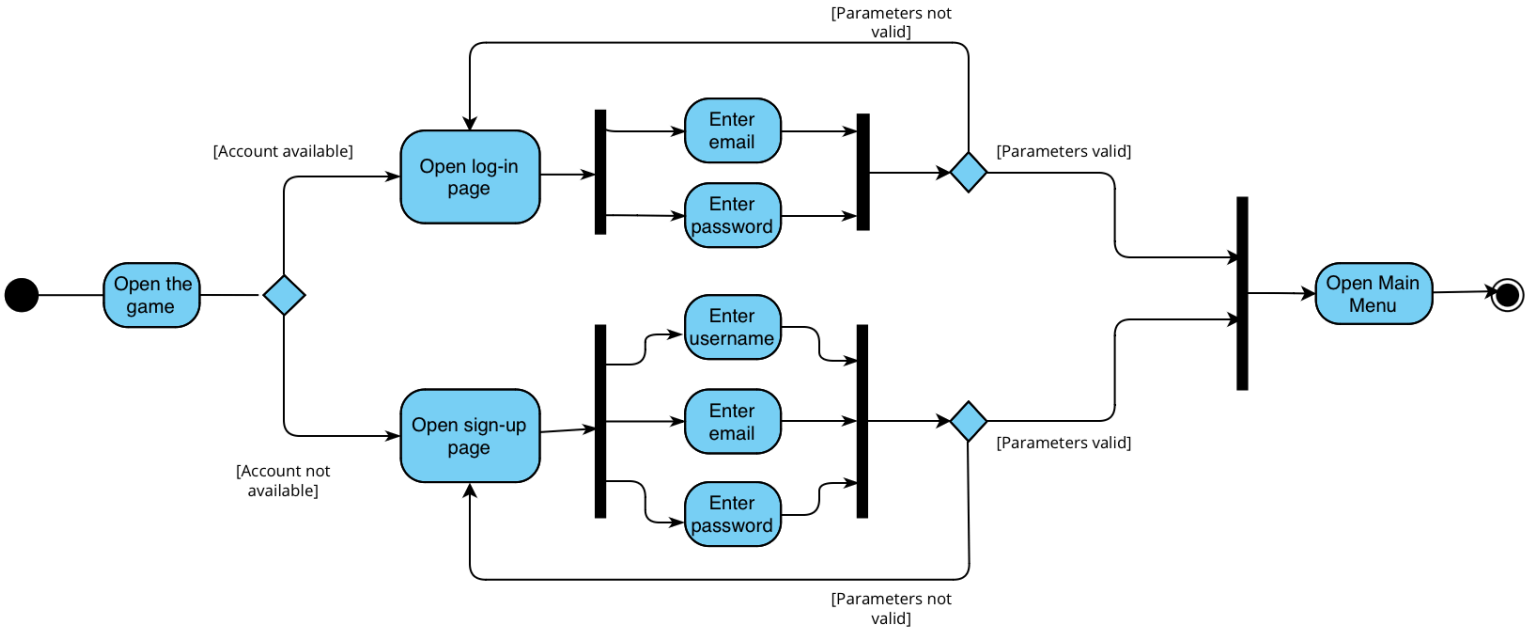
ID	Task	Priority	Estimated Efford
G1	Designing template code/class structure and connecting to template UI	High	3
G2	Designing template user interface with only UI elements	High	3
G3	Designing dummy interface with sci-fi theme including main menu,multiple choice question screen and classical type question screen	Medium	6
G4	Designing profile screen and league screen	Medium	5
G5	Designing game modes menu and load screen	Medium	5
G6	Designing settings menu , question add menu and preferences menu	Low	5
G7	Coding backend for settings menu,question add menu and preferences menu	Medium	8
G8	Coding backend of profile and league screen.	Low	4
G9	Coding backend for playing the competetion and game modes menu	Medium	5
G10	Coding backend for main menu and for the main game manager script	High	9
G11	Deciding and applying language support method	Low	2
G12	Adding full language support	Low	6

3.UML Diagrams

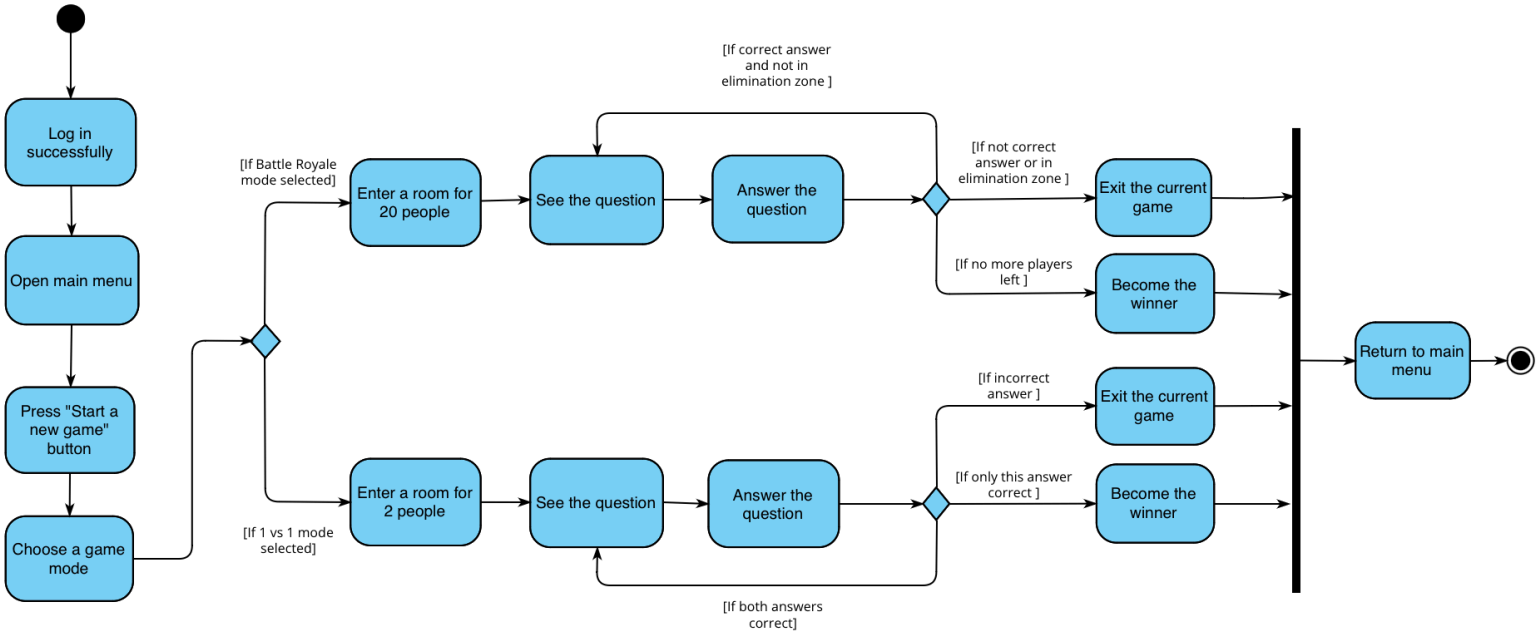
3.1 Context Model

3.2 Process Models

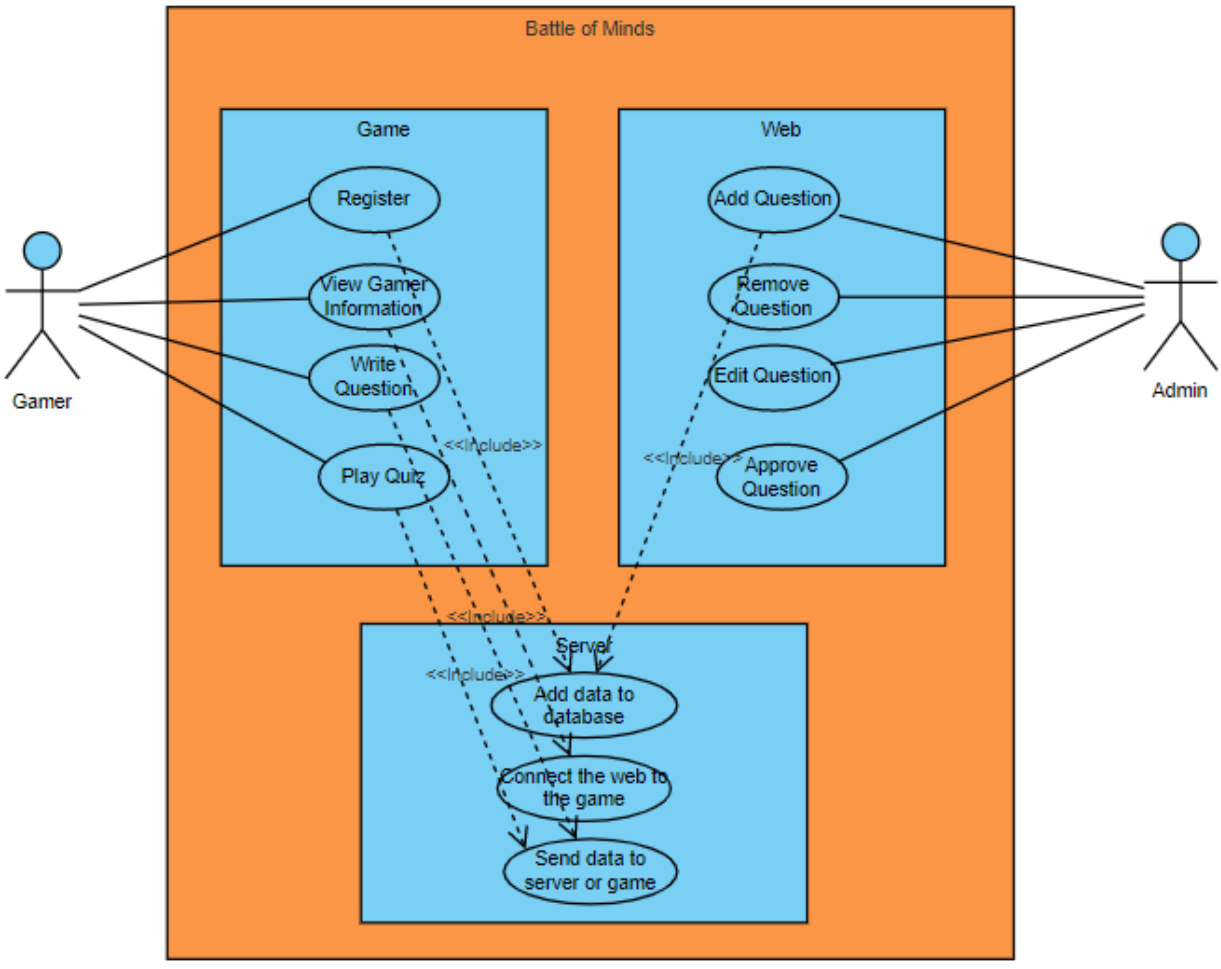
Process Model 1



Process Model 2



3.3 Use Case Diagram and Its Descriptions



Stakeholder List

Gamer:

- must be registered to play the game
- can see their own information.
- can write a new question.
- can play quiz game.

Admin:

- can add new questions.
- can edit questions.
- can remove questions
- can approve questions written by players.

System	*Quiz Game App*
Use Case	*Register to log in to the quiz game*
Actors	*Gamers*
Data	*Players need to register in order to log in to the game. In order to register, they must enter their email address and password. These data are transmitted to the server.*
Stimulus	*If the entered email address and password are valid, the received data is sent to the server and processed into the database.*
Response	*If the email address and password have been added to the database without encountering any problems, the user writes the message that you have successfully registered.*
Comments	*Users who logged in incorrectly should review their data and try to register again.After the registration has been completed, the user can log in by going to the login page and using the application.*

System	*Quiz Game App*
Use Case	*Play quiz game*
Actors	*Gamers*
Data	*The user marks the option that he thinks is correct.*
Stimulus	*It is checked whether the answer to the question drawn from the database is the same as the user's answer.*
Response	*If the answer is correct, the new question is passed.If the answer is wrong, the game ends.*
Comments	*The player has a certain amount of time to be able to answer each question.If he fails to answer the question within this time, he loses the game.*

System	*Quiz Game App*
Use Case	*Write question*
Actors	*Gamers*
Data	*The player writes the question and cavaps he wants to add.*
Stimulus	*It is checked whether the answer to the question drawn from the database is the same as the user's answer.*
Response	*If the answer is correct, the new question is passed.If the answer is wrong, the game ends.*
Comments	*The player has a certain amount of time to be able to answer each question.If he/she fails to answer the question within this time, he loses the game.*

System	*Quiz Game App*
Use Case	*View player information*
Actors	*Gamers*
Data	*User enters her profile to see her own information*
Stimulus	*The player's data is pulled from the database via the server.*
Response	*The profile page with the player's information opens.*
Comments	

System	*Admin Panel Web Page*
Use Case	*Add question*
Actors	*Admins*
Data	*There are two types of questions in the game.Admins should write a question and an answer for classic questions. For multiple choice questions, one question and four stylish and correct answers should be written.*
Stimulus	*After the question is written, a connection is established between the web and the server and the written question is forwarded to the server. The question is added to the database.*
Response	*Admin receives a alert that the problem has been added.*
Comments	*Only admins have permission to write questions from the web panel.Players cannot write questions from the admin panel.*

System	*Admin Panel Web Page*
Use Case	*Remove Question*
Actors	*Admins*
Data	*The admins choose the question they want to remove from the list of questions they have written themselves.*
Stimulus	*The selected question is forwarded to the server and removed from the database.*
Response	*The selected question is removed from the web interface.*
Comments	

System	*Admin Panel Web Page*
Use Case	*Edit Question*
Actors	*Admins*
Data	*Admins select the question they want to edit from the list of questions they have written.*
Stimulus	*The same panel opens as the panel where the questions are written.The selected question is written in it.The corrected version is added instead of the question in the database.*
Response	*The alert is received that your question editing process has been successfully performed.*
Comments	

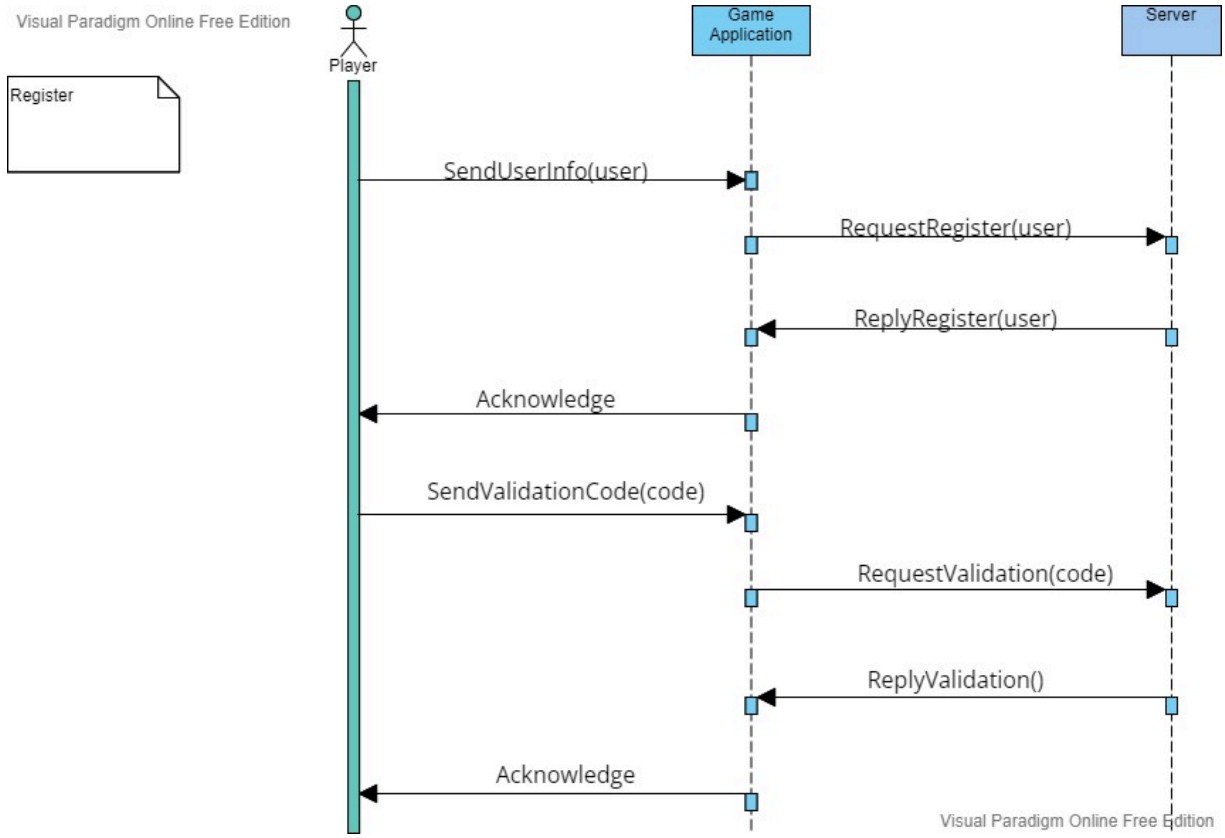
System	*Admin Panel Web Page*
Use Case	*Approve Question*
Actors	*Admins*
Data	*The questions written by the players come to the question confirmation panel. Admins approve or disapprove of the question.*
Stimulus	*If the question is approved, it establishes a web connection with the server and the approved question is added to the database.*
Response	*The alert is received that your question approving process has been successfully performed.*
Comments	*If user questions are not approved, they are permanently deleted and will not be displayed in recycle.*

