

# Quiz Odyssey

## Requirements Specification

Group: 26

**Eren Taşdemir** 150200035

**Berkant Bakışlı** 150200069

**Karol Jan Charchut** 912310003

**Pijus Jonas Navasaitis** 912310007

**Egi Gjineci** 150190910

# Table of Contents

<b>1. Introduction</b>	<b>2</b>
1.1. Goal	2
1.2. Content	2
<b>2. System Requirements</b>	<b>2</b>
2.1. Functional requirements	2
2.2. Non-functional requirements	2
<b>3. Use Cases</b>	<b>3</b>
3.1. User Types	3
3.2. User Scenarios	4
3.3. Use Case Diagram	6
3.3.1. System Use Case Diagram	6
3.3.2. Functionality Specific Diagrams	7
3.4. Use Cases with Flows	12
3.4.1. Registration	12
3.4.2. Logging in	13
3.4.3. Submit a question	13
3.4.4. Change username	13
3.4.5. Change password	14
3.4.6. Play a daily challenge	14
3.4.7. Play trivia game	15
3.4.8. View followers	15
3.4.9. View followed users	15
3.4.10. Review other user's profile	15
3.4.11. Remove user	16
3.4.12. Remove question	16
3.4.13. Review question list	17
3.4.14. Review community questions	17
3.4.15. Compare statistics between users	17
3.4.16. View users list	18

# 1. Introduction

## 1.1. Goal

The goal of this project is to design and implement a trivia style mobile game that offers an engaging and educational experience. The game will allow users to answer questions in various topics, such as art, history, science, sports, entertainment, etc., and earn points based on their performance. The game will also feature daily and weekly challenges, where users can compete with other players in timed quizzes and rank in the leaderboards. The game will be compatible with both Android and iOS platforms and will follow the best practices of user interface design and accessibility.

## 1.2. Content

This document specifies the functional and non-functional requirements for the project. It describes the main features and functionalities of the game, as well as the assumptions that affect its design and development. The document also presents the use cases that illustrate the interactions between the users and the system, as well as the use case diagram that shows the relationships among the use cases. The document is intended for the project stakeholders, such as the client, the developers, and the testers, who need to understand and agree on the requirements of the game.

# 2. System Requirements

## 2.1. Functional requirements

- 2.1.1. The system shall allow users to register
- 2.1.2. The system shall allow previously register users to login
- 2.1.3. The system shall allow users to take a quiz in a chosen category
- 2.1.4. The system shall allow users to take a timed challenge quiz in a chosen category
- 2.1.5. The system shall allow users to participate in daily challenges
- 2.1.6. The system should show the users their streak count
- 2.1.7. The system shall allow users to see their profiles, change their username or password
- 2.1.8. Users shall be able to follow other users
- 2.1.9. Users should be able to submit new questions
- 2.1.10. Users shall be able to view their followers and people that they follow
- 2.1.11. The system shall allow admins to view and ban users
- 2.1.12. The system shall allow admins to promote users to moderators
- 2.1.13. The system shall allow admins and moderators to view, edit, add and remove quiz questions.

## 2.2. Non-functional requirements

- 2.2.1. Performance:
  - 2.2.1.1. Response Time: The response time of the system should always be lower than 5 seconds.

- 2.2.1.2. Scalability: The system should initially support 1000 users and then scale to accommodate 10000 users within 6 months using cloud based options.
- 2.2.1.3. Load Testing: Response time should meet specified requirements when tested with 2000 concurrent users in 24 hours.
- 2.2.2. Security:
  - 2.2.2.1. Data Encryption: User's sensitive data should be encrypted using AES.
  - 2.2.2.2. Authentication: The password should contain a minimum of 8 characters, including uppercase, lowercase, numbers, and special characters.
  - 2.2.2.3. Authorization: Administrators shall have access to specific sections for player management and game monitoring.
  - 2.2.2.4. Security Testing: Different testing methods shall be used throughout and after the development of the project.
  - 2.2.2.5. Compliance: The application should be GDPR-compliant.
- 2.2.3. Usability and User Experience:
  - 2.2.3.1. Accessibility: The system should use voice commands to support users with special needs(low vision).
  - 2.2.3.2. User Interface Consistency: The user interface should provide easy language, familiar terms, and standard interface elements.
  - 2.2.3.3. Error Handling: The system should provide error messages when wrong input is received.

## 3. Use Cases

### 3.1. User Types

<b>Guest:</b>	Can see the landing page Can register
<b>Registered User:</b>	Known by the system Can participate in challenges Can follow other users Can submit new questions Can see the statistics of other users
<b>Moderator:</b>	Can approve submitted questions Can add tags to questions
<b>Administrator:</b>	Can view the list of all users Can do everything a Moderator can do Can ban users Can make other users Moderator or Administrator

## 3.2. User Scenarios

### 3.2.1. The system shall allow users to register

Alice launches the application and is greeted with the welcome screen leading to a **registration form**. She enters her email address, picks a username and chooses her account password. She then sees a message indicating successful registration.

### 3.2.2. The system shall allow previously registered users to login

Alice, an already registered user, opens the app and gets to the login screen. She enters her email and password correctly. She is **authenticated** and redirected to the home screen of the application.

### 3.2.3. The system shall allow users to take a quiz in a chosen category

Alice, on the homepage of the application, sees a **list of categories** to choose from, such as history, science, sports, etc. She selects history and taps on the take quiz button. The app shows her **a question** with four possible answers. She answers the first question correctly, she **earns points** and moves on to the next question. She answers the second question incorrectly and the quiz moves on to the next question without giving any points. She completes the quiz and after the quiz ends, she sees her **score** and **rank** on the leaderboard.

### 3.2.4. The system shall allow users to take a timed challenge quiz in a chosen category

Alice, on the homepage of the application, sees a **list of categories** to choose from, such as history, science, sports, etc. She selects history and taps on the take quiz button. The app shows her **a question** with four possible answers and a **timer**. She has 20 seconds to select the correct answer. She answers the first question correctly, she earns points and moves on to the next question. She answers the second question incorrectly and runs out of time in the third question, the quiz moves on to the next question without giving any points. She completes the quiz and after the quiz ends, she sees her **score** and **rank** on the leaderboard.

