

# Planning Report

## Questions:

1. In what context do you usually use Toohak for and how often?
2. Is it intuitive to use? How?
3. Would you want to be able to sort/ filter your quiz list and what by?
4. Would you want to use and share quizzes without “log in” or “sign up”?
5. Is there any other similar application which you use and find better than this, and were there any specific features you liked?
6. Would you like a history of quiz played? Either as quiz owner or a quiz player?
7. What else would you like to include in Toohak to make it easier for you to use?

## Elicitation:

### **Interview 1**

*Q1. I use kahoot to participate in lecture quizzes*

*Q2. Yes, only a barcode is needed to access the quiz*

*Q3. I would want them to be sorted based on the date last edited/ added*

*Q4. No, I don't want my quiz to be accessed by anyone but me*

*Q5. No*

*Q6. Yes, that would help during practice for a test/exam*

*Q7. Make a graph that shows the percentage of accuracy i have on questions when i take a certain amount of time to answer. For example 60% when i take 1 second, 80 % when i take 2 seconds before answering, 84% when i take 3, and so on.*

### **Interview 2**

*Q1: I mainly use Toohak to take part in quizzes for my classes. I would say the frequency of usage is probably once a month.*

*Q2: Yes it is intuitive to use. I mainly use it for revision.*

Q3: Yes I would like to be able to sort my quizzes based on difficulty and time of creation.

Q4: I would like to use login and signup so I have an idea of who is receiving my quizzes.

Q5: Kahoot and I liked the question-by-question stats.

Q6: Yes, I would.

Q7: Maybe a stat that shows the accuracy rate of a player for all previously attempted quizzes as one cumulative score.

### **Interview 3**

Q1: I have used kahoot in my middle school when teachers used to test us unofficially in our classes. It used to me once every week or once every two weeks

Q2: It is intuitive to use because it can handle a lot of participants at once to test

Q3: Yes. By historical performance in quizzes, low performing participants get easier questions and vice versa

Q4: No

Q5: Yes. Quizizz. They give rewards and power ups randomly which makes the quiz fun and seem like a video game

Q6: Yes.

Q7: with Touch ID or facial recognition instead of user ID and password

## **User stories:**

Title: Interview 1
<b>User Story:</b> This user seems to mainly enjoy the use of kahoot for practicing the academic aspects, with a focus on accuracy and the ability to use the kahoot for practice. In essence to be able to use it to prepare for a test and be able to examine their own current progress in the test in the form of a graph as well as for it to be used for exam practice. This could be a notable avenue, with a single player kahoot being able to 'gameify' exam practice with high scores and whatnot.
<b>Desired features:</b> <ul style="list-style-type: none"><li>- The ability to save results as a graph for each person or everyone at once.</li><li>- The ability for a person to get a graph of average percentages for a quiz.</li></ul>

- A single player kahoot which can be played offline with practice questions and a leaderboard for everyone.

Title: Interview 2

**User Story:**

This user seems fairly similar to the first user, desiring to use it for more academic focusses, with also the discussion of a potential leaderboards to compare one's self to others who attempted the quiz as well as a suggestion of keeping track of difficulty with the ability to sort by that, as well as sorting by time of creation.

**Desired features:**

- A leaderboard keeping track of the best scores and percentages for each player
- The ability to view this leaderboard
- The ability to sort by the creation time, including the date of the creation for a quiz.

Title: Interview 3

**User Story:** This user wants a more accessible and fun version of kahoot with more gamey feel, suggesting a combination of fun power ups, the ability for certain people to get easier questions and a simpler touchId/Facial recognition feature for the app. All of which suggests a preference for greater convenience and gameplay over the academic features, though he still mentions sorting by historical performance (covered in interview 2). Of course, having to write new questions for easier players might be a bit of a problem, so instead, allowing a second attempt on a quiz from a while ago could be possible.

**Desired features:**

- Powerup system for players
- The ability for a catchup for certain players
- The ability to take in a touchId/Facial recognition to log in.

## Use cases:

**Title: Single player Mode (interview 1)**

**Actors:** a player

**Preconditions:**

- The quiz must be first created by an admin
- Trigger: a player chooses to activate a quiz in single player mode.

**Main Flow:**

- Player inputs a quizId they wish to practice
- A single player session is started

- A new quizSession is started with only the player, one which runs automatically without admin input.

**Alternate Flows:**

- **Quiz does not exist:** The system informs the player: "No such quiz exists."

**Postconditions:**

- The single player session starts or an error message is shown.
- Action is logged in the system.
- The data (percentile correct, which answers was correct, overall player score) are stored for later access.

**Acceptance Criteria:**

- The player can play a quiz without the need of an admin to start the quiz.
- The data from the quiz is stored somewhere, which can then be accessed later.

