

## Requirement Engineering

### Elicitation

#### Questions to our targeted audience

Q1: When looking at the login page, what problems do you encounter?

Q2: When you are creating a quiz, what problems do you encounter, what do you find annoying?

Q3: When you are going through an existing list of quizzes, what do you find difficult?

Q4: When you look at your profile, what problems do you encounter, what do you find annoying?

#### Responses from our target audience

Jack	Response
Q1	Can't remember password
Q2	Want an about/help page (could probably at least implement the typing in question part)?
Q3	Need a search bar. See popularity of quizzes.
Q4	Time joined/popularity

Ammay's Mum	Response
Q1.	Forgot Password, remember me option for login (Might want 1 time logins on 2nd account). Verification by text, Login with other apps (google account/phone number instead of username). Specific error message for why login fails. Eye icon to see the password you are typing is right.
Q2.	She said basically kahoot stuff, true false option, longer text response answer that the creator can verify as right or wrong, mcq, fill in the blank, order them best to worst, etc. Add photos, videos and audio to the questions. Add hints. Give the creator the option to preview it.
Q3.	Filter by name, length, description, tag, keywords. Search bar. Thumbnail maybe?? Quiz stats, like avg mark, engagement. Rate quizzes out of 5 stars

#### Solution to problems raised by our target audience

##### Q1:

1. Reset Password Button: Implement a "Reset Password" button on the login page. When clicked, this button should send an email to the user's registered email address. The email will include a link to a password reset function that allows the user to change their password.

2. Phone Number Login: Allow users to add their phone number to their profile. Users should be able to log in using their phone number as an alternative to their email address.
3. Account Linking: Enable users to link their accounts to other applications. Users should be able to log in using their credentials from these linked applications.

**Q2:**

1. Long-Form Answers: Allow users to provide long-form answers instead of selecting from preset multiple-choice options. Quiz creators can review and select the best or correct response from the submitted answers.
2. Hints: Quiz creators can add hints to their quizzes. These hints can be released either:
  - ◆ Upon the creator's command.
  - ◆ According to a predefined schedule. (Note: Students who wait for hints will lose marks for not answering questions quickly.)
3. Help Information: When creating quizzes, users should have access to help information. This can be provided by:
  - ◆ A help button that explains how to fill out quiz fields, including what constitutes a viable answer.
  - ◆ A mini video or YouTube tutorial that guides users on filling out a quiz.
4. Tags: Users can add tags to their quizzes to help categorise and identify them.

**Q3:**

1. Search Bar: Add a search bar to allow users to find their desired quizzes. The search bar can be enhanced with filter options such as creation date and sorting by quiz name.
2. Quiz Popularity: Display the popularity of quizzes, which is determined by ratings, the number of times played, and the users who have played them. This information can be filtered and sorted in ascending or descending order.
3. Quiz Categories: Introduce different categories for quizzes, such as Science, Math, etc., to help users easily find quizzes in specific subjects.

**Q4:**

1. Registration Time: Display the time when the user registered.
2. Quizzes Played: Show the total number of quizzes the user has played.
3. Top 3 Most Popular Quizzes: Highlight the top 3 most popular quizzes created by the user.
4. Number of Wins: Indicate the total number of quizzes the user has won.

## **Analysis & Specification - Use Cases**

### **User Stories**

#### **User story 1:**

As a teacher, I want to organise my quizzes for me so that I know which quizzes are for which classes.

- Teachers should be able to add tags to their quizzes and filter by tags.
- Teachers should be able to search for quizzes by clicking on the search bar.
- Teachers should be able to filter for quizzes via popularity.
- Teachers should be able to sort quizzes by time of creation, name and tags.
- Teachers should be able to put quizzes in different folders for different classification purposes.

#### **User story 2:**

As a teacher, I want to see admin details of my quizzes so that I can see how good I am at making quizzes.

User acceptance criteria

- Teachers should be able to see the number of players who have played a specific quiz that they made.
- Teachers should be able to see the total number of players who played all their quizzes.
- Teachers should be able to see ratings of their quizzes.
- Teachers should see how many people got a particular question right or wrong.

