Final 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

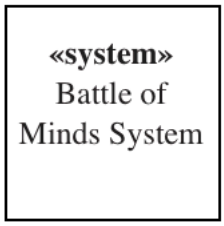
Berry Hibiscus Team

07/01/2023

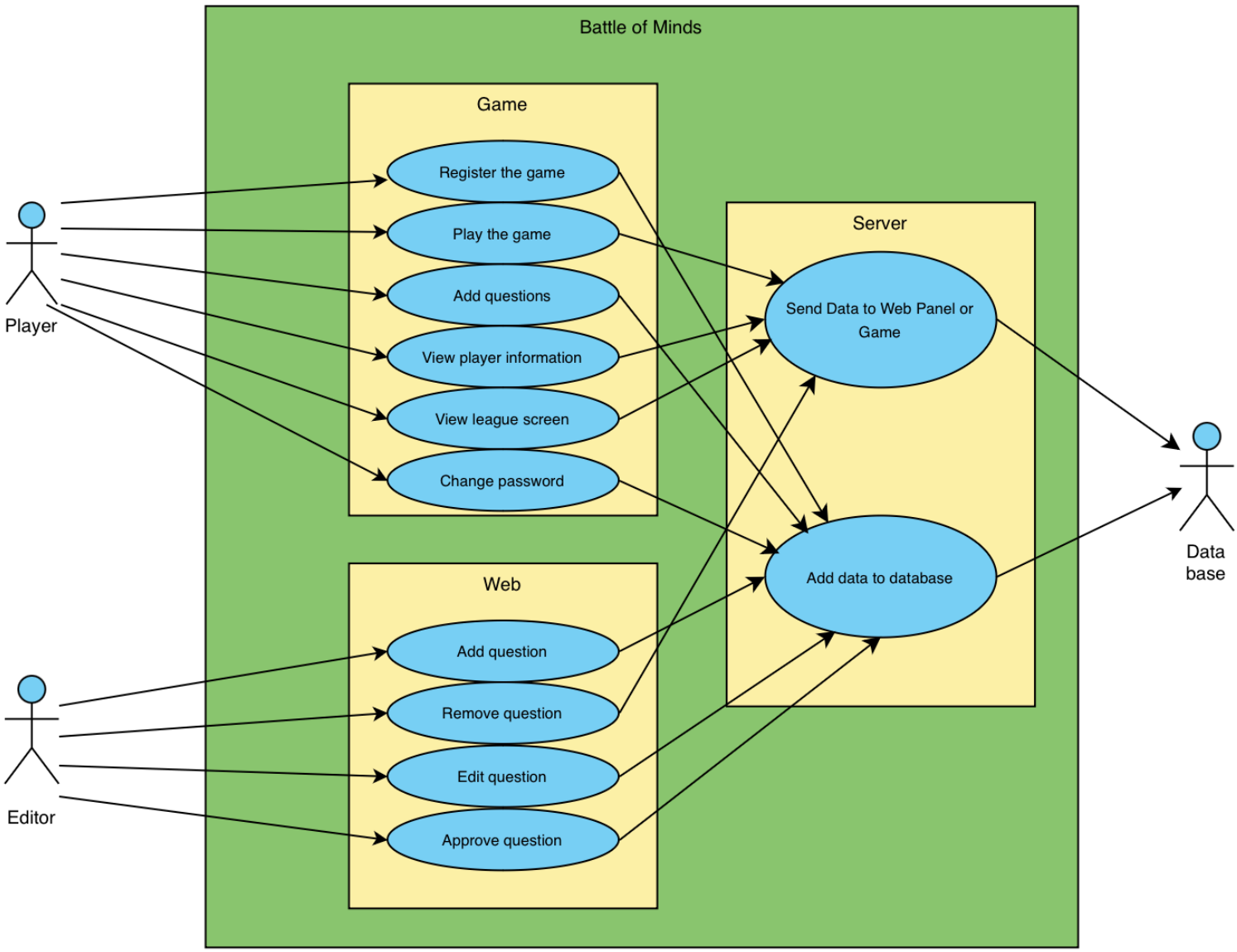
Table of Contents

1.UML Diagrams

1.1 Context Model



1.2 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 the game.

Editor:

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

System	*Battle of Minds-Game*
Use Case	*Register the game*
Actors	*Players*
Data	*Players need to register to play the game. In order to register, they must enter their email address, password and a preferred username.*
Stimulus	*Players' request to register the game*
Response	*If the entered email address, password and username are valid, the received data is sent to the server and processed into the database and player is directed to main screen.*
Comments	*Players who fill the parameters incorrectly should review their data and try to register again. After the registration has been completed, the user can log in by opening the login page.*

System	*Battle of Minds-Game*
Use Case	*Play the game*
Actors	*Players*
Data	*The player's answer*
Stimulus	*Player's request to send the answer*
Response	*It is checked whether the answer of the question fetched from the database is the same as the user's answer. If the answer is correct, the new question appears. If the answer is wrong, the game ends.*
Comments	*The player has certain amount of time to be able to answer each question.If the player fails to answer the question within this time, the player loses the game.*

System	*Battle of Minds-Game*
Use Case	*Add questions*
Actors	*Players*
Data	*The question and its answer which the player wants to add.*
Stimulus	*Players' request to add question to the game*
Response	*If all parts of question are filled, then the given question is accepted and sent to database to be reviewed by editors.*
Comments	*The written question cannot appear after the player sent it. Before displaying the question, it should be reviewed by editors.*

System	*Battle of Minds-Game*
Use Case	*View player information*
Actors	*Players*
Data	*Player's game statistics.*
Stimulus	*Players' request to display the profile screen*
Response	*The player's data is fetched from the database. The profile page with the player's information opens.*
Comments	-

System	*Battle of Minds-Game*
Use Case	*View league screen*
Actors	*Players*
Data	*League statistics.*
Stimulus	*Players' request to display the league screen*
Response	*The league's data is fetched from the database. The league screen opens.*
Comments	-

System	*Battle of Minds-Game*
Use Case	*Change password*
Actors	*Players*
Data	*The new password given by players*
Stimulus	*Players' request to change password*
Response	*The new password is sent to the database.*
Comments	-

System	*Battle of Minds-Web*
Use Case	*Add question*
Actors	*Editors*
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	*Editor's request to add a question*
Response	*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. Editor receives an alert that the problem has been added.*
Comments	*Only editors have permission to write questions from the web panel. Players cannot write questions from the panel.*

System	*Battle of Minds-Web*
Use Case	*Remove question*
Actors	*Editors*
Data	*The editors choose the question they want to remove from the list of questions.*
Stimulus	*Editors' request to remove a question*
Response	*The selected question is forwarded to the server and removed from the database and it is removed from the web interface.*
Comments	*Only editors have permission to remove questions from the web panel. Players cannot remove questions from the panel.*

System	*Battle of Minds-Web*
Use Case	*Edit question*
Actors	*Editors*
Data	*Editors select the question they want to edit from the list of questions .*
Stimulus	*Editors' request to edit a question*
Response	*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. The alert is received that your question editing process has been successfully performed.*
Comments	*Only editors have permission to edit questions from the web panel. Players cannot edit questions from the panel.*

System	*Battle of Minds-Web*
Use Case	*Approve question*
Actors	*Editors*
Data	*Editors select the question they want to approve from the list of questions .*
Stimulus	*Editors' request to approve a question*
Response	*The selected question is approved and sent to database as approved.*
Comments	*Only editors have permission to approve questions from the web panel. Players cannot approve questions from the panel.*

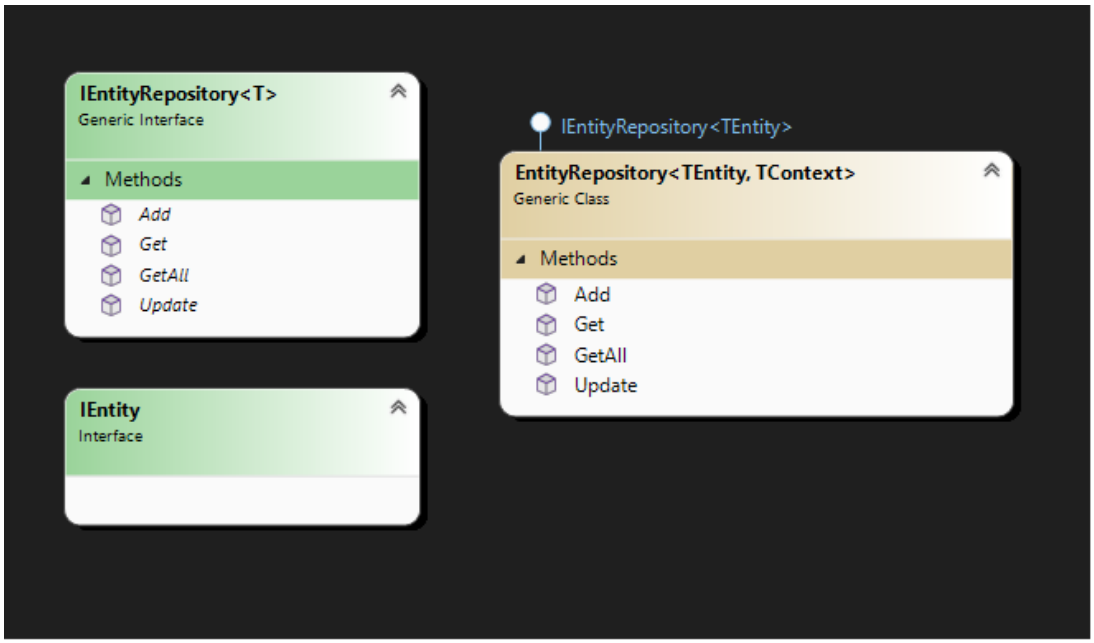
System	*Battle of Minds-Server*
Use Case	*Send Data to Web Panel or Game*
Actors	-
Data	*Information requested from game or web.*
Stimulus	*Web panel or game's request to server*
Response	*The requested data is transferred from the database to the game or the web. The data is seen by the editors on the web or by the players in the game.*
Comments	*If the requested data is not found in the database, it sends data indicating that it is not found.*