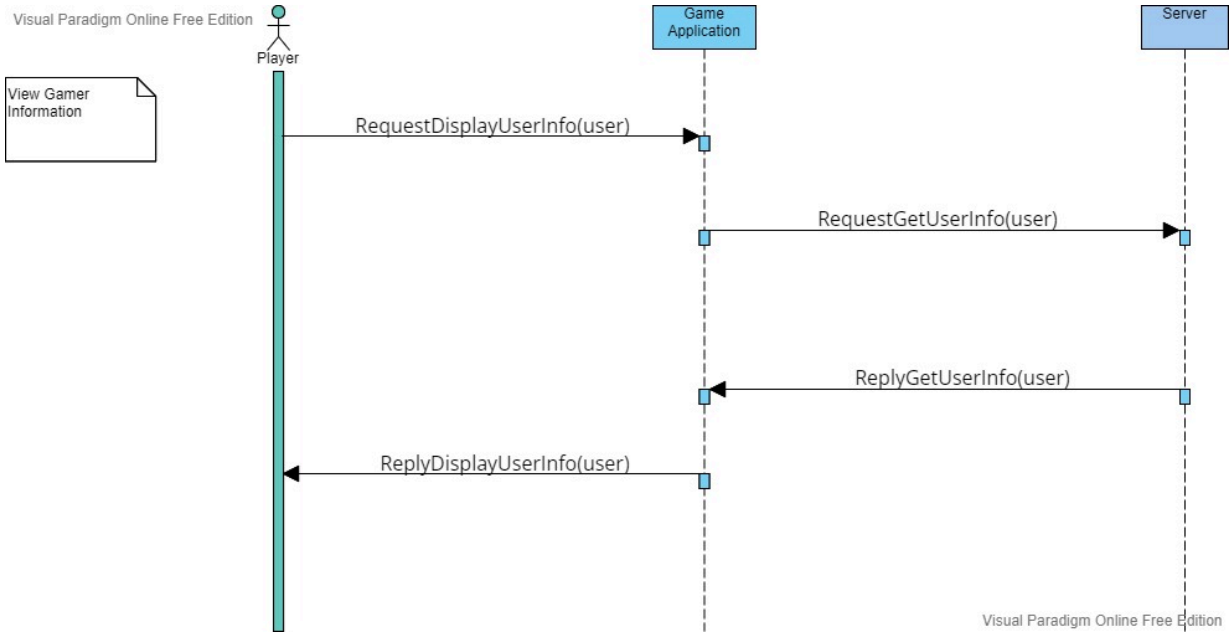
System	*Server*
Use Case	*Send Data to Server or Game*
Actors	-
Data	*Information called from game or web.*
Stimulus	*The requested data is transferred from the database to the game or the web.*
Response	*The data is seen by the administrators on the web or by the players in the game.*
Comments	*If the requested data is not found in the database, it sends data that it is not found.*

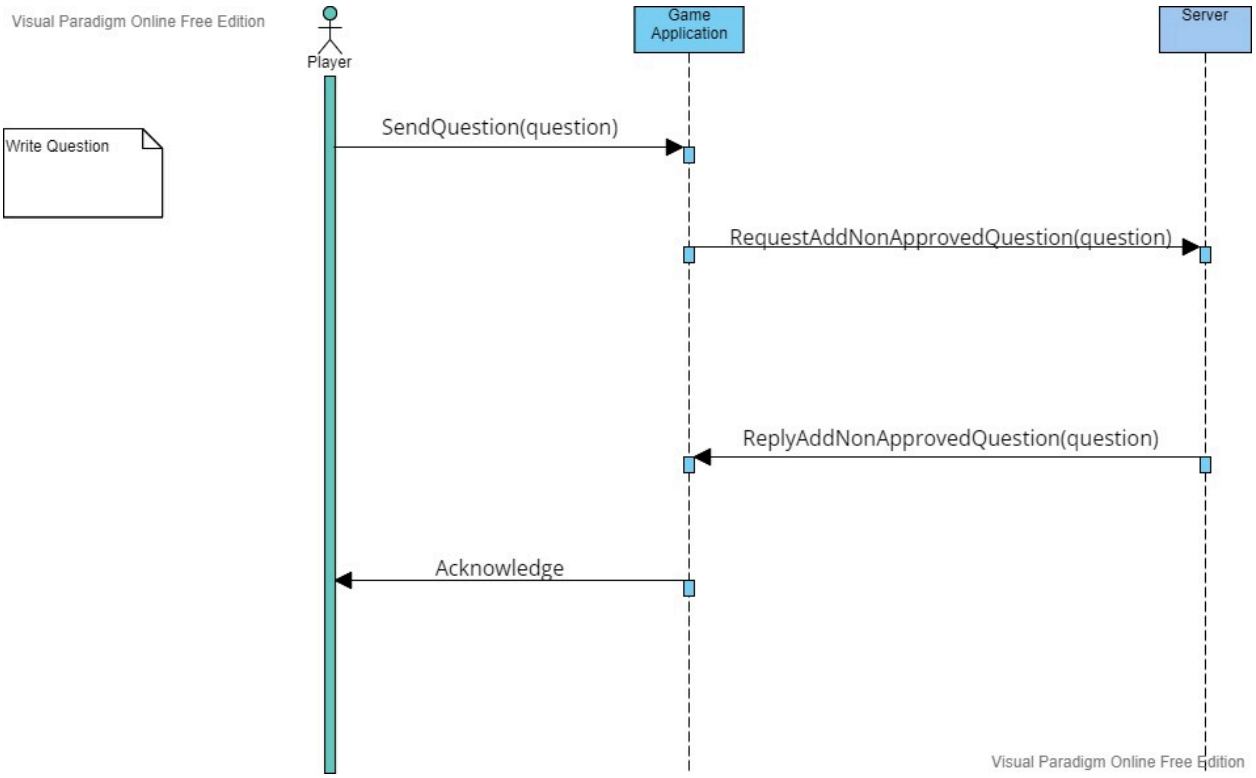
System	*Server*
Use Case	*Connect to web to the game*
Actors	-
Data	*Questions written by the user or admins*
Stimulus	*It transfers the questions written by the players to the web or the questions written by the admins to the game..*
Response	
Comments	

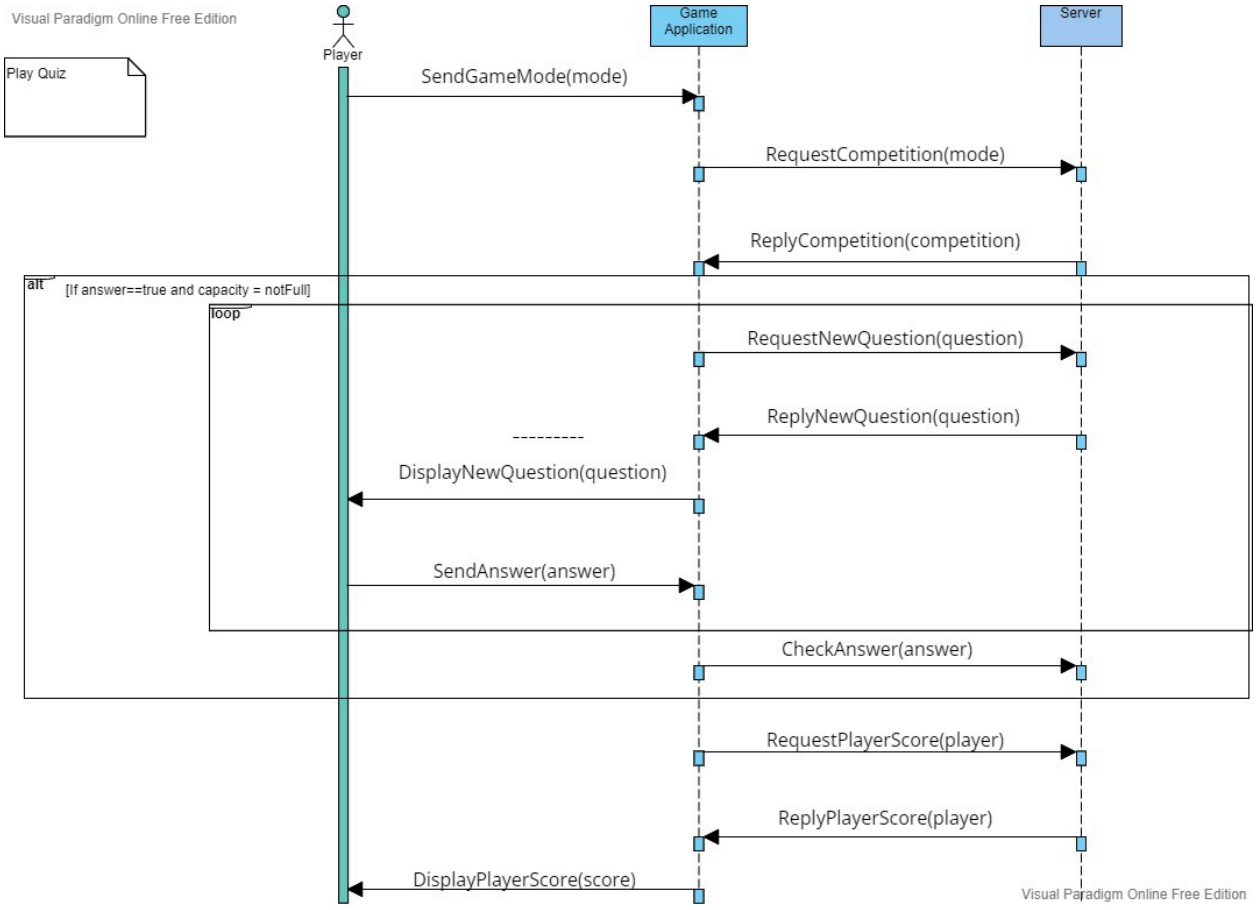
System	*Server*
Use Case	*Add data to database*
Actors	-
Data	*Posts from the web are questions, answers, deletions, corrections.*
Stimulus	*Data sent to the web is added to the database*
Response	*The data added to the database can be called by the web and the game.*
Comments	*If the data is to be deleted, it is not completely removed from the database.*

3.4 Sequence Diagrams for Major State Changes or Activities Defined in 3.3

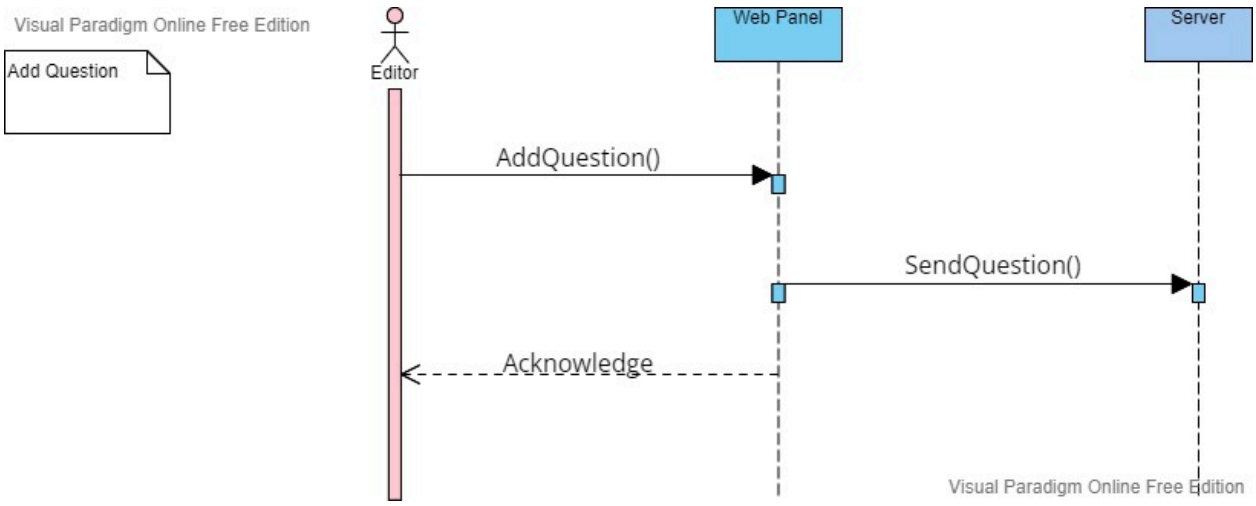


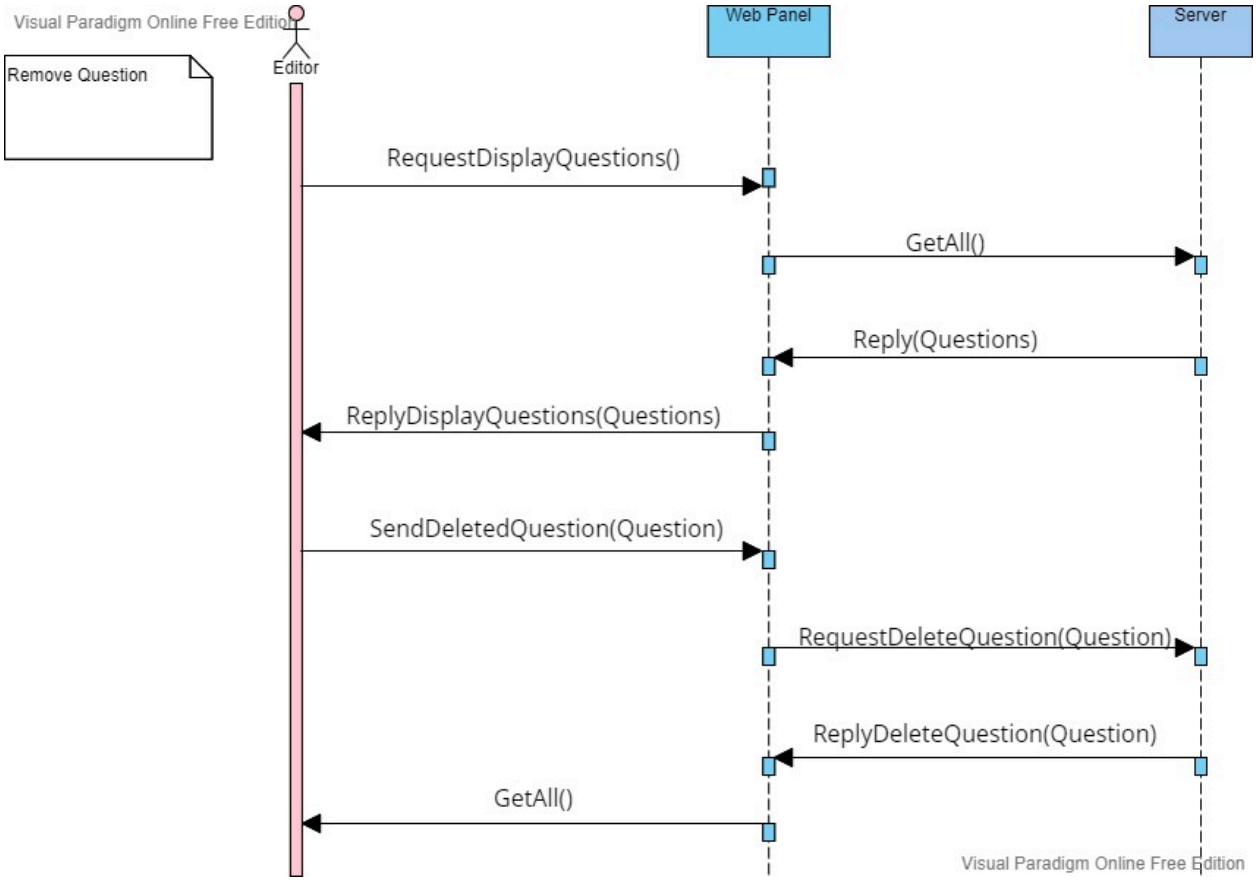


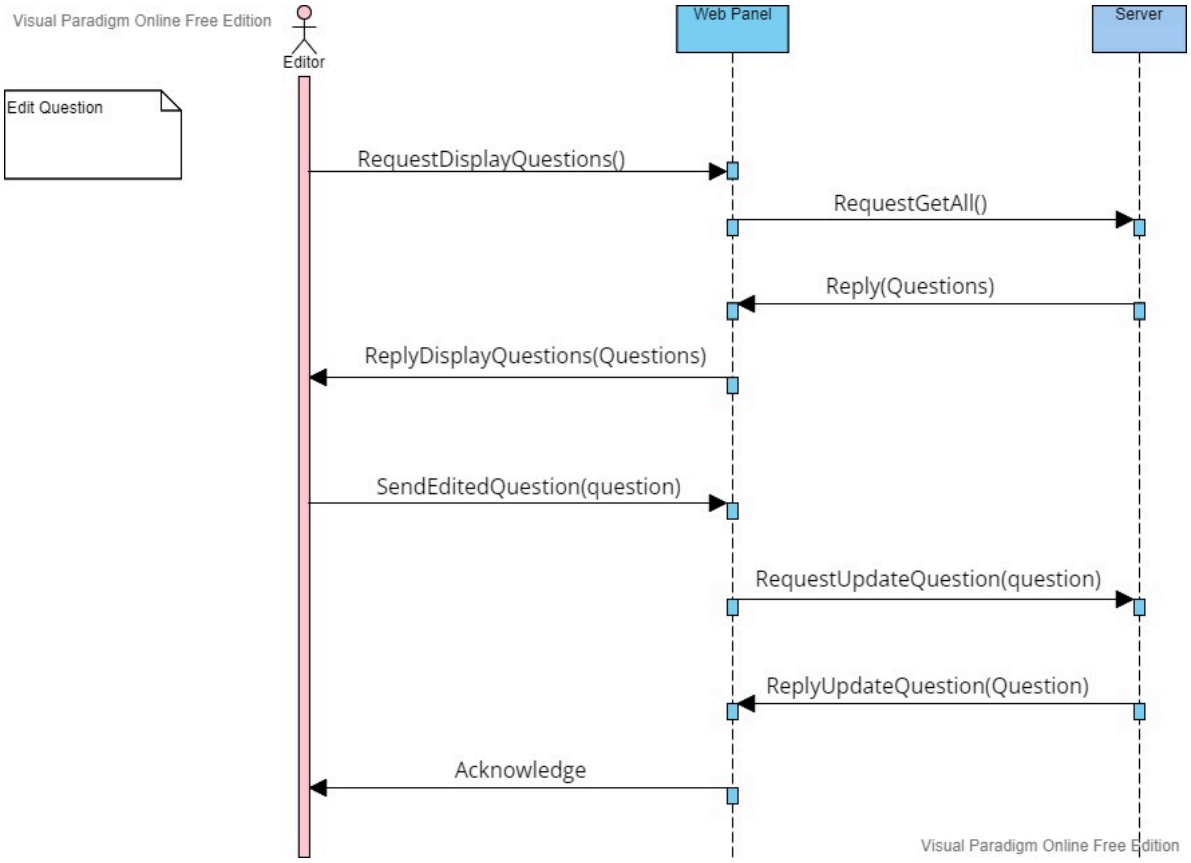

Visual Paradigm Online Free Edition
Player
Game Application
Server
Write Question
SendQuestion(question)
RequestAddNonApprovedQuestion(question)
ReplyAddNonApprovedQuestion(question)
Acknowledge
Visual Paradigm Online Free Edition