#### **User story 3:**

As a student, I want to see details of quizzes I have participated in to see how good I am at answering questions.

User Acceptance Criteria

- Students should be able to see the number of quizzes they have played.
- Students should be able to see the number of quizzes they have won.
- Students should be able to see their average score.
- Students should be able to see the leaderboard for a class over a period of time.

#### **User story 4:**

As a very forgetful person, I want to reset my password so I don't have to make another account or have to do customer service contacting

User Acceptance Criteria

- Users should be able to click on the forgot password button.
- Users should be able to enter an email linked to an account.
- Users should receive email with a link to reset password.
- Users should be able to choose a new password.
- Users should now be able to login with a sign of relief.

## Solution

### Use Case: Organise quizzes

Scenario: A tutor wants to organise quizzes so that they know which quizzes are for which classes.

Scope: A COMP1531 Class

Preconditions: User should be registered and logged in and quizzes should exist

Success End Conditions:

- Can assign multiple tags to quizzes (e.g. different tags for different topics)
- Can create folders
  - By adding in name
- Can change name of folders
- Click on a button to add quiz into folder
  - Have search bar to enter name of quiz
  - Can filter by quiz tags
    - If quizzes have no tags, thing will be shown
  - Can sort quizzes alphabetically
  - Can sort quizzes by time last edited
  - Can sort quizzes by time last created
- See list of folders
  - Can change the order folders are listed in view folders page
- Can open folder to see list of quizzes
- Can delete empty folder
- Move quiz to another folder

Failure End Conditions:

- Did not login and therefore cannot see quizzes
- Did not add name to folder and therefore no folders are created
- New folder has same name as old folder
- Delete a folder with quizzes inside
- Quiz to add does not exist

Primary Actor: Logged in User

Trigger: View folders button

### Validation

Jack

- "The filtering options will significantly improve my experience when navigating through quizzes."
- "Having better organisational tools like tags and folders will make it easier to manage and find relevant quizzes."
- "Definitely something I was looking for"

Ammay's mum

- "The focus on search functionality, sorting, and filtering will greatly enhance usability."
- "Implementing these changes should make the quiz management process smoother and more intuitive."

## Interface Design

### Http routes:

#### - **Make a folder**

Description: Make a new folder

Route name: /v1/admin/folder/create

Method: POST

Parameters:

headers:

token: string

Body:

```
{
    Name: string
}
```

Return:

```
{
    folderId: number
}
```

#### - **Delete a folder**

Description: Delete an empty folder

Route name: /v1/admin/folder/delete

Method: DELETE

Parameters:

headers:

token: string

Body:

```
{
    folderId: number
}
```

Return:

```
{}
```

#### - **Move a quiz to another folder**

Description: Move a quiz to another folder

Route name: /v1/admin/folder/move/{quizid}

Method: PUT

Parameters:

headers:

token: string

Path:

quizid: number

Body:

```
{
    oldFolderId: number,
    newFolderId: number
}
```

Return:

```
{}
```

**- Add a quiz to a folder**

Description: Add a quiz to folder

Route name: /v1/admin/folder/{folderid}/add/quiz/{quizid}

Method: PUT

Parameters:

headers:

token: string

Path:

quizid: number

folderid: number

Body:

{}

Return:

{}

**- Sort quizzes in folder**

Description: Sort quizzes in folder by lexicographical order

Route name: /v1/admin/folder/{folderid}/sort

Method: PUT

Parameters:

headers:

token: string

Path:

folderid: number

Body:

{}

Return:

{}

**- View quizzes in folder**

Description: View quizzes in folder

Route name: /v1/admin/folder/{folderid}/view

Method: GET

Parameters:

headers:

token: string

Path:

folderid: number

Body:

{}

Return:

{

Quizzes: Quiz[]

}

## Conceptual Modelling - State Diagrams

State diagram representing our solution:

