SW Engineering CSC648/848

BeatLink

Section 01, Team 03

Anthony Benjamin - Team Leader

Anna Felipe - Front-end Lead

Guillermo Alcantara - Scrum Master

Warisha Vahidy - Git Master

Zuby Afzal - Back-end Lead

Yash Bhadiyadra - Support

Milestone 2

March 18, 2024

1. Data Definitions:

- Public User
    - Username, profile picture, playlists, bio, status
- Playlists
    - Name, Description, Author, songs
- Comments
    - comment text
- Likes
    - Like Counter
- Personal Profile
    - Access your User settings to change any fields accessible to the user logged in. Password, Username, profile picture, bio, status.
- Share Link
    - A link to share a playlist or profile
- Public Feed
    - Playlist posts, profile activity (likes, new comments, etc), change in status

2. Initial Functional Requirements:

1. Public Feed:
   A public feed of playlist posts that you can interact with. Such as liking, commenting, and following the author of the post.

   Requirement ID: 1
   User story reference: 2
   Priority: 1

2. Import Playlist:
   Ability to import playlists from Spotify and potentially other platforms. This allows users to easily build their profile with minimal effort.

   Requirement ID: 2
   User story reference: 3
   Priority: 1 for Spotify, 4 for other platforms

3. Export Playlist:
   Ability to export playlists from our service into Spotify. Exporting playlists provides users with simple options to expand their existing listening libraries.

   Requirement ID: 3

User story reference: 4
Priority: 4

4. Song Recommendations:
   AI song recommendations based on your playlist's mood, genre, and previous interests.

   Requirement ID: 4
   User story reference: 5
   Priority: 2

5. User Authentication:
   Login authentication will require AES-256 encryption, assuring the protection of personal information.

   Requirement ID: 5
   User story reference: 1
   Priority: 1

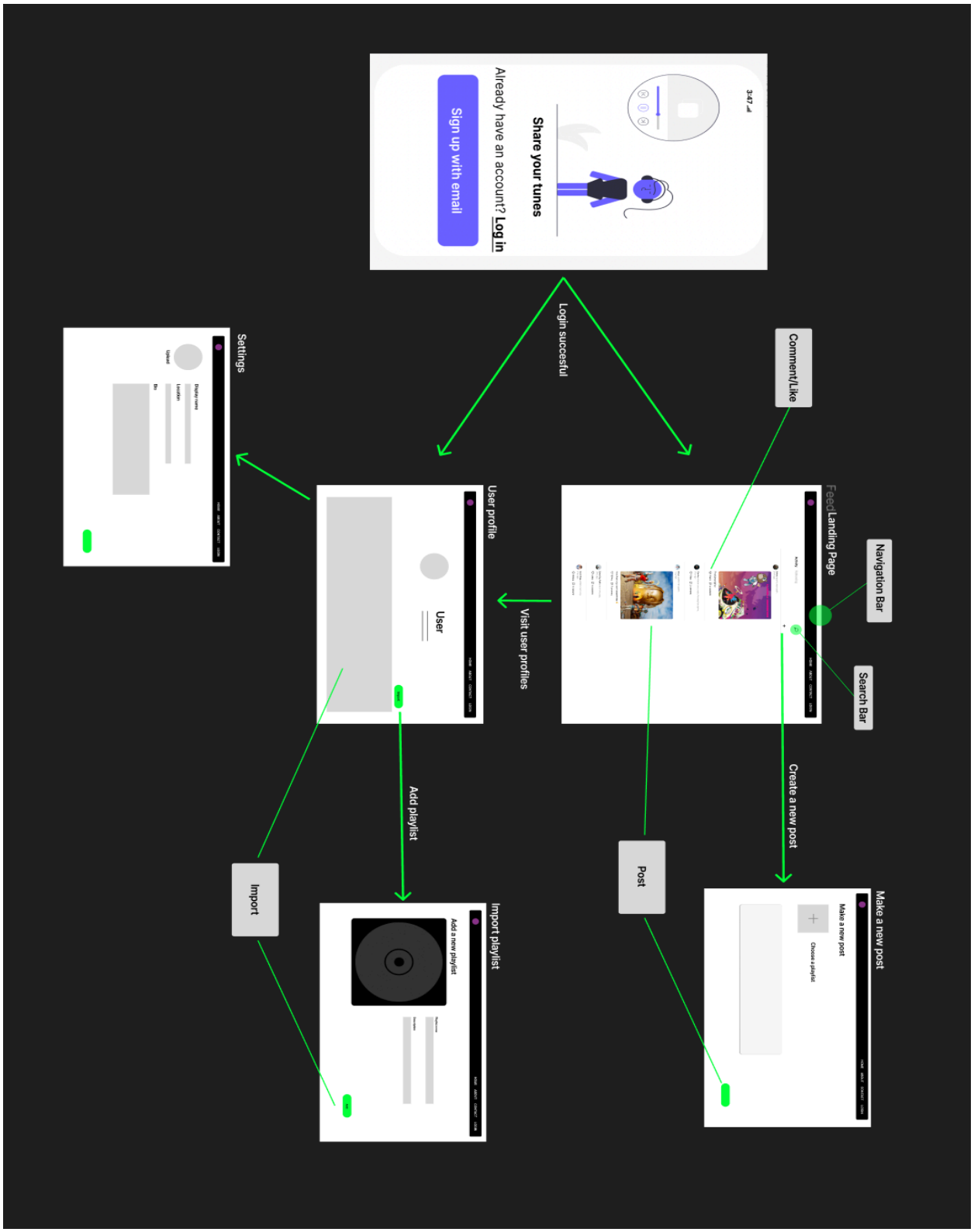
6. Customizability:
   Ability to add a profile picture to your page, filter content, and choose which other users to follow.