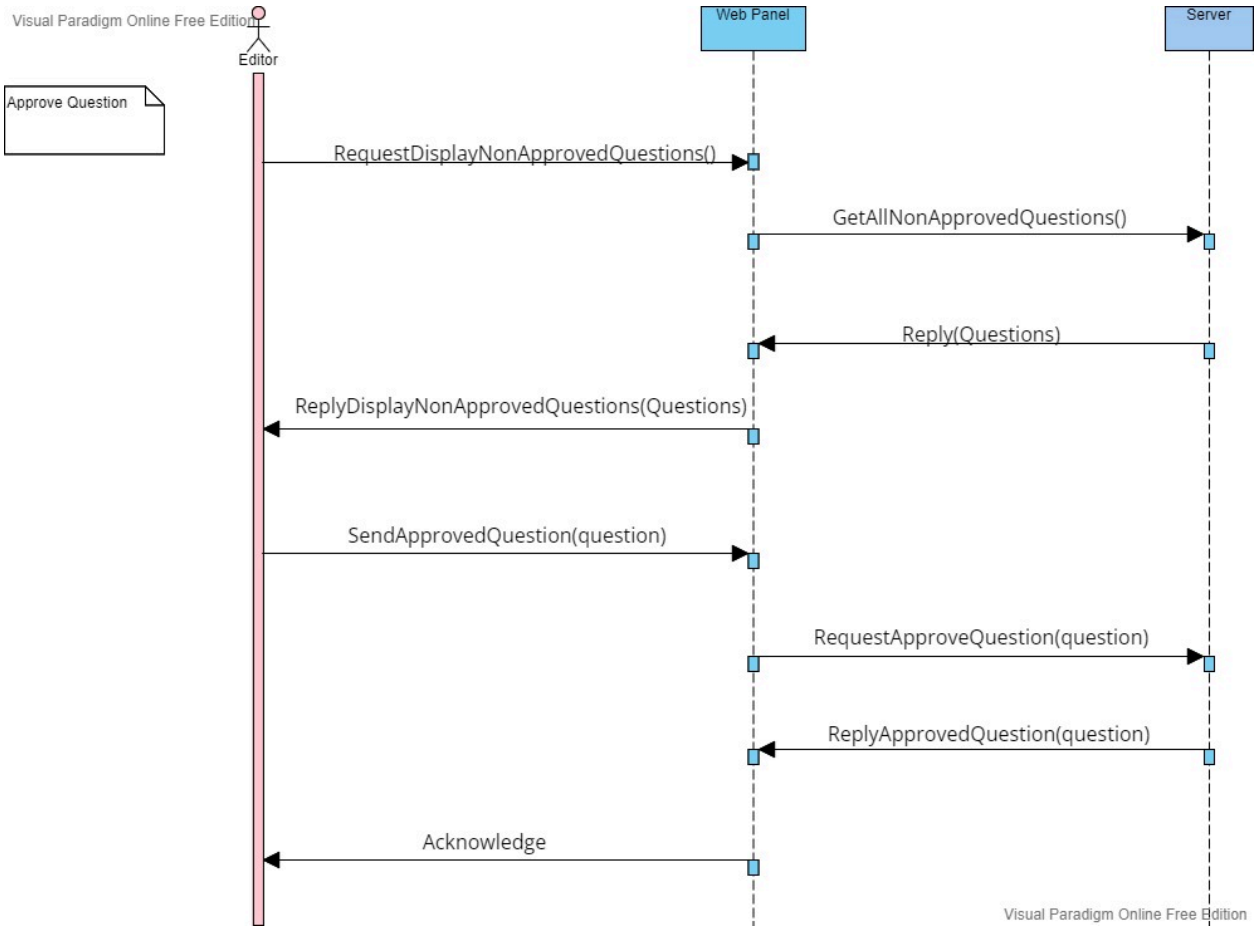

Visual Paradigm Online Free Edition
Player
Game Application
Server
Play Quiz
SendGameMode(mode)
RequestCompetition(mode)
ReplyCompetition(competition)
alt
[If answer==true and capacity = notFull]
loop
RequestNewQuestion(question)
ReplyNewQuestion(question)

DisplayNewQuestion(question)
SendAnswer(answer)
CheckAnswer(answer)
RequestPlayerScore(player)
ReplyPlayerScore(player)
DisplayPlayerScore(score)
Visual Paradigm Online Free Edition


Visual Paradigm Online Free Edition
Add Question
Editor
Web Panel
Server
AddQuestion()
SendQuestion()
Acknowledge
Visual Paradigm Online Free Edition


Visual Paradigm Online Free Edition
Remove Question
Editor
Web Panel
Server
RequestDisplayQuestions()
GetAll()
Reply(Questions)
ReplyDisplayQuestions(Questions)
SendDeletedQuestion(Question)
RequestDeleteQuestion(Question)
ReplyDeleteQuestion(Question)
GetAll()
Visual Paradigm Online Free Edition

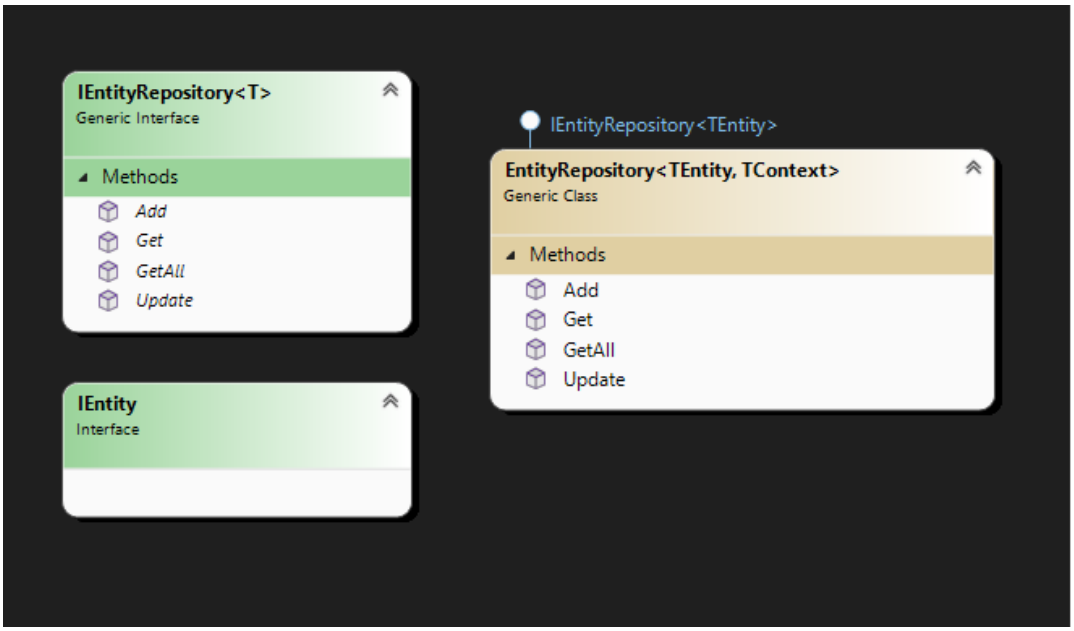

Visual Paradigm Online Free Edition
Editor
Web Panel
Server
Edit Question
RequestDisplayQuestions()
RequestGetAll()
Reply(Questions)
ReplyDisplayQuestions(Questions)
SendEditedQuestion(question)
RequestUpdateQuestion(question)
ReplyUpdateQuestion(Question)
Acknowledge
Visual Paradigm Online Free Edition


Visual Paradigm Online Free Edition
Editor
Web Panel
Server
Approve Question
RequestDisplayNonApprovedQuestions()
GetAllNonApprovedQuestions()
Reply(Questions)
ReplyDisplayNonApprovedQuestions(Questions)
SendApprovedQuestion(question)
RequestApproveQuestion(question)
ReplyApprovedQuestion(question)
Acknowledge
Visual Paradigm Online Free Edition