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

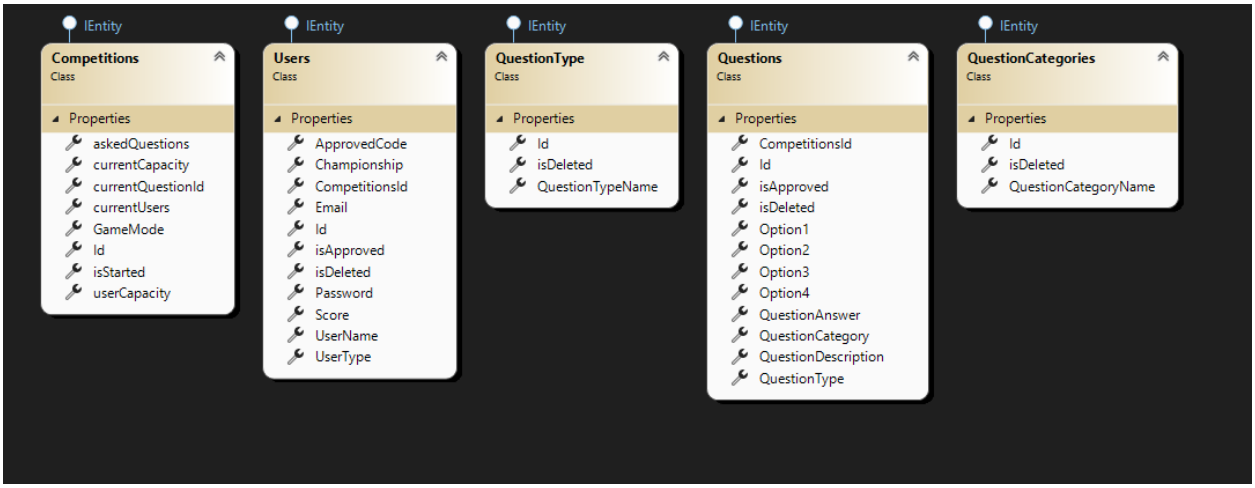
1.3 Class Diagrams

1.3.1 Class Diagrams for Server

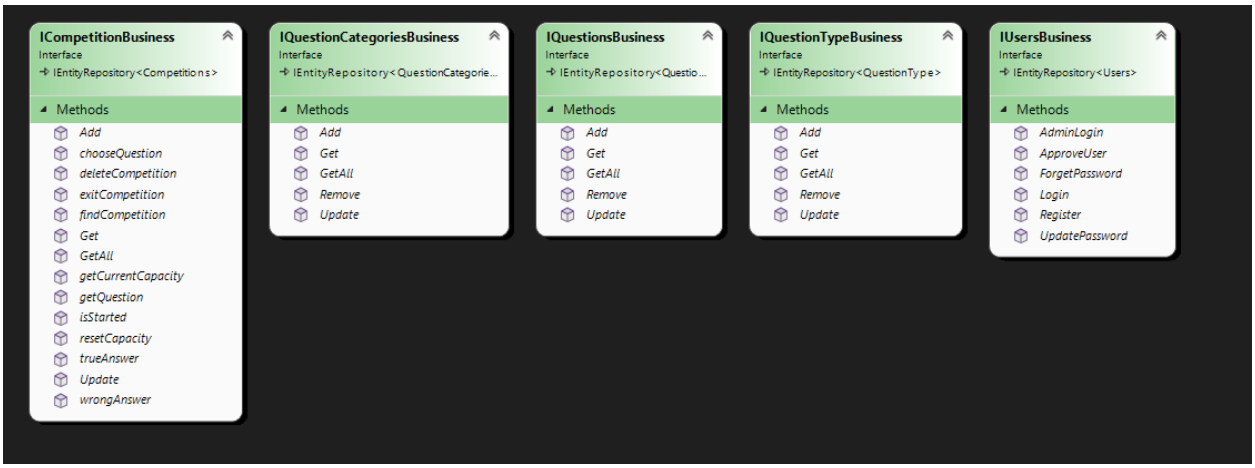
BattleOfMinds.Core

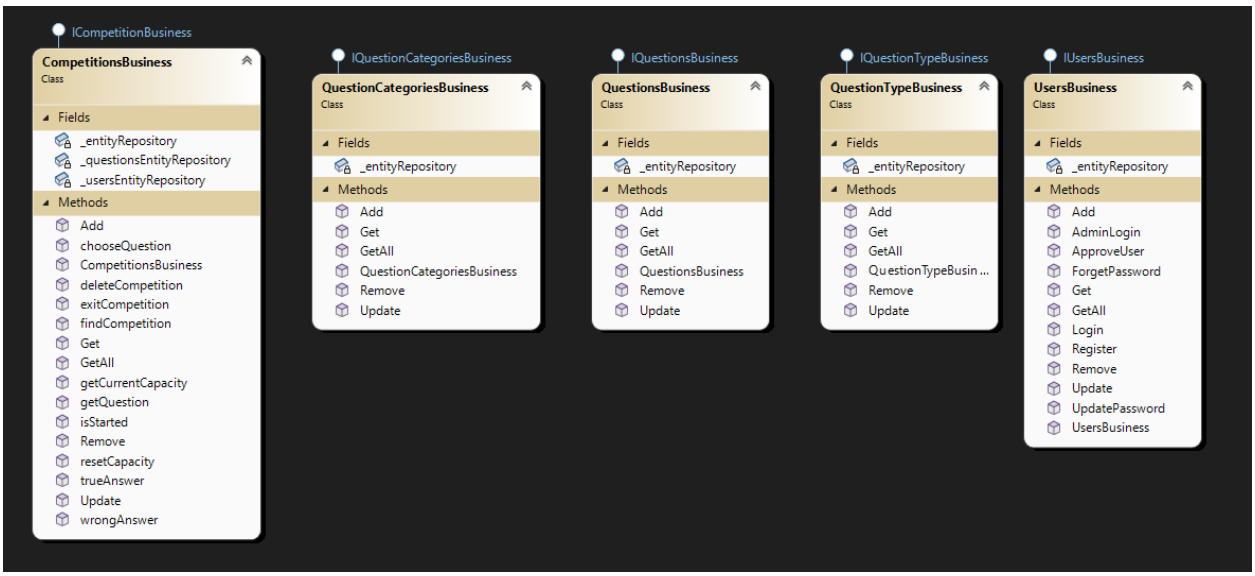


BattleOfMinds.Models

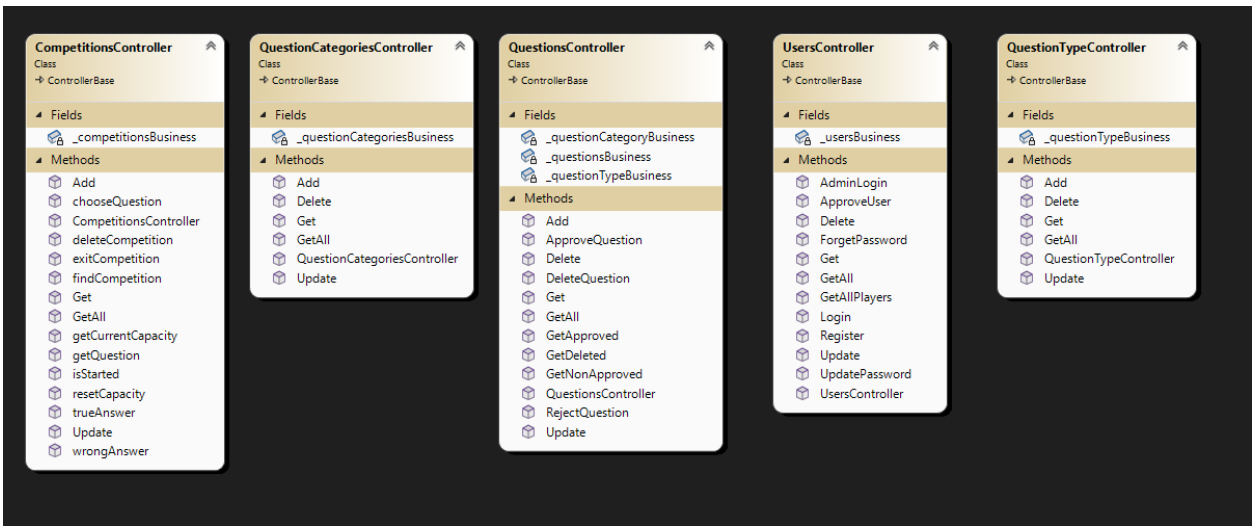


BattleOfMinds.API Business

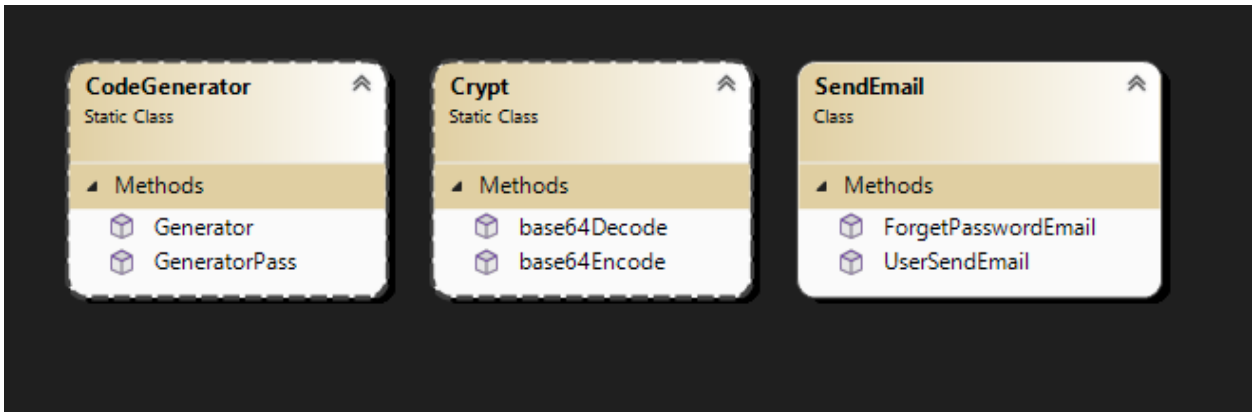




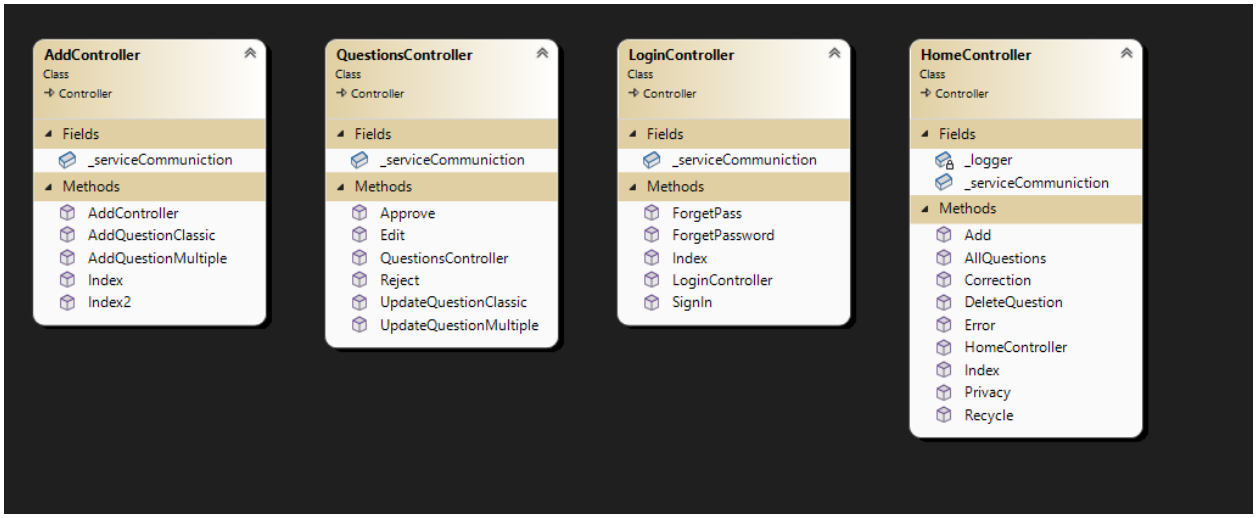
BattleOfMinds.API Controllers



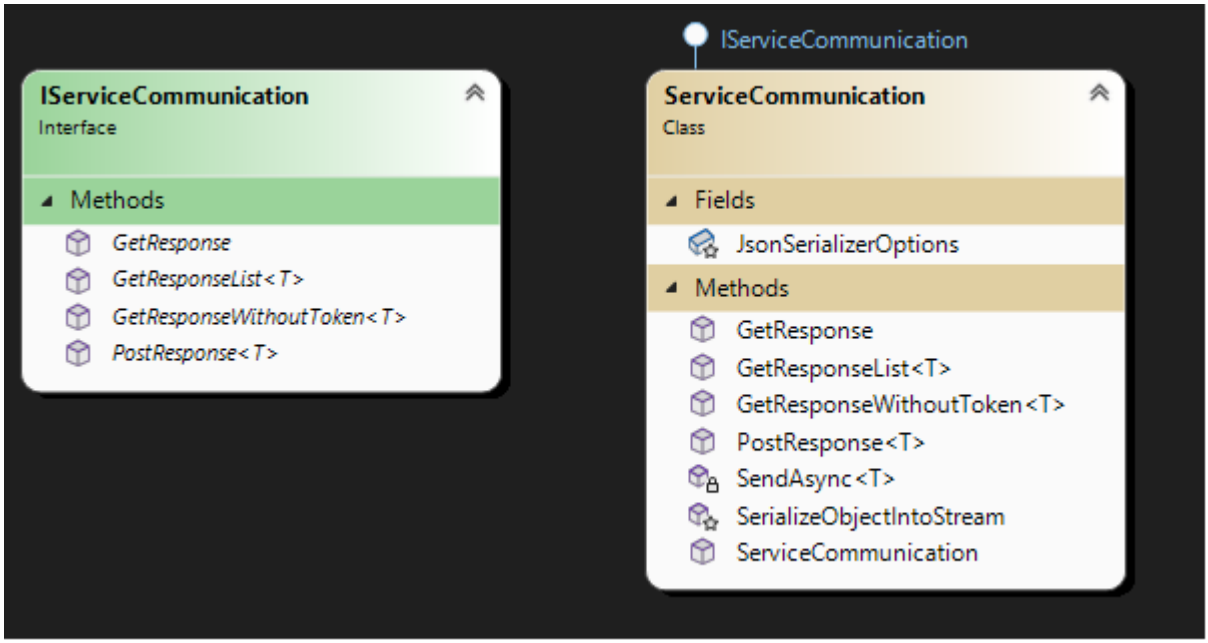
BattleOfMinds.API Helpers



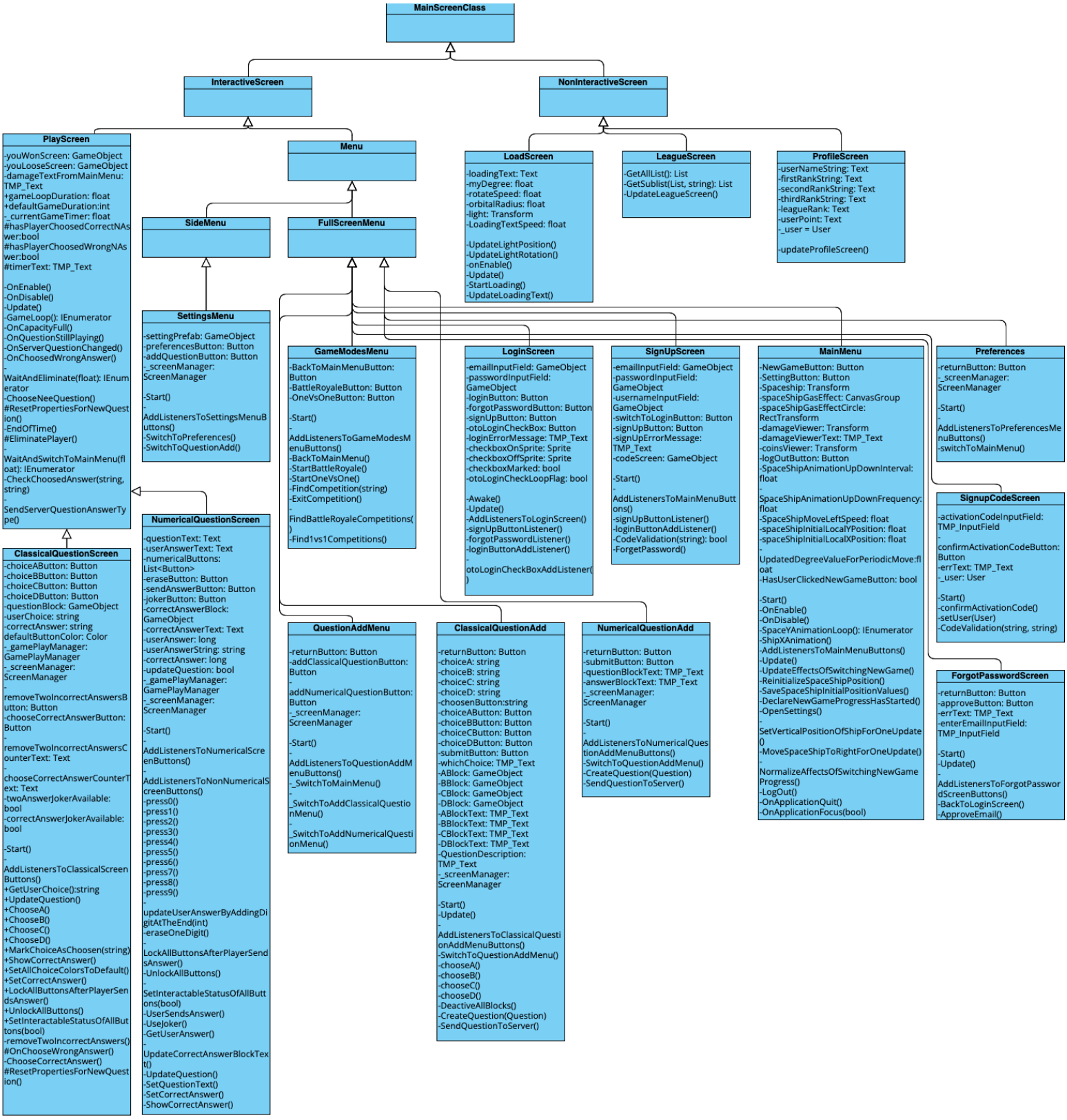
BattleOfMinds.MVC Controller

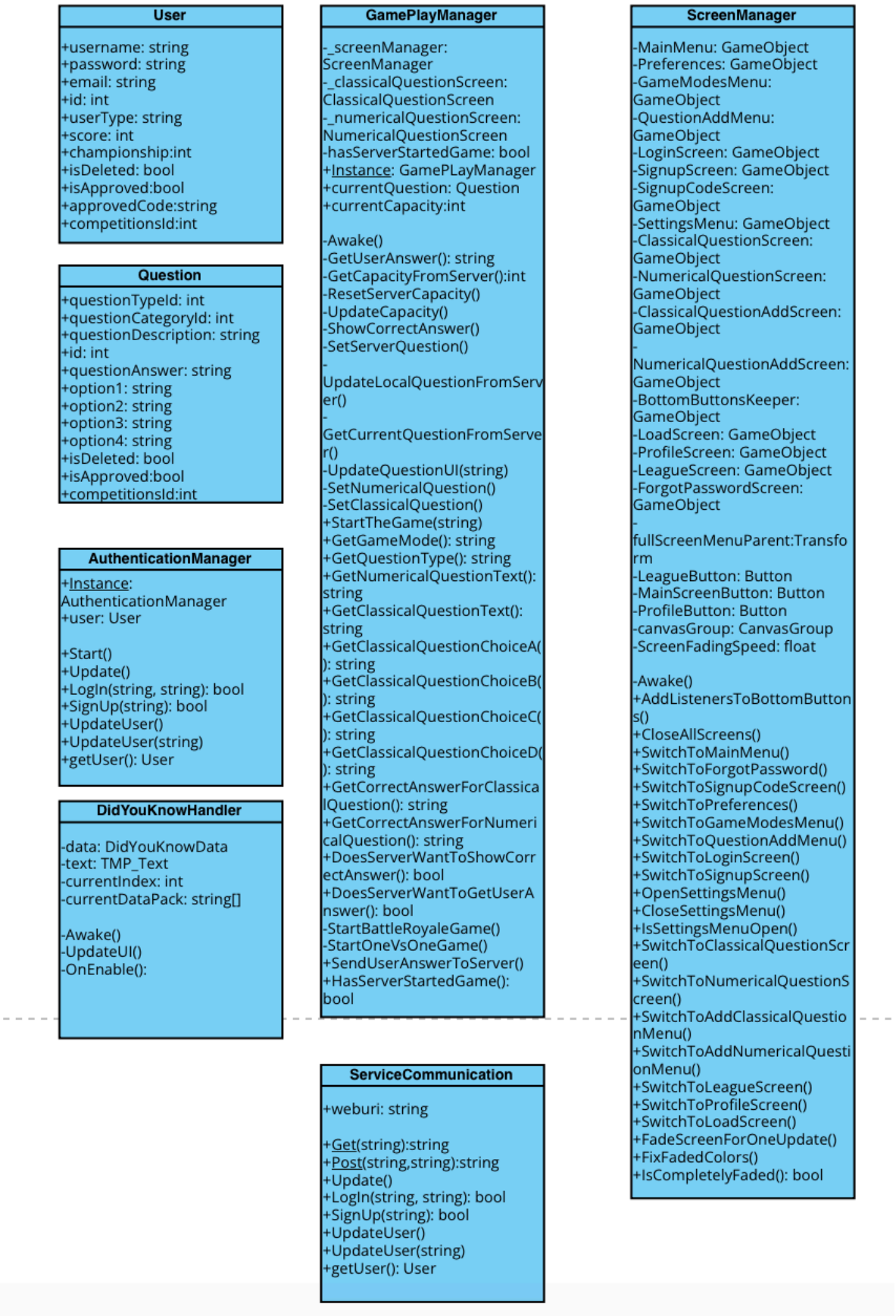


BattleOfMinds.MVC Communication

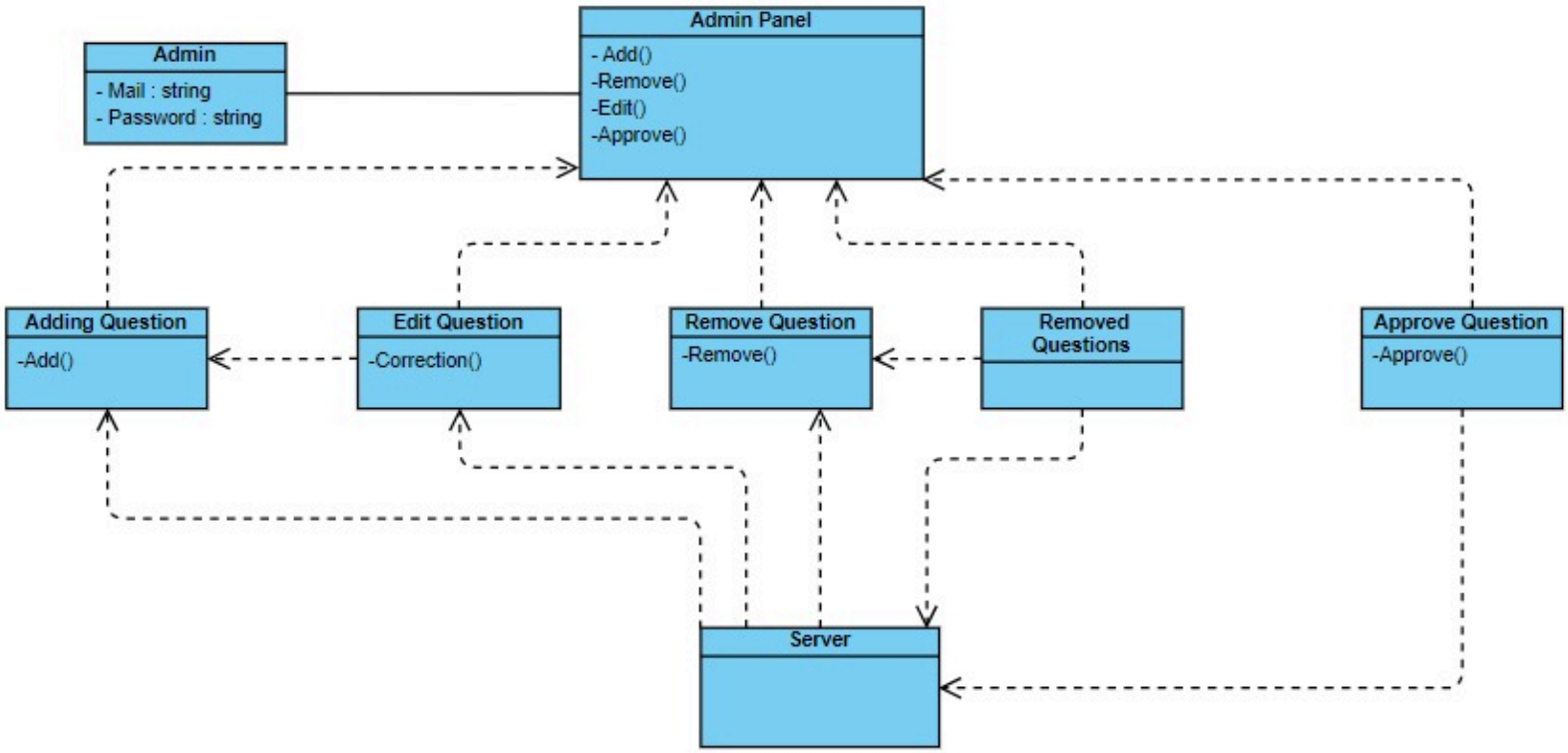


1.3.2 Class Diagrams for Game

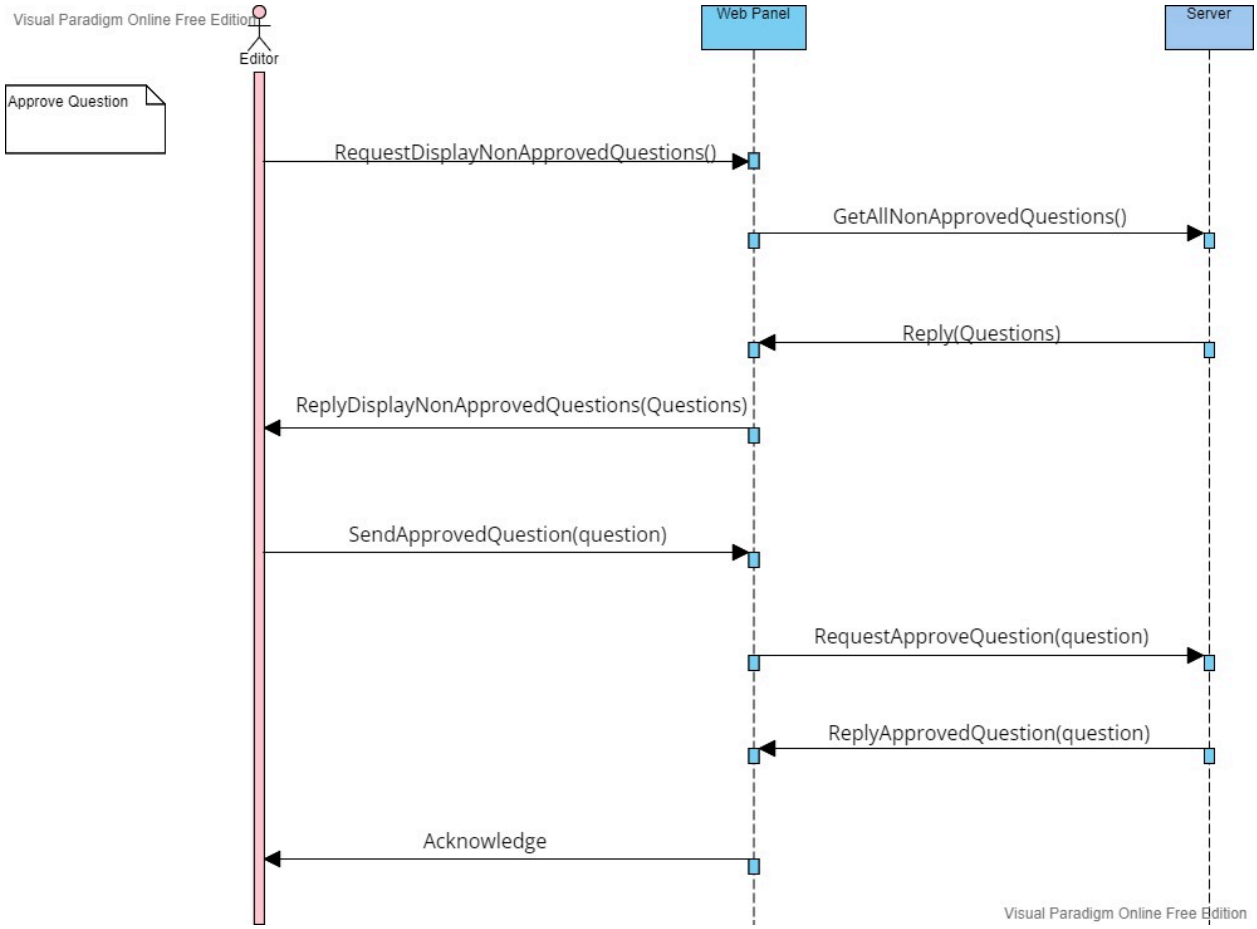


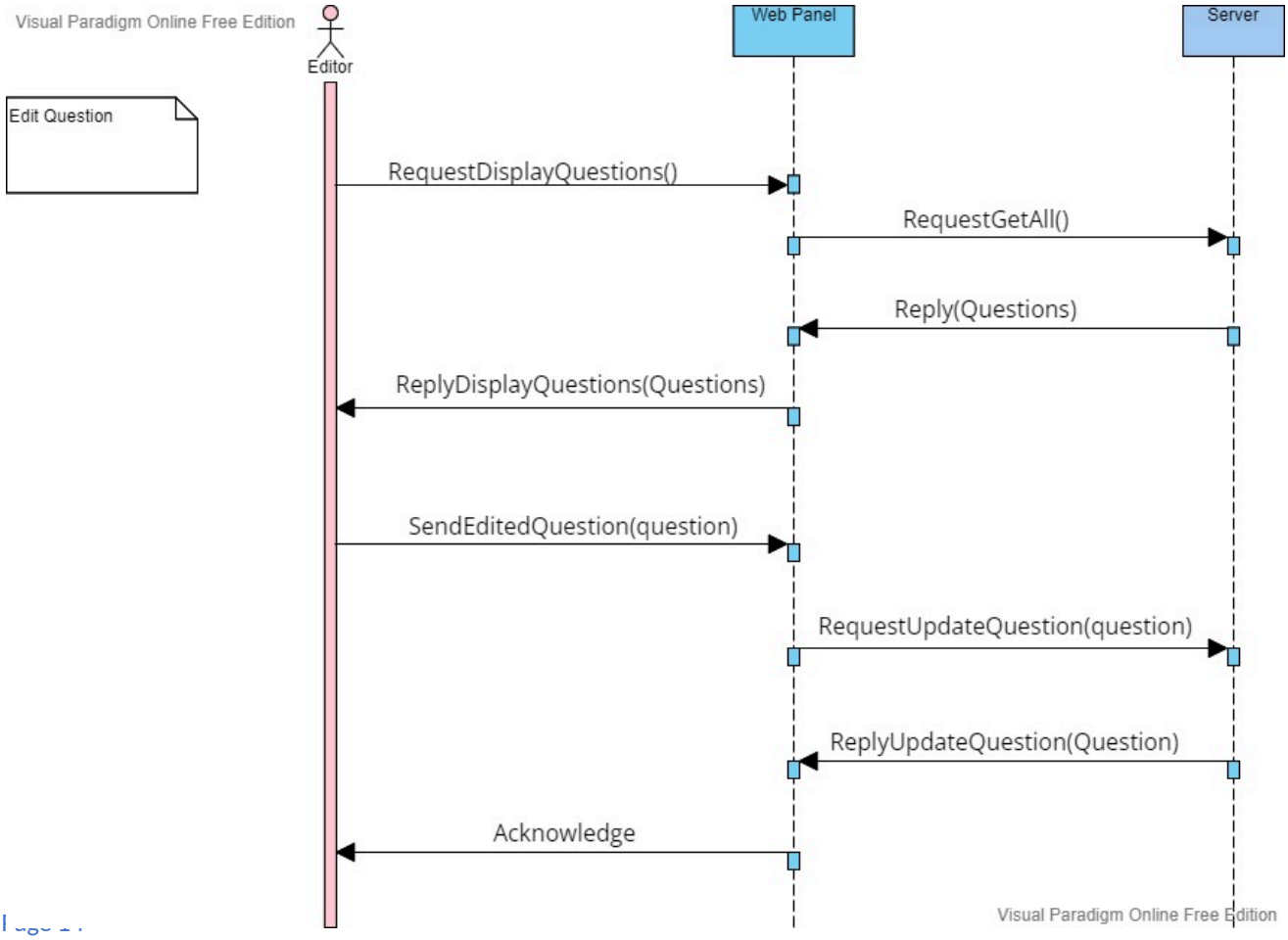

User
+username: string
+password: string
+email: string
+id: int
+userType: string
+score: int
+championship:int
+isDeleted: bool
+isApproved:bool
+approvedCode:string
+competitionsId:int
Question
