

ELICITATION:

Target Users:

Name	Email
Jacky	jackyzibm@hotmail.com
Aaliyah	shayturner759@gmail.com

Interview:

Question 1: For what purpose do you use Kahoot, or a similar quiz platform?	
Jacky	Mostly to participate in quizzes, rarely creating my own.
Aaliyah	To participate in quizzes at school.
Question 2: What features of the quiz platform do you like?	
Jacky	The music, layout, ease of joining, how easy it is to create quizzes.
Aaliyah	It's easy to access by just inputting a code.
Question 3: What features of the quiz platform do you dislike?	
Jacky	Over reliance on everyone being in the same room or needing to be able to see a central display.
Aaliyah	You cannot see the question or answers on your screen.
Question 4: Are there any additional features of the quiz platform you would like to see?	
Jacky	Live/pre quiz betting
Aaliyah	It would be good if you could log in yourself and have all your results saved from each kahoot you have participated in.

Proposed solution:

A common problem identified by the target users is being unable to view both the question and answers on their device as a player. Toohak currently addresses this in the HTTP route '/v1/player/{playerid}/question/{questionposition}', allowing players to get information about a question that they are on. In order to increase engagement, an additional feature of pre-quiz betting is being proposed. This would allow players who are also logged in users to declare an amount of 'Toohak dollars' on which player the user believes will win. Also, in order to track personal progress, a proposed solution is an option to view session result history. This would allow players who are also registered users to view results from previous quiz sessions that were participated in while logged in.

ANALYSIS & SPECIFICATION

User Stories:

User Story	User Acceptance Criteria:
As a player, I want to see the quiz question and answers on my screen, so that I can participate in a quiz session remotely.	<p>Scenario: Wants to view question information Given: The player joined a quiz session And: The quiz session was not in LOBBY, QUESTION_COUNTDOWN, or END state When: The player selected the 'question information' option And: Entered a valid playerId and position of the current question Then: The system displayed a copy of the question information, excluding which answers are correct.</p>
As a logged in player, I want to view my quiz session results, so that I can track my progress over multiple sessions.	<p>Scenario: Wants to view all quiz session results Given: The registered user was a logged in When: The user selected the 'all quiz session results' option And: Entered a valid token Then: The system displayed a list of personal results of quiz sessions participated in whilst logged in.</p>
As a logged in player, I want to engage in pre-quiz betting, so that I can have a higher incentive to achieve greater results.	<p>Scenario: Wants to engage in pre-quiz betting Given: The user is unregistered When: The user selected the 'register' option And: Entered a valid email, password, first name, and last name Then: The system displayed a token. And: An initial \$10 was added to their Tahooq wallet. Given: The user joined a quiz session And: The quiz session was in LOBBY When: The player selected the 'place bet' option And: Entered a valid token, sessionId, dollar amount less than or equal to that in their wallet, and an existing player to win Then: The system returned to LOBBY Given: The quiz session is in FINAL_RESULTS When: The player selected the 'view wallet' option And: Entered a valid token Then: The system displayed the user's wallet with an increase or decrease by the dollar amount, depending on the bet's success.</p>

Use Cases:

Background	Use-Case List
<p>Use Case: previous quiz results Scope: User views all quiz sessions results participated in whilst logged in Preconditions: user is logged in Success End Condition: quiz sessions results are viewed Failed End Condition: quiz sessions results are not viewed Primary Actor: Logged in user</p>	<p>Step 1: Website displays homepage state with the option 'View Results' Step 2: User selects 'View Results' Step 3: Website displays a list of personal results of quiz sessions participated in whilst logged in.</p>
<p>Use Case: pre-quiz betting Scope: player creates a bet on the winner of the quiz Preconditions: player is a logged in user, have successfully joined a quiz session Success End Condition: Bet is made Failed End Condition: Bet is not made due to insufficient funds or invalid winner selection Primary Actor: Logged in player Trigger: quiz session is in LOBBY state</p>	<p>Step 1: Website displays session lobby state with the option 'Place Bet' Step 2: User selects 'Place Bet' Step 3: Website asks for dollar amount and player name Step 4: User enters dollar amount and player name Step 5: Website verifies the validity of inputted information Step 6: Website displays session lobby state Step 7: Website displays session question countdown state Step 8: Website displays session question open state with different answer options Step 9: User selects an answer Step 10: Website displays session question close state Step 11: Website displays session question close state Step 12: Step 7-11 repeated for all quiz questions Step 13: Website displays session final results state with the option 'View Bet Results' Step 14: User selects 'View Bet Results' Step 15: Website displays wallet balance with an increase or decrease by the dollar amount, depending on the bet's success.</p>

VALIDATION:

Comments:

Jacky (in response to pre-quiz betting user case) - 'I think you did it perfectly, that's exactly what I was thinking.'

Aaliyah (in response to previous quiz results user case) - 'This looks great, it would be very helpful for study.'

