Criterion B: Design

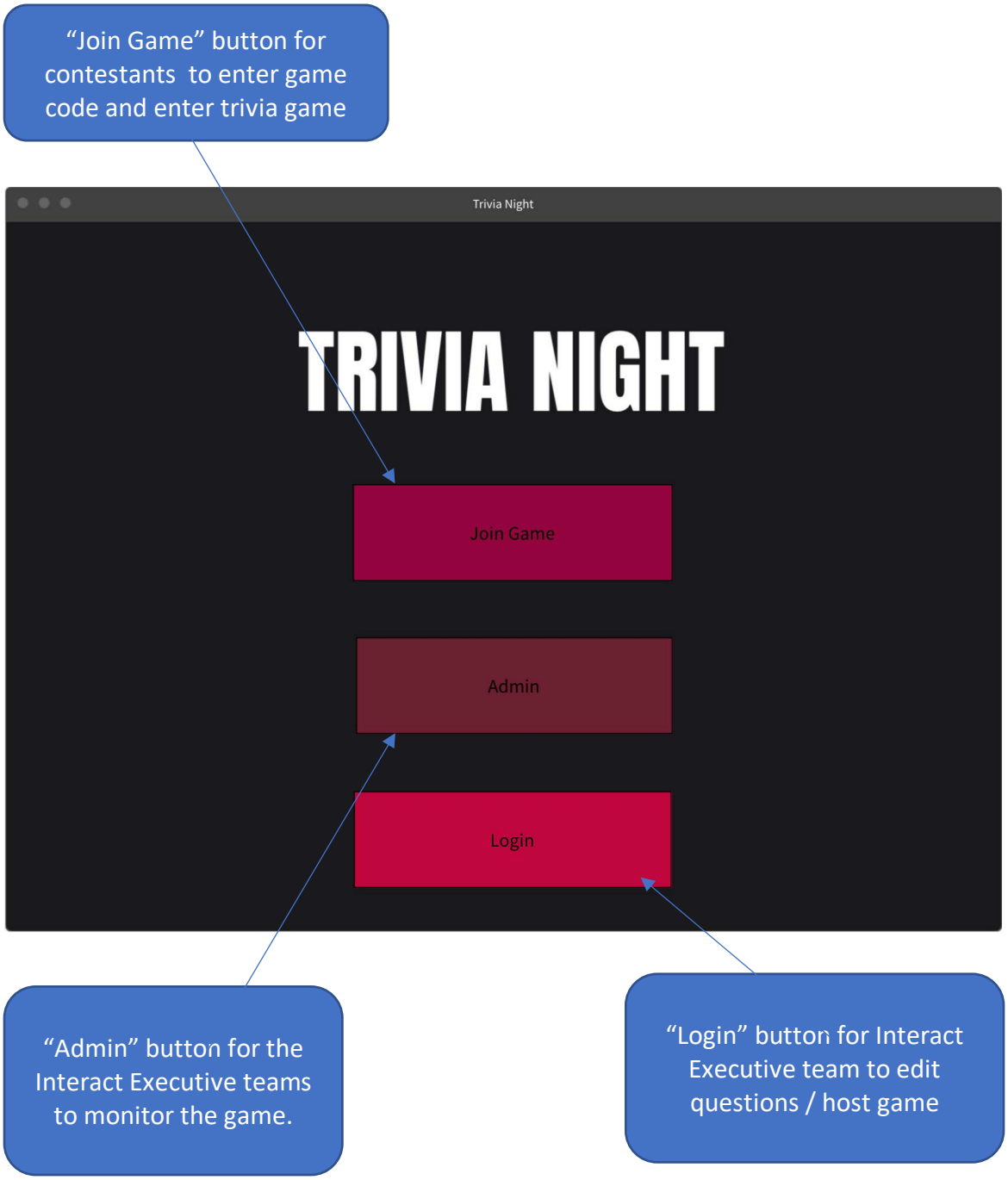
1.UI mockup, this part describes the UI for the system.

Color palate

This page describes the color palate, which will be used by the system.

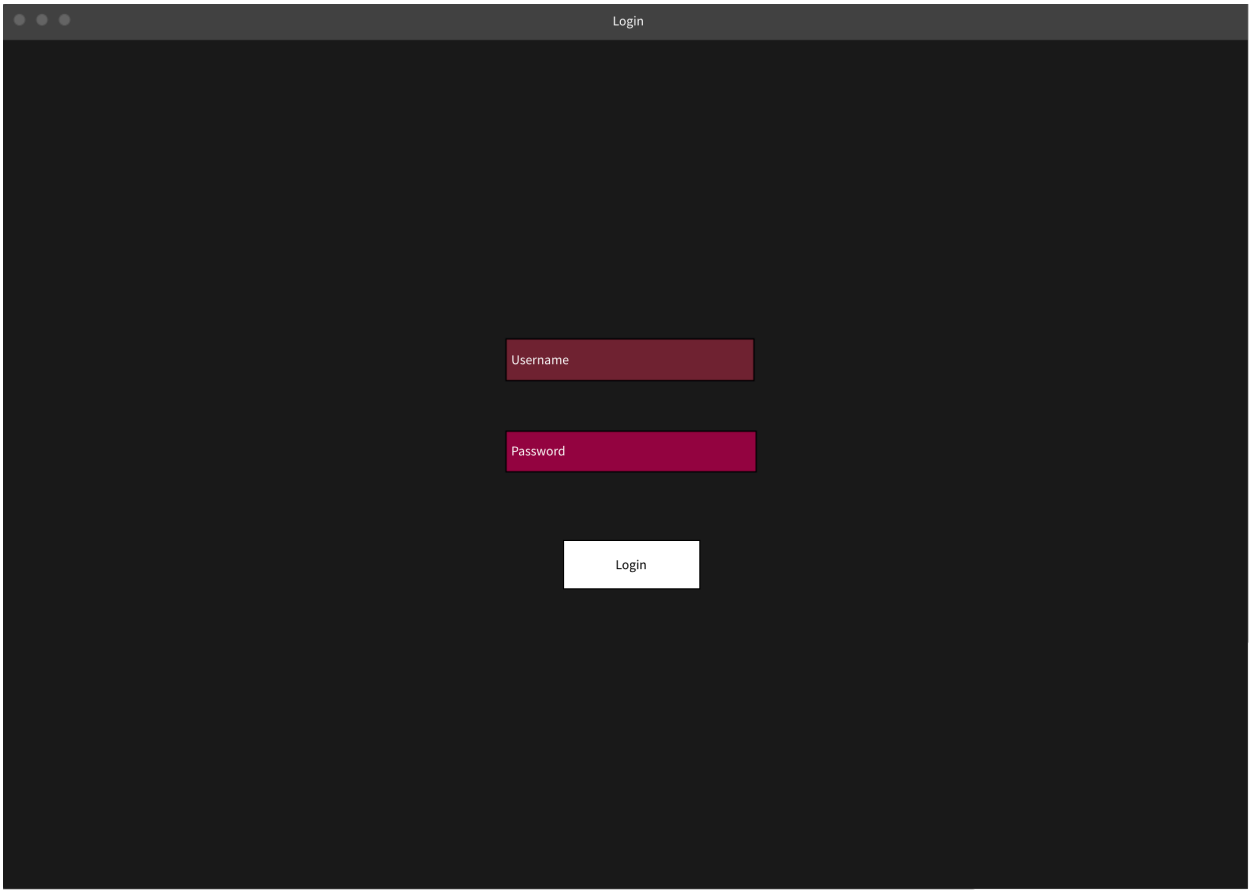
Front Page

This page will be the entry for all the users who is using the system.