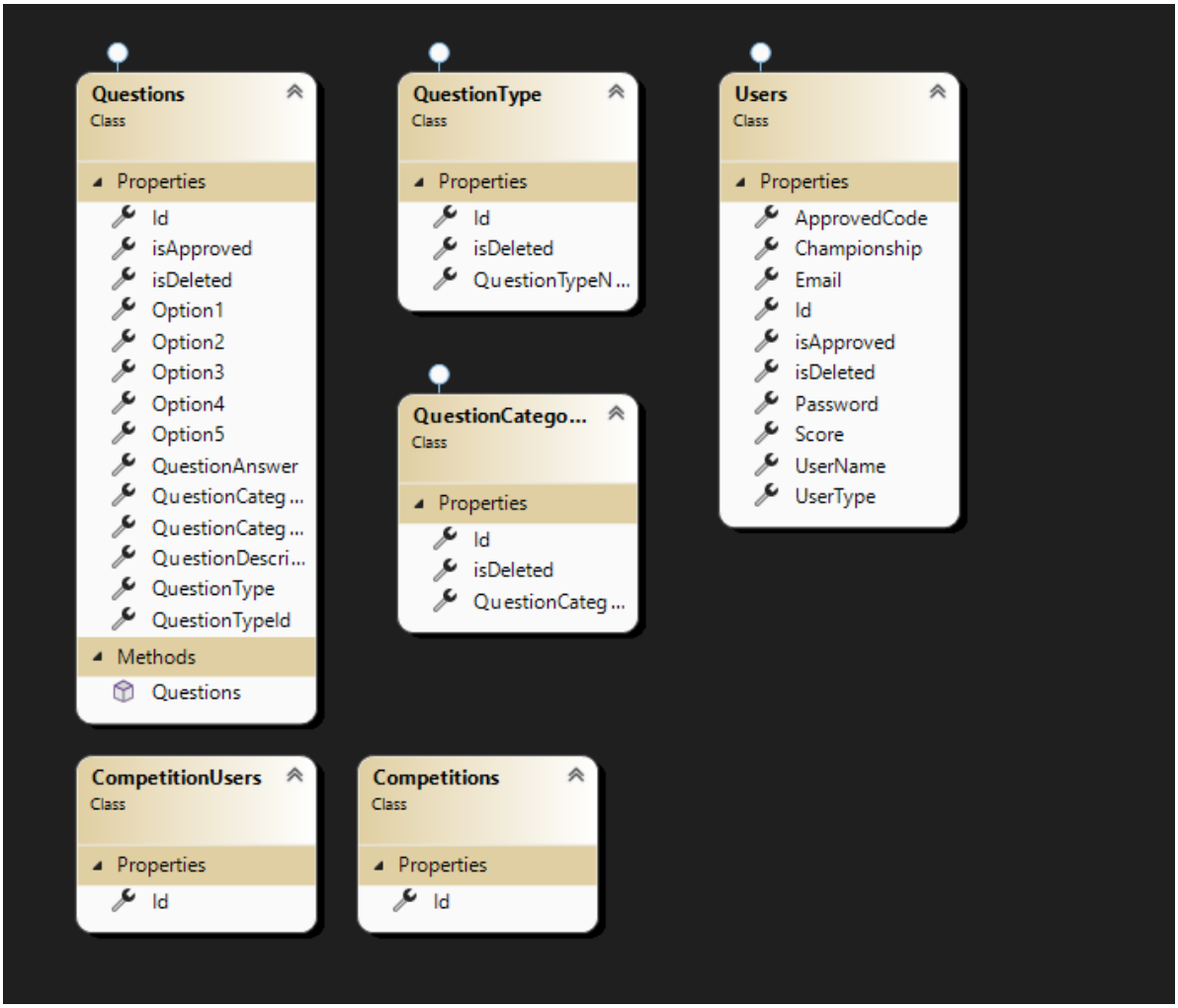
3.5 Class Diagrams

Class Diagrams for Server

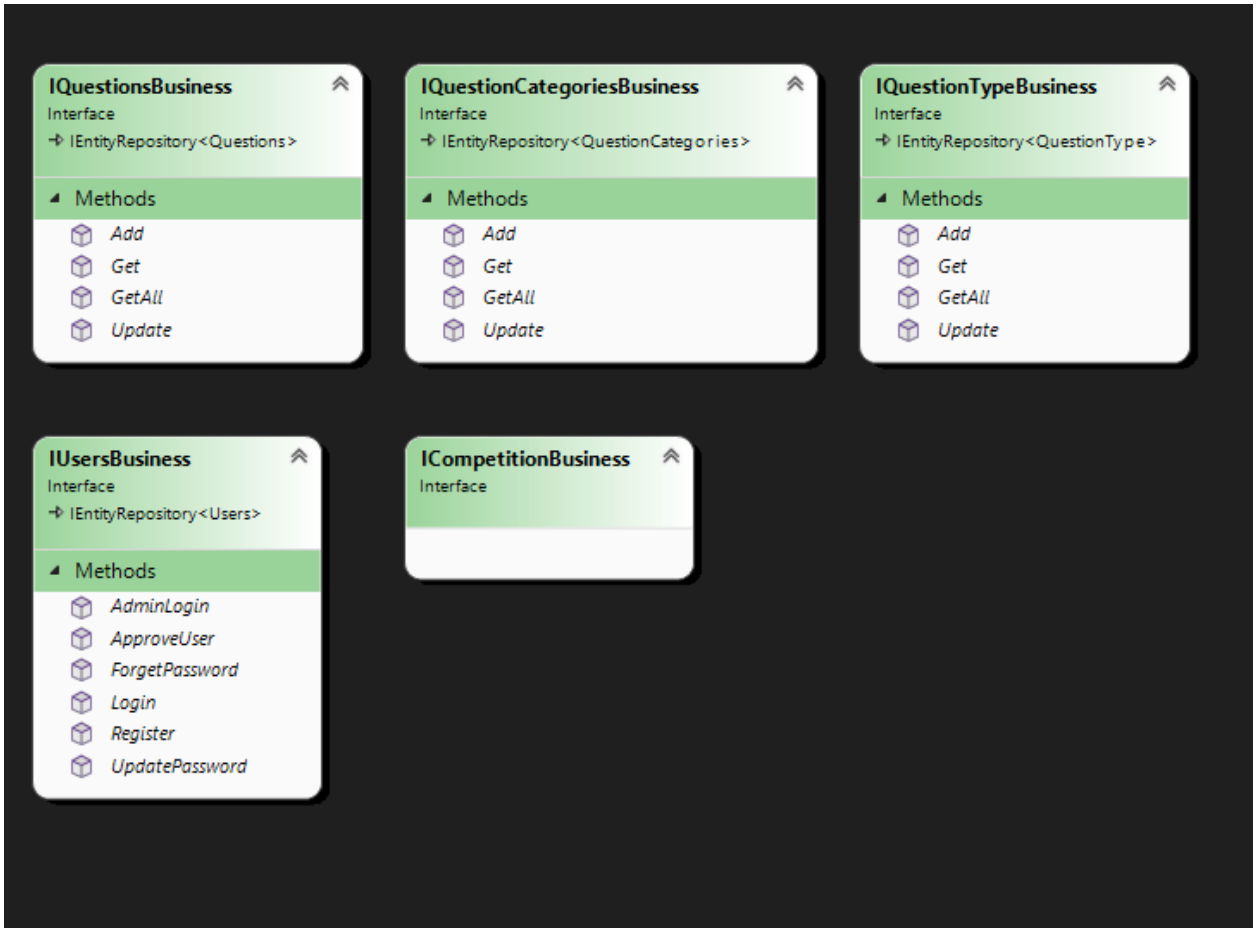
BattleOfMinds.Core

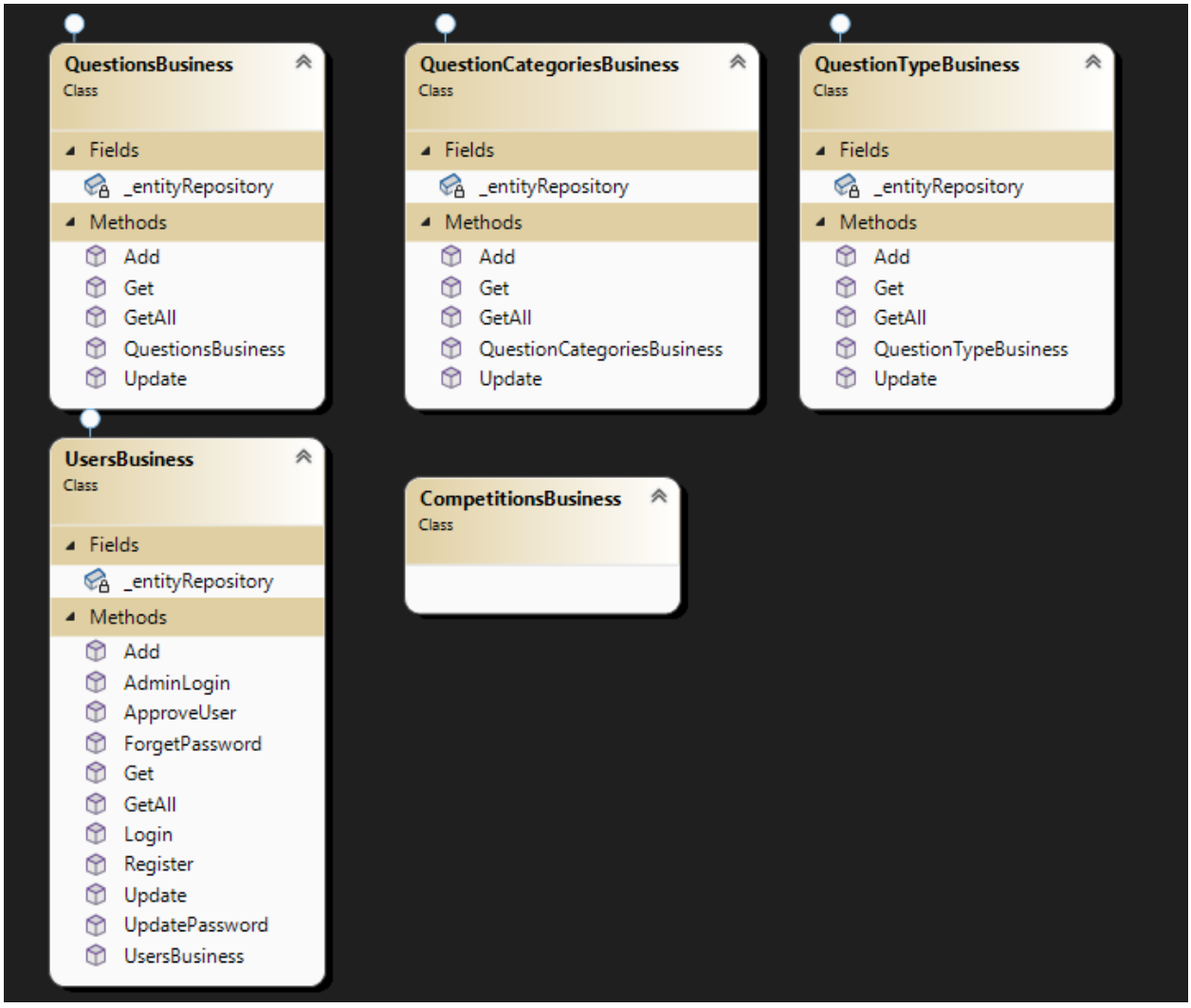


BattleOfMinds.Models

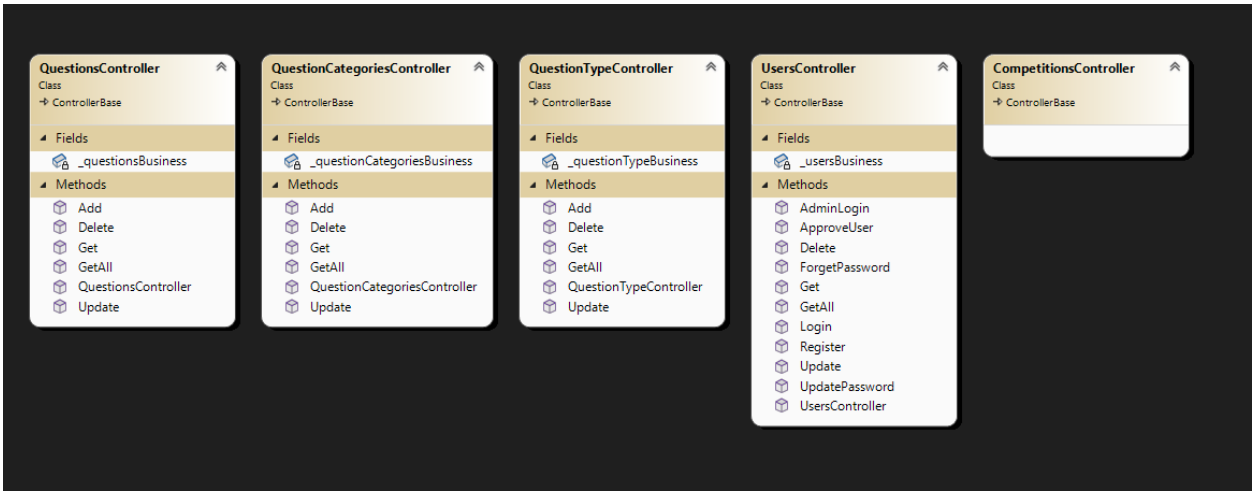


BattleOfMinds.API Business

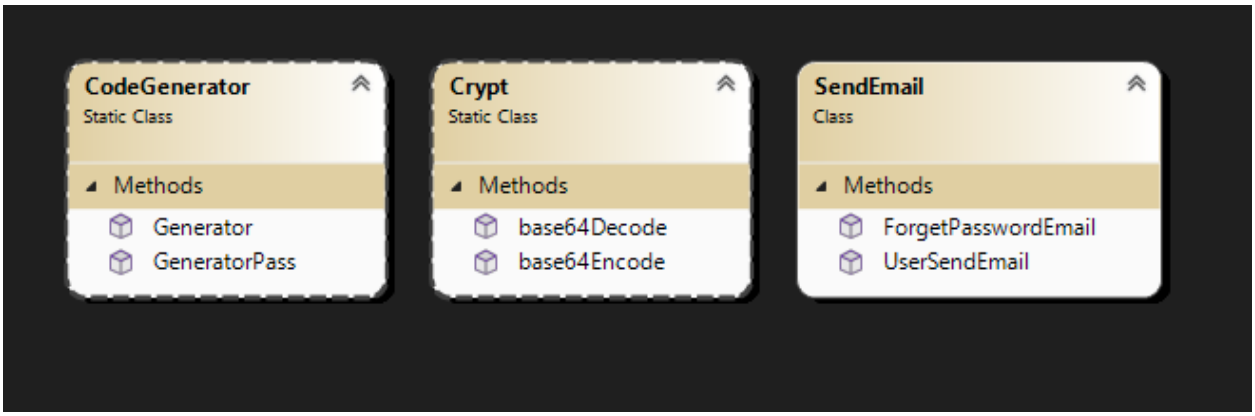




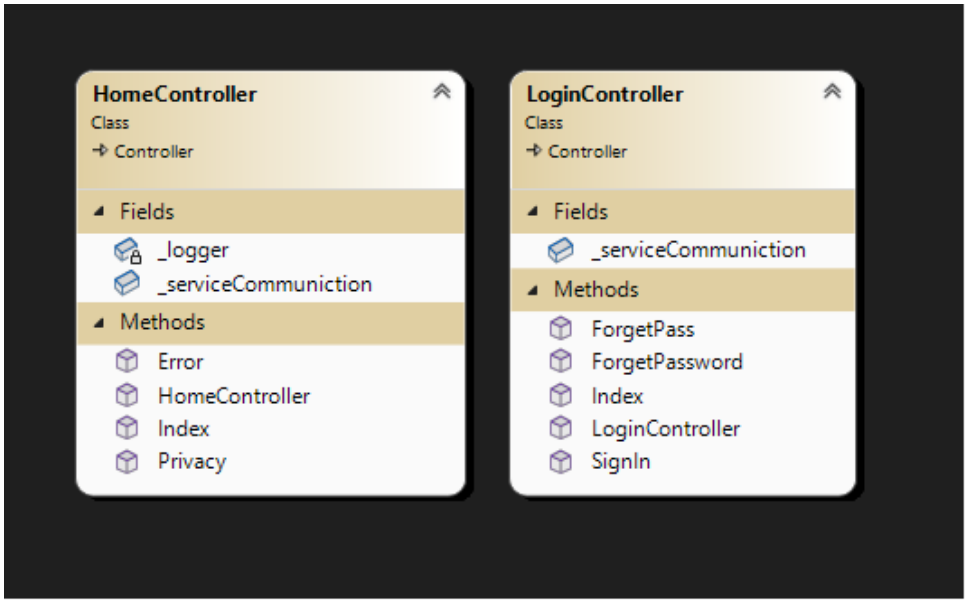
BattleOfMinds.API Controllers

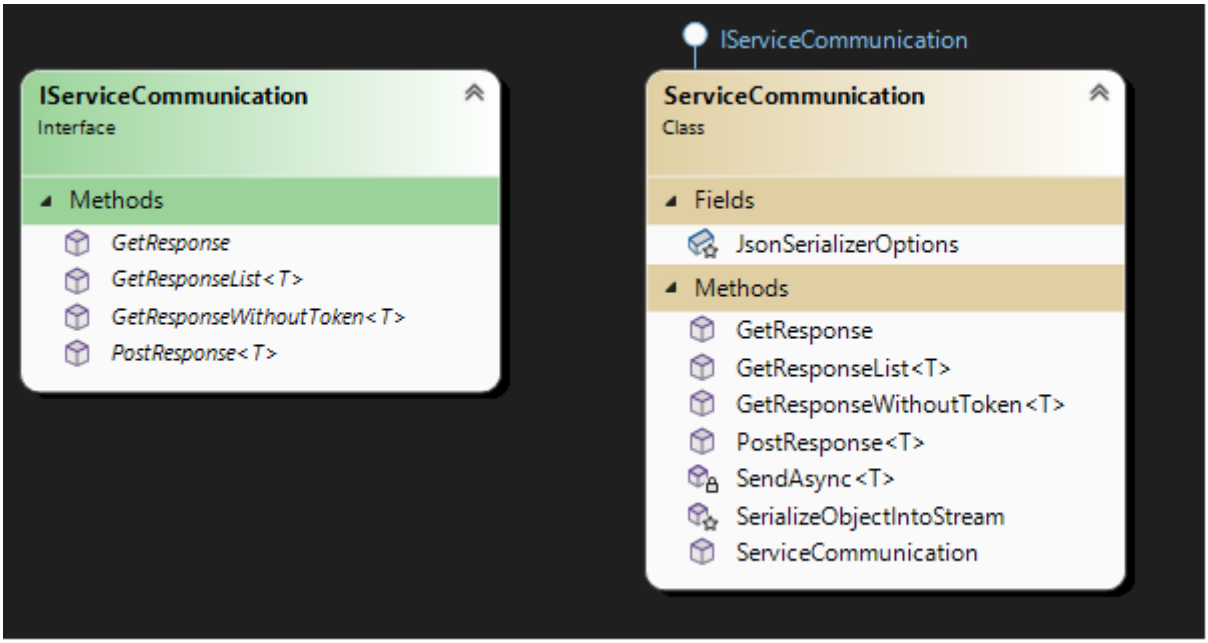


BattleOfMinds.API Helpers

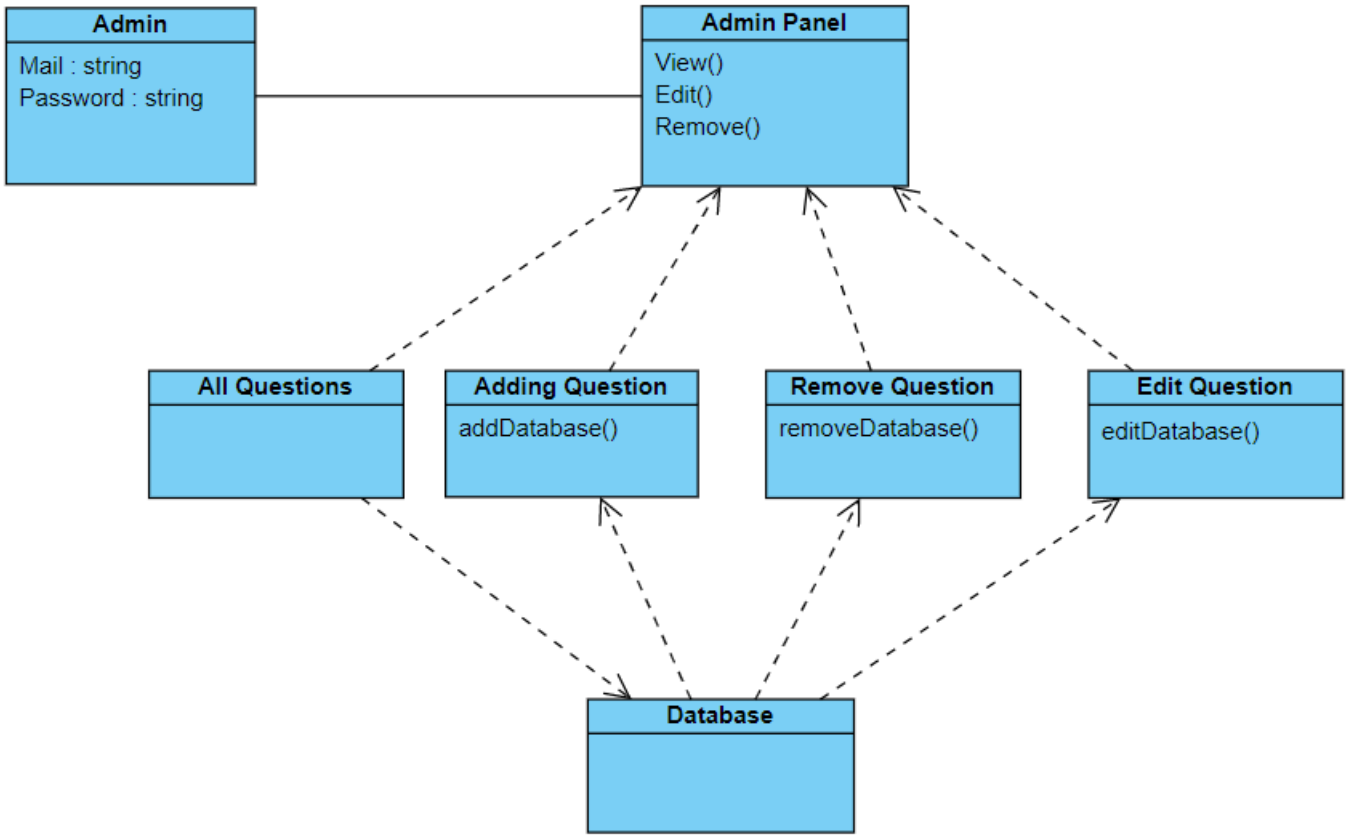


BattleOfMinds.MVC Controller

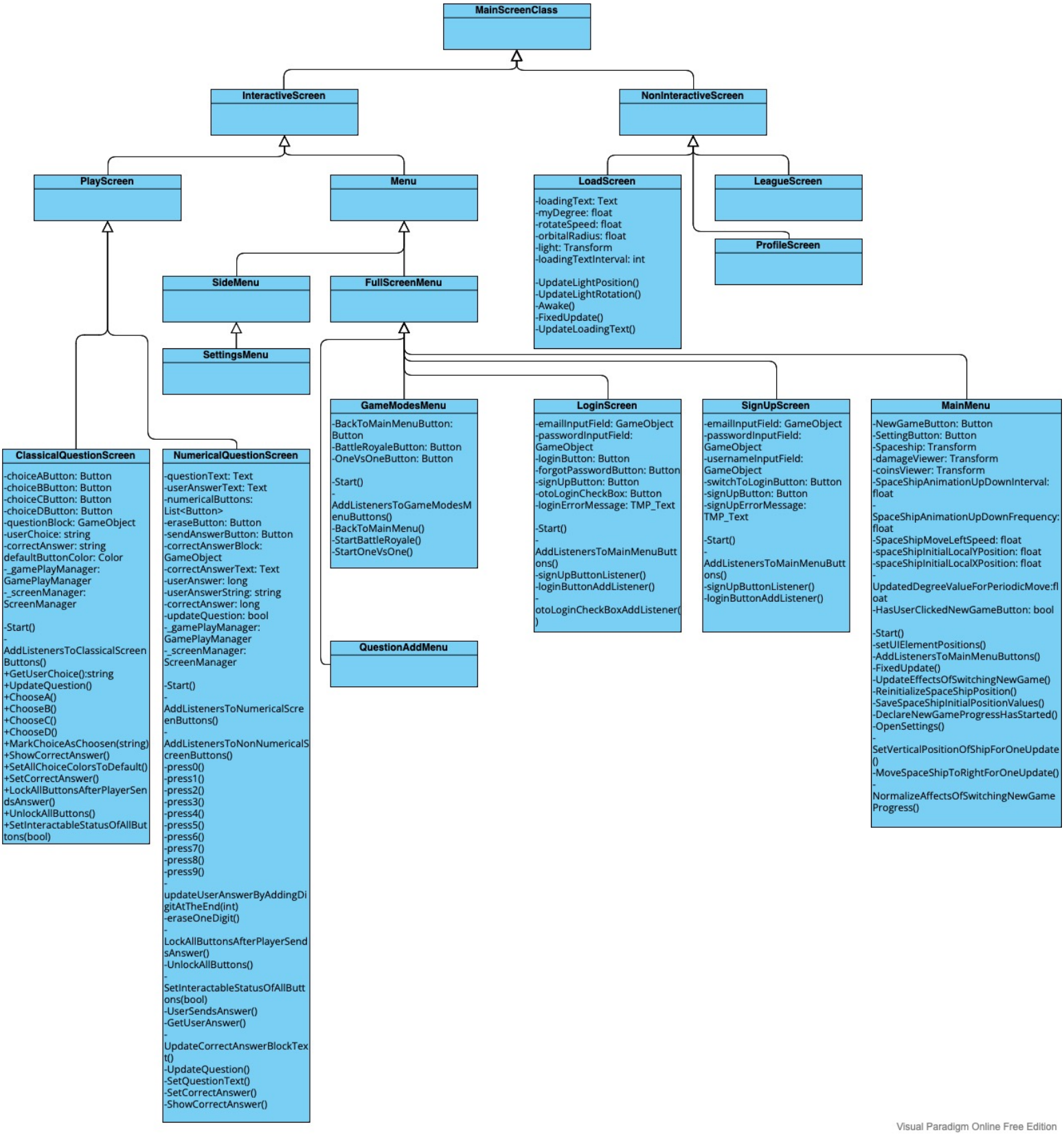


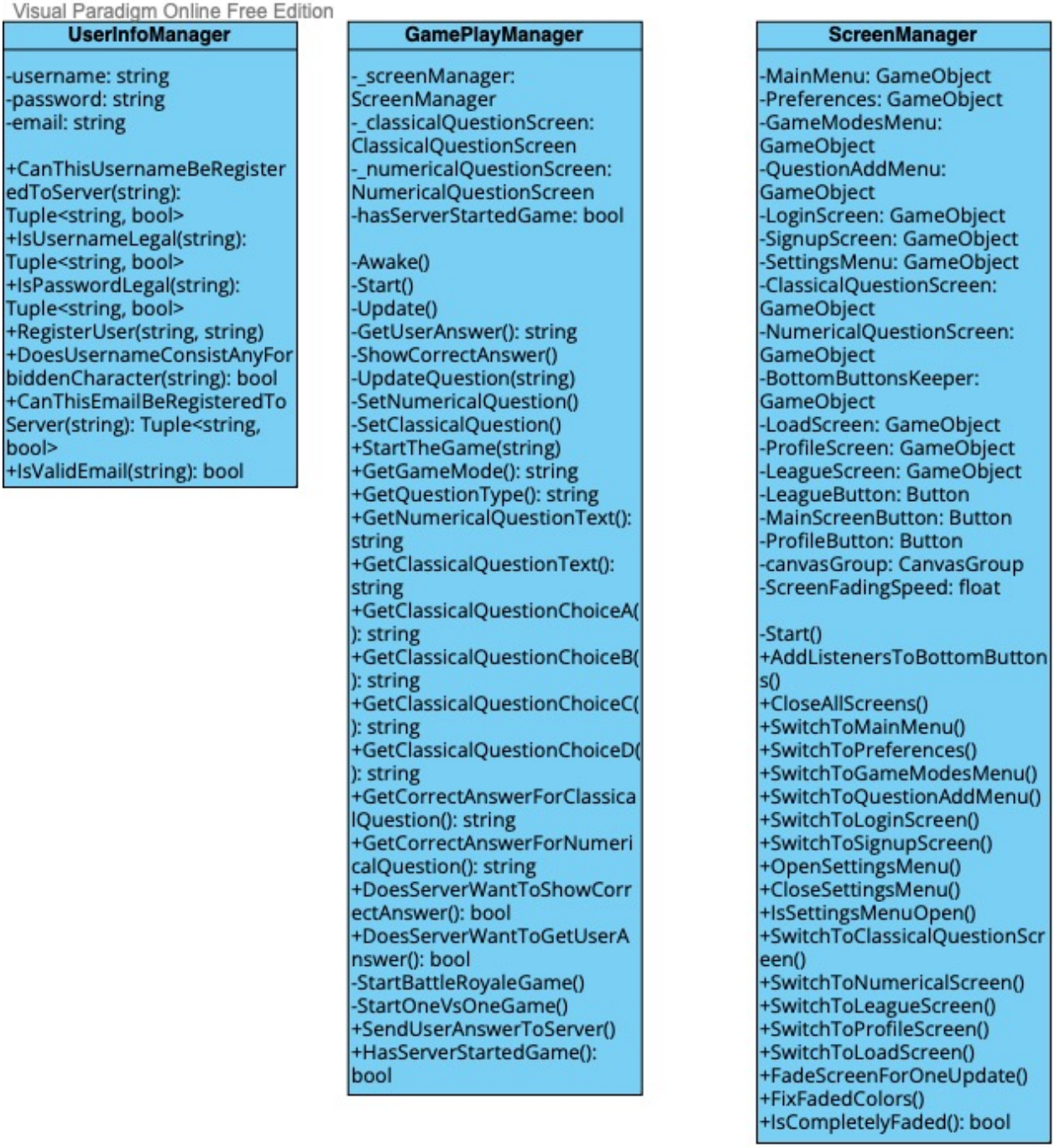


Class Diagram for Web Editor



Class Diagram for Game





4. Examples From Working Graphical Interface

4.1 Examples from Game

Log-in Screen

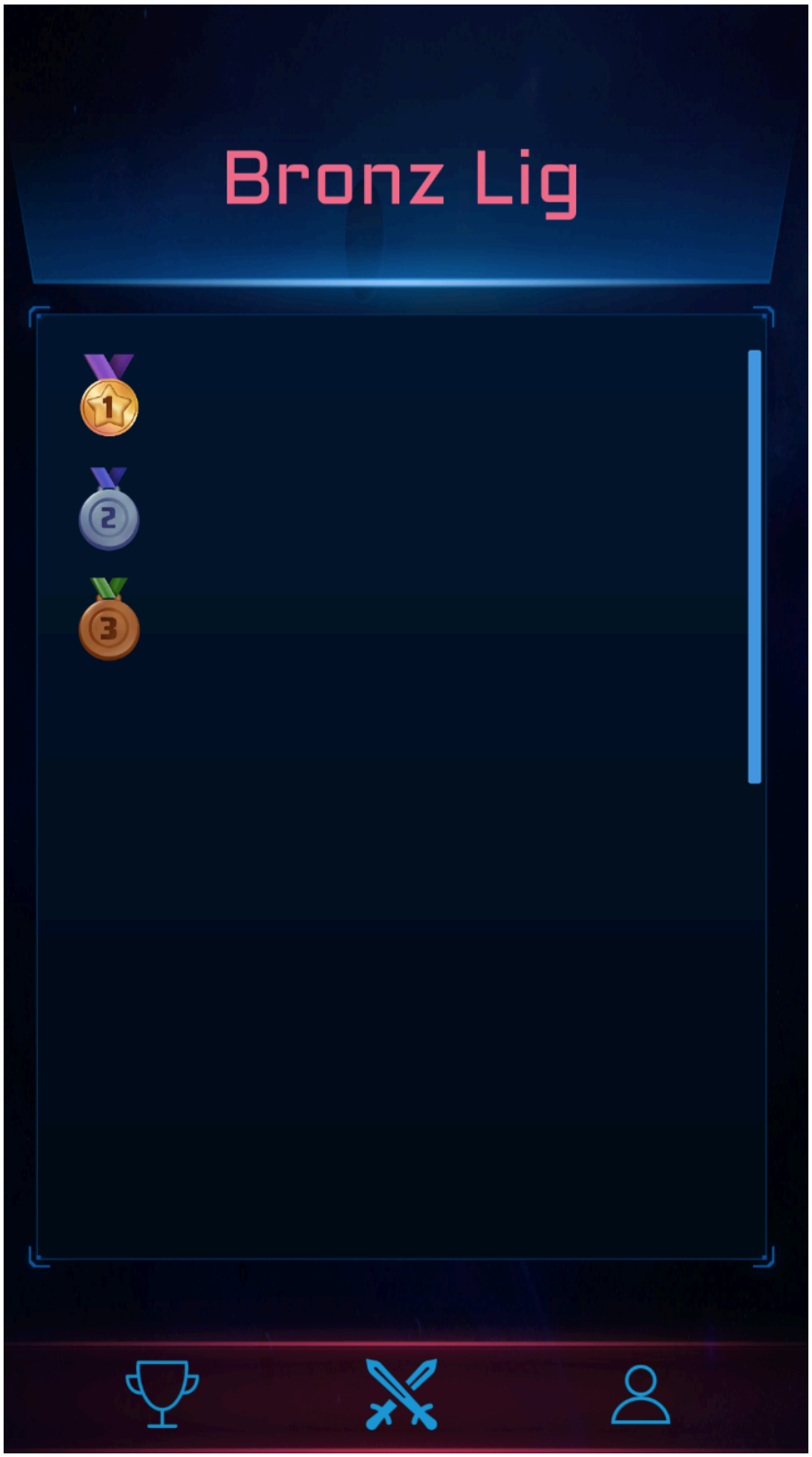
Sign-up Screen

Main Menu

Settings Side Menu

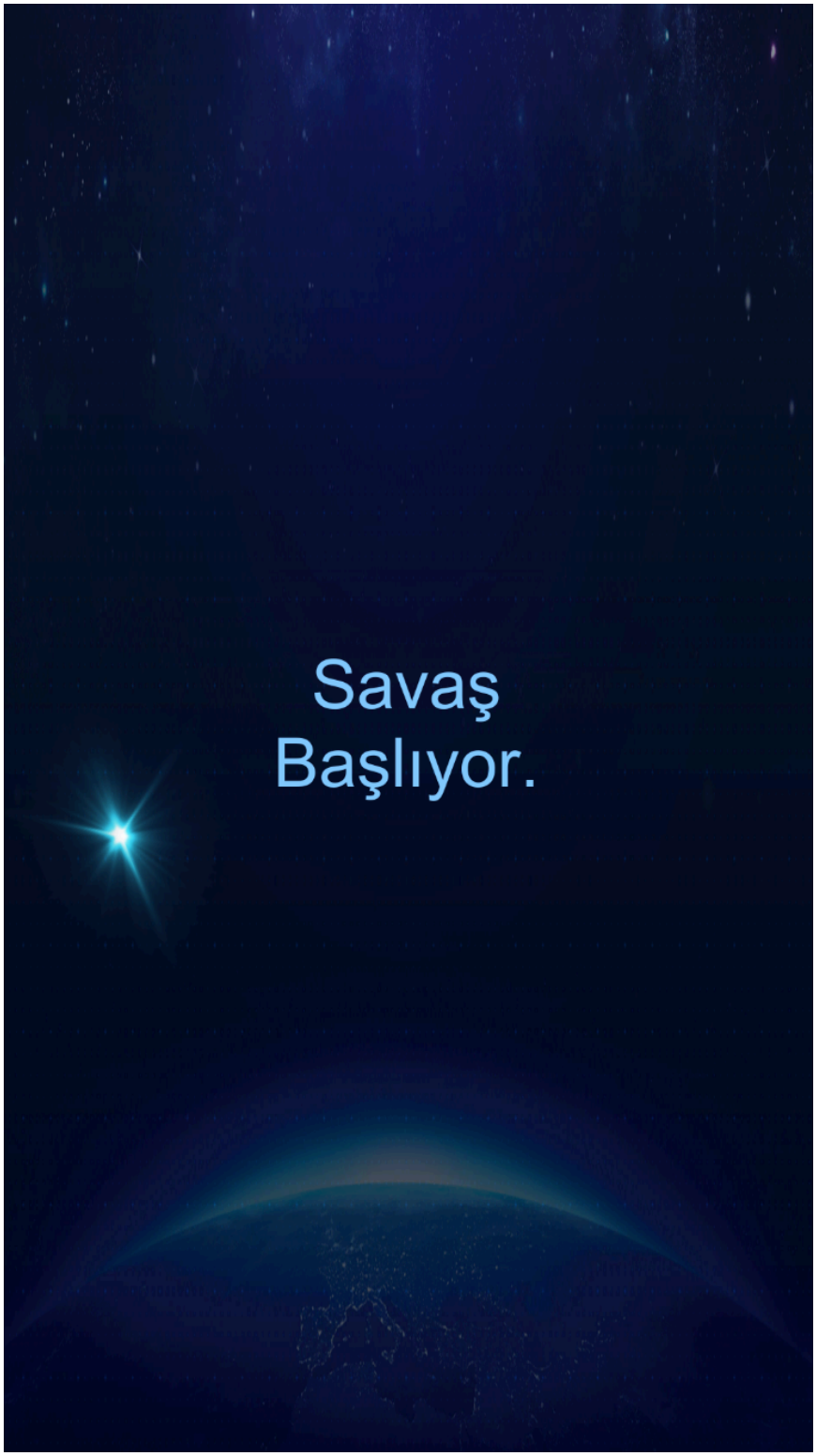
Profile Screen

League Screen



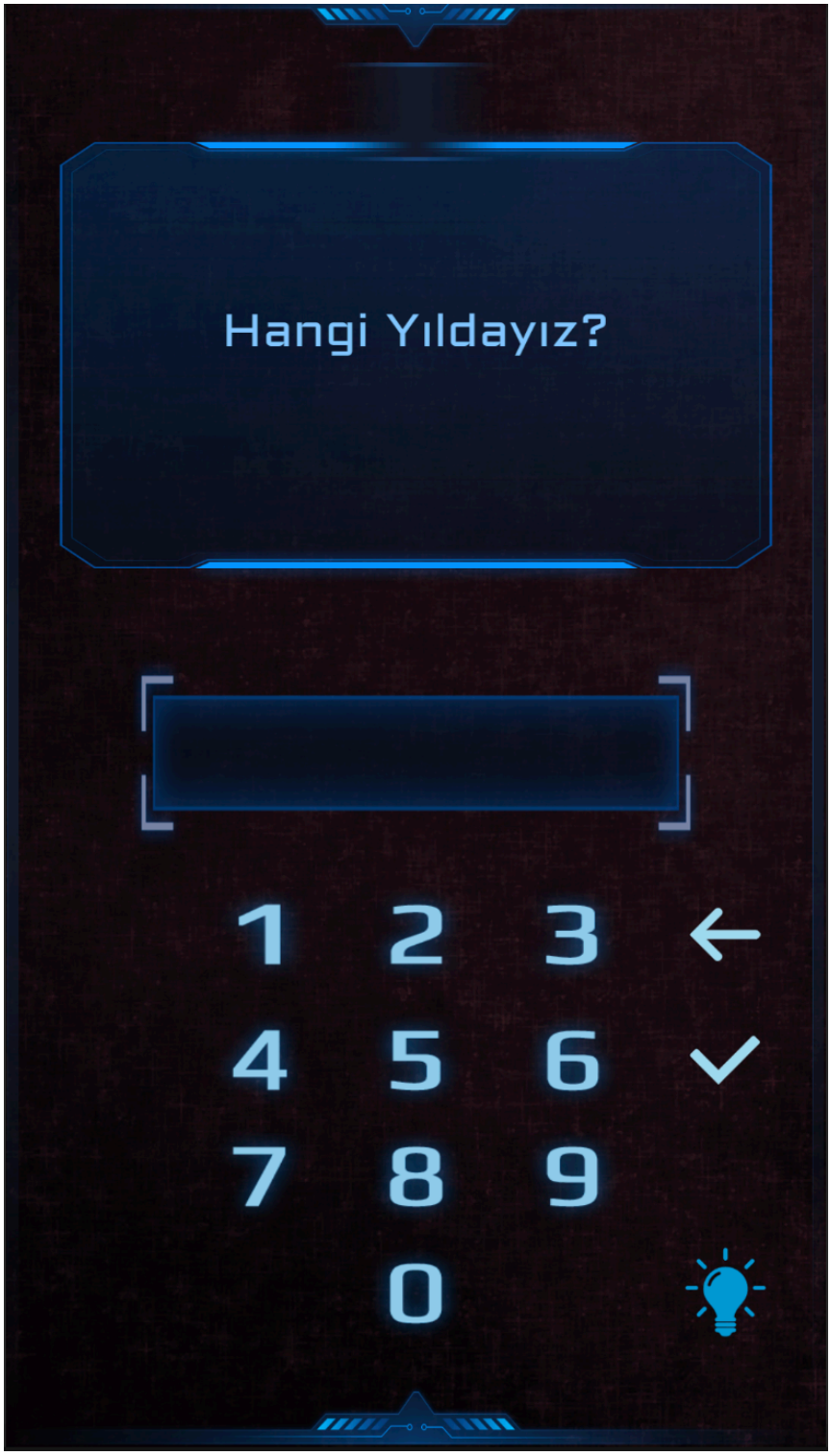
Game Mode Selection Screen

Load Screen

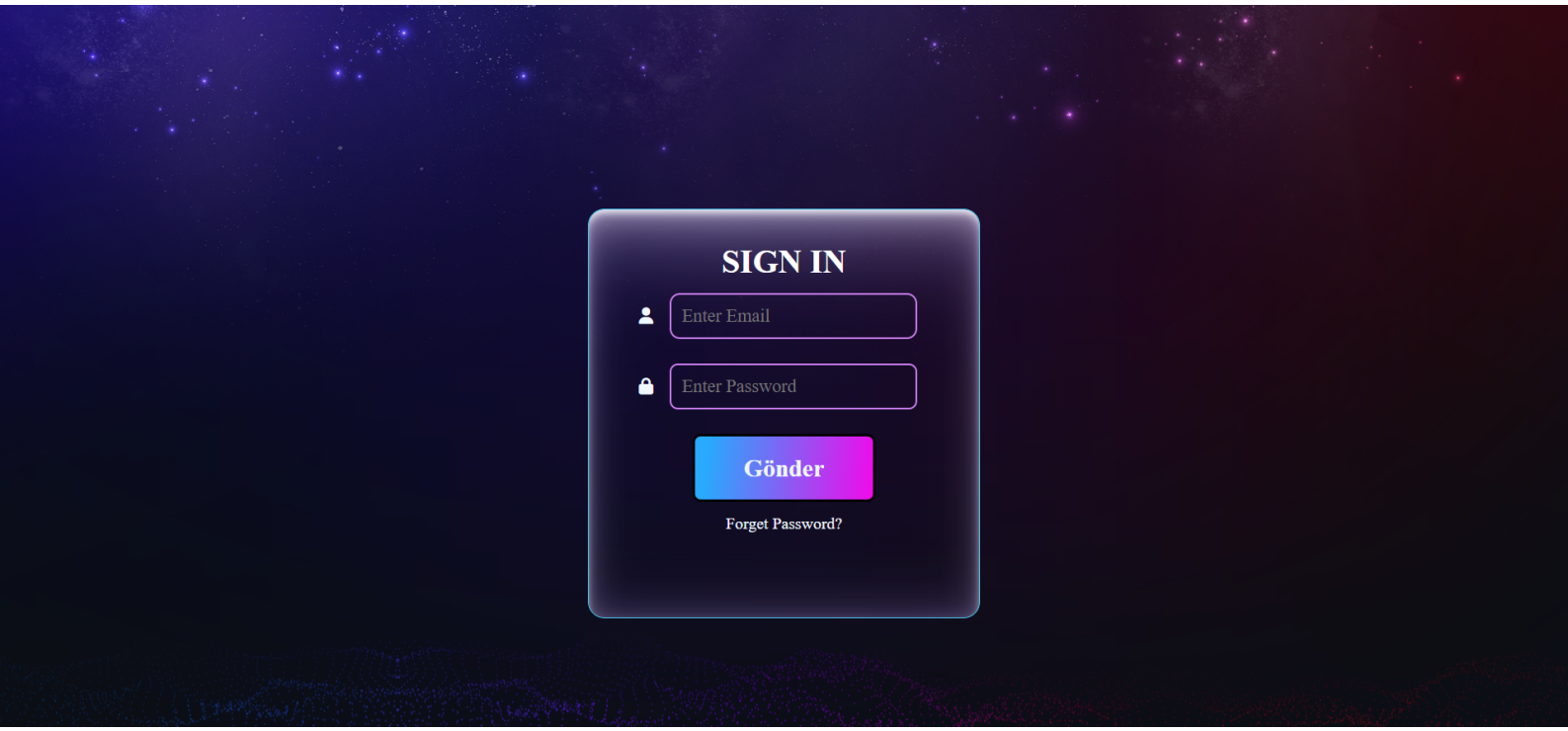


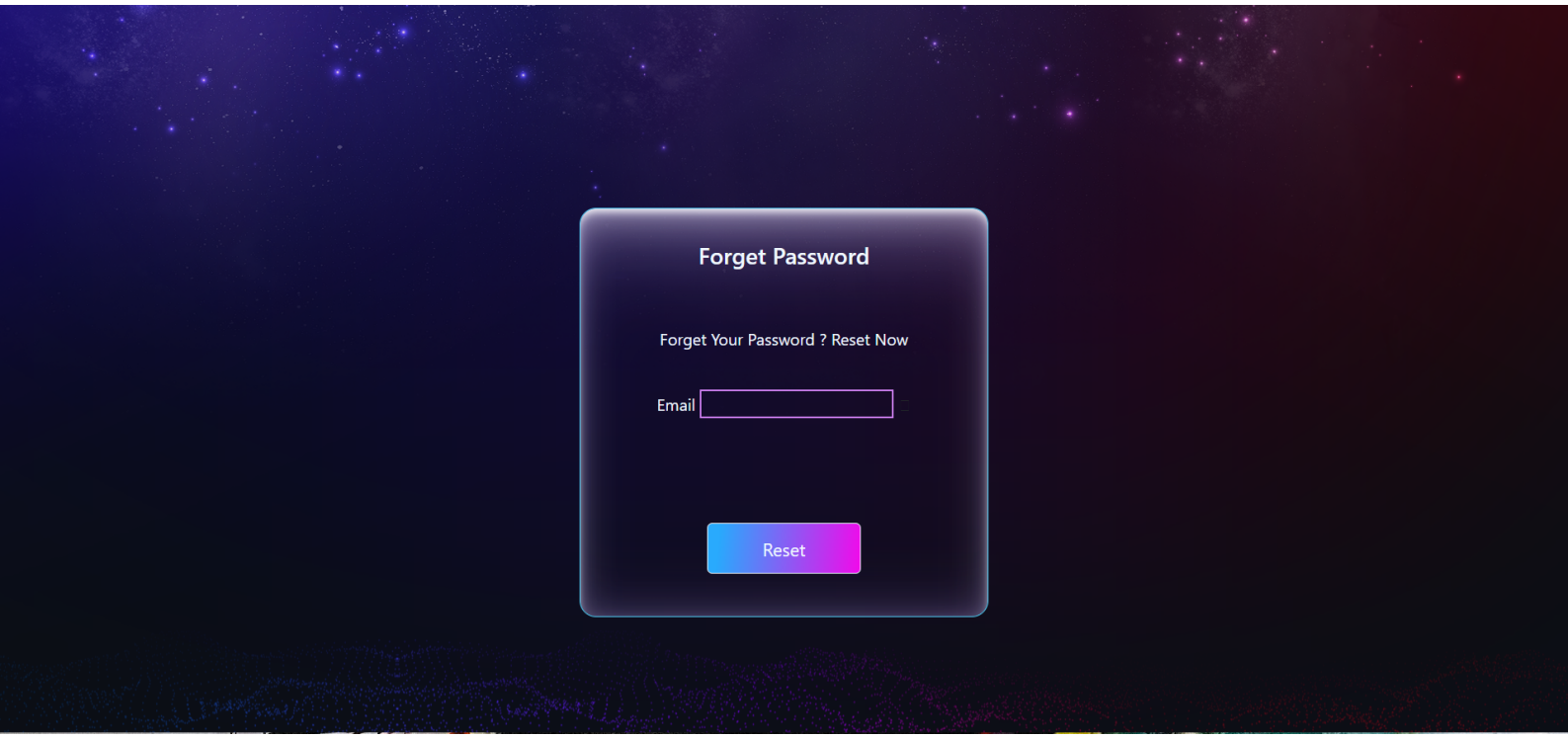
Classical Question Screen

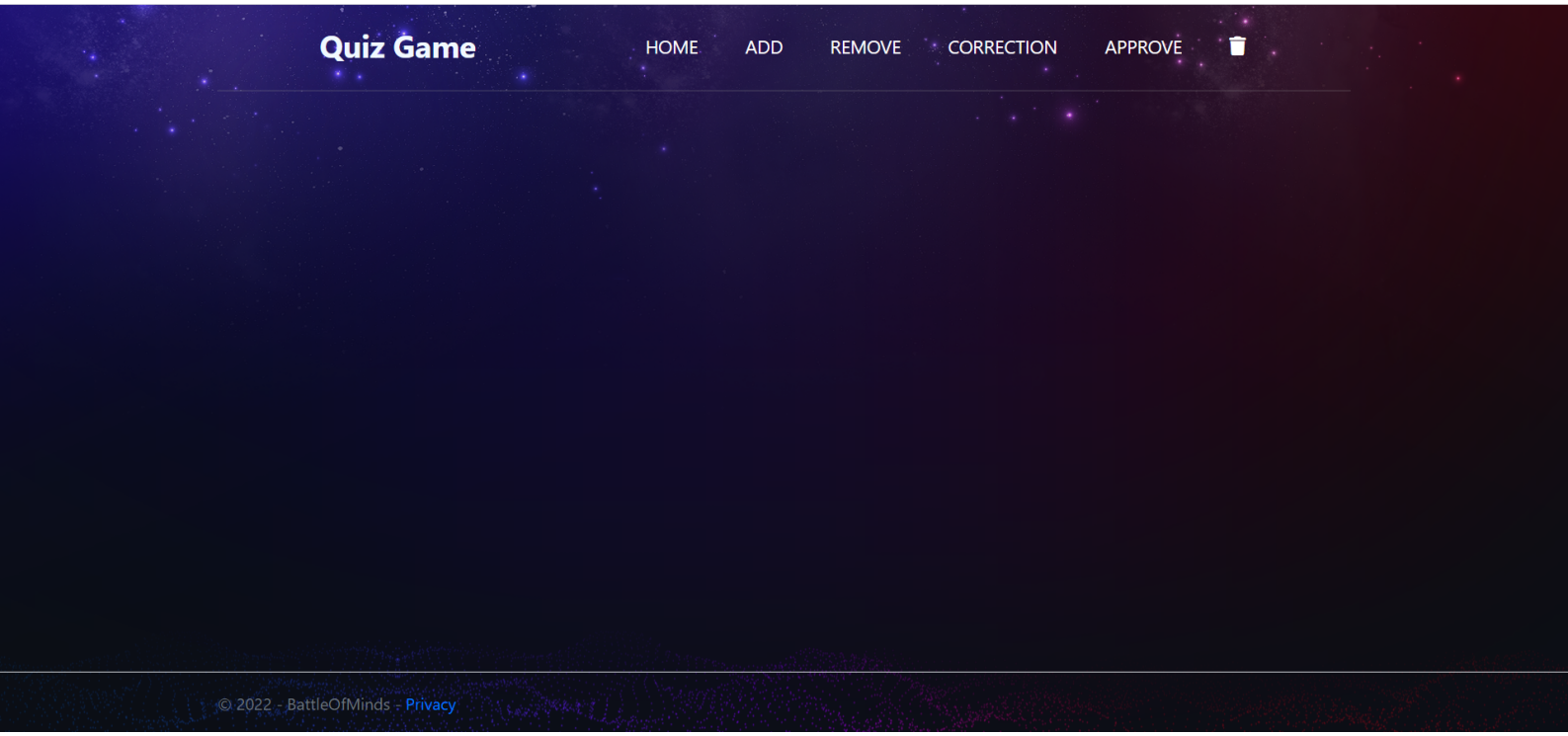
Numerical Question Screen

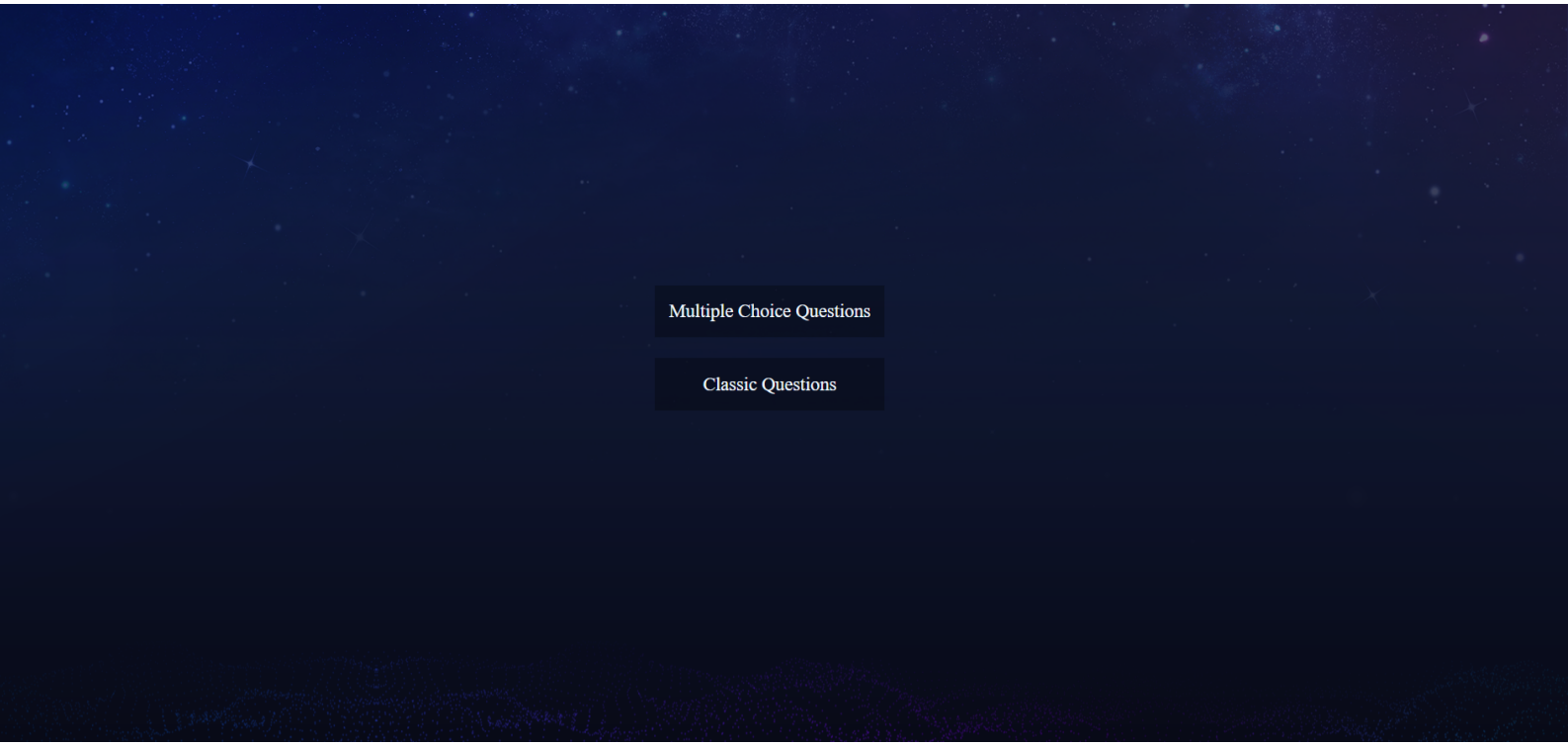


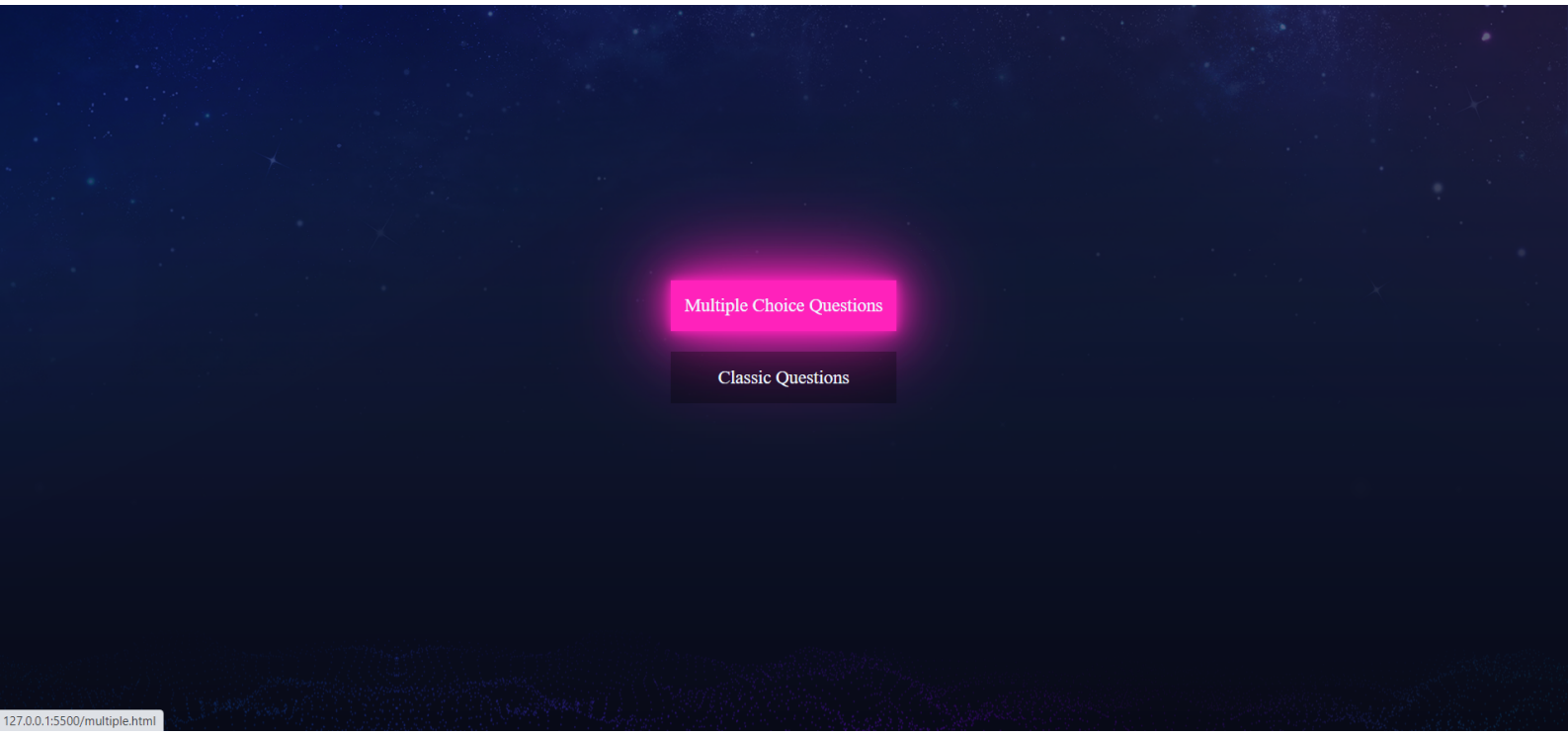
4.2 Examples from Web Editor

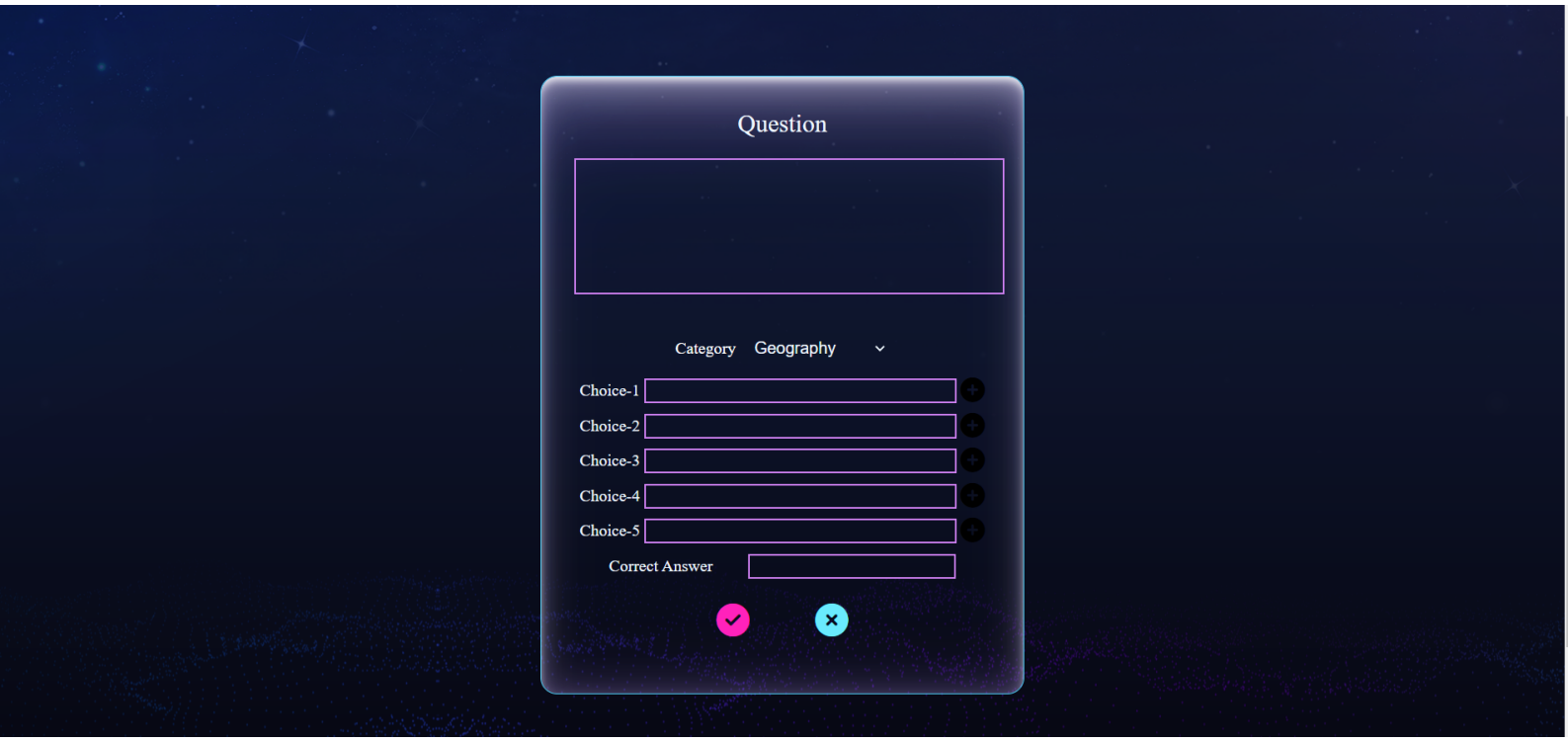


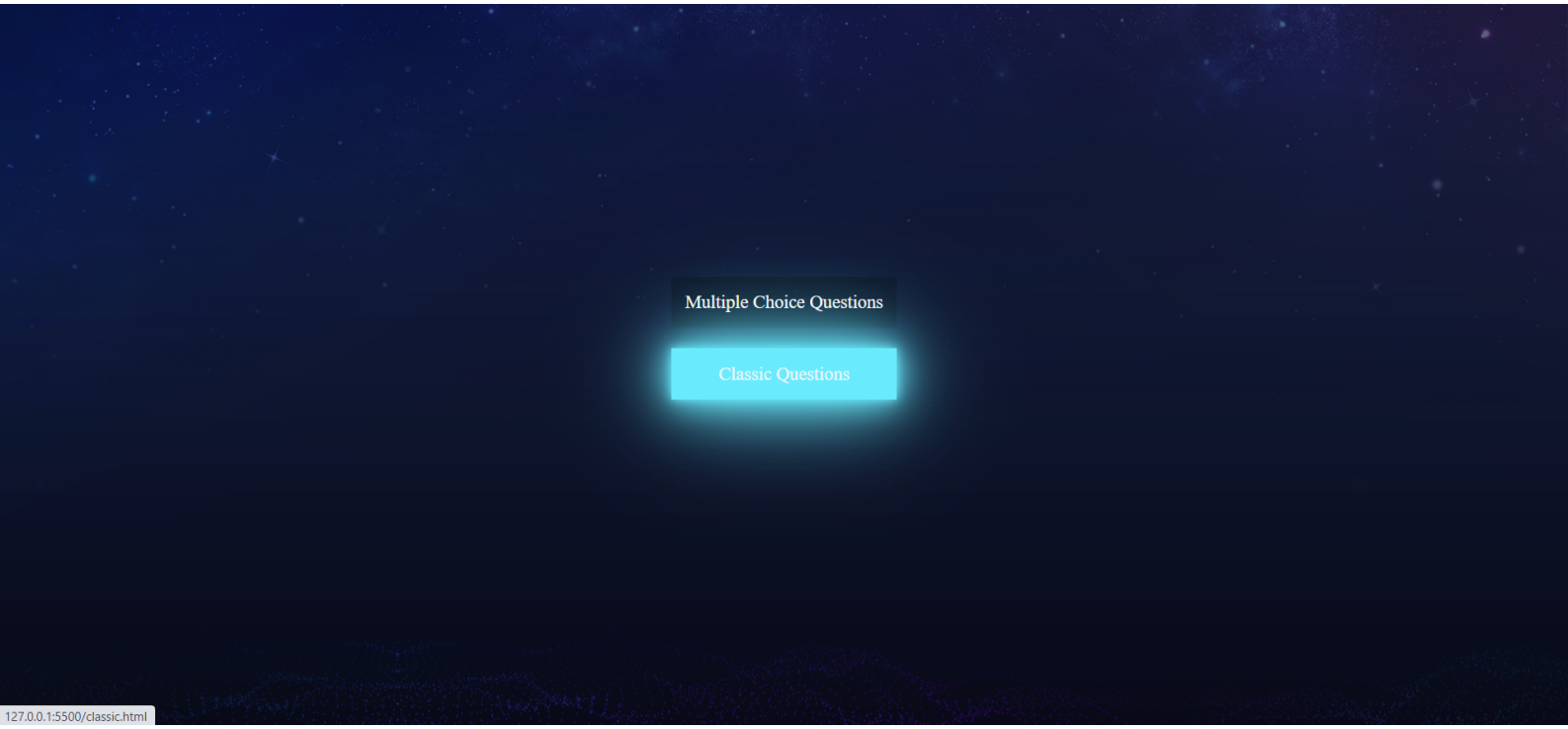


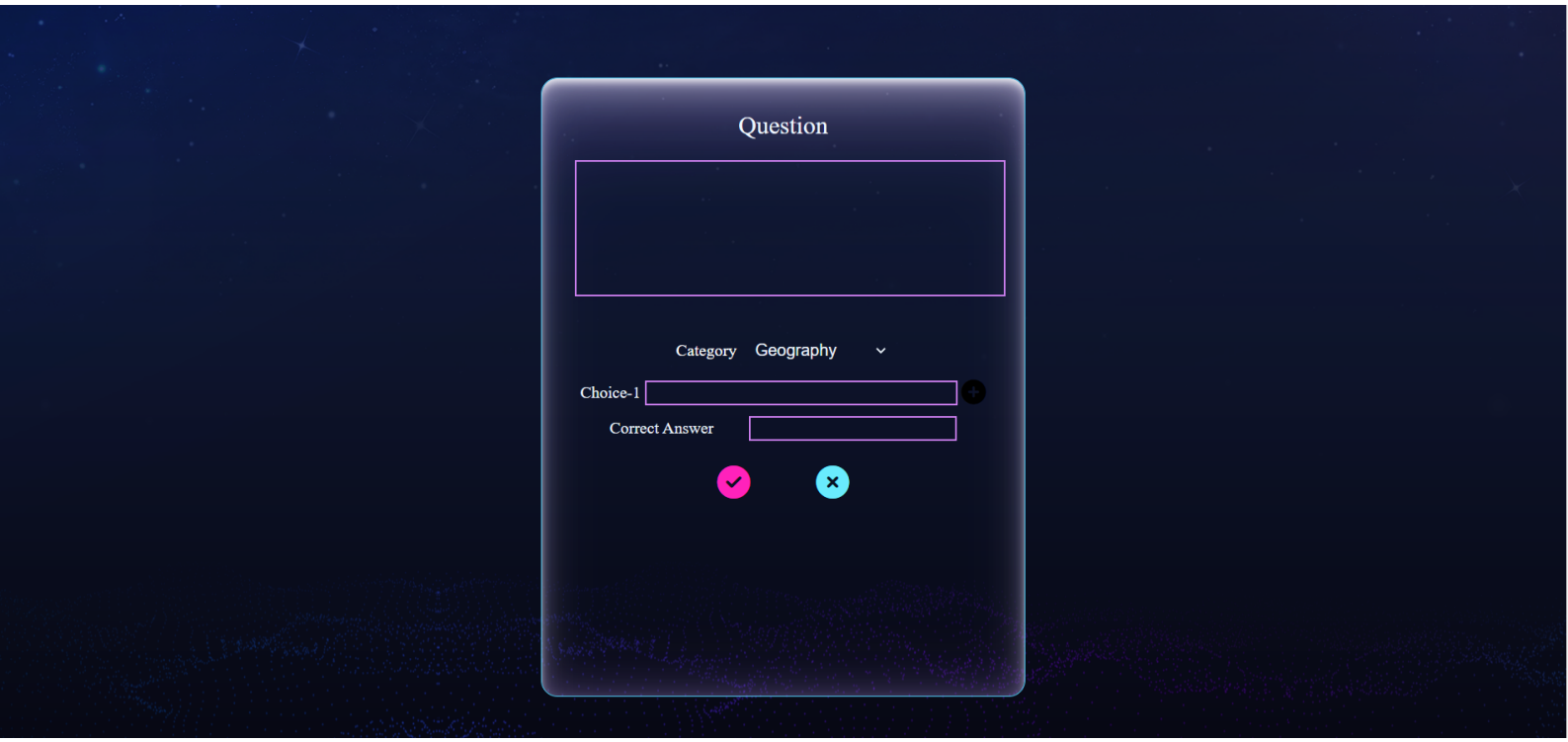

Quiz Game
HOME
ADD
REMOVE
CORRECTION
APPROVE
© 2022 - BattleOfMinds - Privacy

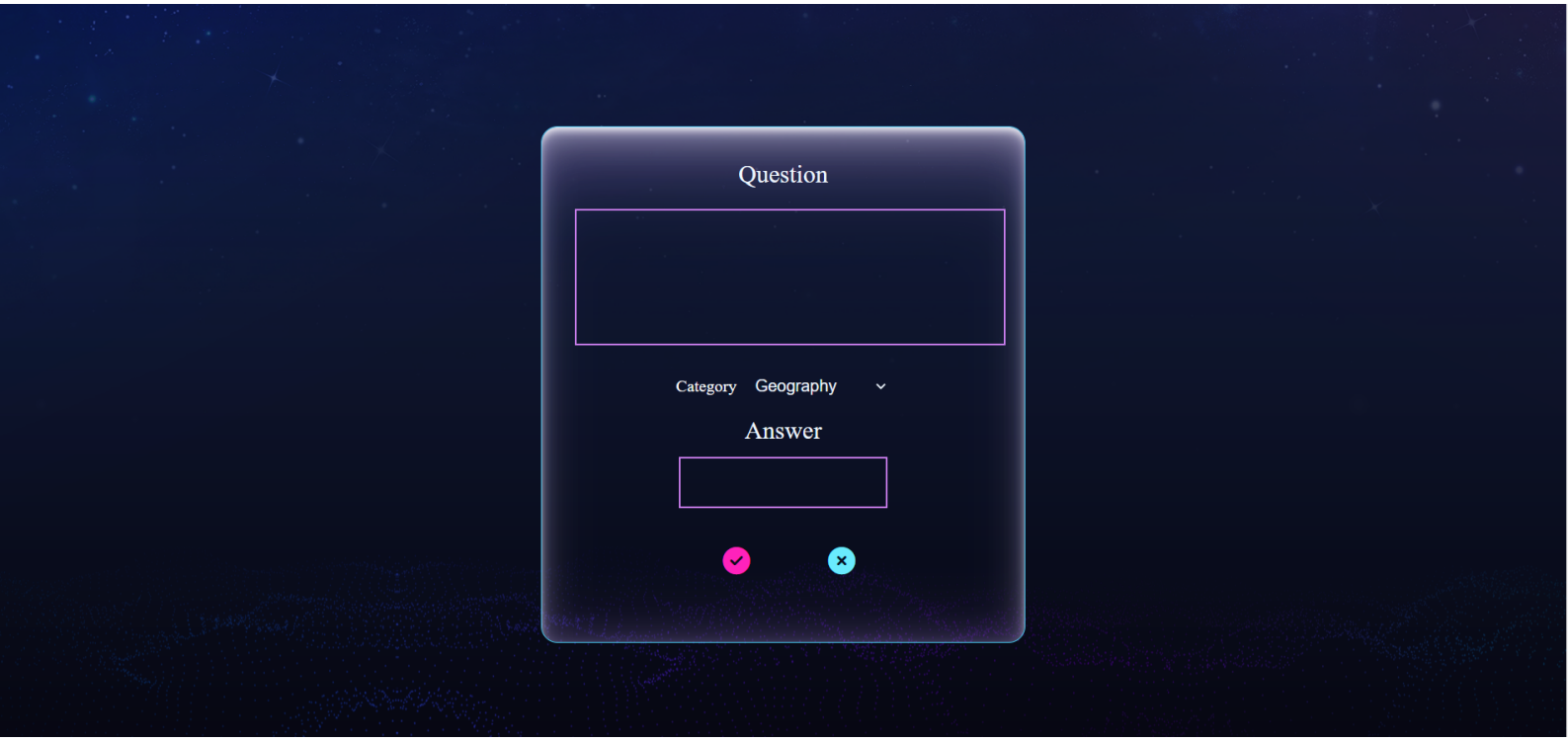

Multiple Choice Questions
Classic Questions


Multiple Choice Questions
Classic Questions
127.0.0.1:5500/multiple.html


Question
Category
Geography
Choice-1
Choice-2
Choice-3
Choice-4
Choice-5
Correct Answer


Multiple Choice Questions
Classic Questions
127.0.0.1:5500/classic.html


Question
Category
Geography
Choice-1
Correct Answer


Question
Category
Geography
Answer

5. Test Reports

5.1 Test Reports for Web Editor

Test Case ID	Test Case Objective	Prequisitite	Steps	Input Data	Expected Output	Actual Output	Status
TC_01	Choosing the type of question to adding	Admins registered in the database will be able to access	1) Choose the type of question you want to add (classic or multiple choice)	N/A	Go to the add questions page according to the selected question type	Go to the add questions page according to the selected question type	Pass
TC_02	Adding classical question	Admins registered in the database will be able to access	1)Add question 2)Add answer 3) Click on the confirmation icon	Questions and answers	The question and answer are added to the database.	The question and answer are added to the database.	Pass
TC_03	Adding multiple choice question	Admins registered in the database will be able to access	1)Add question 2)Add choice 3)Add answers 3) Click on the confirmation icon	Questions, choices and answers	The question,choices and answer are added to the database.	The question,choices and answer are added to the database.	Pass
