

## **Report for Toohak(Dream)**

### **Introduction**

Toohak's software development is an iterative process, and we are constantly striving to improve our products. After developing and testing a feature, the next step is to understand user requirements and needs to design the next set of features. This report outlines our approach to eliciting user questions, developing requirements, and creating early designs for upcoming iterations.

### **[Requirements] Elicitation**

To understand the challenges faced by our target users, we interviewed two people who were either using a tool like Toohak or planning to use it. The participants' information is below:

1. Name: Ikun Cai

Email: [Ikun.lovebasketball@gmail.com.cn](mailto:Ikun.lovebasketball@gmail.com.cn)

2. Name: GGbond

Email: [GGbond@qq.com](mailto:GGbond@qq.com)

### **Questions For Asked:**

1. What features or function do you believe are missing in existing quiz tools that hinder your experience?
2. Are there any specific challenges you encounter while using quiz tools, and how do you think they can be solved?
3. In what ways do you think Toohak can be improved to better meet your needs?
4. Are there any additional tools or integrations you would like to see in Toohak that would improve the overall experience?

### **Responses:**

#### **Interview with Ikun Cai:**

1. I often find that existing quizzes are limited in the types of questions they can ask. It would be great if there were more options such as multiple choice, right/wrong and short answer questions to make quizzes more engaging and dynamic. In addition, some tools lack flexibility in configuring question options, which can be frustrating.
2. A common challenge is the lack of comprehensive analyses and reports. I want to track student performance on tests and identify areas where they need to improve. Having an analytical dashboard with completion rates, grade point averages, and time spent on each problem would be very helpful.
3. Toohak is already a great tool, but it could be further improved by enhancing the user interface and making it more intuitive. Adding support for a wider range of question types and providing customisable options for each question would be great. In addition, if Toohak could be seamlessly integrated with popular learning management systems, the process of importing and exporting tests would become even easier.
4. It would be great if Toohak could be integrated with other educational tools such as interactive whiteboards or classroom response systems. This integration would create a more interactive and immersive learning experience for teachers and students.

#### **Interview with GGBond:**

1. Some assessment tools do not provide enough question types and students find them monotonous. I think a wider choice of question types and interactive elements, such as image-based questions, would make the tests more interesting and engaging for students.
2. Scoring and analysing test results can be very time-consuming, especially when dealing with many students. Implementing an efficient analysis dashboard that automatically calculates and displays results would be very helpful.
3. Toohak has a lot of potential and I believe it can be made even better by improving the collaboration features. Adding features that allow teachers and students to interact during tests (e.g., instant feedback or peer assessment) would be valuable in enhancing the learning experience.
4. It would be great if Toohak could integrate with popular learning management systems. Seamless integration would save time and effort in administering tests and scores, making it a more user-friendly platform for educators.

#### **[Requirements] Analysis & Specification - Use Cases**

Based on the information obtained, we identified the following user stories to address the target user's problem:

##### **User Story 1: Enhanced Question Diversity**

As a user, I would like Toohak to support a wider range of question types (e.g., multiple choice, right/wrong, short answer) to create more engaging quizzes.

##### **User acceptance criteria:**

1. Toohak should support at least five different question types.
2. Each question type must be configurable with associated options and validations.
3. Users should be able to preview each question type before it is finalised.

##### **User Story 2: Improved Analytics and Reporting**

As a user, I would like to see a comprehensive analytics dashboard in Toohak to track test performance and identify areas for improvement.

##### **User acceptance criteria:**

1. The analytics dashboard should show the quiz completion rate, average score and time spent per question.
2. Users should be able to filter and compare results based on different criteria.
3. Data should be presented in graphical form for easy understanding.

##### **User Story 3: Seamless LMS Integration**

As a user, I want Toohak to seamlessly integrate with popular learning management systems (LMS) to simplify the process of importing and exporting tests.

##### **User Acceptance Criteria:**

1. Toohak should support integration with at least three major LMS platforms.
2. Users should be able to import tests directly from the LMS into Toohak.
3. Tests should be exportable to the LMS for easy scoring and analysis.

## **Use Case: Conducting a Quiz**

### **People:**

- Player
- Admin

### **Target:**

Administrators create and organise quizzes for players to participate in and complete.

### **Main Scenario:**

1. The administrator logs in to Toohak.
2. The administrator selects the "Create Quiz" option.
3. The administrator enters the name of the quiz, question, question type, and options.
4. The administrator configures other settings for the quiz, such as time limits and participation conditions.
5. The administrator publishes the quiz for players to take.
6. Players can access the Toohak system without logging in and enter a unique quiz identifier to take part in the quiz.
7. The quiz begins, and players answer questions one by one until all questions are completed or the time limit is reached.
8. Players submit their answers to the quiz.
9. The quiz ends and the administrator collects and analyses the player's answers.

### **Other possible scenarios:**

- In step 5, the administrator may choose to schedule the release of a quiz so that a player cannot access the quiz until a specific time.
- In step 7, the administrator may choose to end the quiz early if the player completes all questions before the time limit, even if the time limit has not yet been reached.