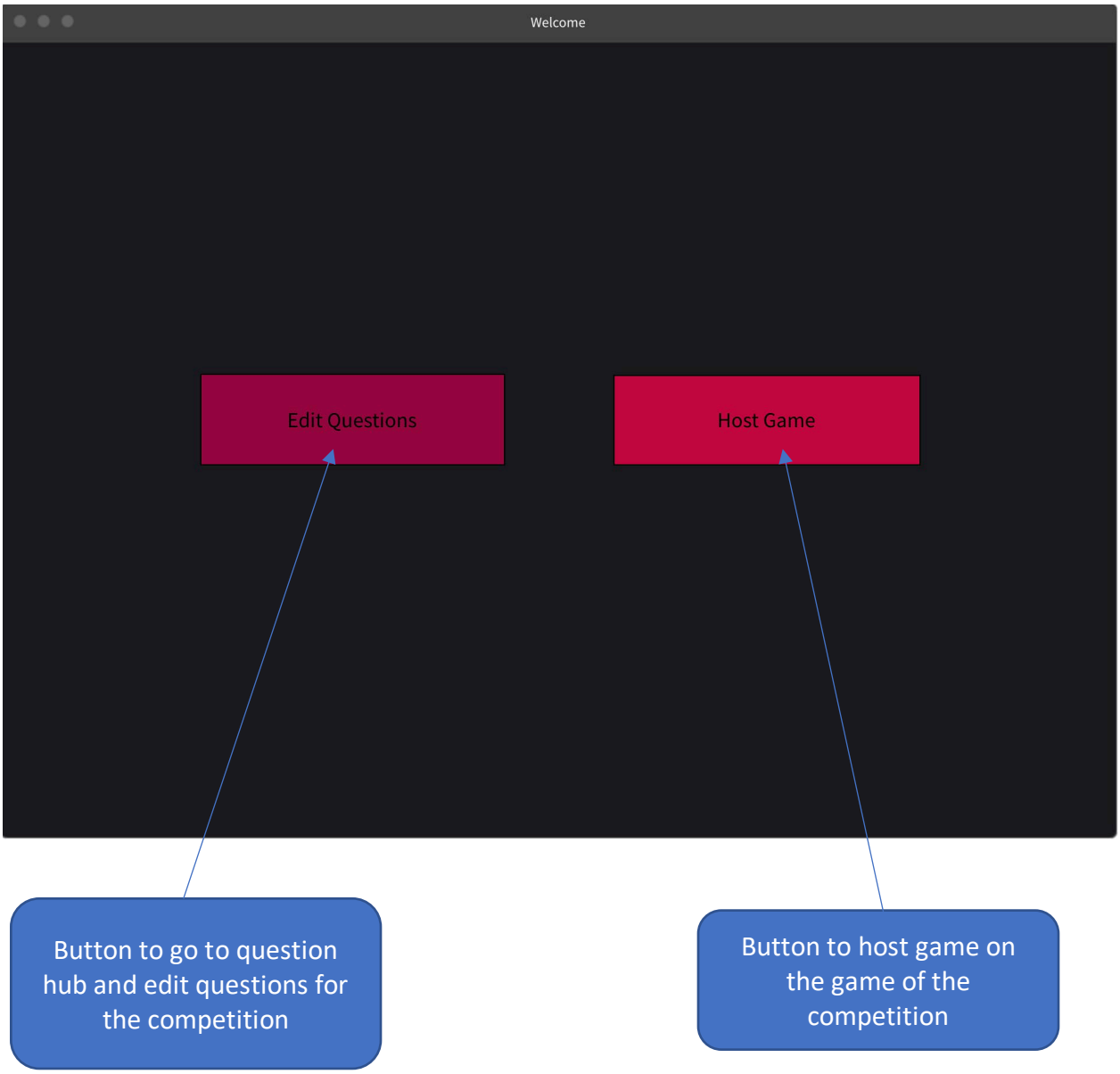
Login Page

Executive team login page



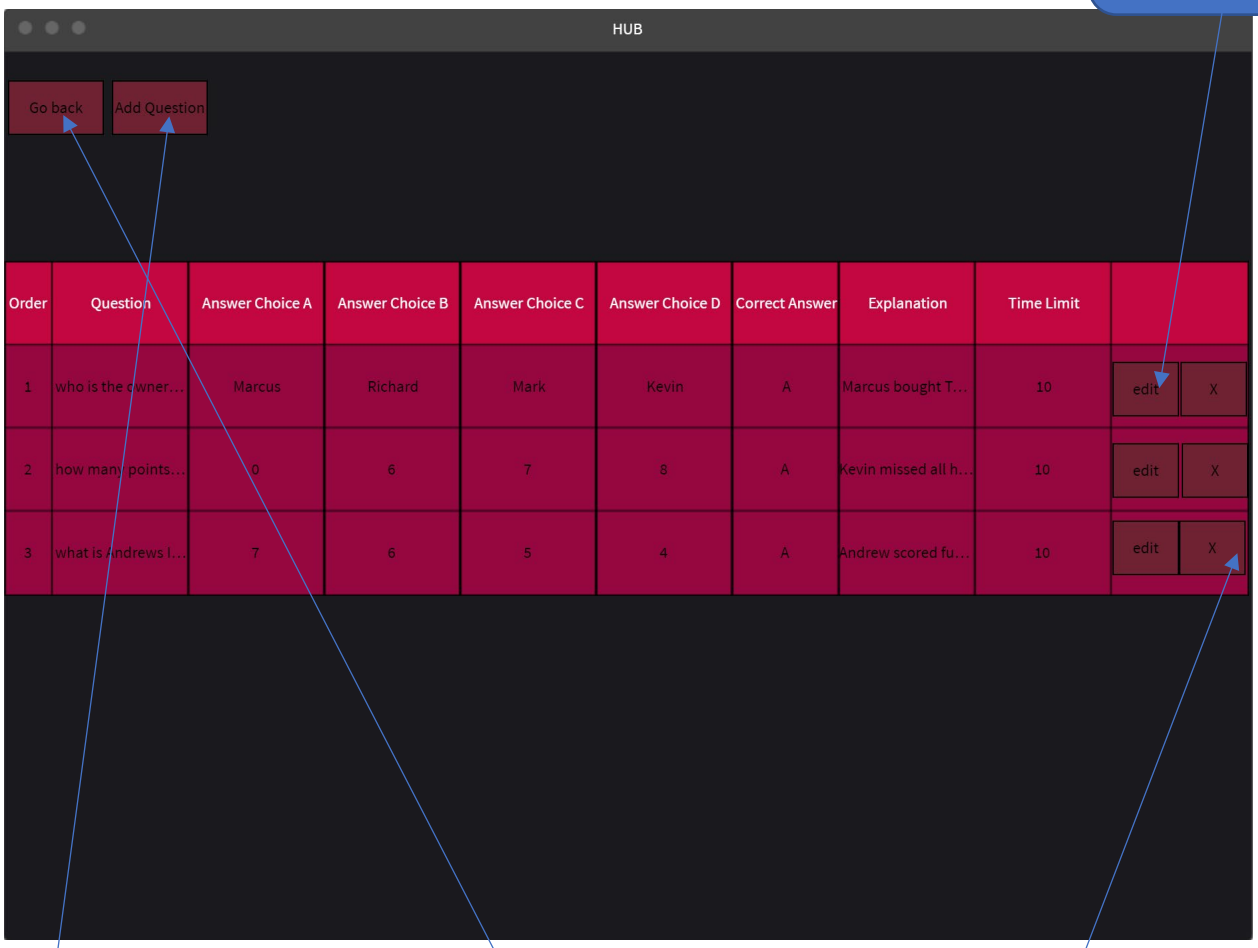
Page after login

This page displays the functions which Interact Executive team can be used.



Question Hub

Page provides the functions for user to add / edit / delete questions.