### 3.2.5. The system shall allow users to participate in daily challenges

Alice chooses to enter the daily challenge quiz and clicks on the **daily challenge** button in the homepage. The app shows her 10 mixed category questions one by one. For every correct answer she earns points. After she completes all the questions, she sees the quiz score and her rank on the daily **leaderboard**.

### 3.2.6. The system shall allow users to see their profiles, change their username or password

Alice, thinking that her password is no longer secure, wants to change her password. She clicks on the profile button in the tab bar. She can see her username and several buttons named change username and change password. She clicks on the **change password button** and she is shown a form where she needs to enter her current password and the new password. After she correctly inputs passwords her password is changed.

**3.2.7. The system should show the users their streak count**

Alice, on her profile page, sees her streak count. She continues her streak every day she completes at least one quiz.

**3.2.8. Users shall be able to follow other users**

Alice can follow other users. She entered the profile section of another user and clicked the follow button.

**3.2.9. Users should be able to submit new questions**

Alice submits a custom question to be reviewed and potentially included into the question list.

**3.2.10. Users shall be able to view their followers and people that they follow**

Alice can view people who follow her and people she follows. She goes to her profile and heads to the appropriate section where these people are listed. To select between followers and followings, she just switches between tabs.

**3.2.11. The system shall allow admins to view and ban users**

Bob, an admin user, enters the system management interface. He can see **all the users in the system**. He sees that one of the users has an offensive username. He clicks on the **ban user** button, shown alongside the user in the list. The user with the offensive username can no longer access the application.

**3.2.12. The system shall allow admins to promote users to moderators**

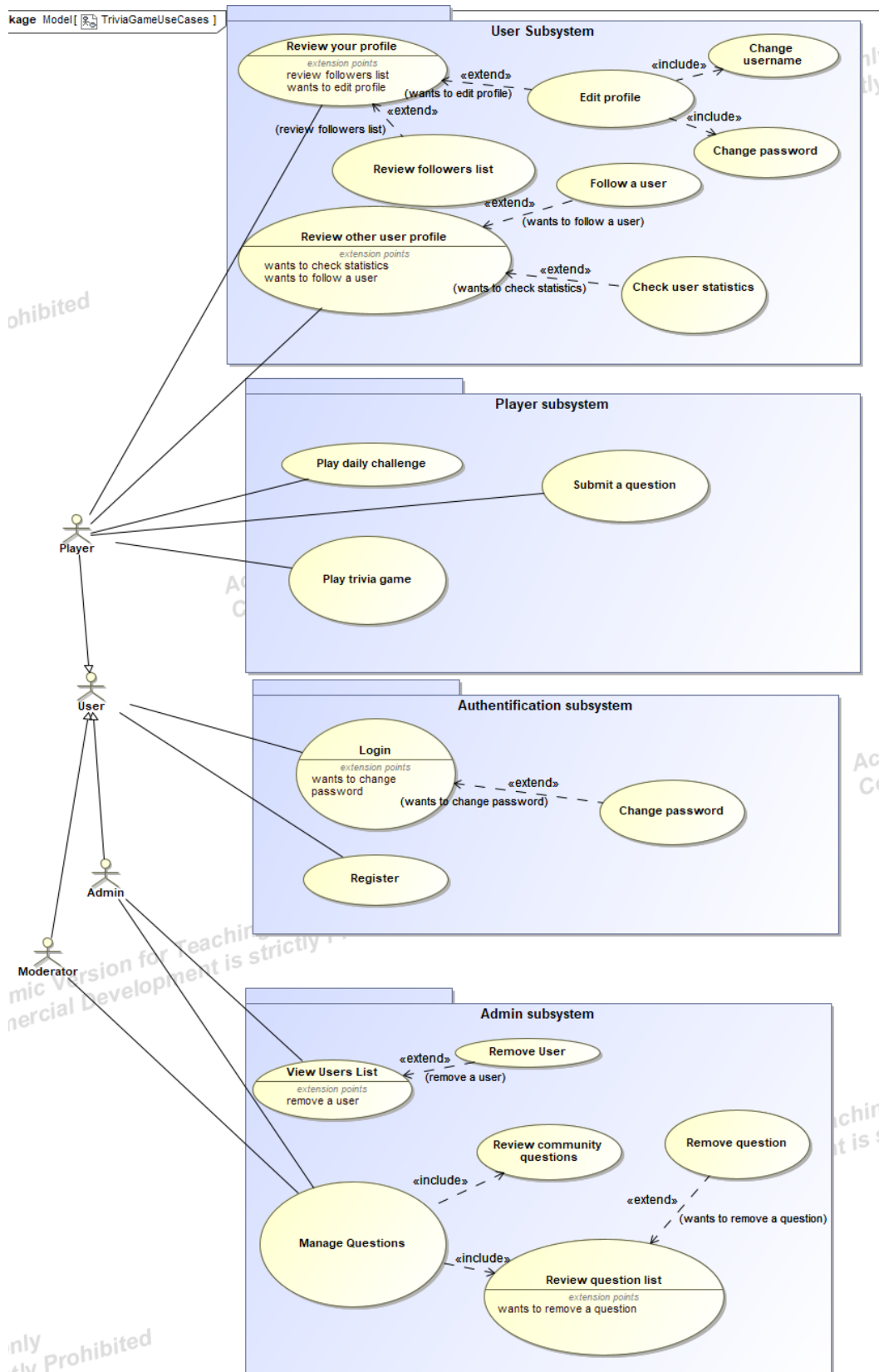
Bob, an admin user, wants to have someone as a moderator in the system. He opens up the user list in the management interface and clicks the **make moderator button** of the user Charlie. Charlie is now a moderator.

**3.2.13. The system shall allow admins and moderators to view, edit, add and remove quiz questions.**

Charlie, a moderator, wants to **remove a question** as he sees that the question has an ambiguous answer. He enters the management interface and clicks on the questions button. He is shown a **list of questions** with **category filters**. He finds the question and clicks on the remove question button. The question is now removed from the system and **will no longer be included** in the quizzes.

