

Analysis Document

Name: Tony Jiang

Class: RB04

Semester: 6

Version	Date	Description
0.1	18 Sept 24	Initial documentation.
0.2	23 Sept 24	Redo non-functional with more detailed.

Contents

Introduction	3
User story	3
Player	3
1. Daily guess	3
2. Login	4
3. Replay song	4
4. 5 guess opportunities	4
5. Autocomplete search	4
6. Skippable guess	5
7. Show previous answers	5
8. Statistic	5
9. Create playlist	6
Admin	6
1. Login	6
2. Add song	6
3. See list of songs	7
4. Delete song	7
5. Modify song	7
Non-functional requirements	8

Introduction

This document contains the user stories and non-functional requirements for this project. It helps us define, review, and understand the importance of both functional and non-functional requirements.

User story

The user stories are divided by user roles. Currently, there are three user roles: player, admin, and moderator. The player guesses the song by participating in the game. The admin manages the songs, and the moderator oversees the gameplay.

Each user story is prioritized using the MUSCOW technique, which determines which user stories to focus on first and which are most important to implement in the system. MUSCOW stands for:

Must have

Should have

Could have

Won't have (this time)

Player

1. Daily guess

As a player, I want to play a daily guessing game, so that I can challenge myself to see if I know the song of the day.

Acceptance Criteria

- Players can play this without logging in.
- After guessing the song, the player has to wait until the next day to play the next daily song.
- A timer is displayed when the next daily song is ready.

MUSCOW Prioritization

Must

2. Login

As a player, I want to log in, so that I can guess more songs and challenge my knowledge.

Acceptance Criteria

- Player has to provide user credentials to log in.
- Logging in will redirect the player to a targeted page.

MUSCOW Prioritization

Must

3. Replay song

As a player, I want to be able to replay the song that I just listened to, so that I can secure my answer

Acceptance Criteria

- Players can interact with the replay button, to replay the song.
- The system will replay the song from the starting point.

MUSCOW Prioritization

Must

4. 5 guess opportunities

As a player, I want to have five opportunities to guess the song, so that I can challenge myself and test my knowledge.

Acceptance Criteria

- Player can only listen to a set duration of the song in seconds.
- Each attempt to guess gets an additional number of seconds to guess the song.
- Player can see how many attempts they have.

MUSCOW Prioritization

Must

5. Autocomplete search

As a player, I want to autocomplete my answer, so that I know if it's the correct answer I'm thinking of.

Acceptance Criteria

- A max certain amount of autocomplete should be shown.
- Autocomplete should be displayed when the player starts typing.
- Autocomplete is displayed under the answer.
- Player can select the autocomplete answer.
- Autocomplete should not be displayed when there isn't a match to what the player has typed.

MUSCOW Prioritization

Must

6. Skippable guess

As a player, I want to be able to skip a song guess, so that I can listen to the rest of the song and make a guess.

Acceptance Criteria

- Player can interact with the skip button to skip a guess of a song.
- The duration of the song is extended when the skip button is activated.
- The system displays which guess was skipped.

MUSCOW Prioritization

Must

7. Show previous answers

As a player, I want to see my previous wrong guesses, so that I know which answers I have already submitted.

Acceptance Criteria

- Players can see they previous wrong answers.

MUSCOW Prioritization

Must

8. Statistic

As a player, I want to see my statistics, so that I can understand at which point in the guessing process I know the answer.

Acceptance Criteria

- Players can see the total songs they have guess.

- Players can see the total correct answer of each guess attempt of all the songs they have guessed.

MUSCOW Prioritization

Should

9. Create playlist

As a player, I want to be able to create a song playlist, so I can challenge other players to see if they know the songs I added to the playlist

Acceptance Criteria

- Player has to be logged in to create the playlist
- Player would have to fill all the inputs on the form to create the playlist.
- A notification should be displayed when the playlist is successfully submitted.
- Player has to wait for approval after the playlist is submitted.