+questionTypeId: int
+questionCategoryId: int
+questionDescription: string
+id: int
+questionAnswer: string
+option1: string
+option2: string
+option3: string
+option4: string
+isDeleted: bool
+isApproved:bool
+competitionsId:int
AuthenticationManager
+Instance: AuthenticationManager
+user: User
+Start()
+Update()
+LogIn(string, string): bool
+SignUp(string): bool
+UpdateUser()
+UpdateUser(string)
+getUser(): User
DidYouKnowHandler
-data: DidYouKnowData
-text: TMP_Text
-currentIndex: int
-currentDataPack: string[]
-Awake()
-UpdateUI()
-OnEnable():
GamePlayManager
-_screenManager: ScreenManager
-_classicalQuestionScreen: ClassicalQuestionScreen
-_numericalQuestionScreen: NumericalQuestionScreen
-hasServerStartedGame: bool
+Instance: GamePLayManager
+currentQuestion: Question
+currentCapacity:int
-Awake()
-GetUserAnswer(): string
-GetCapacityFromServer():int
-ResetServerCapacity()
-UpdateCapacity()
-ShowCorrectAnswer()
-SetServerQuestion()
-UpdateLocalQuestionFromServer()
-GetCurrentQuestionFromServer()
-UpdateQuestionUI(string)