Click on edit button to edit question

Button to add questions

Button to go back to page after login

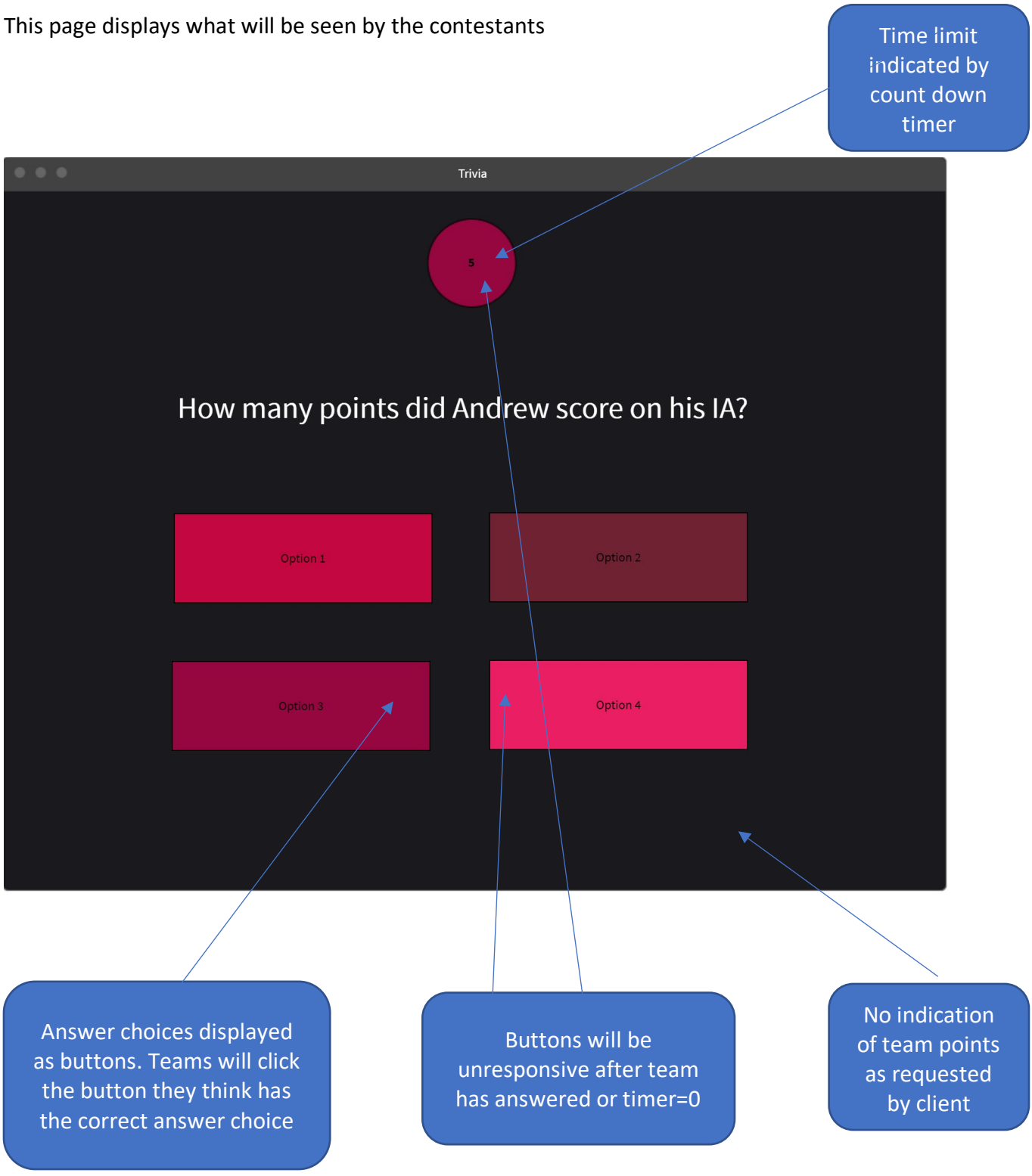
Delete button to delete questions

Question Editing

This shows how users can add or edit questions for the game. This example is when “add question” button is pressed.