TC_04	Remove the question, options and answers	Admin must have written the question but not approved	1)Press the reject icon	N/A	Deleting the questions and choices written on the screen	The questions and options written on the screen are deleted.	Pass
TC_05	Viewing the questions written by the admin herself	Questions must be registered in the database	1) Click on the question display menu from the admin menu	N/A	The admin sees all the questions s/he has written.	The admin sees all the questions s/he has written.	Pass

TC_06	Remove the question	Question must be previously added in order to be deleted	1) Go to the panel where all questions are seen 2) Click the delete sign next to the questions	N/A	The previously written question with a database added is extracted from the questions written using the soft deletion method.	The previously written question with a database added is extracted from the questions written using the soft deletion method.	pass
TC_07	Edit the question	Question must be previously added in order to be edit	1) Go to panel showing all questions 2) Press the edit icon 3) Edit the question on the screen where the question comes in written form	N/A	It should be possible to edit the questions written before.	Previously written questions are edited. It replaces the old version in the database.	Pass

5.2 Test Reports for Server

Test Case Id	Test Case Objective	Prequistite	Steps	Input Data	Expected Output	Actual Output	Status
G_TC_01	Test Users Controller GetAll Function	All Users should be available in the database	1. Create UsersController object 2. Call GetAll function	N/A	A Users List that includes all users available on the system.	A Users List that includes all users available on the system.	Pass
G_TC_02	Test Users Controller Get Function	Given Email should be available on the system	1. Create UsersController object 2. Call Get function 3. Pass an email to the function	String email = “User@hotmail.com”	A Users object that has the given email.	A Users object that has the given email.	Pass
G_TC_03	Test Users Controller Delete Function	Given Users object should be available on the system	1. Create UsersController object 2. Call Delete function 3. Pass a Users object to the function	Users user	A Users object that is deleted from the system.	A Users object that is deleted from the system.	Pass
G_TC_04	Test Users Controller Update Function	Given Users object should be available on the system	1. Create UsersController object 2. Call Update function 3. Pass a Users object to the function	Users user	A Users object that is updated.	A Users object that is updated.	Pass