-SetNumericalQuestion()
-SetClassicalQuestion()
+StartTheGame(string)
+GetGameMode(): string
+GetQuestionType(): string
+GetNumericalQuestionText(): string
+GetClassicalQuestionText(): string
+GetClassicalQuestionChoiceA(): string
+GetClassicalQuestionChoiceB(): string
+GetClassicalQuestionChoiceC(): string
+GetClassicalQuestionChoiceD(): string
+GetCorrectAnswerForClassicalQuestion(): string
+GetCorrectAnswerForNumericalQuestion(): string
+DoesServerWantToShowCorrectAnswer(): bool
+DoesServerWantToGetUserAnswer(): bool
-StartBattleRoyaleGame()
-StartOneVsOneGame()
+SendUserAnswerToServer()
+HasServerStartedGame(): bool
ScreenManager
-MainMenu: GameObject
-Preferences: GameObject
-GameModesMenu: GameObject
-QuestionAddMenu: GameObject
-LoginScreen: GameObject
-SignupScreen: GameObject
-SignupCodeScreen: GameObject
-SettingsMenu: GameObject
-ClassicalQuestionScreen: GameObject
-NumericalQuestionScreen: GameObject
-ClassicalQuestionAddScreen: GameObject
-NumericalQuestionAddScreen: GameObject
-BottomButtonsKeeper: GameObject
-LoadScreen: GameObject