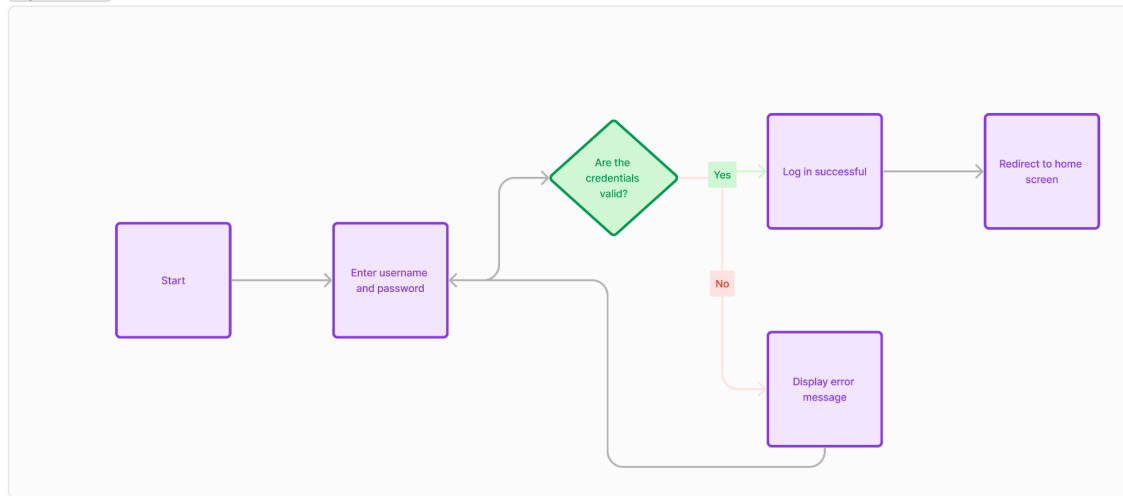
### 3.3. Use Case Diagram

#### 3.3.1. System Use Case Diagram

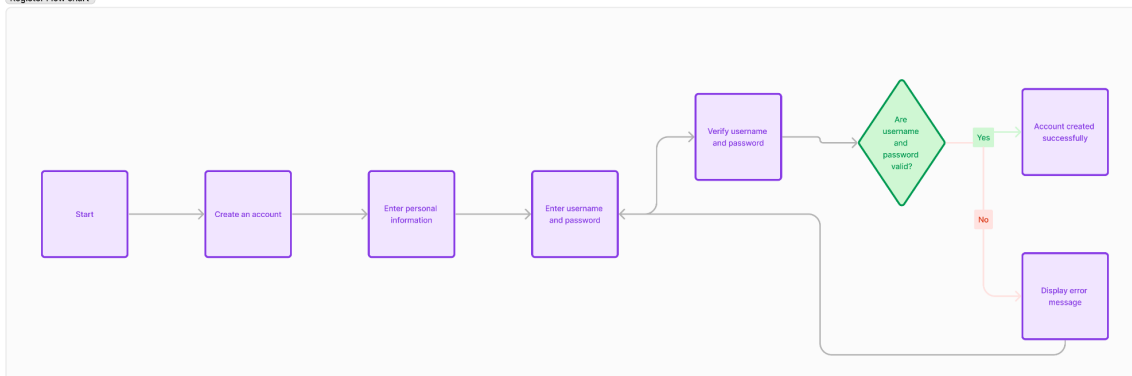


### 3.3.2. Functionality Specific Diagrams

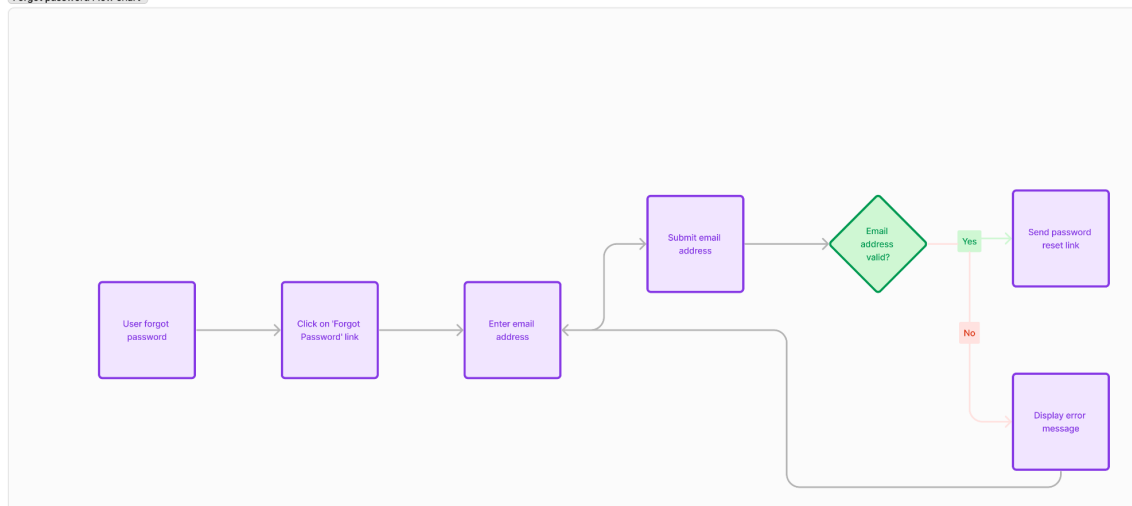
Log in flow chart



Register Flow chart

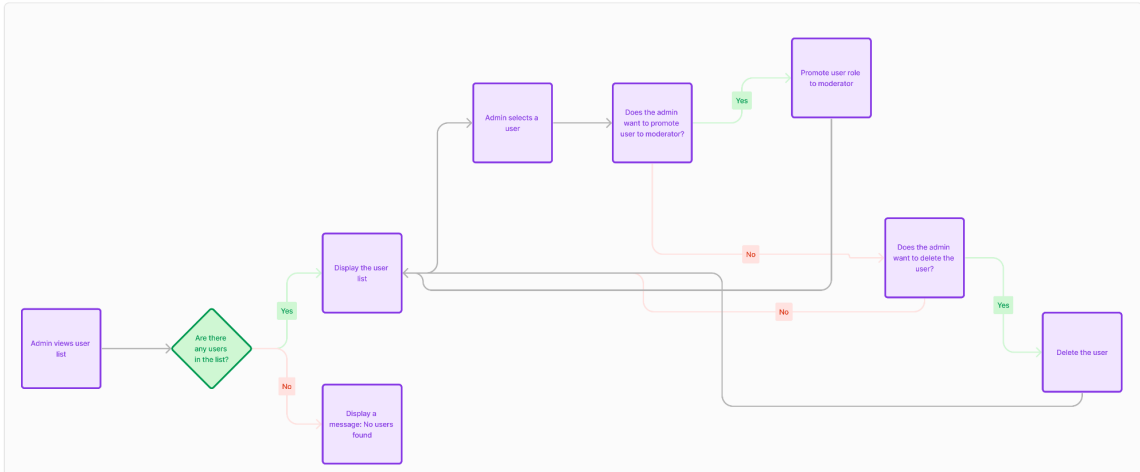


Forgot password Flow chart