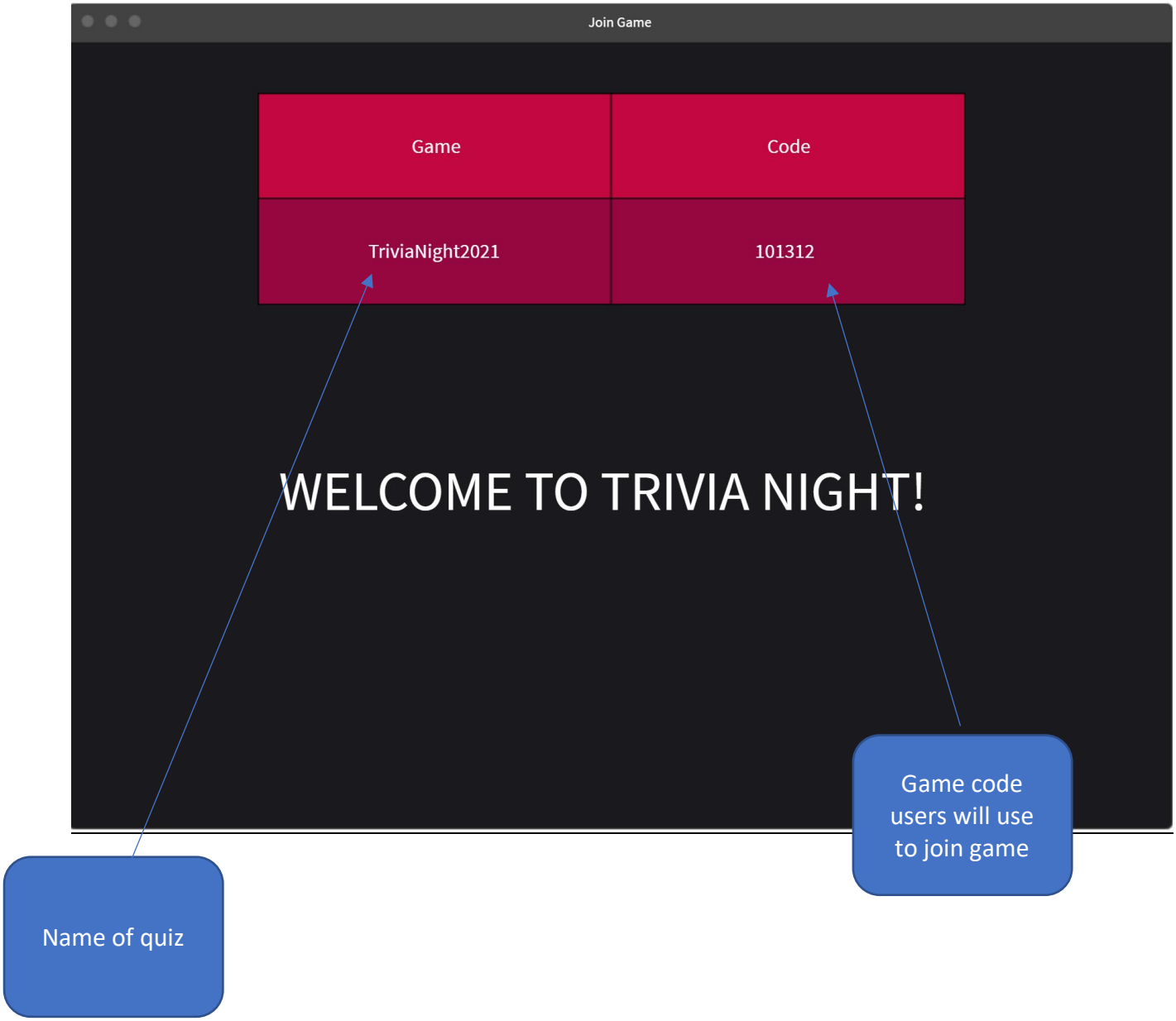
Contestant Page

This page displays what will be seen by the contestants



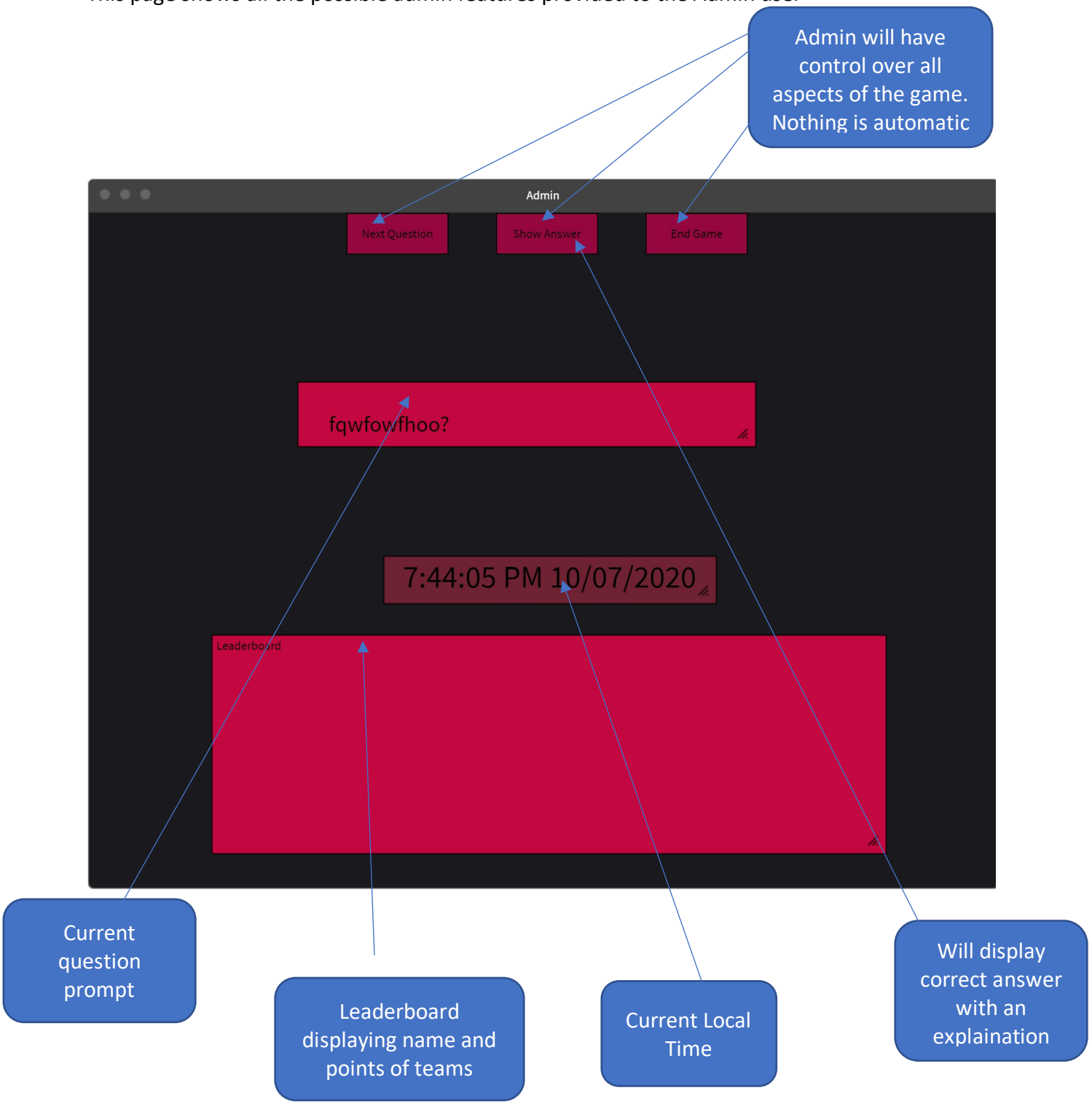
Page after game is hosted:

This page provides the provides the contestants with the game code they need to join the quiz