-ProfileScreen: GameObject
-LeagueScreen: GameObject
-ForgotPasswordScreen: GameObject
-fullScreenMenuParent:Transform
-LeagueButton: Button
-MainScreenButton: Button
-ProfileButton: Button
-canvasGroup: CanvasGroup
-ScreenFadingSpeed: float
-Awake()
+AddListenersToBottomButtons()
+CloseAllScreens()
+SwitchToMainMenu()
+SwitchToForgotPassword()
+SwitchToSignupCodeScreen()
+SwitchToPreferences()
+SwitchToGameModesMenu()
+SwitchToQuestionAddMenu()
+SwitchToLoginScreen()
+SwitchToSignupScreen()
+OpenSettingsMenu()
+CloseSettingsMenu()
+IsSettingsMenuOpen()
+SwitchToClassicalQuestionScreen()
+SwitchToNumericalQuestionScreen()
+SwitchToAddClassicalQuestionMenu()
+SwitchToAddNumericalQuestionMenu()
+SwitchToLeagueScreen()
+SwitchToProfileScreen()
+SwitchToLoadScreen()
+FadeScreenForOneUpdate()
+FixFadedColors()
+IsCompletelyFaded(): bool
ServiceCommunication
+weburi: string
+Get(string):string
+Post(string,string):string
+Update()
+LogIn(string, string): bool
+SignUp(string): bool
+UpdateUser()
+UpdateUser(string)
+getUser(): User