G_TC_05	Test Users Controller Register Function	Given Users object should not be available on the system	1. Create UsersController object 2. Call Register function 3. Pass a Users object to the function	Users user	A success or error message about the register status.	A success or error message about the register status.	Pass

G_TC_06	Test Users Controller ApproveUser Function	Given userId should be registered to the system	1. Create UsersController object 2. Call ApproveUser function 3. Pass a userId and a code value	İnt UserId = 1 String code = “ADF5R7Y”	A boolean value about the validation status.	A boolean value about the validation status.	Pass
G_TC_07	Test Users Controller Login Function	Given Email and password should be available on the system	1. Create UsersController object 2. Call Login function 3. Pass an email and a password	String email = User@hotmail.com String password = Password123	A boolean value about the login status.	A boolean value about the login status.	Pass
G_TC_08	Test Users Controller AdminLogin Function	Given Email and password should be available as admin on the system	1. Create UsersController object 2. Call AdminLogin function 3. Pass an email and a password	String email= Admin@gmail.com String password = Password123	A boolean value about the admin login status.	A boolean value about the admin login status.	Pass
G_TC_09	Test Users Controller Forget Password Function	Given Email should be available on the system	1. Create UsersController object 2. Call ForgetPassword function 3. Pass an email to the function	String email = User@hotmail.com	A boolean value about the forget password status.	A boolean value about the forget password status.	Pass
G_TC_10	Test Users Controller Update Password Function	Given userId should be available on the system	1. Create UsersController object 2. Call UpdatePassword function 3. Pass an email and a password	İnt UserId = 1 String oldPassword = Password123 String newPassword = pass123	A boolean value about the update password status.	A boolean value about the update password status.	Pass

G_TC_11	Test Questions Controller GetAll Function	All Questions should be available in the database	1. Create QuestionsController object 2. Call GetAll function	N/A	A Questions List that includes all questions available on the system.	A Questions List that includes all questions available on the system.	Pass
G_TC_12	Test Questions Controller Get Function	Given Question Id should be available on the system	1. Create QuestionsController object 2. Call Get function 3. Pass an id to the function	İnt id = 1	A Questions object that has the given id.	A Questions object that has the given id.	Pass
G_TC_13	Test Questions Controller Add Function	Given Questions object should not be available on the system	1. Create QuestionsController object 2. Call Add function 3. Pass a Questions object to the function	Questions question	A Questions object that is added to the system.	A Questions object that is added to the system.	Pass