**Title: Graphs (interview 1)**

**Actors:** an admin

**Preconditions:**

- The quiz must be first created by an admin
- At least one quiz session (single player or not) has been ran before
- Trigger: the admin requests for a graph specifically

**Main Flow:**

- Admin inputs a quizid and either chooses to request for all or that of a specific player
- The specific data is accessed
- The data is processed in the backend
- A graph is returned

**Alternate Flows:**

- **Token does not exist:** The system informs the admin: "Invalid token."
- **Quiz does not exist/is in bin/is not owned:** The system informs the admin: "Invalid quiz."
- **No sessions for quiz:** the system informs the admin: "there have yet to be any sessions for the quiz."

**Postconditions:**

- The graph is displayed or an error message is shown.
- The data (percentile correct, which answers was correct, overall player score) are stored for later access.

**Acceptance Criteria:**

- The system successfully stores the data related to the graph
- A graph is generated as a .jpg file displayed on the server, which the user can download

**Title: Leaderboard (interview 2)**

**Actors:** admin

**Preconditions:**

- The quiz must be first created by an admin, though any admin can access the leaderboards
- Trigger: the admin checks the leaderboard

**Main Flow:**

- Admin calls a function, which returns the leaderboard, either all or for just the top three of a quiz
- The data should be then processed and returned as a list.

**Alternate Flows:**

- **Token does not exist:** The system informs the admin: "Invalid token."
- **Quiz does not exist/is in bin/is not owned:** The system informs the admin: "Invalid quiz."
- **No sessions for quiz:** the system informs the admin: "there have yet to be any sessions for the quiz."

**Postconditions:**

- Returns a sorted array or an error message is shown.

**Acceptance Criteria:**

- Returns a sorted array of the best performing names at a quiz with their specific scores and percentilecorrect.

**Title: better sorting (interview 2 & 3)**

**Actors:** admin

**Preconditions:**

- At least one quiz which exists, this function will sort all quizzes, not just those created by the specific admin.
- Trigger: specifically called by the admin for information.

**Main Flow:**

- Admin calls one of two functions, with either HIGH or LOW to return the creation dates from highest to lowest or the average percentile (tracked by leaderboard and graph).
- The data should be then processed and returned as a list.

#### **Alternate Flows:**

- **Token does not exist:** The system informs the admin: "Invalid token."
- **No quizzes:** the system informs the admin: "there are no quizzes in the system/"

#### **Postconditions:**

- Returns a sorted array or an error message is shown.

#### **Acceptance Criteria:**

- Returns a sorted array of the best performing names at a quiz with their specific scores and percentile correct.
- The function(s) successfully detect "HIGH" and "LOW" as enums.

### **Title: catchup system (interview 3)**

**Actors:** admin

#### **Preconditions:**

- A session must already be started and be within the LOBBY state.
- Trigger: set up by the admin beforehand

#### **Main Flow:**

- Admin calls the catchup system, with parameters for how common and strong they want the catchup system to be as well as the bottom percentile which the admin wants to grant the catchup bonus to.
- The quiz continues as usual, however during the countdown phase, the system also calls a series of functions to generate a powerup for usage by the players who are struggling.
- The players then may activate these powerups during the next and all subsequent countdown phases when given the options, but they may also only end up with just percentile (round up) point bonuses to a question they get right.
- The system lasts for the duration of the quiz session, after which it expires.

#### **Alternate Flows:**

- **Token does not exist:** The system informs the admin: "Invalid token."
- **Quiz does not exist/is in bin/is not owned:** The system informs the admin: "Invalid quiz."
- **No active session for quiz:** the system informs the admin: "there are no active sessions for quiz".
- **No active session for quiz in LOBBY state:** the system informs the admin: "the quiz is not in lobby state".

- Chance of powerup, strength of powerup or percentile of powerup is less than 0: the system informs the admin “you cannot set chance, strength or percentile to be less than 0.”
- **Strength of powerup greater than 3:** the system informs the admin “strength of powerup is between 1 and 3.”
- **Chance or percentile is greater than 100:** the system informs the admin “chance and percentile must be between 1 and 100.”

**Postconditions:**

- Adds new parameters to the quizsession interface, noting the chance of powerup, strength of powerup, whether it is enabled, the bottom percentile which will get it.
- Changes the quiz session such that such parameters are tracked over the course of the quiz.
- Changes the way the scores are presented for leaderboards and whatnot.
- Allows the players to activate power-ups and sends a message to players when they receive a powerup/score boost.

**Acceptance Criteria:**

- Successfully adds a new setting for quiz sessions.
- Automatically can be called if on and depending on a series of variables.
- Can be activated by players during the countdown, adding a risk and reward factor.

**Title: touchId/FacelId login (interview 3)**

**Actors:** admin

**Preconditions:**

- There must be an account for the admin with such already set up
- Trigger: an alternate form of login

**Main Flow:**

- Admin sets up the touchId/FacelId using a third party app.
- Admin can enter email and select touchId/FacelId login.
- Then when attempting to login, we call this third party app to scan them and allow logins if it is a match.