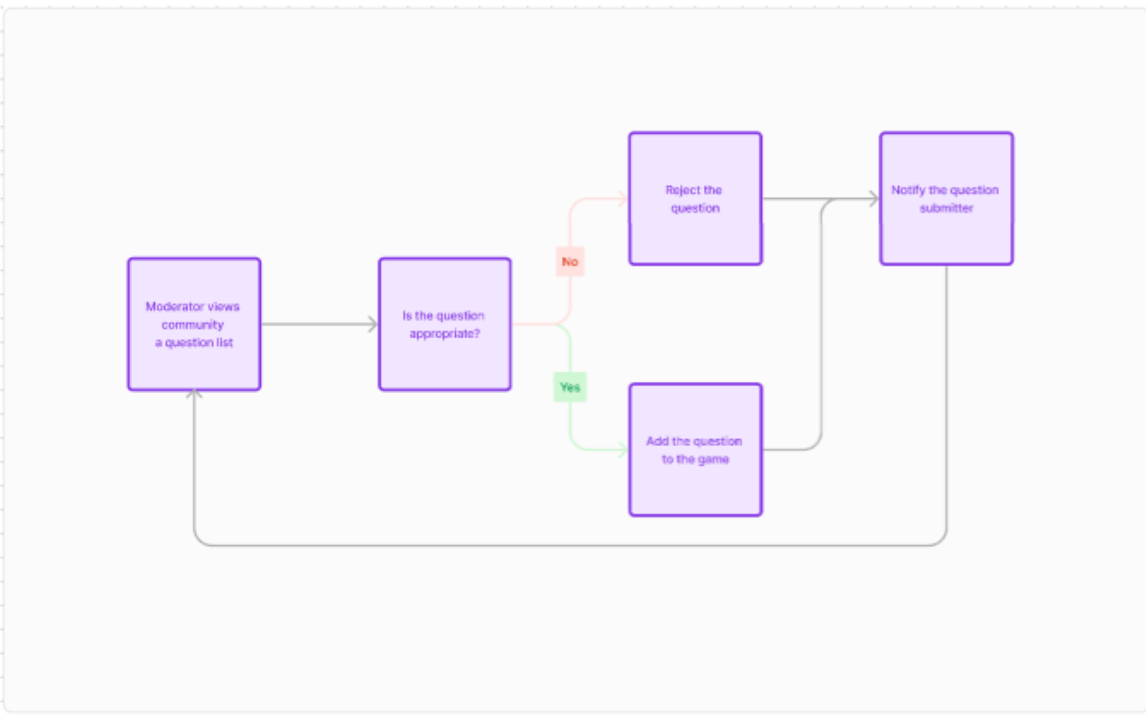




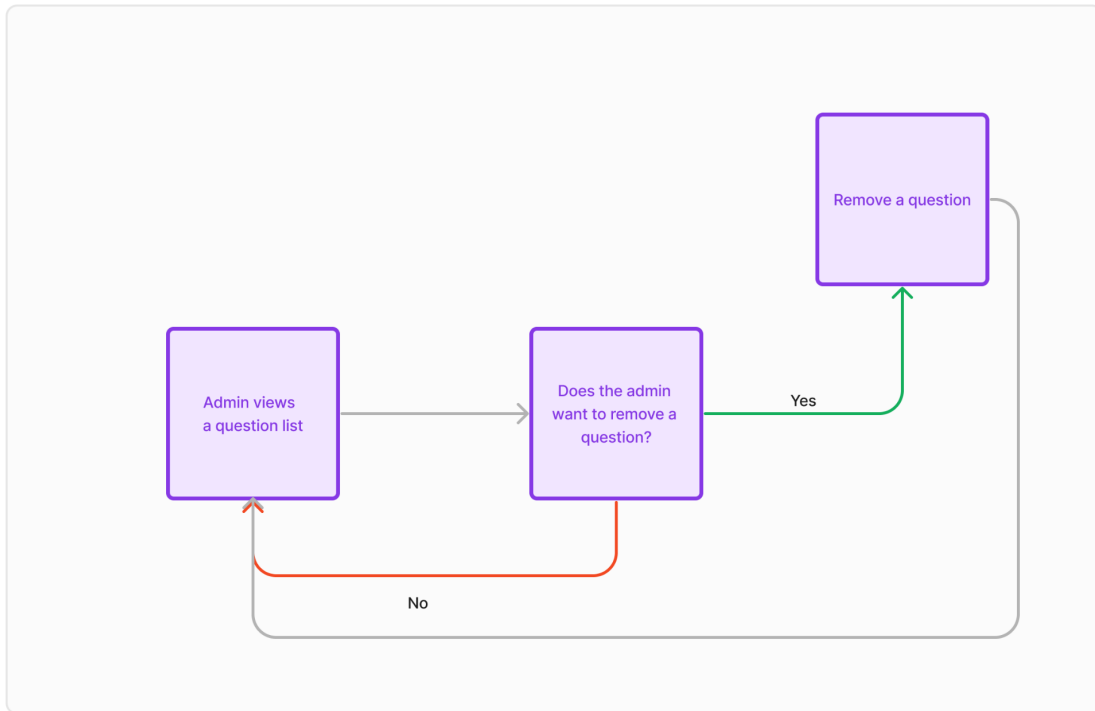
View user list Flow chart



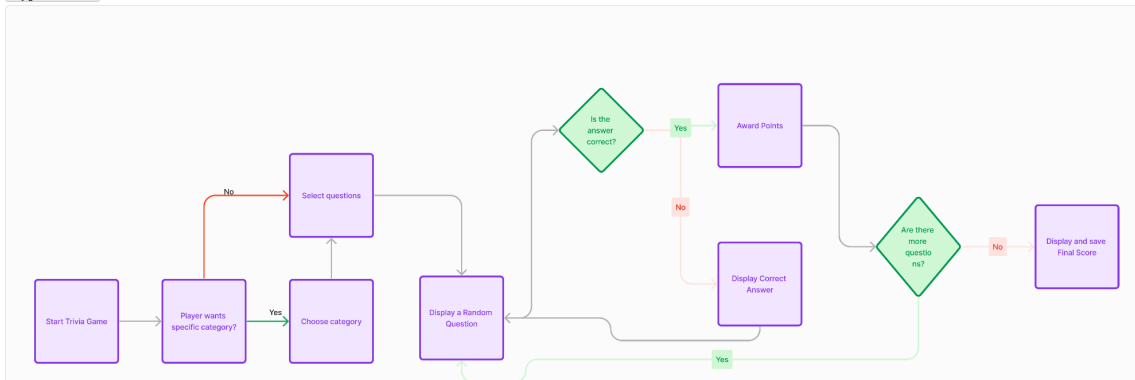
review community questions Flow chart



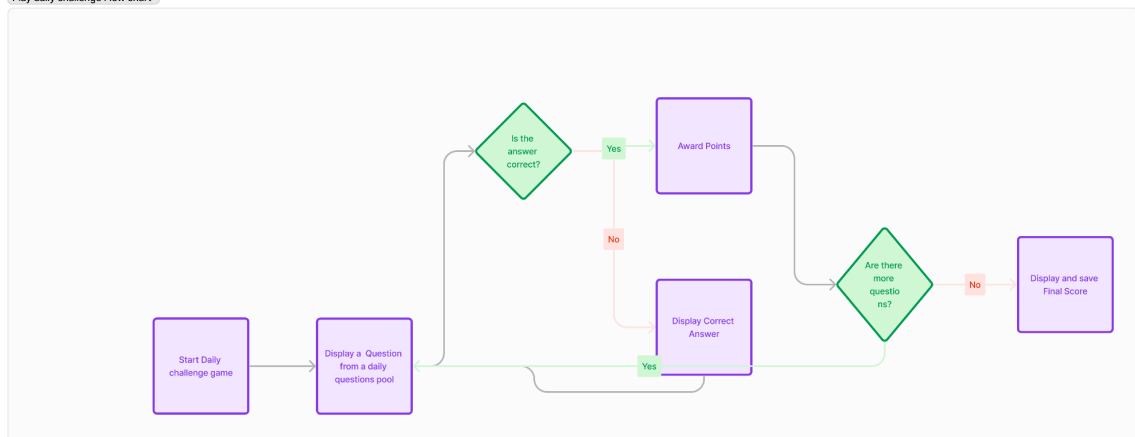
review questions list Flow chart



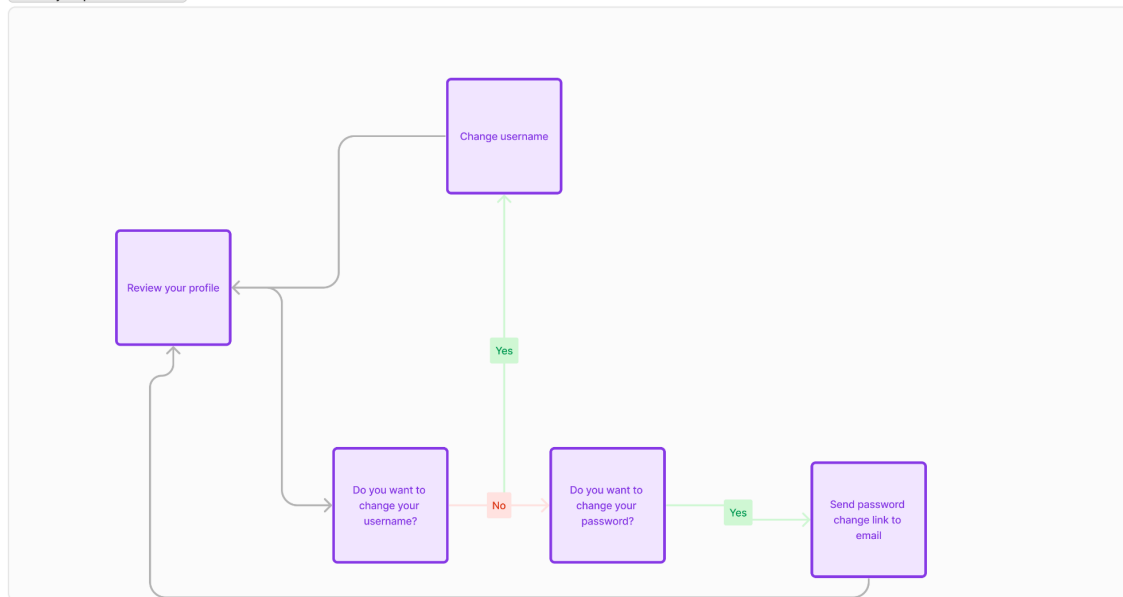
Play game Flow chart