INTERFACE DESIGN:

Solution 1 - Previous Quiz Results

POST	/v2/player/join
If the name entered is an empty string, a name must be randomly generated that conforms to the structure "[5 letters][3 numbers]" (e.g. valid123, ifjru483, ofijr938) where there are no repetitions of numbers or characters within the same name. <i>Allows both logged in and unregistered players.</i>	
Parameters	
Name	Description
token : - optional - (header)	token
body : - required - object - (body)	{ "sessionId": number, "Name": string }
Responses	
Code	Description
200	OK { "playerId": number }
400	If any of the following are true: <ul style="list-style-type: none"> - Name of user entered is not unique (compared to other users who have already joined) - Session Id does not refer to a valid session - Session is not in LOBBY state { "error": string }
401	Token is invalid (does not refer to valid logged in user session - do not throw error if token is empty) { "error": string }

PUT	/v2/player/{playerid}/question/{questionposition}/answer
Allow the current player to submit answer(s) to the currently active question. Question position starts at 1. <i>If a player is logged in, a copy of answers and results is made.</i> Note: An answer can be re-submitted once first selection is made, as long as game is in the right state	
Parameters	
Name	Description
token : - optional - (header)	token
playerId: - required - (path)	number
questionPosition: - required - (path)	number
body : - required - object - (body)	{ "answerIds": number[] }
Responses	
Code	Description
200	OK {}
400	If any of the following are true: <ul style="list-style-type: none"> - If player ID does not exist - If question position is not valid for the session this player is in - Session is not in QUESTION_OPEN state - If session is not yet up to this question - Answer IDs are not valid for this particular question - There are duplicate answer IDs provided - Less than 1 answer ID was submitted { "error": string }
403	Player did not join with given token (do not throw error if token is empty) { "error": string }

401	<p>Token is invalid (does not refer to valid logged in user session - do not throw error if token is empty)</p> <pre>{ "error": string }</pre>
-----	--

GET	/v1/admin/quiz/results
For the given admin user that is logged in, return all personal results of quizzes participated in whilst logged in.	
Parameters	
Name	Description
token : - required - (header)	token
Responses	
Code	Description
200	<p>OK</p> <pre>{ "results": [{ "sessionId": number, "questions": [{ "questionPosition": number, "question": string, "answers": [{ "answer": string, "correct": boolean }], "score": number }] }] }</pre>
401	<p>Token is empty or invalid (does not refer to valid logged in user session)</p> <pre>{ "error": string }</pre>

Solution 2 - Pre-Quiz Betting:

POST	/v2/admin/auth/register
Takes in information about a new admin user and registers them in the system. This route is not relevant to guests who want to play a particular quiz, but is used for the creation of accounts of people who manage quizzes. A wallet is initialised with a starting balance of \$10.	
Parameters	
Name	Description
body : - required - object - (body)	{ "email": string, "Password": string, "nameFirst": string, "nameLast": string }
Responses	
Code	Description
200	OK { "token": string }
400	If any of the following are true: <ul style="list-style-type: none"> - Email address is used by another user - Email does not satisfy this: https://www.npmjs.com/package/validator (validator.isEmail) - NameFirst contains characters other than lowercase letters, uppercase letters, spaces, hyphens, or apostrophes - NameFirst is less than 2 characters or more than 20 characters - NameLast contains characters other than lowercase letters, uppercase letters, spaces, hyphens, or apostrophes - NameLast is less than 2 characters or more than 20 characters - Password is less than 8 characters - Password does not contain at least one number and at least one letter { "error": string }
GET	/v3/admin/user/details
For the given admin user that is logged in, return all of the relevant details.	
Parameters	

Name	Description
token : - required - (header)	token
Responses	
Code	Description
200	<ul style="list-style-type: none"> - "name" is the first and last name concatenated with a single space between them - numSuccessfulLogins includes logins direct via registration, and is counted from the moment of registration starting at 1 - numFailedPasswordsSinceLastLogin is reset every time they have a successful login, and simply counts the number of attempted logins that failed due to incorrect password, only since the last login <pre>{ "user": { "userId": number, "name": string, "email": string, "numSuccessLogins": number, "numFailedPasswordsSinceLastLogin": number, "wallet": number } }</pre>
401	<p>Token is empty or invalid (does not refer to valid logged in user session)</p> <pre>{ "error": string }</pre>

POST	/v2/player/join
See 'Solution 1 - Previous Quiz Results'	

POST	/v1/player/{playerid}/bet
Allow logged in players to place a bet with Toohak dollars on the predicted player with the highest score.	
Parameters	
Name	Description
token : - required - (header)	token

body : - required - object - (body)	<pre>{ "dollarAmount": number, "winner": string }</pre>
Responses	
Code	Description
200	<p>OK</p> <pre>{}</pre>
400	<p>If any of the following are true:</p> <ul style="list-style-type: none"> - dollarAmount is less than 0 - sum of bets made exceeds what is in the players wallet - winner is not the name of a current player in the session - session is not in LOBBY state <pre>{ "error": string }</pre>
401	<p>Token is empty or invalid (does not refer to valid logged in user session)</p> <pre>{ "error": string }</pre>

GET	/v1/player/{playerid}/bet/results
Allow logged in players to place a bet with Toohak dollars on the predicted player with the highest score.	
Parameters	
Name	Description
token : - required - (header)	token
Responses	
Code	Description
200	<p>OK</p> <ul style="list-style-type: none"> - betResult is positive for wins and negative for losses <pre>{ "betResult": number, "wallet": number }</pre>

400	If any of the following are true: - session is not in FINAL_RESULTS or END state { “error”: string }
401	Token is empty or invalid (does not refer to valid logged in user session) { “error”: string }

CONCEPTUAL MODELLING:

State Diagram