**Alternate Flows:**

- **Account does not exist:** the system informs the user “Account does not exist.”
- **Incorrect Face/TouchId:** the system informs the user “Invalid Face/TouchId”

**Postconditions:**

- Adds the new sessiontoken in sessions if successfully login, returning such, else gives and error.

## Acceptance Criteria:

- Allows for the calls to the app to save a profile for an account.
- Allows calls for the app to verify profiles for accounts.

## Validation

**Interviewee 1:** Yes a more practice-oriented mode for toohak would make it more encompassing

**Interviewee 2:** it would also be good if the admin has the ability to sort quizzes based off difficulty levels

**interviewee 3 :** These features would make toohak on par with its competitor quizziz in terms of power ups and it would stand out from it with all the other features mentioned

## Interface design

See the updated swagger.yaml files for determining the specific return and input for each function. This will describe the routes and what each of the additional nine functions will do and how they interact with the greater function.

Interview 1 functions:

- **Result graph all:** functions like the list function, requiring a number array in terms of percentile correct to be stored in the datastore. Uses the route of GET `/v1/admin/session/{sessionId}/graph` requiring just the sessionId in parameters, the token, returning the percent correct of each question.
- **Result graph player:** functionally similar to graph all, but getting the graph of a specific player from playerId. Uses the route of GET `/v1/admin/session/{sessionId}/graph/{playerid}` requiring just the sessionId in parameters, the token, returning the percent correct of each question.
- **Result average overall:** requiring a new variable in the datastore to be updated, tracking the percentile correct for a quiz in the form of a graph denoting correctness on each question. Uses the route of GET `/v1/admin/quiz/{quzid}/graph`, requiring just the token and functioning similar to list.
- **Single player:** a special version of the quiz, where only one person plays in a session with no others, with it adding the result into a specific part of the quiz in terms of data and returning a result graph at the end. Uses the route of POST `/v1/play/mode/single`, requiring only the name string to allow non-admin players to practice the single player mode.

Interview 2 functions

- **Podium:** returns the best scores achieved for each quiz, alongside with the name, which likely requires a new element in the datastore or quiz interface to be updated to keep track of. Uses the route of GET `/v1/admin/quiz/{quzid}/podium` requiring just the sessionId in parameters and the token, returning the sorted array of the three best scores.



- **Leaderboard:** returns the best scores achieved for each quiz, alongside with the name, this one instead returns all of them sorted rather than just three. Uses the route of GET /v1/admin/quiz/{quizid}/leaderboard requiring just the sessionId in parameters and the token, returning the sorted array of all scores ever achieved in the quiz.
- **Time sorted list:** functions like quiz list either in terms of earliest created to latest created. Uses the route of GET /v1/admin/quiz/list/sorted/time, this takes in a token and a string of either "LATEST" or "OLDEST" denoting how it should be sorted.
- **Difficulty sorted list:** sorts by quiz depending on the percentile correct, something that would be generated by result average overall. Uses the route of GET /v1/admin/quiz/list/sorted/difficulty, this takes in a token and a string of either "EASIEST" or "HARDEST" denoting how it should be sorted.

### Interview 3 functions

- **Powerup generation:** a special function which can be called at certain points providing special power-ups for players, such as being able not lose points, getting double points at a certain time can be specifically given by the session starter to a player. Uses the route of PUT /v1/admin/powerup/{playerid}/give taking in the token, playerid and a string which functionally is an ENUM being either the name of a powerup or "RANDOM".
- **Powerup activate:** allows a player to activate their powerup, can only be used in the countdown or quiz\_closed states. Uses the route of PUT /v1/player/{playerid}/powerup and automatically activates whichever powerup the player has.
- **Catchup SET:** sets up the catchup mechanics for a quiz session whilst it is in lobby, allowing for the generation of power-ups and extra points to be given, essentially a settings menu to allow for powerup generation and whatnot to be non-manual and decided before the quiz. Uses the route of PUT /v1/admin/session/{sessionId}/catchup/set, taking in the token, sessionId, either "YES" or "NO" to determine if it's on and a number from 1 to 5 to determine how many get this benefit (5 means everyone, 4 for bottom 80% and so on) and a second number from 1 to 3 to determine how notable this is (1 for just point boosts, 2 for rare power-ups and point boosts and 3 for all powerups).
- **Touch login:** calls an outer service to attempt to login to an account using either touch id or face id. Uses the route of POST /v1/admin/auth/login/{service} where service is a string denoting either "TOUCH" or "FACE", taking in also the email of the account which the user wishes to (attempt) login.
- **Touch setup:** calls an outer app which sets up a profile for the user and then saves a profile, likely an access key to the third party app, which can then be checked by the program. Uses the route of PUT /v1/admin/auth/{service}/setup, with type either being "TOUCH" or "FACE" and taking in the token, which then calls the app allowing the user to set such up.

# State diagram