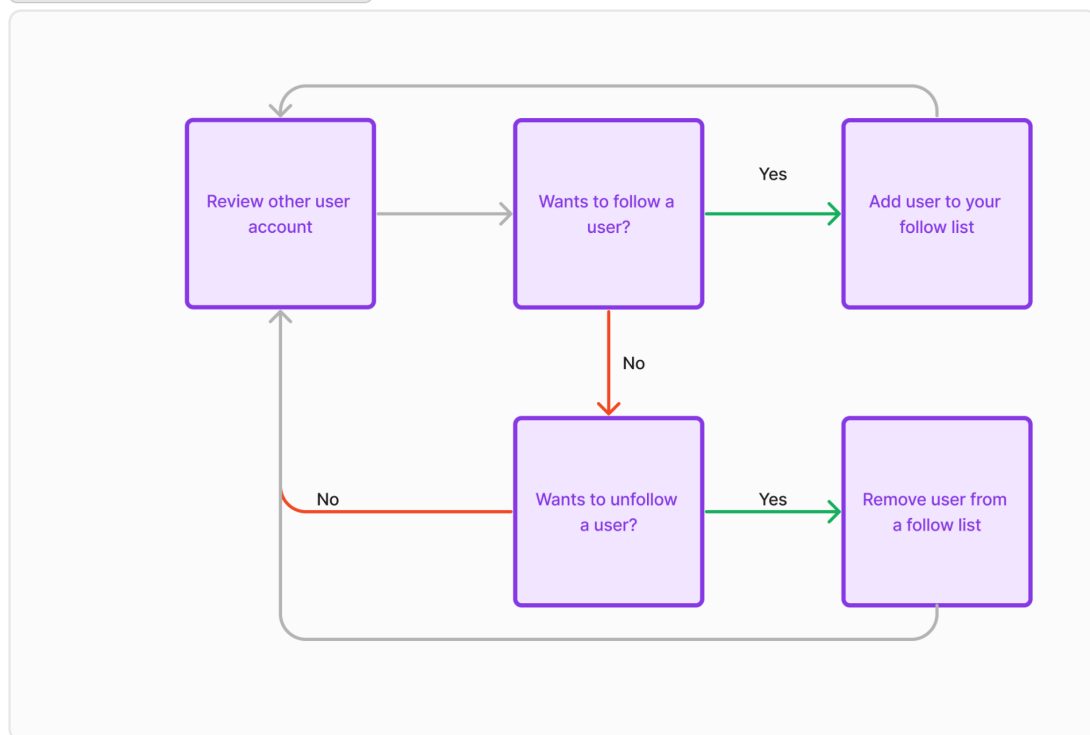
Play daily challenge Flow chart



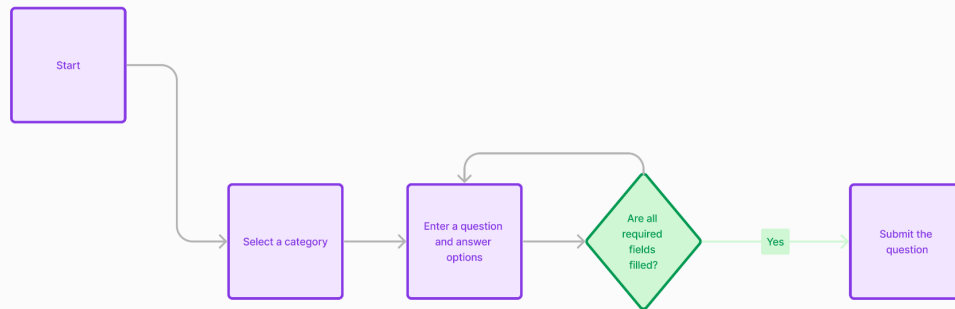
Review your profile Flow chart



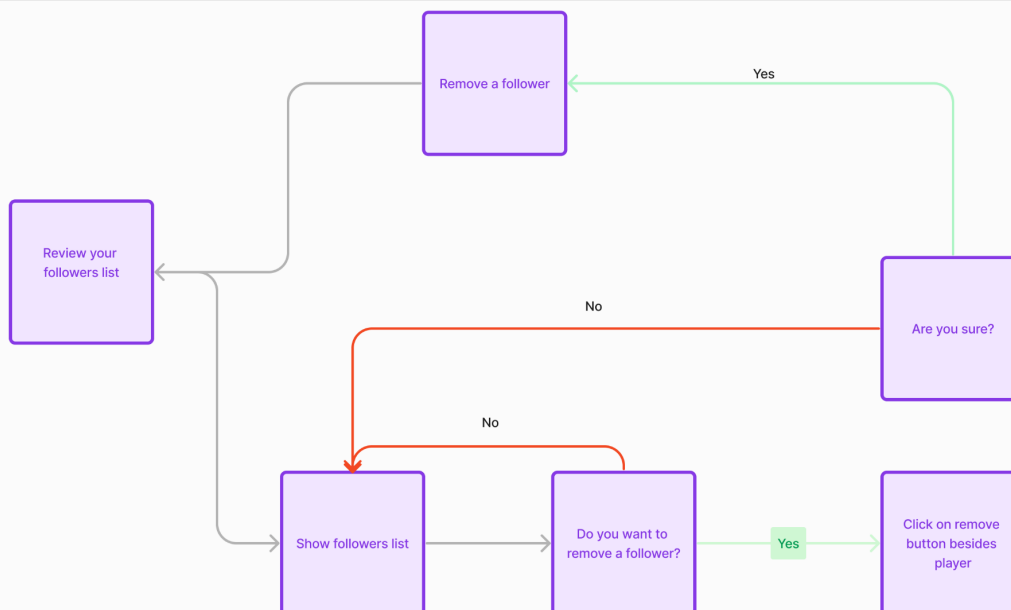
Review other user account Flow chart

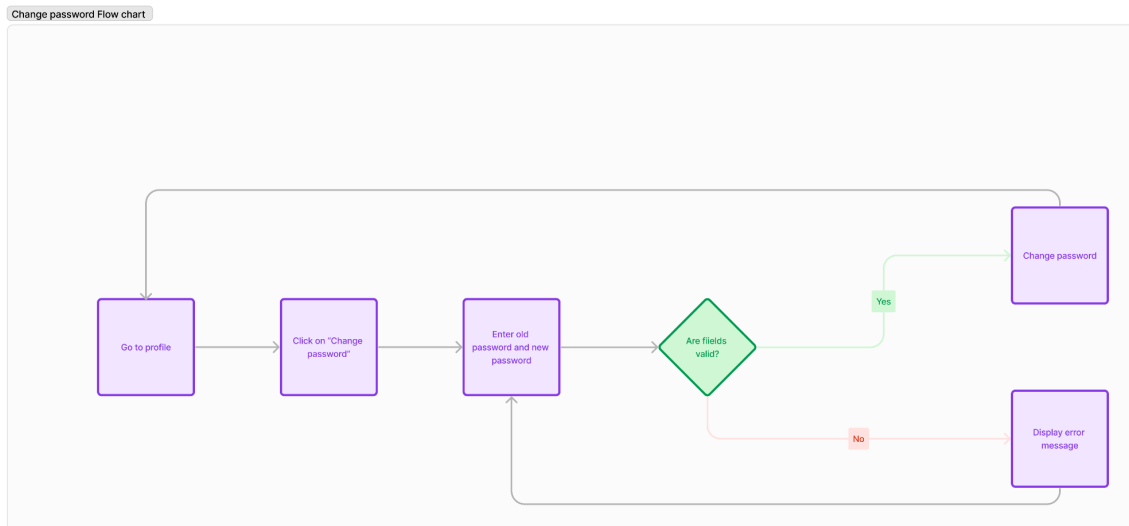


Submit a question Flow chart



Review your followers list Flow chart





### 3.4. Use Cases with Flows

Register	Change password	View followed users	Review question list
Login	Play a daily challenge	Review other user's profile	Review community questions
Submit a question	Play trivia game	Remove user	Compare statistics between users
Change username	View followers	Remove question	View users list