### [Requirements] Validation

We contacted the participants we previously interviewed and shared the use cases we developed. We collected the following feedback:

Ikun Cai: These use cases capture the problems I've encountered with other testing tools and the proposed features seem promising to enhance the overall experience."

GGbond: The Analytics Dashboard and LMS integration is exactly what I need to track student performance and streamline the grading process."

### [Design] Interface Design

Based on the requirements, the following are the **Interface Design**:

#### 1.Admin Sending Email Notification

Name & Description	Data & Type	Error returns
<p>adminSendEmailNotification</p> <p>This interface allows the administrator to send email notifications to specific users. The email will contain important information, announcements or updates related to the test platform.</p>	<p>Parameters:</p> <ul style="list-style-type: none"><li>recipientEmail (string): The email address of the user who will receive the email notification.</li><li>subject (string): The subject of the email notification.</li><li>body (string): The content of the email notification.</li></ul> <p>Return Type (if no error):</p> <ul style="list-style-type: none"><li>success (Boolean): A Boolean value indicating whether the email notification was successfully sent.</li></ul>	<p>Return object {error: 'specific error message here' } when any of the following conditions are met:</p> <ol style="list-style-type: none"><li>The recipientEmail is not a valid email address.</li><li>The subject is empty or exceeds a certain character limit.</li><li>The body of the email is empty or exceeds a certain character limit.</li></ol>