G_TC_14	Test Questions Controller Delete Function	Given Question Id should be available on the system	1. Create QuestionsController object 2. Call Delete function 3. Pass an id to the function	İnt id = 1	A Questions object that is deleted from the system.	A Questions object that is deleted from the system.	Pass
G_TC_15	Test Questions Controller Update Function	Given Questions object should be available on the system	1. Create QuestionsController object 2. Call Update function 3. Pass a Questions object to the function	Questions question	A Questions object that is updated.	A Questions object that is updated.	Pass

5.3 Test Reports for Game

Test Case Id	Test Case Objective	Prequistite	Steps	Input Data	Expected Output	Actual Output	Status
G_TC_01	Create ScreenManager Class	A ScreenManager object has to be created	1.Create an object of ScreenManager class 2.Assign to ScreenManager game object 3.Check if any screen reference is null	None	None of the screen references are null	None of the screen references are null	Passed
G_TC_02	Check ScreenManager Class CloseAllScreens method	A ScreenManager object has to keep reference of all screens of the game	1.Call CloseAllScreens method. 2.Check if all screens are disabled	None	All screens gets disabled,screen goes dark	All screens gets disabled,screen goes dark	Passed
G_TC_03	Check ScreenManager Class SwitchToMainMenu method	A ScreenManager object has to keep reference of the main menu screen	1.Call CloseAllScreens method. 2.Check if main menu and bottom buttons get opened	None	All screens gets disabled,main menu and bottom buttons gets enabled	All screens gets disabled,main menu and bottom buttons gets enabled	Passed
G_TC_04	Check ScreenManager Class SwitchToLeagueScreen method	A ScreenManager object has to keep reference of the league screen	1.Call SwitchToLeagueScreen method. 2.Check if league screenand bottom buttons get opened	None	All screens gets disabled,league screen and bottom buttons gets enabled	All screens gets disabled,league screen and bottom buttons gets enabled	Passed
G_TC_05	Check ScreenManager Class SwitchToProfileScreen method	A ScreenManager object has to keep reference of the league screen	1.Call SwitchToProfileScreen method. 2.Check if league screenand bottom buttons get opened	None	All screens gets disabled,league screen and bottom buttons gets enabled	All screens gets disabled,league screen and bottom buttons gets enabled	Passed
G_TC_06	Check ScreenManager Class SwitchToPreferences method	A ScreenManager object has to keep reference of the league screen	1.Call SwitchToPreferences method. 2.Check if preferences gets opened	None	All screens gets disabled,league screen and preferences screen gets enabled	All screens gets disabled,league screen and preferences screen gets enabled	Passed