MUSCOW Prioritization

Could

Admin

1. Login

As an admin, I want to be able to login, so that I can manage the songs of the system.

Acceptance Criteria

- Admin should be able to log in with their credentials.
- Logging in will redirect the admin to a targeted page.

MUSCOW Prioritization

Must

2. Add song

As an admin, I want to be able to add songs, so that the system has more songs for players to guess.

Acceptance Criteria

- Admin has to fill a form to add a song.
- A notification should be displayed when the song is successfully added

MUSCOW Prioritization

Must

3. See list of songs

As an admin, I want to be able to see a list of songs, so that I know what has been added to the system.

Acceptance Criteria

- Admin can see the songs by alphabetic order.
- Admin can see the total number of songs.

MUSCOW Prioritization

Must

4. Delete song

As an admin, I want to be able to delete a song, so that I can remove it if I make a mistake.

Acceptance Criteria

- Admin can interact with the delete button to delete the song.
- A notification should appear when the admin tried to delete a song to confirm if the admin wants to delete the song.
- A notification should appear when the song is successfully deleted.

MUSCOW Prioritization

Must

5. Modify song

As an admin, I want to be able to modify a song, so that I can correct any mistakes that were made.

Acceptance Criteria

- Admin can interact with the modify button to modify the song.
- A notification should appear when the song is successfully saved.

MUSCOW Prioritization

Must

Non-functional requirements

Here are the define non-functional requirements for this project and their criteria.

Scalability:

The system should be able to handle 15,000 concurrent players during peak hours, with a maximum response time of under 2 seconds.

Explanation

There was no freely available data on daily active users for a music guessing website. To access this information, I would need to create an account and provide a lot of personal information, which I prefer not to do. So, I conducted some research and estimated the number of users needed if I include advertisements on my website.

If I want to earn a minimum of 2,000 euros per month from ads, and the ads earn 5 euros per 1,000 views (CPM, Cost Per Mille – “Mille” is a Latin for “thousand”, a common CPM), the calculation is as follows:

- Monthly Revenue Goal: 2,000 euros.
- Daily Revenue Goal: 2,000 euros / 30 days \approx 67 euros per day.

To calculate the daily impressions needed:

- Daily Revenue Goal: 67 euros.
- CPM Rate: 5 euros per 1,000 impressions.

Daily Impressions Needed:

Daily Impressions = Daily Revenue Goal \times 1,000 / CPM

Daily Impressions = $67 \times 1,000 / 5 = 13,400$ impressions / users / players

Therefore, I would need 13,400 daily users / players for 1 page to meet the daily revenue goal of 67 euros, assuming a CPM of 5 euros.

A maximum response time of 2 seconds is acceptable since it's not a multiplayer game, but rather a knowledge challenge. In this context, a 2 second response time is sufficient, as users won't feel that the delay negatively impacts their experience.

Security:

The system will enforce a strong password policy for user login and integrate OAuth with trusted third-party providers, such as Google. Role-based access control will ensure that users only have access to their own data, while sessions will automatically expire after a defined period of inactivity. Additionally, an anti-cheat system will be implemented to detect and prevent malicious behavior or cheating within the platform. All user inputs will be validated and sanitized to prevent security

vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

Explanation

Here is some information I found for web security considerations:

<https://www.cogentinfo.com/resources/web-security-latest-security-threats-and-best-practices-for-securing-web-applications>

<https://www.fortinet.com/resources/cyberglossary/web-security-threats>

Here are some web game security considerations I found:

<https://www.linkedin.com/advice/1/what-most-important-online-game-security-measures-yd6gc>

<https://www.kaspersky.co.in/resource-center/threats/top-10-online-gaming-risks>

Privacy:

The system will comply with GDPR guidelines by securing and encrypting all user data. Users will have full control over their data, including the ability to:

- Access their personal data at any time.
- Correct inaccurate or incomplete data.
- Request data deletion in compliance with the "right to be forgotten."

Performance:

The system must successfully complete 99.9% of transactions within an average response time of less than 2 seconds during peak load.