#### 3.4.1. Registration

<b>Primary Actor:</b>	Guest
<b>Brief:</b>	User should register in order to use the application
<b>Basic Flow:</b>	<ol style="list-style-type: none"> <li>1. Guest directed to the registration page</li> <li>2. Requested credentials (Email, Username, Password) supplied by user and submitted</li> <li>3. Credentials validated, added to database and user redirected to the page for registered users</li> </ol>
<b>Alternative Flow:</b>	<ol style="list-style-type: none"> <li>1. Guest directed to the registration page</li> <li>2. Requested credentials (Email, Username, Password) supplied by user and submitted</li> <li>3. Credentials invalid (i.e invalid email, password not long enough, username already used etc.)</li> <li>4. An alert stating the error is shown to the user.</li> </ol>

### 3.4.2. Logging in

<b>Primary Actor:</b>	Registered User
<b>Brief:</b>	User should login in order to use the application
<b>Basic Flow:</b>	<ol style="list-style-type: none"><li>1. User directed to the login page</li><li>2. User enters username and password</li><li>3. Credentials valid</li><li>4. User redirected to the page for registered users</li></ol>
<b>Alternative Flow:</b>	<ol style="list-style-type: none"><li>1. User directed to the login page</li><li>2. User enters username and password</li><li>3. Credentials invalid</li><li>4. An alert saying invalid credentials is shown to users</li></ol>

### 3.4.3. Submit a question

<b>Primary Actor:</b>	Registered User
<b>Brief:</b>	User should be able to submit questions to be included in the game
<b>Basic Flow:</b>	<ol style="list-style-type: none"><li>1. User clicks the "Submit a Question" button</li><li>2. User selects a category from the available ones</li><li>3. User is asked to type the question, and the answer options</li><li>4. User clicks the submit button</li><li>5. A check is done to see if all the required fields are filled</li><li>6. Check passed</li><li>7. Question is sent to review by the moderator, and a success message is shown to the user</li></ol>
<b>Alternative Flow:</b>	<ol style="list-style-type: none"><li>1. User clicks the "Submit a Question" button</li><li>2. User selects a category from the available ones</li><li>3. User is asked to type the question, and the answer options</li><li>4. User clicks the submit button</li><li>5. A check is done to see if all the required fields are filled</li><li>6. Check failed</li><li>7. An error message is shown to the user</li></ol>

### 3.4.4. Change username

<b>Primary Actor:</b>	Registered User
<b>Brief:</b>	User should be able to change their username
<b>Basic Flow:</b>	<ol style="list-style-type: none"><li>1. User clicks the edit button next to their username</li><li>2. A new username is typed by the user</li><li>3. The username availability is checked</li><li>4. A green tick is shown next to the field if the username is available</li><li>5. User clicks confirm button</li><li>6. Username is changed</li></ol>

<b>Alternative Flow:</b>	<ol style="list-style-type: none"> <li>1. User clicks the edit button next to their username</li> <li>2. A new username is typed by the user</li> <li>3. The username availability is checked</li> <li>4. A red cross is shown next to the field if the username is not available</li> <li>5. Confirm button is not clickable</li> </ol>
--------------------------	--

### 3.4.5. Change password

<b>Primary Actor:</b>	Registered User
<b>Brief:</b>	User should be able to change their password
<b>Basic Flow:</b>	<ol style="list-style-type: none"> <li>1. User clicks the change password button</li> <li>2. The old password and new password is asked to the user, along with a confirmation of the new password</li> <li>3. User clicks confirm button</li> <li>4. The old password is validated and the typed passwords match</li> <li>5. Password is changed and a message is shown to the user</li> </ol>
<b>Alternative Flow:</b>	<ol style="list-style-type: none"> <li>1. User clicks the change password button</li> <li>2. The old password and new password is asked to the user, along with a confirmation of the new password</li> <li>3. User clicks confirm button</li> <li>4. The old password is not validated, or the typed passwords do not match</li> <li>5. Password is not changed and the relevant error message is shown to user</li> </ol>

### 3.4.6. Play a daily challenge

<b>Primary Actor:</b>	Registered User
<b>Brief:</b>	User should be able to participate in daily challenges
<b>Basic Flow:</b>	<ol style="list-style-type: none"> <li>1. User clicks on the daily challenge button</li> <li>2. A question is displayed to the user until all questions are over (10 questions)</li> <li>3. User clicks on their answer for every question</li> <li>4. If their answer is correct, points are added. If not, the correct answer is shown and no points are added.</li> <li>5. Quiz finishes and the quiz score is displayed, along with the user's rank in the daily leaderboard.</li> </ol>
<b>Alternative Flow:</b>	<ol style="list-style-type: none"> <li>1. User clicks on the daily challenge button</li> <li>2. A question is displayed to the user until all questions are over (10 questions)</li> <li>3. User decides to quit by pressing the proper button any time throughout the quiz</li> <li>4. No points are added to account stats, and the user is returned to the home page.</li> </ol>

### 3.4.7. Play trivia game

<b>Primary Actor:</b>	Registered User
<b>Brief:</b>	User should be able to start playing a quiz.
<b>Basic Flow:</b>	<ol style="list-style-type: none"><li>1. User enters quiz categories section</li><li>2. User chooses category they want to play</li><li>3. The quiz starts</li><li>4. User selects answers and submits it for each individual question</li><li>5. Correct answer is shown after submitting the answer</li><li>6. Quiz finishes and the summary is displayed.</li><li>7. User is awarded points for each correctly answered question.</li><li>8. User score is updated - point for the quiz added to his score.</li></ol>
<b>Alternative Flow:</b>	<ol style="list-style-type: none"><li>1. User enters quiz categories section</li><li>2. User chooses category he wants to play</li><li>3. The quiz starts</li><li>4. User decides to quit by pressing the proper button</li><li>5. No points are added to account stats</li></ol>