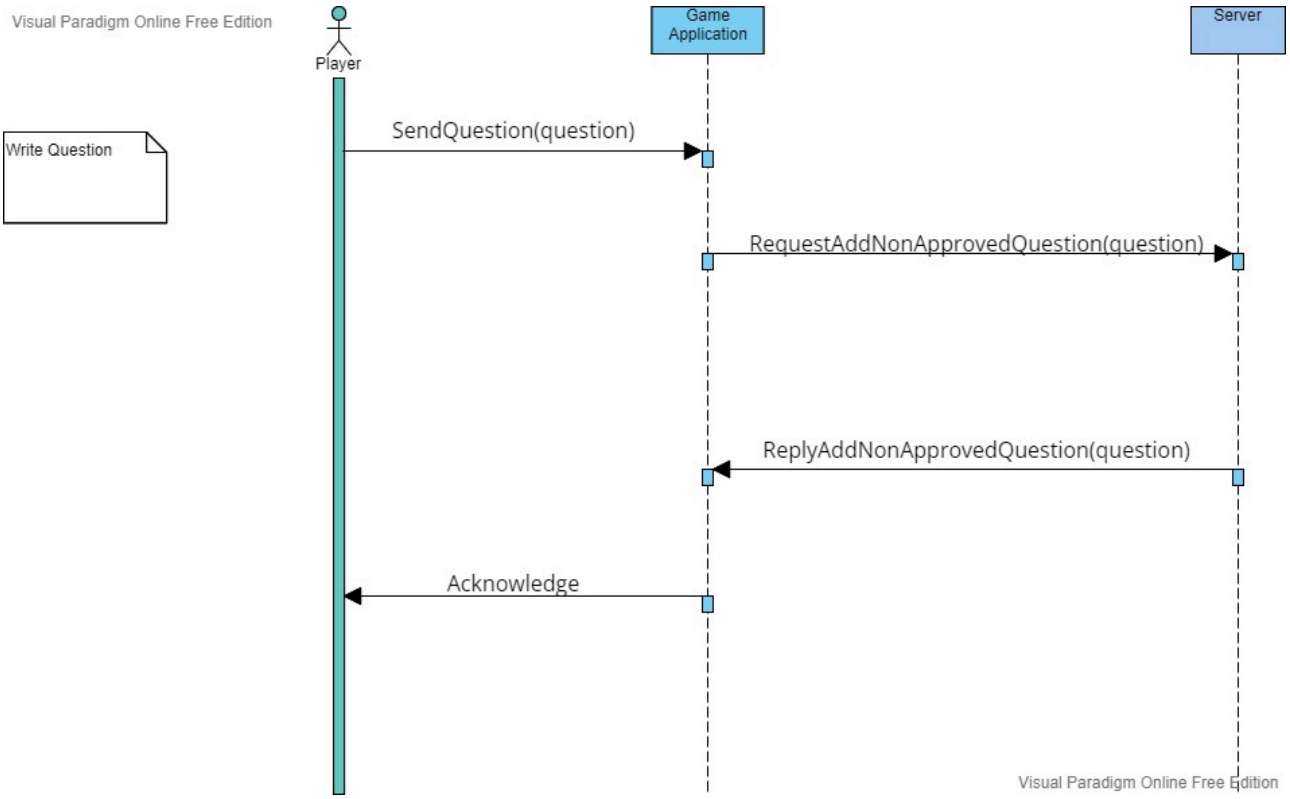
1.3.3 Class Diagrams for Web

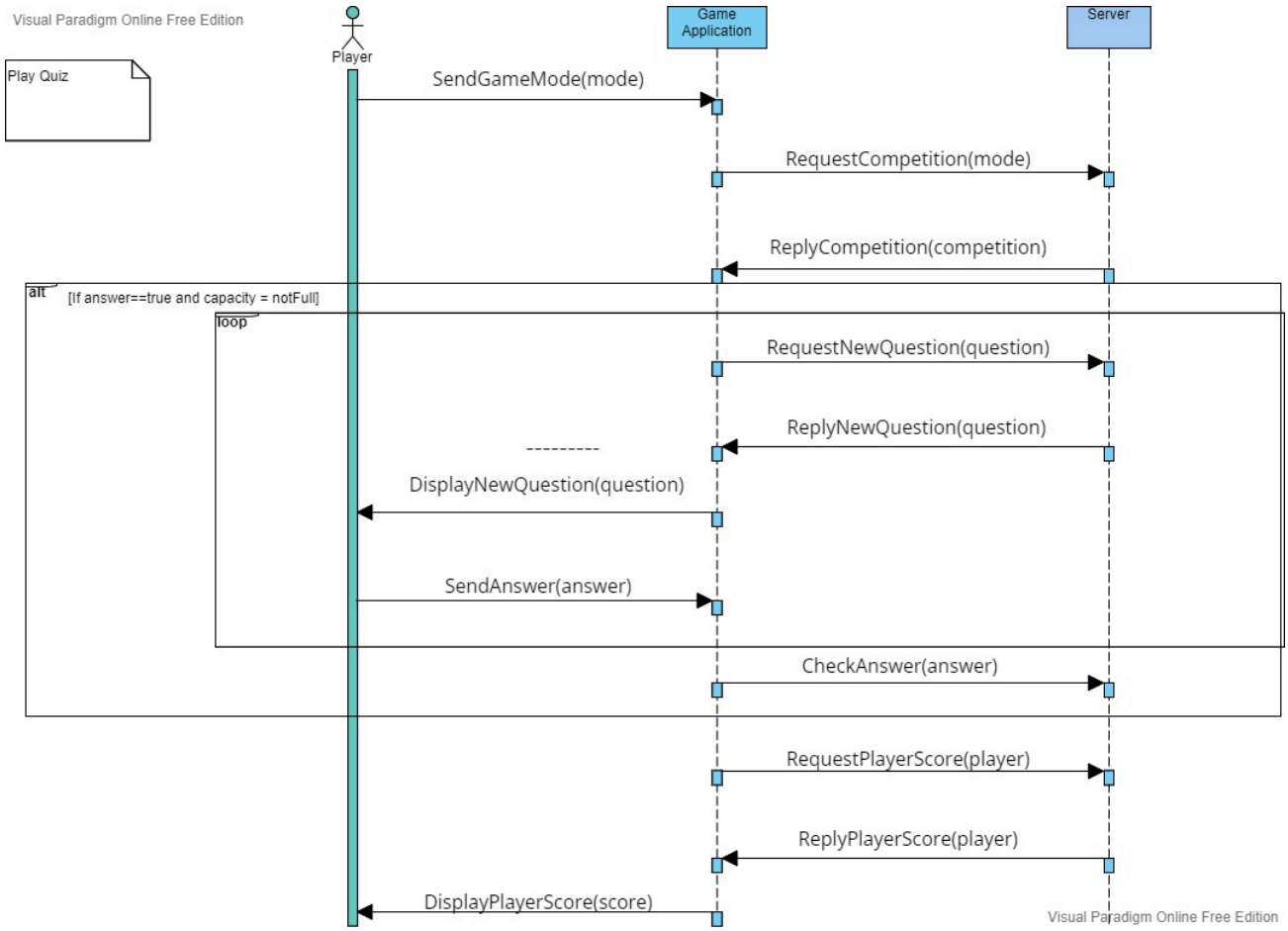


1.4 Sequence Diagrams for All the Use Cases (5 Major Use Cases)

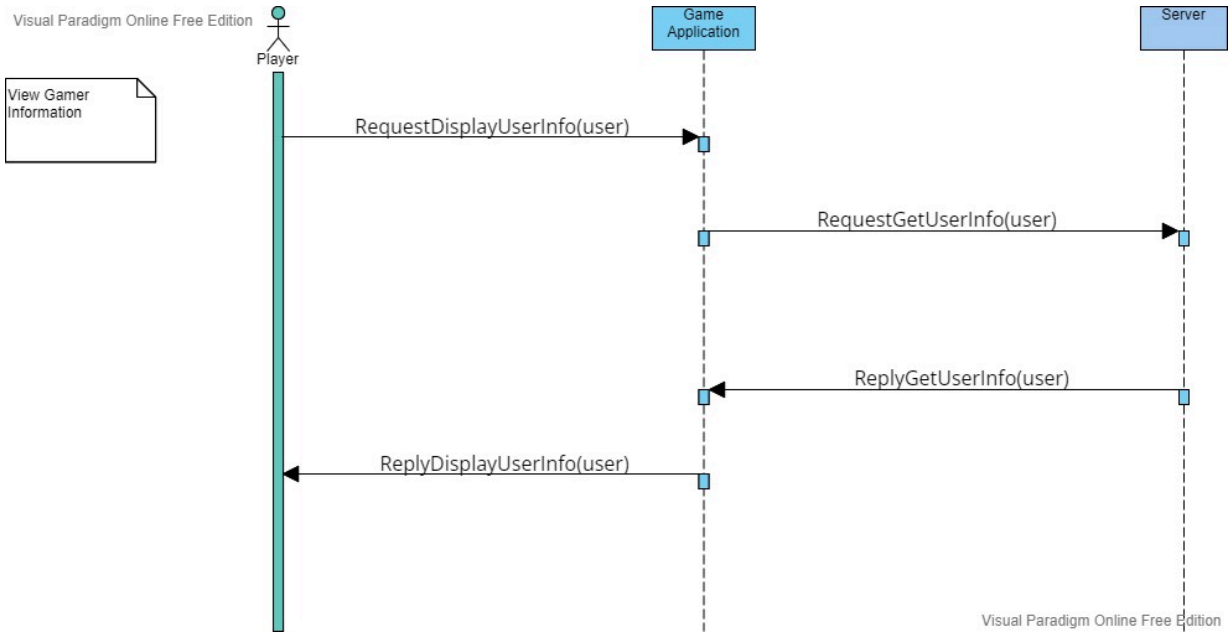




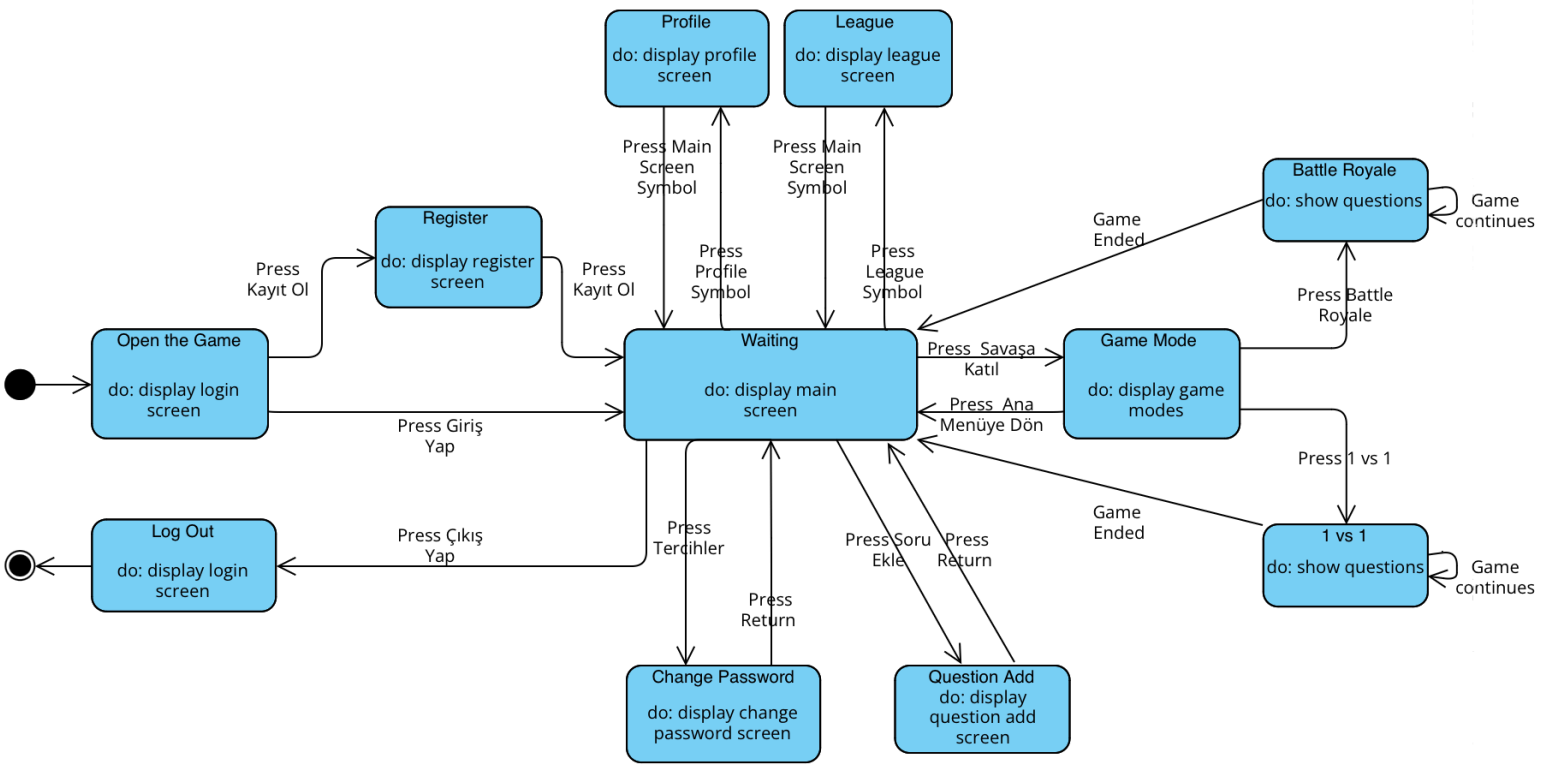

Visual Paradigm Online Free Edition
Player
Game Application
Server
Write Question
SendQuestion(question)
RequestAddNonApprovedQuestion(question)
ReplyAddNonApprovedQuestion(question)
Acknowledge
Visual Paradigm Online Free Edition


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)