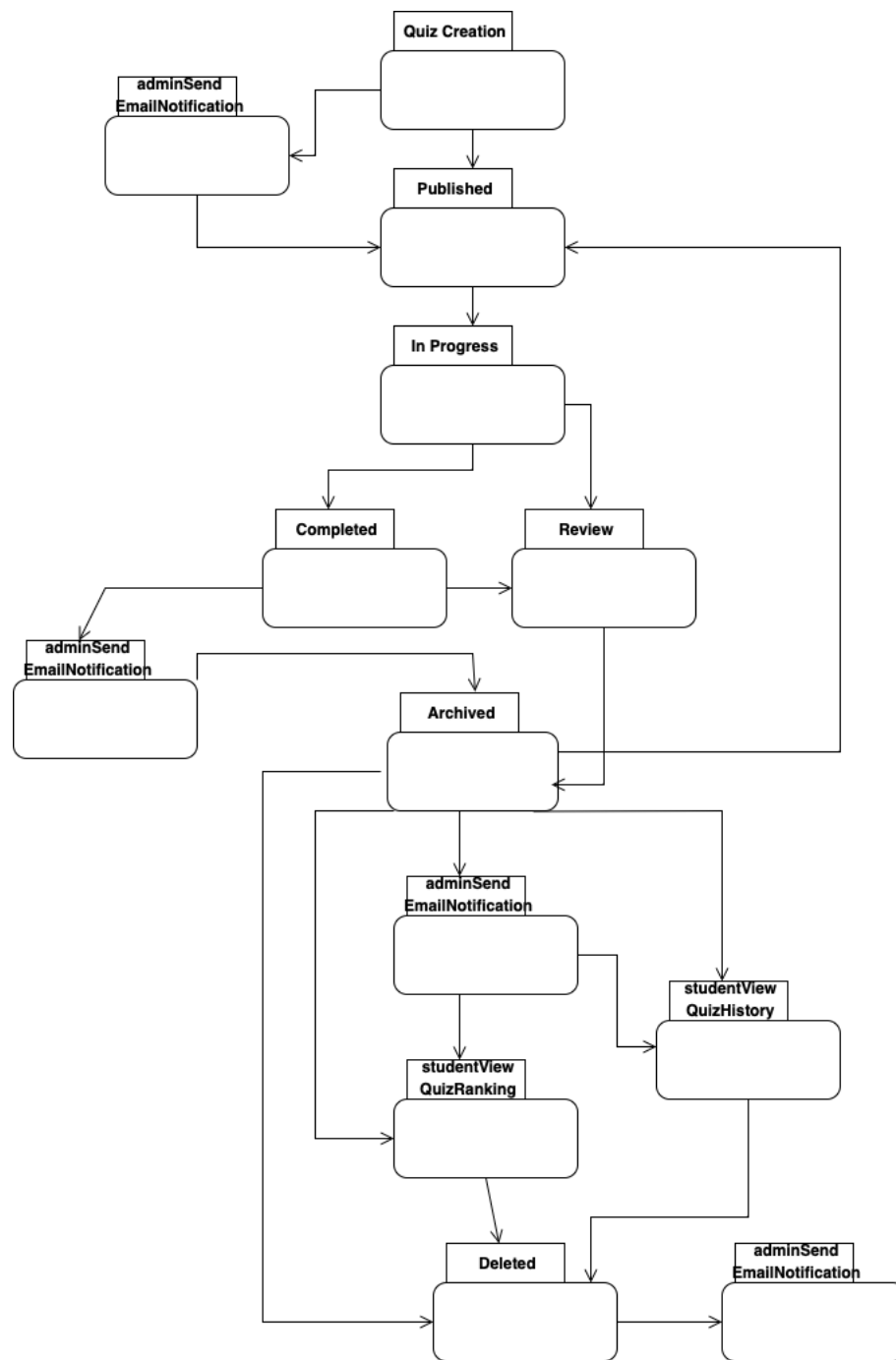
## 2. Student Viewing Quiz History

Name & Description	Data & Type	Error returns
<p>studentViewQuizHistory</p> <p>This screen allows students to view their test history. The Test History provides information about the tests the student has taken, including the test name, date, and performance.</p>	<p>Parameters:</p> <ul style="list-style-type: none"><li>studentId (string): The unique identifier of the student.</li></ul> <p>Return Type (if no error):</p> <ul style="list-style-type: none"><li>quizHistory (array of objects): An array of objects, each representing a quiz in the student's history. Each object contains the following properties:<ul style="list-style-type: none"><li>quizId (string): The unique identifier of the quiz.</li><li>quizName (string): The name of the quiz.</li><li>date (string): The date when the student participated in the quiz.</li><li>score (number): The score achieved by the student in the quiz.</li></ul></li></ul>	<p>Return object { error: 'specific error message here' } when any of the following conditions are met:</p> <ol style="list-style-type: none"><li>The studentId does not exist in the system.</li><li>There is no quiz history available for the student.</li></ol>

### 3. Student Viewing Quiz Ranking

Name & Description	Data & Type	Error returns
<p>studentViewQuizRanking</p> <p>The interface allows students to view a student's ranking based on the student's performance on a particular test. The rankings show the top performing students and their respective scores.</p>	<p>Parameters:</p> <ul style="list-style-type: none"><li>quizId (string): The unique identifier of the quiz for which the student wants to view the ranking.</li></ul> <p>Return Type (if no error):</p> <ul style="list-style-type: none"><li>quizRanking (array of objects): An array of objects, each representing a student in the quiz ranking. Each object contains the following properties:<ul style="list-style-type: none"><li>studentId (string): The unique identifier of the student.</li><li>studentName (string): The name of the student.</li><li>score (number): The score achieved by the student in the quiz.</li></ul></li></ul>	<p>Return object { error: 'specific error message here' } when any of the following conditions are met:</p> <ol style="list-style-type: none"><li>The quizId does not exist in the system.</li><li>There is no ranking available for the specified quiz.</li></ol>

## [Design] Conceptual Modelling - State Diagrams



The state diagram illustrates the different states that the quiz undergoes during its lifecycle. There is a textual representation of the state diagram:

### State 1: Quiz Creation

Description: This is the initial state when the administrator creates the quiz.

#### Transitions:

- Can transit to "adminSendEmailNotification."
- Can transit to "Published."

**State 2: Published**

Description: The quiz is now available for players to join and participate in.

**Transition:**

- When the player starts to join the quiz, the quiz will transition to the "In Progress"

**Status 3: In Progress**

Description: Players are actively participating in the quiz and answering questions.

**Transition 1:** If all players have completed the quiz, it will go to "Completed" status.

**Transition 2:** If the administrator decides to end the quiz early, the quiz will go to "Completed" status.

**Status 4: Completed**

Description: The quiz has ended, and all player responses have been recorded.

**Transitions:**

- If the administrator decides to view the quiz results, the quiz will transition to the "Review" state.
- Can transit to "adminSendEmailNotification."
- Can transit to "Archived."

**State 5: Review**

Description: The administrator is reviewing the quiz results and analysing the player's performance.

**Transition:**

- The administrator can choose to archive the quiz and move it to the "Archived" state.

**State 6: Archived**

Description: The quiz is saved for historical purposes, but the player cannot participate in it.

**Transitions:**

- The administrator can choose to restore the quiz, moving it back to the Published state.
- The administrator can choose to permanently delete it, moving it to the Deleted state.

**State 7: ViewQuizHistory**

Description: The state where students can view their quiz history.

**Transitions:**

- Can transit to "Deleted."

**State 8: ViewQuizRanking**

Description: The state where students can view the quiz rankings.

**Transitions:**

- Can transit to "Deleted."

**State 9: EmailNotification**

Description: The state where the administrator can send email notifications.

**Transitions:**

- Can transit to "Published."
- Can transit to "Archived."
- Can transit to "studentViewQuizRank" or "studentViewQuizHistory."
- Can transit to "Deleted."

**Status 10: Deleted**

The quiz will be permanently deleted and can no longer be accessed in the system.

**Transitions:**

- Can transit to "adminSendEmailNotification."
- This state is the final state, and no further transitions are allowed.



**The Conclusion :**

The planning report outlines the steps taken to understand user issues, development requirements, and create an early design for the next iteration of Toohak. By combining user feedback and specific use cases, we aim to improve the quiz game platform and enhance the user experience. The HTTP endpoints and state charts will serve as a guide for future development and testing phases.