   Requirement ID: 6
   User story reference: 5
   Priority: 7

7. Personalized Content:
   User-specific feed for when browsing other pages and customized feeds for when people click on your page.

   Requirement ID: 7
   User story reference: 4
   Priority: 6

3. UI Mockups and UX Flow



4. High level Architecture, Database Organization

- **DB Organization:**
    - User Schema (Add, Delete, and Search):
        - Id: String
        - Email: String
        - Username: String
        - Bio: String
        - Status: String
        - Password: String (hashed)
        - Following: Number
        - Followers: Number
        - Playlists: Array<Playlist.id>

    - Playlist Schema (Add, Delete, Search):
        - Id: Number
        - Name: String
        - Description: String
        - Author: String
        - Songs: Array<Object>
        - Users: Array<User.id>

    - Post Schema (Add, Delete, Search):
        - Id: Number
        - User: User.id
        - Playlist: Playlist
        - Likes: Number
        - Comments: Array<String>

- **APIs:**
    - **Login:** POST method to check if the credentials (username and hashed password) entered match the credentials stored in the database. If credentials match, a token is created to validate session status for other pages.
    - **Logout:** Post method to remove the token from the user session.
    - **Registration:** Post request form used to register users. This api extracts users info (email, username, password, DOB) ensures that none of the information on the sign up page is missing or the user will not be able to sign up. Once the user is successfully signed up the user will be redirected to the login page. Uses bycrypt to hash users passwords to

make it secure before storing in the database. Calls createuser() to store user objects in the database.
    - **GetPlaylists:** GET method that'll display all playlists in our database
    - **PlaylistInfo:** POST method that takes in a "playlistURL" in its body and creates a document for the playlist inside of our database. The playlist is stored in our Playlist table with all the relevant info in our Schema. We use the spotify API to translate the Spotify playlist URL into a document in our database.
    - **DeletePlaylist:** DELETE method that deletes a "playlistId" in it's body and deletes that specified resource in the database.
    - **CreatePost:** POST method that takes in fields that are associated with the Post Schema, and stores the post within our database.
    - **GetPosts:** GET method that will display all the posts that are stored in our database. This also has optional body fields to search specific query parameters,
    - **DeletePosts:** DELETE method that will delete a specific post signified by a postID in the request's body.
    - **Follow:** POST method that adds a follow to a specific user specified by the request's body.
    - **RemoveFollow:** POST method that removes a follower from a specified user
    - **AddLike:** POST method that adds a like for a post on the public feed, and increments the like number in the database.
    - **RemoveLike:** POST method that removes a like from a given post
    - **AddComment:** POST method that adds a comment in our database
    - **MakeSongRecommendation:** POST method that gives song recommendations based on a list of artists inside of a playlist. We leverage OpenAI's Assistant API to make these recommendations.

6. Identify actual key risks for your project at this time

Identify only actual and specific risks in your current work such as

- **Skills risks and mitigation plan**
    - ❖ Do you have a proper study plan to cover all the necessary technologies?

    - ➢ In addressing the skills risks associated with our project, we have implemented a comprehensive study plan covering all the necessary technologies. Each technology vital to the project has been meticulously outlined, including designated team members responsible for learning and mastering it. We have prioritized a well-rounded approach, ensuring that

every team member possesses a foundational understanding of all project-relevant technologies.

➢ Our study plan includes detailed timelines for each technology, allowing us to pace our learning effectively and allocate resources efficiently. By setting deadlines for mastering specific concepts within each technology, we aim to maintain a steady progression without compromising depth or breadth of knowledge.

➢ Furthermore, we have compiled a list of key concepts for each technology, providing clear objectives for our learning efforts. This structured approach enables us to focus on essential skills and competencies necessary for project success.

➢ Our proactive approach to skills development mitigates potential risks associated with knowledge gaps, ensuring that our team is well-equipped to tackle the challenges ahead.

- **Schedule risks**
  - ❖ Does your team have a team schedule for every member including their detailed task?

  ➢ We conduct two team meetings every week to facilitate communication, collaboration, and progress tracking. The first meeting takes place during the second half of our software engineering class on Mondays, providing an opportunity to reflect on the previous week's accomplishments, challenges encountered, and plans for the upcoming weeks. Additionally, we hold a second meeting on Sundays from 12 pm to 1 pm, allowing for in-depth discussions and problem-solving sessions.

  ➢ During these meetings, each team member shares their achievements, challenges, and future plans, fostering transparency and accountability within the team. We actively engage in addressing challenges by collectively brainstorming solutions and offering assistance to overcome obstacles. This collaborative approach strengthens team cohesion and ensures that all members are supported in their endeavors.

  ❖ If change happens, does it update transparently? Does your team use project management tools (e.g. Jira, Trello)?

- ➢ In conjunction with regular meetings, we utilize Trello as our project management tool to streamline task allocation, tracking, and updates. Our scrum master oversees the organization of tasks within Trello, assigning them to team members based on their skills and expertise. Clear deadlines are set for each task to maintain momentum and accountability.

- ➢ Our scrum master diligently updates the Trello board to reflect task progress, ensuring transparency and real-time visibility into project developments. This centralized platform serves as a repository for tasks, milestones, and updates, facilitating effective coordination and communication among team members.

- ➢ Our proactive approach to scheduling and project management minimizes schedule risks and enhances the overall efficiency and effectiveness of our project execution.

- **Teamwork risks** (any issues related to teamwork)
  - ❖ Everybody is in the meeting regularly?

  - ➢ All team members demonstrate a high level of commitment by consistently attending all scheduled meetings. During these gatherings, each member actively participates by sharing updates on their assigned tasks, challenges encountered, and progress made. This regular exchange of information fosters transparency, accountability, and alignment among team members.

  - ❖ Everybody keeps his/her pace? If not, what is your plan to mitigate the risks?

  - ➢ Our team excels in maintaining individual pace while ensuring collective progress toward project goals. Every member takes ownership of their responsibilities and strives to meet deadlines effectively. In instances where a team member encounters difficulty or falls behind schedule, the team rallies together to provide support and guidance. Our scrum master plays a pivotal role in monitoring individual progress and facilitating discussions to address any roadblocks encountered.

  - ➢ Furthermore, our team culture prioritizes camaraderie and well-being. Our scrum master goes the extra mile by bringing snacks to each meeting, creating a relaxed and enjoyable atmosphere conducive to productive

collaboration. This thoughtful gesture fosters team cohesion and alleviates stress, allowing us to work harmoniously towards achieving our objectives.

- **Legal/content risks**
  - ❖ Can you obtain content/SW you need legally with proper licensing, copyright)?

  - ➢ Our project adopts a strategic approach by predominantly leveraging open-source software and freely available content. By utilizing resources that are openly accessible and distributed under permissive licenses, we minimize the risk of encountering copyright infringement or licensing issues. Open-source software provides us with the flexibility, transparency, and legal certainty necessary to develop and deploy our project without constraints.

  - ➢ Furthermore, our adherence to open-source principles aligns with our commitment to fostering collaboration, innovation, and community-driven development. By embracing open-source methodologies, we contribute to a culture of sharing knowledge, code, and resources for the benefit of the broader software engineering community.

  - ➢ Our deliberate choice to prioritize open-source software and free content mitigates legal and content-related risks, ensuring compliance with licensing requirements while fostering a culture of openness, collaboration, and innovation.

7. Project Management:

- In our project management approach for M2 tasks, we adopted an agile methodology with 2 weekly scrum meetings to ensure effective communication and progress tracking among each team member. During each meeting, each member shared updates on their current tasks, resources that they looked at for help, shared links for key resources, outlined any obstacles they were facing, discussed their plans for the following week, and if any issues were faced we tried to solve them during the meeting. This transparent sharing mechanism fostered a culture of openness and prompt assistance, enabling team members to offer help, advice, resources and provide updates, and ensure everyone is completing a task. To facilitate a structured and efficient task management

process, we utilized a Trello Board, where the scrum master and other members were able to create, assign, and track tasks in a transparent manner, allowing every team member to view the progress of their tasks as well as the overall project development. Trello’s dashboard and reporting features provided us with insights into our progress, including completed tasks, ongoing activities, and pending items, thus ensuring that everyone was aligned with the project's objectives and timelines.By leveraging these tools and practices, we were able to maintain a high level of transparency and collaboration among team members to deliver the M2 progress.