G_TC_07	Check ScreenManager Class SwitchToGameModes-Menu method	A ScreenManager object has to keep reference of the game modes menu	1.Call SwitchToGameModes-Menu method. 2.Check if game modes menu gets opened	None	All screens gets disabled,game modes menu gets enabled	All screens gets disabled,game modes menu gets enabled	Passed
G_TC_08	Check ScreenManager Class SwitchToFriendAddMenu method	A ScreenManager object has to keep reference of the friend add menu	1.Call SwitchToFriendAddMenu method. 2.Check if the friend add gets opened	None	All screens gets disabled,friend add menu gets enabled	All screens gets disabled,friend add menu gets enabled	Passed
G_TC_09	Check ScreenManager Class SwitchToQuestionAdd Menu method	A ScreenManager object has to keep reference of the question add menu	1.Call SwitchToFriendAddMenu method. 2.Check if the question add gets opened	None	All screens gets disabled, question add menu gets enabled	All screens gets disabled, question add menu gets enabled	Passed
G_TC_10	Check ScreenManager Class SwitchToLoginScreen method	A ScreenManager object has to keep reference of the login screen	1.Call SwitchToLoginScreen method. 2.Check if the login screen gets opened	None	All screens gets disabled, question login screen gets enabled	All screens gets disabled, question login screen gets enabled	Passed
G_TC_11	Check ScreenManager Class SwitchToSignupScreen method	A ScreenManager object has to keep reference of the sign up screen	1.Call SwitchToSignupScreen method. 2.Check if the sign up screen gets opened	None	All screens gets disabled, question sign up screen gets enabled	All screens gets disabled, question sign up screen gets enabled	Passed
G_TC_12	Check ScreenManager Class SwitchToLoadScreen method	A ScreenManager object has to keep reference of the load screen	1.Call SwitchToLoadScreen method. 2.Check if the login screen gets opened	None	All screens gets disabled, question load screen gets enabled	All screens gets disabled, question load screen gets enabled	Passed
G_TC_13	Check ScreenManager Class SwitchToClassicalQuestion-Screen method	A ScreenManager object has to keep reference of the classical question screen	1.Call SwitchToClassicalQuestion-Screen method. 2.Check if the classical question screen gets opened	None	All screens gets disabled, question classical question screen gets enabled	All screens gets disabled, question classical question screen gets enabled	Passed
G_TC_14	Check ScreenManager Class SwitchToNumericalQuestion-Screen method	A ScreenManager object has to keep reference of the numerical question screen	1.Call SwitchToNumericalQuestion-Screen method. 2.Check if the numerical question screen gets opened	None	All screens gets disabled, question numerical question screen gets enabled	All screens gets disabled, question numerical question screen gets enabled	Passed

G_TC_15	Check ScreenManager Class OpenSettingsMenu method	A ScreenManager object has to keep reference of the settings menu	1.Call OpenSettingsMenu method. 2.Check if the settings menu gets opened	None	Settings menu gets opened on the main menu	Settings menu gets opened on the main menu	Passed
G_TC_16	Check ScreenManager Class CloseSettingsMenu method	A ScreenManager object has to keep reference of the settings menu	1.Call CloseSettingsMenu method. 2.Check if the settings menu gets closed	None	Settings menu gets closed on the main menu	Settings menu gets closed on the main menu	Passed
G_TC_17	Check ScreenManager Class CloseSettingsMenu method	A ScreenManager object has to keep reference of the settings menu	1.Call CloseSettingsMenu method. 2.Check if the method returns correct result	None	Returns true if settings menu is opened or vice versa	Returns true if settings menu is opened or vice versa	Passed
G_TC_18	Check ScreenManager Class FadeScreenForOneUpdate method	A ScreenManager object has to keep reference of the canvas group	1.Call FadeScreenForOneUpdate method for one time. 2.Check if screen gets darker	None	Fades screen a bit for every call,amount can be set from inspector panel.	Fades screen a bit for every call,amount can be set from inspector panel.	Passed
G_TC_19	Check ScreenManager Class FixFadedColors method	A ScreenManager object has to keep reference of the canvas group	1.Call FixFadedColors method for one time. 2.Check if screen gets to default brightness again	None	Sets screen to default brightness once it gets called	Sets screen to default brightness once it gets called	Passed
G_TC_20	Check ScreenManager Class IsCompletelyFaded method	A ScreenManager object has to keep reference of the canvas group	1.Call IsCompletelyFaded method for one time. 2. Check if the method returns correct result	None	Returns true if screen is completely or vice versa	Returns true if screen is completely or vice versa	Passed
G_TC_21	Create LoadScreen Class and assign to LoadScreen game object	A LoadScreen object has to be created	1.Create an object of ScreenManager class 2.Assign to ScreenManager game object 3.Check if any ui element of load screen is connected or not	None	All of the ui element of load screen is connected.	All of the ui element of load screen is connected.	Passed
G_TC_22	Check ScreenManager Class UpdateLightPosition method	Light ui element has to be assigned to the script	1. Call IsCompletelyFaded method for one time. 2.Check if the position of the light has changed	None	Loading light position has to change one unit on the orbit for one update	Loading light position has to change one unit on the orbit for one update	Passed

G_TC_23	Check ScreenManager Class UpdateLightRotation method	Light ui element has to be assigned to the script	1. Call IsCompletelyFaded method for one time. 2.Check if the rotation of the light has changed	None	Loading light rotation has to change one unit on the orbit for one update	Loading light rotation has to change one unit on the orbit for one update	Passed
G_TC_24	Check ScreenManager Class UpdateLoadingTest method	Loading text ui element has to be assigned to the script	1. Call UpdateLoadingTest method for one time. 2.Check if loading text has changed	None	Loading text changes as : Loading. Loading.. Loading...	Loading text changes as : Loading. Loading.. Loading...	Passed
G_TC_25	Create MainMenu Class and assign to MainMenu game object	A MainMenu object has to be created	1.Create an object of MainMenu class 2.Assign to MainMenu game object 3.Check if any ui element of main menu is connected or not	None	All of the ui element of main menu is connected.	All of the ui element of main menu is connected.	Passed
G_TC_26	Check MainMenu Class UpdatEffectsOfSwitching-NewGame method	SpaceShip ui element has to be attached and ScreenManager class has to exist in game	1. Call UpdatEffectsOfSwitching-NewGame method for one time. 2.Check if SpaceShip position and screen brightness changes	None	Spaceship moves one unit to the left,screen fades for one unit,after spaceship goes,screen gets bright back	Spaceship moves one unit to the left,screen fades for one unit,after spaceship goes,screen gets bright back	Passed
G_TC_27	Check MainMenu Class ReinitializeSpaceShipPosition method	SpaceShip ui element has to be attached	1.Call ReinitializeSpaceShipPosition 2.Check if spaceship has come to it's initial position	None	Spaceship gets located to it's initial position at the main menu again	Spaceship gets located to it's initial position at the main menu again	Passed
G_TC_28	Check MainMenu Class SaveSpaceShipInitial-PositionValues method	SpaceShip ui element has to be attached	1.Call ReinitializeSpaceShipPosition 2.Check if spaceship's initial position is saved to corresponding members	None	Spaceship's initial position gets saved to corresponding members	Spaceship's initial position gets saved to corresponding members	Passed
G_TC_29	Check MainMenu Class DeclareNewGameProgress-HasStarted method	NewGame button gets clicked	1.Call DeclareNewGameProgress - HasStarted method 2.Save information of the if game has started	None	Information of if game has started gets saved to corresponding members	Information of if game has started gets saved to corresponding members	Passed

G_TC_30	Check MainMenu Class SetVerticalPositionOfThe-ShipForOneUpdate method	SpaceShip ui element has to be attached	1.Call SetVerticalPositionOfThe-ShipForOneUpdate method 2.Check if spaceship has moved in vertical direction	None	Spaceship moves on vertical axis for one unit and makes sinusodial move	Spaceship moves on vertical axis for one unit and makes sinusodial move	Passed
G_TC_31	Check MainMenu Class MoveSpaceShipToRight-ForOneUpdate method	SpaceShip ui element has to be attached	1.Call SetVerticalPositionOfThe-MoveSpaceShipToRight-ForOneUpdate method 2.Check if spaceship has moved to right direction	None	Spaceship moves to right for one unit	Spaceship moves to right for one unit	Passed
G_TC_32	Check MainMenu Class NormalizeAffectsOf-SwitchingNewGameProgress method	SpaceShip ui element has to be attached and ScreenManager class has to exist in game	1.Call SetVerticalPositionOfThe-NormalizeAffectsOf-SwitchingNewGameProgress method 2.Check if affects of new game progress gets reseted	None	Information of if game has started,spaceship position and colors gets to initial values.	Information of if game has started,spaceship position and colors gets to initial values.	Passed
G_TC_33	Create GameModesMenu Class and assign to GameModesMenu game object	A GameModesMenu object has to be created	1.Create an object of MainMenu class 2.Assign to GameModesMenu game object 3.Check if any ui element of main menu is connected or not	None	All of the ui element of game modes menu is connected.	All of the ui element of game modes menu is connected.	Passed
G_TC_34	Check GameModesMenu Class BackToMainMenu method	ScreenManager object and it's script has to exist in the game	1.Call BackToMainMenu method 2.Check if screen switches to main menu.	None	Screen switches to main menu.	Screen switches to main menu.	Passed
G_TC_35	Check StartBattleRoyale Class BackToMainMenu method	ScreenManager / GameManager object and those script has to exist in the game	1.Call StartBattleRoyale method 2.Check if switches to load screen and starts the game	None	Switches to load screen,starts the game when server connection is available	Switches to load screen,starts the game when server connection is available	Passed
G_TC_36	Check StartOneVsOne Class BackToMainMenu method	ScreenManager object and it's script has to exist in the game	1.Call StartOneVsOne method 2.Check if switches to load screen and starts the game	None	Switches to load screen,starts the game when server connection is available	Switches to load screen,starts the game when server connection is available	Passed
G_TC_37	Create LogInScreen Class and assign to LogInScreen game object	ScreenManager / UserInfoManager objects and those script has to exist in the game	1.Create an object of LogInScreen class 2.Assign to LogInScreen game object 3.Check if user can log in	None	All of the ui element of login screen is connected.	All of the ui element of login screen is connected.	Passed

G_TC_38	Check LogInScreen class button add-listeners	ScreenManager / UserInfoManager objects and those script has to exist in the game	1.Press any of the buttons of the LoginScreen 2.Check if user logins	None	Uses UserInfoManager class to handle log-in	Uses UserInfoManager class to handle log-in	Passed
G_TC_39	Create SignUpScreen Class and assign to LogInScreen game object	ScreenManager / UserInfoManager objects and those script has to exist in the game	1.Create an object of SignUpScreen class 2.Assign to SignUpScreen game object 3.Check if user can sign up	None	All of the ui element of signup screen is connected.	All of the ui element of signup screen is connected.	Passed
G_TC_40	Check LogInScreen class button add-listeners	ScreenManager / UserInfoManager objects and those script has to exist in the game	1.Press any of the buttons of the LoginScreen 2.Check if user signs up	None	Uses UserInfoManager class to handle signup	Uses UserInfoManager class to handle signup	Passed
G_TC_41	Create ClassicalQuestionScreen Class and assign to ClassicalQuestionScreen game object	A ClassicalQuestionScreen object has to be created	1.Create an object of ClassicalQuestionScreen class 2.Assign to ClassicalQuestionScreen game object	None	All of the ui element of classical question screen screen is connected.	All of the ui element of classical question screen screen is connected.	Passed
G_TC_42	Check ClassicalQuestionScreen Class GetUserChoice method	None	1.Call StartOneVsOne method 2.Check if it returns user choice string	None	Returns user choice as string	Returns user choice as string	Passed
G_TC_43	Check ClassicalQuestionScreen Class ChooseA/ ChooseB/ChooseC/ ChooseD methods	Choice buttons has to be attached	1.Call ChooseA/ChooseB/ ChooseC/ChooseD methods 2. ClassicalQuestionScreen Class MarkAsChoosen method gets called	None	Call MarkChoiceAsChoosen method of ClassicalQuestion Screen with corresponding choice as input	Call MarkChoiceAsChoosen method of ClassicalQuestion Screen with corresponding choice as input	Passed
G_TC_44	Check ClassicalQuestionScreen Class MarkAsChoosen method	Choice buttons has to be attached, ChooseA/ ChooseB/ ChooseC/ChooseD methods of the ClassicalQuestionScreen	1.Gets choice as string input 2.Marks user choice by changing color 3.Saver user choice as string within class.	String = userChoice	Mark choosen choice with navy blue color	Mark choosen choice with navy blue color	Passed

G_TC_45	Check ClassicalQuestionScreen Class ShowCorrectAnswer method	Choice buttons has to be attached	1.Call ShowCorrectAnswer method 2.Mark correct answer	None	Mark correct choice with green color	Mark correct choice with green color	Passed
G_TC_46	Check ClassicalQuestionScreen Class SetInteractableStatusOfAllButtons method	Choice buttons has to be attached	1.Call SetInteractableStatusOfAllButtons method 2.Lock/unlock choice buttons	None	Lock/unlock choice buttons according to input	Lock/unlock choice buttons according to input	Passed
G_TC_47	Create NumericalQuestionScreen Class and assign to ClassicalQuestionScreen game object	A NumericalQuestionScreen object has to be created	1.Create an object of NumericalQuestionScreen class 2.Assign to NumericalQuestionScreen game object	None	All of the ui element of numerical question screen screen is connected.	All of the ui element of classical numerical screen screen is connected.	Passed
G_TC_48	Check ClassicalQuestionScreen Class UpdateUserAnswerByAddingDigitAtTheEnd method	Digit buttons has to be attached	1.Call UpdateUserAnswerByAddingDigitAtTheEnd method 2.Check if userAnswer text changes	None	Given integer digit gets added at the end of userAnswer text	Given integer digit gets added at the end of userAnswer text	Passed
G_TC_49	Check ClassicalQuestionScreen Class EraseOneDigit method	Delete button has to be attached	1.Call EraseOneDigit method 2.Check if userAnswer text changes	None	Last entered digit gets erased from the end of userAnswer text	Last entered digit gets erased from the end of userAnswer text	Passed
G_TC_50	Check ClassicalQuestionScreen Class UserSendsAnswer method	Send answer button has to be attached	1.Call UserSendsAnswer method 2.Check if correctAnswer text changes 3.Check if digit buttons gets locked	None	Correct answer text gets updated and digit buttons gets locked	Correct answer text gets updated and digit buttons gets locked	Passed
G_TC_51	Check ClassicalQuestionScreen Class SetCorrectAnswer method	UpdateCorrectAnswer method must be called	1. Call UserSendsAnswer method with string input 2.Check if correctAnswer has changed	String = correctAnswerString	correctAnswer long gets assigned from input string	correctAnswer long gets assigned from input string	Passed
G_TC_52	Check ClassicalQuestionScreen Class ShowCorrectAnswer method	None	1.Call ShowCorrectAnswer method 2.Check if correct answer appears	None	Correct answer appears on the screen	Correct answer appears on the screen	Passed

G_TC_53	Check ClassicalQuestionScreen Class UpdateQuestion method	None	1.Call UpdateQuestion method 2.Check if buttons are unlocked 3.Check if correct answer part is dissapeared 4.Check if question is updated	None	Buttons gets unlocked,correct answer button gets dissapeared and question gets updated	Buttons gets unlocked,correct answer button gets dissapeared and question gets updated	Passed
G_TC_54	Create GamePlayManager Class and assign to GamePlayManager game object	ScreenManager, ClassicalQuestionScreen, NumericalQuestionScreen has to exist	1.Create GamePlayManager object 2.Assign to GamePlayManager game object	None	ScreenManager, ClassicalQuestionScreen, NumericalQuestionScreen are interactable from this class.	ScreenManager, ClassicalQuestionScreen, NumericalQuestionScreen are interactable from this class.	Passed
G_TC_55	Check GamePlayManager class getter methods	Server connection has to be assured	1.Call one of the getter functions 2.Check if returned value is correct	None	Get corresponding data from the server,return it	Get corresponding data from the server,return it	Passed
G_TC_56	Check GamePlayManager class UpdateQuestion method	Server connection has to be assured, ClassicalQuestionScreen, NumericalQuestionScreen has to exist	1.Call UpdateQuestion function with question type 2.Check if screen is switched to question screen with correct question	String = questionType	Get question from the server,set question screen with it	Get question from the server,set question screen with it	Passed
G_TC_57	Check GamePlayManager class sender methods	Server connection has to be assured	1.Call one of the sender functions 2.Check if current corresponding data to the server has sent	String = data	Send data to the server to make interactive gameplay	Send data to the server to make interactive gameplay	Not checked
G_TC_58	Create UserInfoManager Class and assign to UserInfoManager game object	Server connection has to be assured	1.Create UserInfoManager object 2.Assign to UserInfoManager game object	None	Username information are interactable between server and this class	Username information are interactable between server and this class	Passed
G_TC_59	Check UserInfoManager class getter functions	None	1.Call one of the getter functions 2.Return user data	String = user data	User data gets returned	User data gets returned	Passed
G_TC_60	Check UserInfoManager class sender functions	Server connection has to be assured	1.Call one of the sender functions 2.Send user data to the server	None	User data gets sent to the server	User data gets sent to the server	Passed

G_TC_61	ServiceCommunication Get method	Given url has to be valid	1. Call Get method 2. Pass a url to the method	String url = https://url/	A json string taken from the server	A json string taken from the server	Passed
G_TC_62	ServiceCommunication Post method	Given url shoul be valid and given data should be in json format	1.Call Post Method 2.Pass url and data to the method	String url = https://url/ String data = {jsondata}	A json string taken from the server	A json string taken from the server	Passed