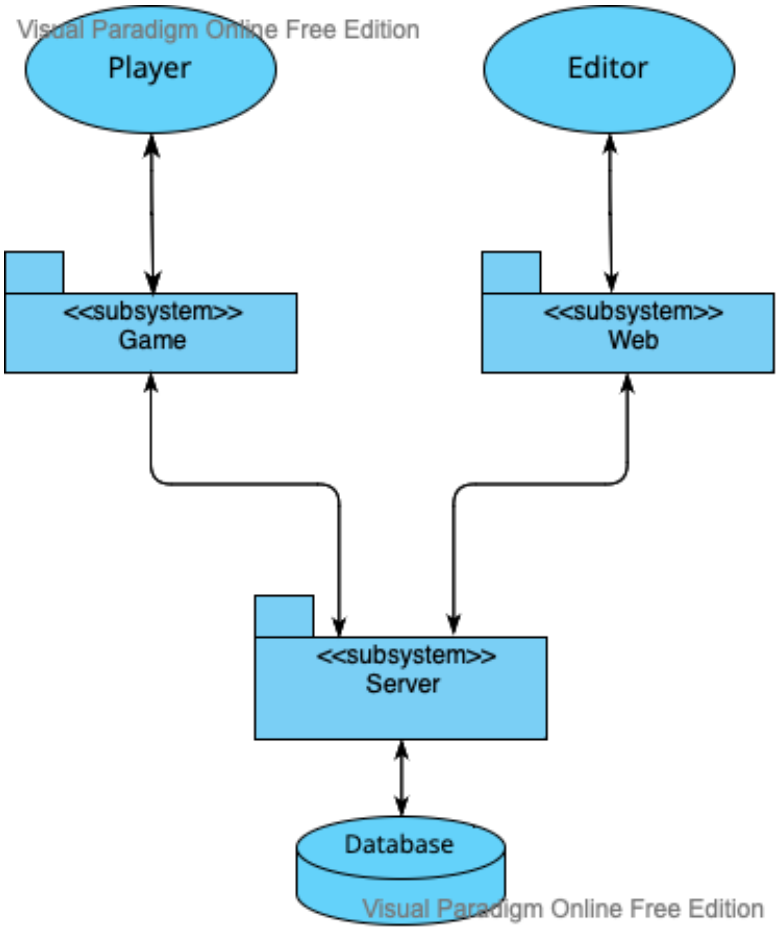
DisplayNewQuestion(question)
SendAnswer(answer)
CheckAnswer(answer)
RequestPlayerScore(player)
ReplyPlayerScore(player)
DisplayPlayerScore(score)
Visual Paradigm Online Free Edition



1.5 State Diagrams



2. Simple System Architecture Document



Our system architecture consists of 3 subsystems: Game, Web and Server. This system architecture is based on client-server model.

Game subsystem is the game itself. It consists of the UI of the game and backend of the game flow. One of the clients, player uses this subsystem in order to create a connection for the game. Every transaction is evaluated and required data is fetched from database or sent to database via Server.

Web subsystem is for administration parts. This system also provides UI for another client, editors to create a connection for the system. Addition/removal/edit operations for questions are handled via this subsystem. Questions added by players are also evaluated and approved by editors via Web subsystem.

Having two distinct subsystems for two different clients is important because both subsystems are responsible for distinct part of the whole system. Therefore, they should be located independently but connected somehow in order to provide contiguous data flow.

There is another system to connect these two subsystems to database and make data transfer between them. This subsystem is called Server. This is responsible for data flow between database and other subsystems. Since there is a contiguous data flow and this data flow determines the game flow, this is the most critically-positioned subsystem.

Using client-server model is the best option for our project because the main part of the project, gameplay part is totally controlled by server and there exists real time data flow during the gameplay.

3. Test Reports

3.1 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

G_TC_16	Test Questions Controller GetApproved Function	All Approved Questions should be available in the database	1. Create QuestionsController object 2. Call GetAll function with the filter isApproved	N/A	A Questions List that includes all approvedquestions available on the system.	A Questions List that includes all approved questions available on the system.	Pass
G_TC_17	Test Questions Controller GetNonApproved Function	All Non Approved Questions should be available in the database	1. Create QuestionsController object 2. Call GetAll function with the filter isApproved	N/A	A Questions List that includes all non approvedquestions available on the system.	A Questions List that includes all non approved questions available on the system.	Pass
G_TC_18	Test Questions Controller GetDeleted Function	All Deleted Questions should be available in the database	1. Create QuestionsController object 2. Call GetAll function with the filter isDeleted	N/A	A Questions List that includes all deleted approvedquestions available on the system.	A Questions List that includes all deleted approved questions available on the system.	Pass
G_TC_19	Test Questions Controller ApproveQuestion Function	Given Question Id should be available on the system	1. Create QuestionsController object 2. Change isApproved field 3. Update the new object	İnt id = 1	A Questions object is approved.	A Questions object is approved.	Pass
G_TC_20	Test Questions Controller RejectQuestion Function	Given Question Id should be available on the system	1. Create QuestionsController object 2. Change isApproved field 3. Update the new object	İnt id = 1	A Questions object is rejected and deleted from the database.	A Questions object is rejected and deleted from the database.	Pass

G_TC_21	Test CompetitionsController findCompetition Function	Gived UserId must be available on the system.	1. Create Competitions object 2. Create a new competition if there is no avaliable one.	İnt id = 1 String GameMode = “BattleRoyals”	A new competition is created on the database.	A new competition is created on the database.	Pass
G_TC_22	Test CompetitionsController isStarted Function	Given CompetitionId must be available on the system.	1. Create Competitions object 2. Check if the given competition is started or not.	İnt id = 1	Competition status is returned.	Competition status is returned.	Pass
G_TC_23	Test CompetitionsController chooseQuestion Function	Given CompetitionId must be available on the system.	1. Create Competitions object 2. Choose a new question fort he competition.	İnt i = 1	New Question is added to the competition.	New Question is added to the competition.	Pass
G_TC_24	Test CompetitionsController getQuestion Function	Given CompetitionId must be available on the system.	1. Create Competitions object 2. Get the current question on the competition.	İnt i = 1	Current question on the competition is returned.	Current question on the competition is returned.	Pass
G_TC_25	Test CompetitionsController exitCompetition Function	Gived UserId must be available on the system.	1. Create Competitions object 2. Delete the given user from the competition.	İnt id = 1	User is deleted from the competition’s players list.	User is deleted from the competition’s players list.	Pass
G_TC_26	Test CompetitionsController resetCapacity Function	Given CompetitionId must be available on the system	1. Create Competitions object 2. Calculate the new capacity for the competition.	İnt id = 1	New capacity is returned.	New capacity is returned.	Pass

G_TC_27	Test Competitions Controller getCurrentCapacity Function	Given CompetitionId must be available on the system	1. Create Competitions object 2. Return the current capacity for the competition.	İnt id = 1	Current capacity is returned.	Current capacity is returned.	Pass
G_TC_28	Test Competitions Controller trueAnswer Function	Given Users object should be available on the system	1. Create Competitions object 2. Check if the capacity is enough. 3. Return if user is eliminated or not.	Users user	If capacity is full return false to indicate user is eliminated otherwise return true.	If capacity is full return false to indicate user is eliminated otherwise return true.	Pass
G_TC_29	Test Competitions Controller wrongAnswer Function	Given Users object should be available on the system	1. Create Competitions object 2. Eliminate the user.	Users user	Delete the user from the competition.	Delete the user from the competition.	Pass
G_TC_30	Test Competitions Controller deleteCompetition Function	Given CompetitionId must be available on the system.	1. Create Competitions object 2. Delete the given competition from the system.	İnt id = 1	Given competition is deleted from the system.	Given competition is deleted from the system	Pass

3.2 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 SwitchToQuestionAddMenu 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 ReinitializeSpaceShipPositio n 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 DeclareNewGameProgr ess-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 SetVerticalPositionOfTh e-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-SwitchingNewGamePro gress method	SpaceShip ui element has to be attached and ScreenManager class has to exist in game	1.Call SetVerticalPositionOfThe-NormalizeAffectsOf-SwitchingNewGameProgres s 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 MarkChoiceAsChosen method of ClassicalQuestionScreen with corresponding choice as input	Call MarkChoiceAsChosen method of ClassicalQuestionScreen 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 has to be called	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, NumericalQuestion Screen 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

3.3 Test Reports for Web

Test Case ID	Test Case Objective	Prequistite	Steps	Input Data	Expected Output	Actual Output	Status
TC_01	Check admin login with valid data	Admin information must be registered in the database	1)Enter mail 2)Enter password 3) Click on send button	Email adress and password	Access to the admin panel is provided	Access to the admin panel is provided.	Pass
TC_02	Reset password	In order to perform this function, it is necessary to be an admin beforehand.	1)Click on Forget password 2)Enter e-mail address 3)Click on send button	Emaill adress of admin	Admin's password is changed.	Admin's password is changed.	Pass
TC_03	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_04	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_05	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

Test Case ID	Test Case Objective	Prequistite	Steps	Input Data	Expected Output	Actual Output	Status
TC_06	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_07	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_08	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_09	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

Test Case ID	Test Case Objective	Prequistite	Steps	Input Data	Expected Output	Actual Output	Status

TC_10	Approve user-typed questions	The user must have written a question in the app and sent it to the server.	1)Go to the correction section from menu 2)Click on the confirmation icon	N/A	The approved question is added to the database and is seen among the questions of the admin.	The question is added to the database and the question reaches the users when the game will be played.	Pass
TC_11	Viewing the questions deleted by the admin	Admin must have deleted some of the questions he wrote before.	1)Click on the recycle icon from the menu	N/A	The administrator sees the deleted questions.	The administrator sees the deleted questions.	Pass

4. Accessibility of Battle of Minds

Our game is now accessible to potential users via APK version since we use a cloud database so you can easily access to our game and start playing. It is also attached to the assignment page. Note: Battle Royale mode requires at least 4 players to play.