### 3.4.8. View followers

<b>Primary Actor:</b>	Registered User
<b>Brief:</b>	Allows user to view people who follow him
<b>Basic Flow:</b>	<ol style="list-style-type: none"><li>1. User enters his profile section</li><li>2. User clicks followers button</li><li>3. List of followers is displayed</li></ol>
<b>Alternative Flow:</b>	None

### 3.4.9. View followed users

<b>Primary Actor:</b>	Registered User
<b>Brief:</b>	Allows user to view people that he follows
<b>Basic Flow:</b>	<ol style="list-style-type: none"><li>1. User enters his profile section</li><li>2. User clicks following button</li><li>3. List of followed users is displayed</li></ol>
<b>Alternative Flow:</b>	None

### 3.4.10. Review other user's profile

<b>Primary Actor:</b>	Registered User
<b>Brief:</b>	Allows to view other user's profile page – information about that user



<b>Basic Flow:</b>	<ol style="list-style-type: none"> <li>1. User searches for and finds other user</li> <li>2. User clicks on that user</li> <li>3. User is redirected to the other user's profile</li> <li>4. The profile displayed statistics about that user as well as comparisons between them.</li> </ol>
<b>Alternative Flow:</b>	<ol style="list-style-type: none"> <li>1. User finds the user in his followers</li> <li>2. User clicks on that user</li> <li>3. User is redirected to the other user's profile</li> <li>4. The profile displayed statistics about that user as well as comparisons between them.</li> </ol>

#### 3.4.11. Remove user

<b>Primary Actor:</b>	Administrator
<b>Brief:</b>	User requests to delete their account or administrator removes a user due to violations, inactivity or other reasons.
<b>Basic Flow:</b>	<ol style="list-style-type: none"> <li>1. Administrator logs into the admin dashboard and navigates to the user management section.</li> <li>2. Administrator searches for a user and removes them.</li> <li>3. System asks for confirmation.</li> <li>4. Administrator confirms the removal.</li> <li>5. System deletes the account and associated data from the database.</li> </ol>
<b>Alternative Flow:</b>	<ol style="list-style-type: none"> <li>1. Administrator logs into the admin dashboard and navigates to the user management section.</li> <li>2. Administrator searches for a user to remove.</li> <li>3. User does not exist or has already been removed, so the system prompts an error message.</li> </ol>

#### 3.4.12. Remove question

<b>Primary Actor:</b>	Moderator
<b>Brief:</b>	Moderator removes a question from the game due to inaccuracies or irrelevance.
<b>Basic Flow:</b>	<ol style="list-style-type: none"> <li>1. Moderator accesses the moderator dashboard and navigates to the question management section.</li> <li>2. System displays the list of questions.</li> <li>3. Moderator searches and selects the question to remove.</li> <li>4. System asks for confirmation and the moderator confirms the removal.</li> <li>5. System removes the question from the database.</li> </ol>
<b>Alternative Flow:</b>	<ol style="list-style-type: none"> <li>1. Moderator accesses the moderator dashboard and navigates to the question management section.</li> <li>2. System displays the list of questions.</li> <li>3. Moderator searches for the question to remove.</li> <li>4. Question does not exist so the system prompts an error</li> </ol>

	message.
--	----------

#### 3.4.13. Review question list

<b>Primary Actor:</b>	Moderator
<b>Brief:</b>	Moderator reviews the list of available questions in the app.
<b>Basic Flow:</b>	<ol style="list-style-type: none"> <li>1. Moderator logs into the moderator dashboard and navigates to the question management section.</li> <li>2. System displays the list of available questions.</li> <li>3. Moderator views the questions, sorted by categories or other filters.</li> <li>4. Moderator may edit, add, or remove questions as necessary.</li> </ol>
<b>Alternative Flow:</b>	None

#### 3.4.14. Review community questions

<b>Primary Actor:</b>	Moderator
<b>Brief:</b>	Moderator reviews user-submitted questions before inclusion in the question database.
<b>Basic Flow:</b>	<ol style="list-style-type: none"> <li>1. Moderator accesses the community question review section in the moderator dashboard.</li> <li>2. System displays a list of pending community-submitted questions.</li> <li>3. Moderator reviews each question for accuracy and relevance.</li> <li>4. Question is approved and is added to the database.</li> <li>5. The user that has submitted the question gets notified.</li> </ol>
<b>Alternative Flow:</b>	<ol style="list-style-type: none"> <li>1. Moderator accesses the community question review section in the moderator dashboard.</li> <li>2. System displays a list of pending community-submitted questions.</li> <li>3. Moderator reviews each question for accuracy and relevance.</li> <li>4. Question is rejected.</li> <li>5. User gets notification for the rejected question.</li> </ol>

#### 3.4.15. Compare statistics between users

<b>Primary Actor:</b>	Registered user
<b>Brief:</b>	User checks for statistical comparison with another user.
<b>Basic Flow:</b>	<ol style="list-style-type: none"> <li>1. User visits another user's profile</li> <li>2. User chooses to compare their statistics.</li> <li>3. Weekly comparison visually displayed.</li> </ol>
<b>Alternative Flow:</b>	<ol style="list-style-type: none"> <li>1. User visits another user's profile</li> <li>2. User chooses to compare their statistics.</li> </ol>

	3. System prompts a message that no comparison can be made due to lack of statistics from the other user.
--	---

#### 3.4.16. View users list

<b>Primary Actor:</b>	Administrator
<b>Brief:</b>	Administrator accesses the complete list of users registered in the app.
<b>Basic Flow:</b>	<ol style="list-style-type: none"> <li>1. Administrator logs into the admin dashboard and navigates to the user management section.</li> <li>2. System displays the complete list of registered users and the associated data.</li> <li>3. Administrator can sort, filter, search, delete specific users or choose to promote a user to moderator.</li> </ol>
<b>Alternative Flow:</b>	<ol style="list-style-type: none"> <li>1. Administrator logs into the admin dashboard and navigates to the user management section.</li> <li>2. System prompts a message that there are no registered users.</li> </ol>