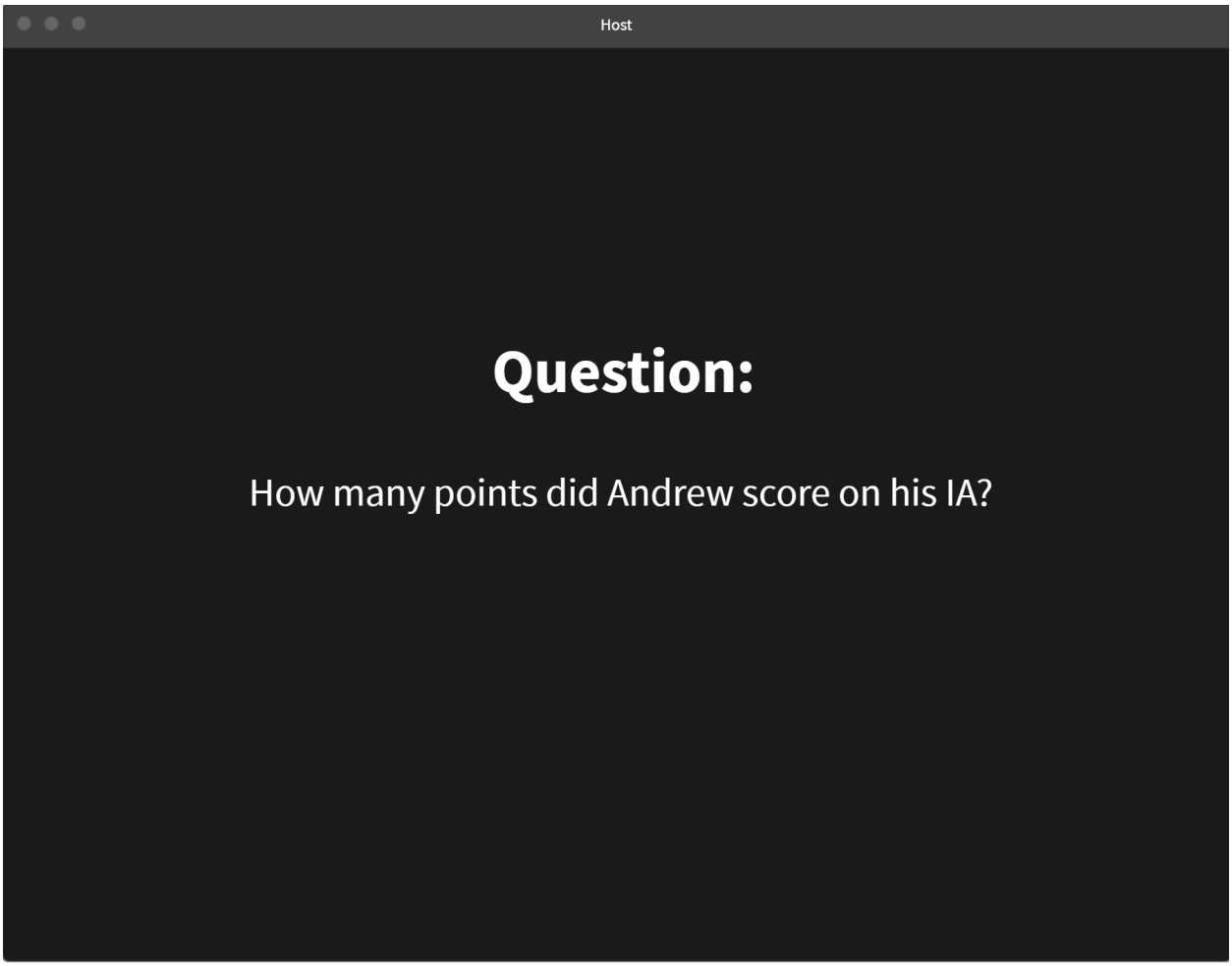
Game Admin

This page shows all the possible admin features provided to the Admin user



Game Hosting

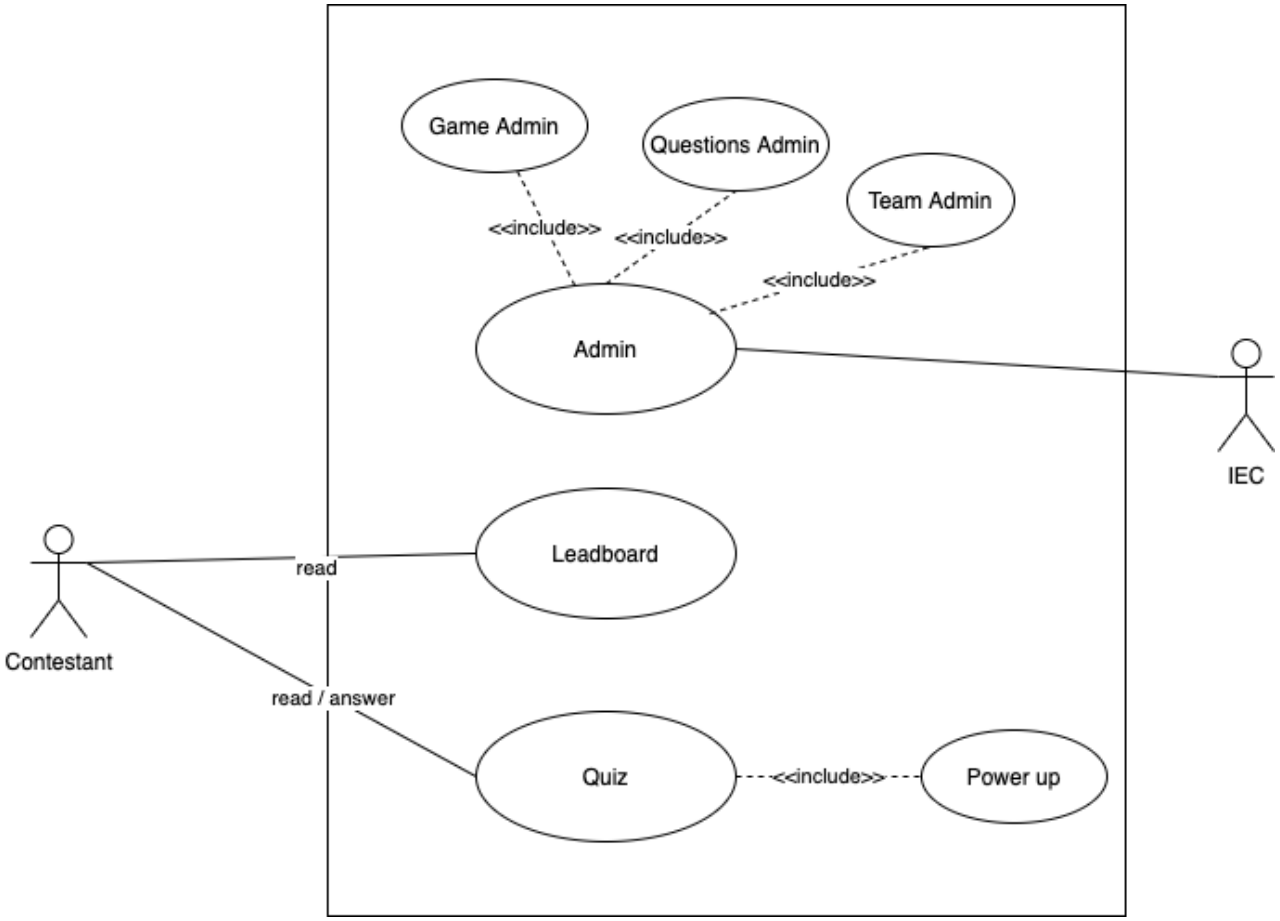
Page displayed on host's screen during the quiz



2.This part describes the design of the system.

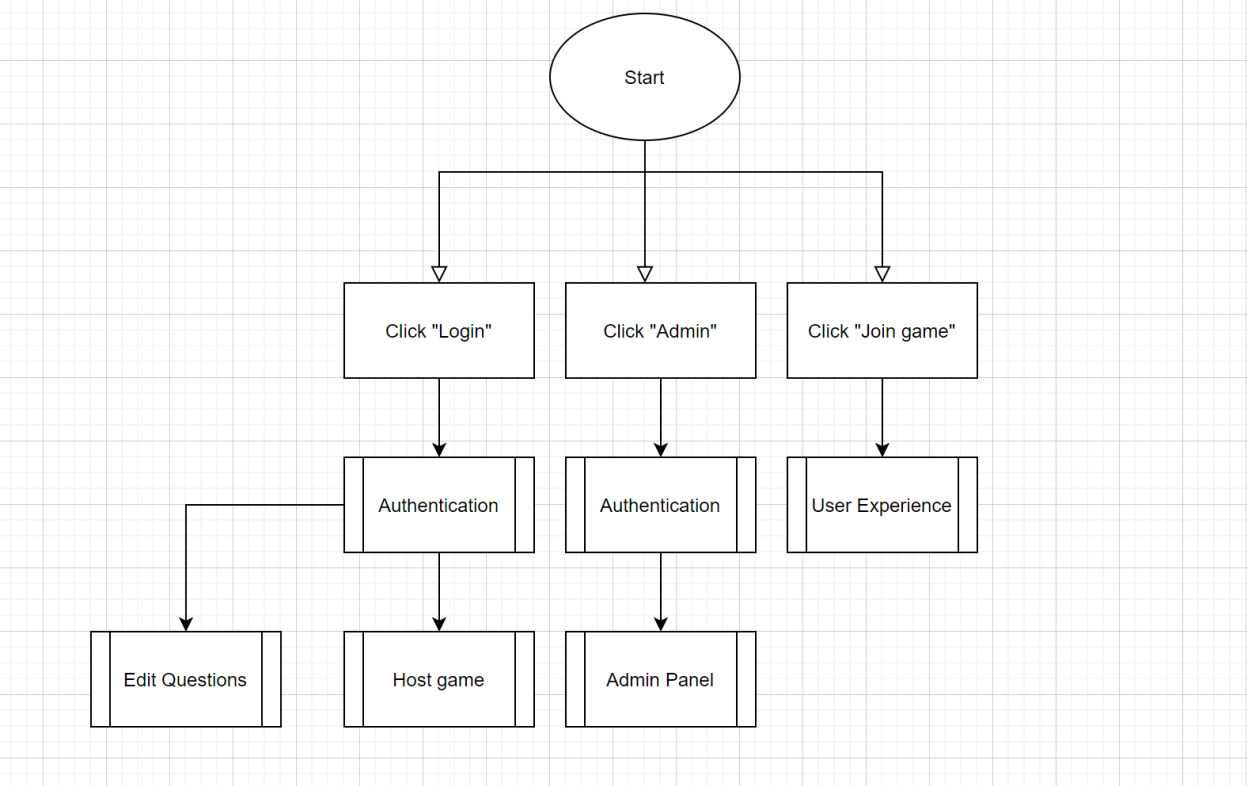
Use case diagram

IEC is short for **I**nteract **E**xecutives

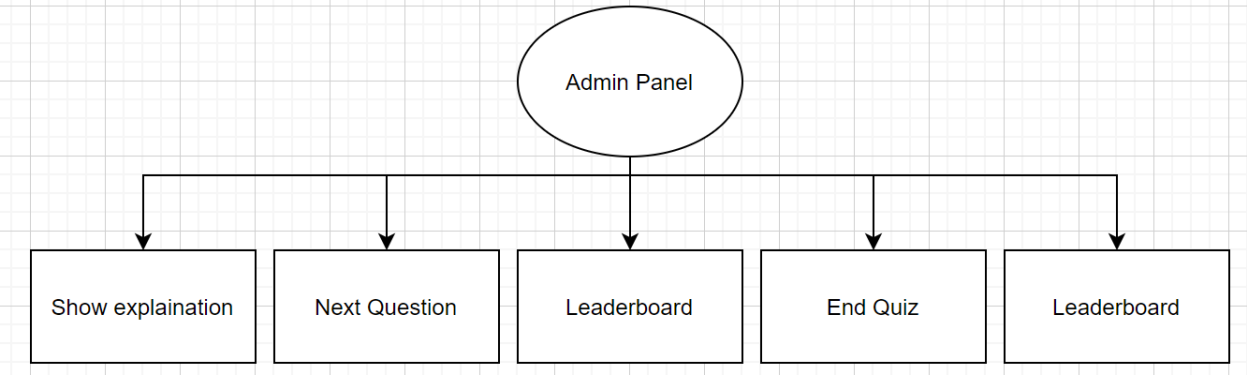


Data flow diagram:

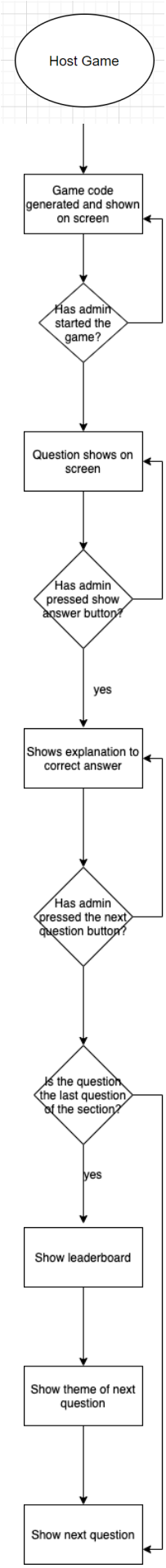
Flow diagram of program:



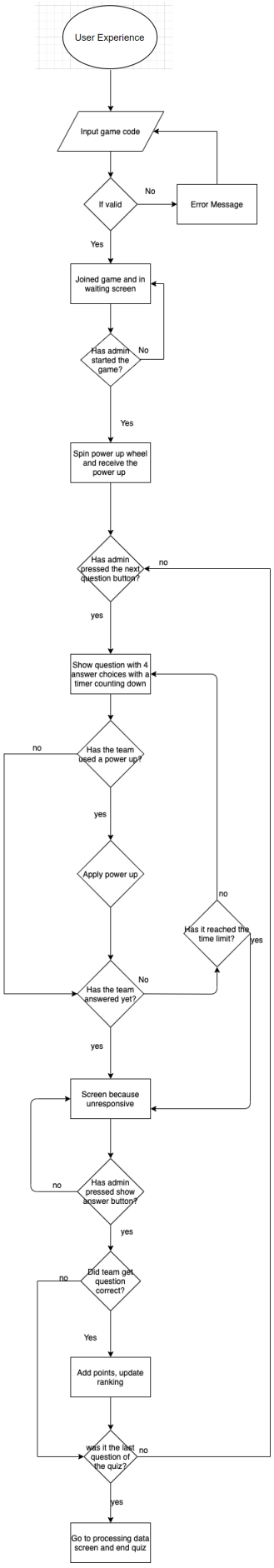
Admin panel subroutine:



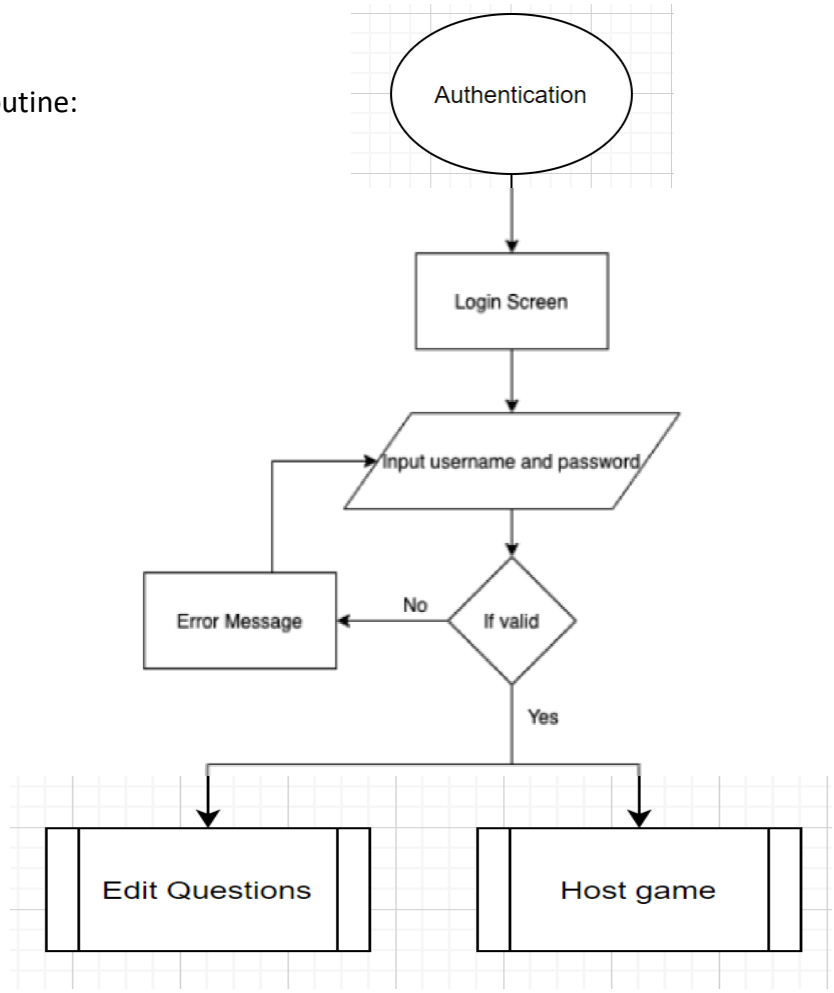
Host game subroutine:



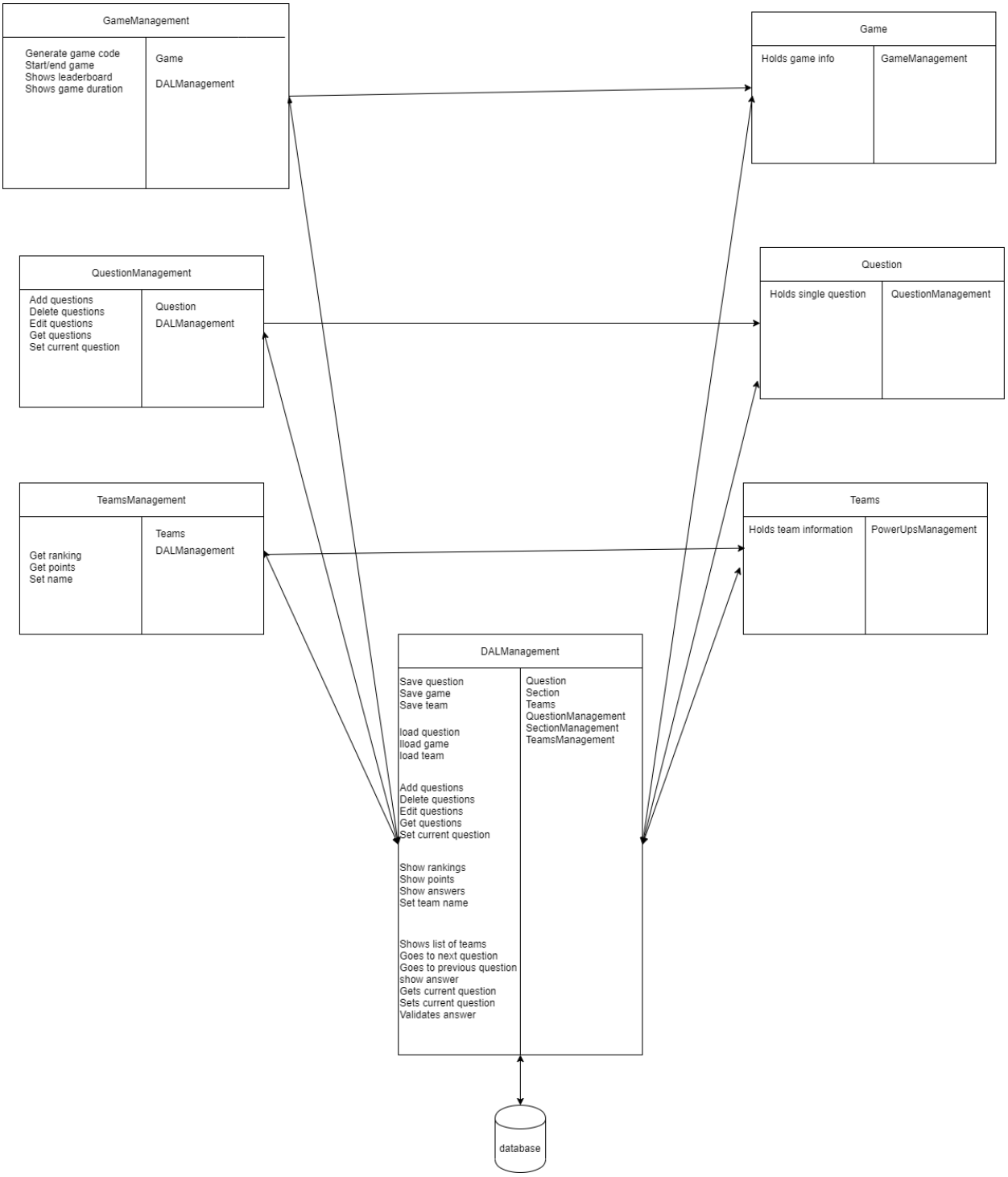
Contestant’s usage of program:



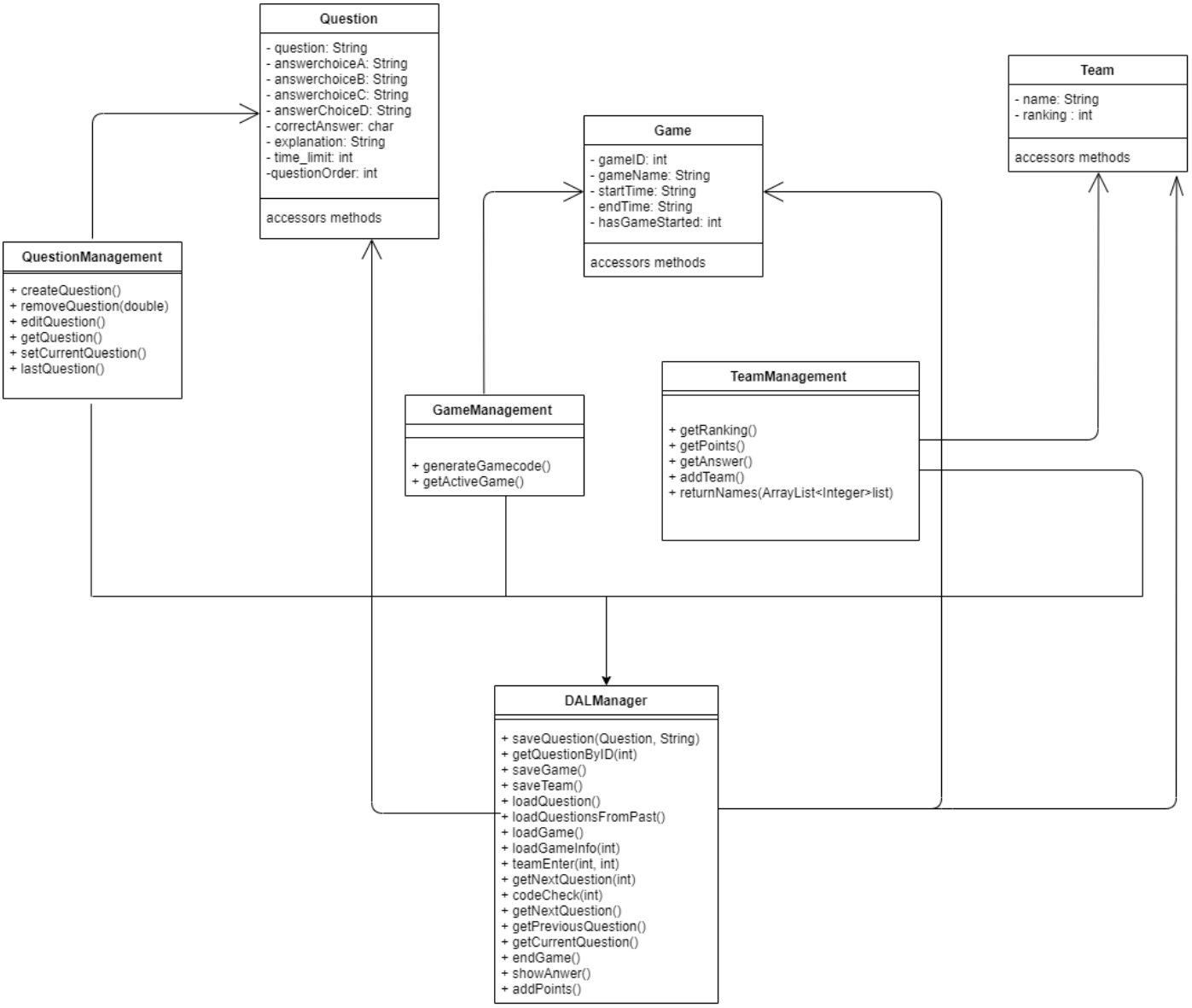
Authentication subroutine:



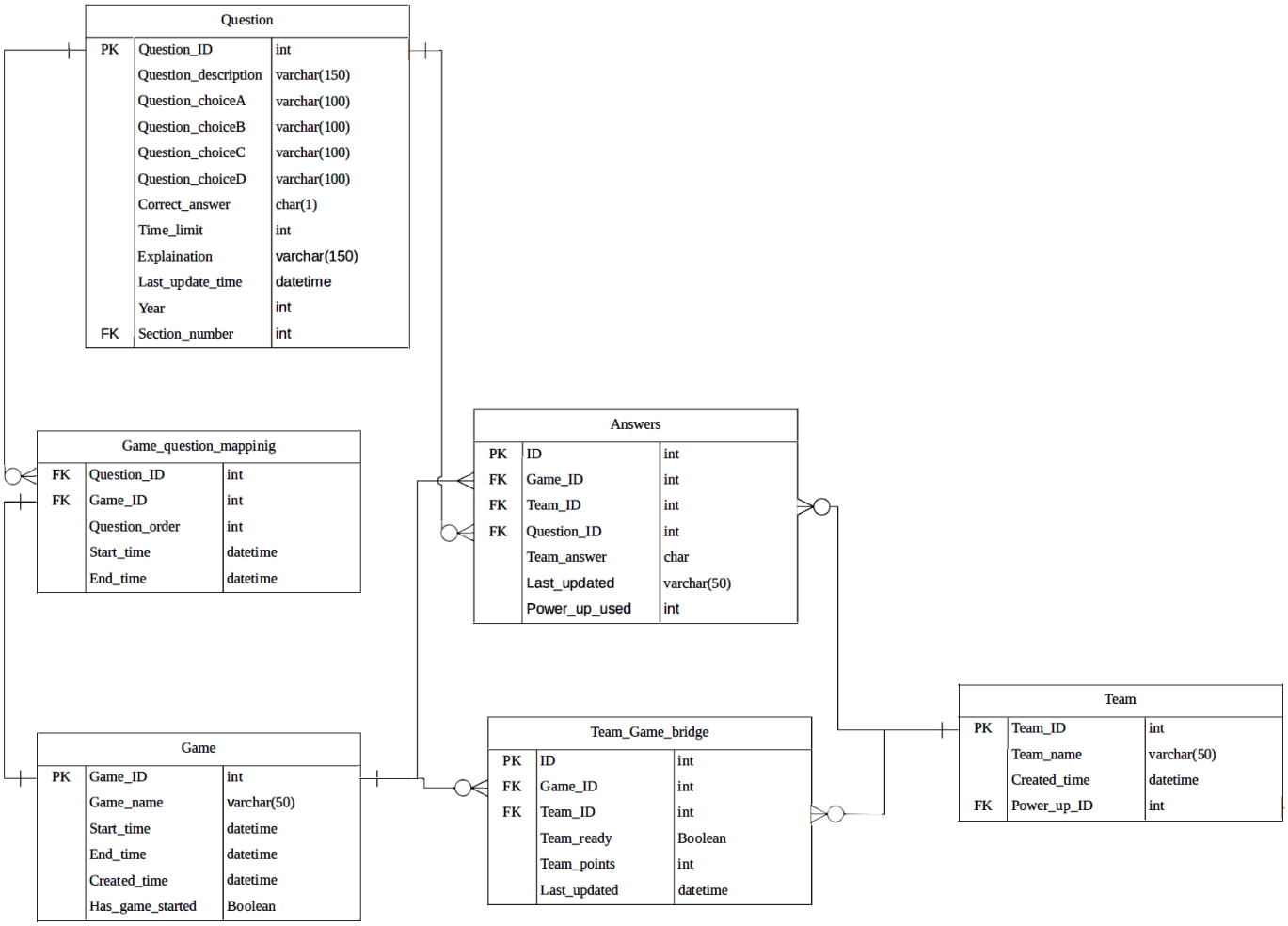
CRC Diagram


GameManagement
Generate game code
Start/end game
Shows leaderboard
Shows game duration
Game
DALManagement
Game
Holds game info
GameManagement
QuestionManagement
Add questions
Delete questions
Edit questions
Get questions
Set current question
Question
DALManagement
Question
Holds single question
QuestionManagement
TeamsManagement
Get ranking
Get points
Set name
Teams
DALManagement
Teams
Holds team information
PowerUpsManagement
DALManagement
Save question
Save game
Save team
load question
lload game
load team
Add questions
Delete questions
Edit questions
Get questions
Set current question
Show rankings
Show points
Show answers
Set team name
Shows list of teams
Goes to next question
Goes to previous question
show answer
Gets current question
Sets current question
Validates answer
Question
Section
Teams
QuestionManagement
SectionManagement
TeamsManagement
database

Class diagram



ERD Diagram


Question
PK Question_ID int
Question_description varchar(150)
Question_choiceA varchar(100)
Question_choiceB varchar(100)
Question_choiceC varchar(100)
Question_choiceD varchar(100)
Correct_answer char(1)
Time_limit int
Explaination varchar(150)
Last_update_time datetime
Year int
FK Section_number int
Game_question_mappinig
FK Question_ID int
FK Game_ID int
Question_order int
Start_time datetime
End_time datetime
Answers
PK ID int
FK Game_ID int
FK Team_ID int
FK Question_ID int
Team_answer char
Last_updated varchar(50)
Power_up_used int
Team
PK Team_ID int
Team_name varchar(50)
Created_time datetime
FK Power_up_ID int
Game
PK Game_ID int
Game_name varchar(50)
Start_time datetime
End_time datetime
Created_time datetime
Has_game_started Boolean
Team_Game_bridge
PK ID int
FK Game_ID int
FK Team_ID int
Team_ready Boolean
Team_points int
Last_updated datetime

Pseudocode:

Leaderboard Sorting algorithm

```
// Algorithm returns a collection of collections with one collection being the name of the teams and the other being the number of points of each team.

ranking = new Hashmap<Integer, Integer>()
leaderboard = new Collection()
ResultSet= SELECT everything from Game_Team_bridge database WHERE GameID = id of current game  // Using the SQLite-jdbc library to get results from database

loop while ResultSet.hasNext()
        ranking.put(ResultSet.get("Team_ID"), ResultSet.get("Team_Points"))
        //HashMAP mapping Team ID to Team Points
end loop

temp= new Collection(ranking.keySet()) // Collection of all Team ID
points = new Collection(ranking.values()) // Collection of all Team Points
teams = new Collection()
Collections.sort(Points) // sorts the points in ascending order

loop while !temp.isEmpty
        loop i from 0 to points.size()-1
                loop j from 0 to temp.size()-1
                        if ranking.get(temp.get(j))==points.get(i)
                                teams.addItem(temp.get(j))
                                teams.remove(j)
                                skip to next iteration of i
                        end if
                end loop
        end loop
end while
// loop allows the team ID to be stored in the same order as the points which is sorted previously


ranking.addItem(Team)
ranking.addItem(Points)
Return ranking // returns a collection of collections
```

Algorithm to save question.

```
// parameters of method takes in a Question object and a String called process
question= Question
process;
if process = “add”
        insert into database:
        question_id = question.getQuestionID()
        question_description = question.getQuestion()
        question_choiceA = question.getChoiceA()
        question_choiceB = question.getChoiceB()
        question_choiceC = question.getChoiceC()
        question_choiceD = question.getChoiceD()
        correct_answer = question.getCorrectAnswer()
        explanation = question.getExplaination()
        time_limit = question.getTimeLimit()
end if
if process = “edit”
        update database where question_id = question.getQuestionID()
        // question ID cannot be changed so we can find the question to edit by using it
        question_description = question.getQuestion()
        question_choiceA = question.getChoiceA()
        question_choiceB = question.getChoiceB()
        question_choiceC = question.getChoiceC()
        question_choiceD = question.getChoiceD()
        correct_answer = question.getCorrectAnswer()
        explanation = question.getExplaination()
        time_limit = question.getTimeLimit()

end if
if process= “delete”
        delete from database where question_id= question.getQuestionID()
        // question ID cannot be changed so we can find the question to edit by using it
end if
```

3.This part describes testing with test cases for the system.

Blue = Contestant
Green = Host
Orange = Admin

Criteria	Testing Method	Expected Result
Be able to join the game with a game code	Click “Join Game” of the front page	The system should be directed the join game webpage
	providing the generated game code and try joining	System should show “You are in game” text appear on screen if code is valid The system should show “invalid code” if code is entered incorrectly
Be able to see and answer the questions	Join a hosted game, after game started	System should show the current question with answer choices in which users should be able to press and submit
Be able to see time remaining for each question shown on screen	Join a hosted game, after game started	System should show a countdown timer which indicates the time remaining for answering this question
User will not be able to answer questions after time limit	Join a hosted game and not answer a question within the limited time	Buttons should turn unresponsive and users will not be able to interact with the page
User will not be able to change their answers after answering	Join a hosted game and answer a question within limited time	Buttons should turn unresponsive and users will not be able to interact with the page
Be able to see the explanation of the correct answers	Join a hosted game and answer a question. Admin then presses “show explanation” button	System should show why the explanation to the correct answer when admin presses “show answer” button
Be able to see an end game page indicating that the quiz has ended on screen	End a hosted game	System should redirect constants to an end game screen after admin presses “end game” button
Be able to show the game code and game name on screen	Host a game	System should show the game code and game name when the admin hosts the game

Be able to display the questions on screen	Host a game and start the game	The current question should be shown on the hosts screen
	Admin clicks “next question” button	The question displayed on screen should change to the next question
Be able to display a page indicating that the quiz has ended on screen.	End a hosted game	System should redirect the host’s screen to an end game page
Be able to add, delete, edit questions (including the duration, correct answer, explanation of why etc)	Go into question hub and try adding, deleting, and editing questions	A question should be added by pressing the add button When deleting questions, system should display a confirmation message when pressing the delete button The answer choices, question, explanation and time limit for each question can be changed and edited
Be able to see questions from past years	Click on “View past years questions”	System should redirect the host’s screen to an end game page
Be able to host game	Press “host game” button	The system should host a game when the “host game” button is clicked An auto generated game code should then appear

Be able to start game or end game	Go to game admin page and press “next question” and “end game” button	The system should start the game when the next question button is first pressed The system should end game when the end game button is pressed
Be able to see the teams and their position on the leaderboard	Go to game admin page	leaderboard displayed to the admins for them to monitor the game
Be able to see the question that is currently being answered	Go to game admin page	System should show the question currently being answered by the contestants Refreshes every time the admin presses the next question button.
Be able to see the local time	Go to game admin page	System should show the local time.
Be able to go to the next question of the quiz	Go to game admin page and press “next question” button.	The “current question” on the admin page should change as well as the host and contestants screen.

Be able to see the final leaderboard when game is ended	Go to game admin page and press “end game” button	System should redirect from admin page to a page displaying the final leaderboard and rankings.