PHASE 1: PROJECT PLANNING SUBMISSION
Project Title: Music Cataloging System
Team Name: BrightByte
Team Members:

- Bach Nguyen (Project Manager)
- Massi Afzal (Technical Manager)
- Vincent Tseng (Front-End Lead)
- Jason Guan (Back-End Lead)
- Moksh Patel (Software Quality Lead)

---

1. Project Overview

1.1 Brief Summary of the Project's Purpose and Scope

Our "Music Cataloging System" aims to help users organize and explore music albums. The software enables creation, browsing, and editing of album information (artist, genre, release date, tracklist, etc.). Through robust search and filter features, users can quickly find music of interest. An account-management component allows users to sign up, log in, and edit personal profiles, potentially with the option to favorite albums.

This project will be delivered as a web-based platform with a user-friendly interface, leveraging an Agile workflow (Kanban) over three two-week iterations.

1.2 High-Level Goals and Expected Outcomes

1. Provide a comprehensive database of music albums with rich metadata.
2. Allow users to add, edit, and delete album entries.
3. Enable advanced search and filtering (by artist, album name, genre, etc.).
4. (Optional) Offer personalized user recommendations and/or track previews.
5. Adhere to an Agile workflow with clearly defined roles and responsibilities.

---

2. Team Agreements & Elicitation Documentation

2.1 Summary of Team Contract

- **Team Roles & Responsibilities**
    - **Project Manager (Bach Nguyen):** Oversees project timeline and deliverables.

    - **Technical Manager (Massi Afzal):** Guides architectural decisions and technology usage.
    - **Front-End Lead (Vincent Tseng):** Designs and implements user interface.
    - **Back-End Lead (Jason Guan):** Manages server-side logic and database operations.
    - **Software Quality Lead (Moksh Patel):** Oversees testing, QA processes, and code reviews.
    - **All Members:** Serve as developers in addition to their primary roles.
- **Communication**
    - Dedicated Discord server with daily/bi-daily updates posted by 9pm.
    - Team members respond to messages within 24 hours.
    - Weekly in-person meeting (Fridays) to coordinate tasks.
- **Work Methods**
    - Pair programming is encouraged outside of group meetings.
    - All code changes must be reviewed by a teammate before merging to the main branch.
- **Conflict Resolution & Accountability**
    - The team commits to respectful reminders and mutual support if tasks fall behind.
    - Major project decisions require team consensus.

2.2 Overview of Elicitation Methods

- **Domain Research:** Investigated well-known cataloging services (Discogs, MusicBrainz, AllMusic) for best practices.
- **Interviews & Q&A with "Customer":** Clarified feature priorities, MVP requirements, and potential data sources.
- **Stakeholder Analysis:** Identified who will fund, use, or maintain the system, includes sponsors, lawyers, and end-users.

2.3 Key Findings from Elicitation

1. **Core Requirement:** Store and manage album metadata (artist, genre, release date, tracklist).
2. **High Priority Features:** User account system (create, edit, delete) and robust search.
3. **Secondary Features:** Additional filters, personalized recommendations, track previews.
4. **Preferred Platform:** Web-based UI rather than a command-line interface.

2.4 Meeting Notes Summary (Customer Interactions)

- **Must-Have:** Basic CRUD on albums, search functionality, user login.
- **Nice-to-Have:** Album cover art, track samples, recommendation system, favorites.
- **Data Source:** External APIs (Spotify, MusicBrainz) or Kaggle datasets for initial catalog seeding.

---

3. User Stories & Prioritization

Below is a consolidated list of user stories, grouped by priority.

3.1 Must-Have

1. **Account Management**
    - **User Story:** “As a user, I want to create, log into, and delete my account, so that I can securely manage my personal profile.”
    - **Justification:** Critical for user-based access and personalization.
2. **Catalog Viewing**
    - **User Story:** “As a music enthusiast, I want to browse the album catalog, so that I can discover music easily.”
    - **Justification:** Central to the system’s purpose.
3. **Search Functionality**
    - **User Story:** “As a user, I want to search for albums by title, artist, or genre, so that I can quickly find specific albums.”
    - **Justification:** Essential for discovering content in large datasets.
4. **Basic CRUD on Albums**
    - **User Story:** “As a user, I want to add, edit, or delete albums, so that the catalog is always accurate and up-to-date.”
    - **Justification:** Maintains an up-to-date music library.

3.2 Should-Have

5. **Filtering Albums**
    - **User Story:** “As a user, I want to filter albums by genre, artist, or release year, so that I can narrow down my results.”
    - **Justification:** Enhances user experience but not strictly necessary for the MVP.
6. **Detailed Album Information**
    - **User Story:** “As a user, I want to see each album’s tracklist, release date, and cover art, so that I can fully explore the album’s details.”
    - **Justification:** Adds depth to the user’s browsing experience.
7. **Favorite Album Feature**
    - **User Story:** “As a user, I want to mark certain albums as favorites, so that I can quickly revisit them later.”
    - **Justification:** Personalization that enriches the platform’s utility.

3.3 Nice-to-Have

8. **Personalized Recommendations**

    - **User Story:** “As a user, I want to see recommended albums based on my preferences, so that I can discover new music easily.”
    - **Justification:** Adds extra value but requires more complex data analysis.
9. **Sample/Picture Playback**
    - **User Story:** “As a user, I want to play short previews or see album covers, so that I can quickly decide if an album interests me.”
    - **Justification:** Multimedia support that can boost user engagement.

---

4. Effort Estimation & Iteration Planning

4.1 Estimated Effort

- Ranges from ~3–5 hours for smaller features to ~8–10 hours for more complex tasks.
- Derived from Planning Poker sessions and group consensus.

Examples:

- **Account Management:** ~8 hours total (front-end + back-end).
- **Catalog Viewing:** ~5 hours.
- **Search Functionality:** ~5 hours.
- **Album CRUD:** ~6–8 hours, front-end forms and back-end endpoints.

4.2 Task Assignments

Task	Assigned Member(s)	Effort (hrs)
Account Management (Front-End)	Vincent	4
Account Management (Back-End)	Jason	4
Album Data Model & CRUD	Jason, Moksh	16
Search & Filter UI	Vincent, Bach	4
Recommendation Engine	Massi	2
Testing & QA	Moksh (all contribute)	6 (ongoing)

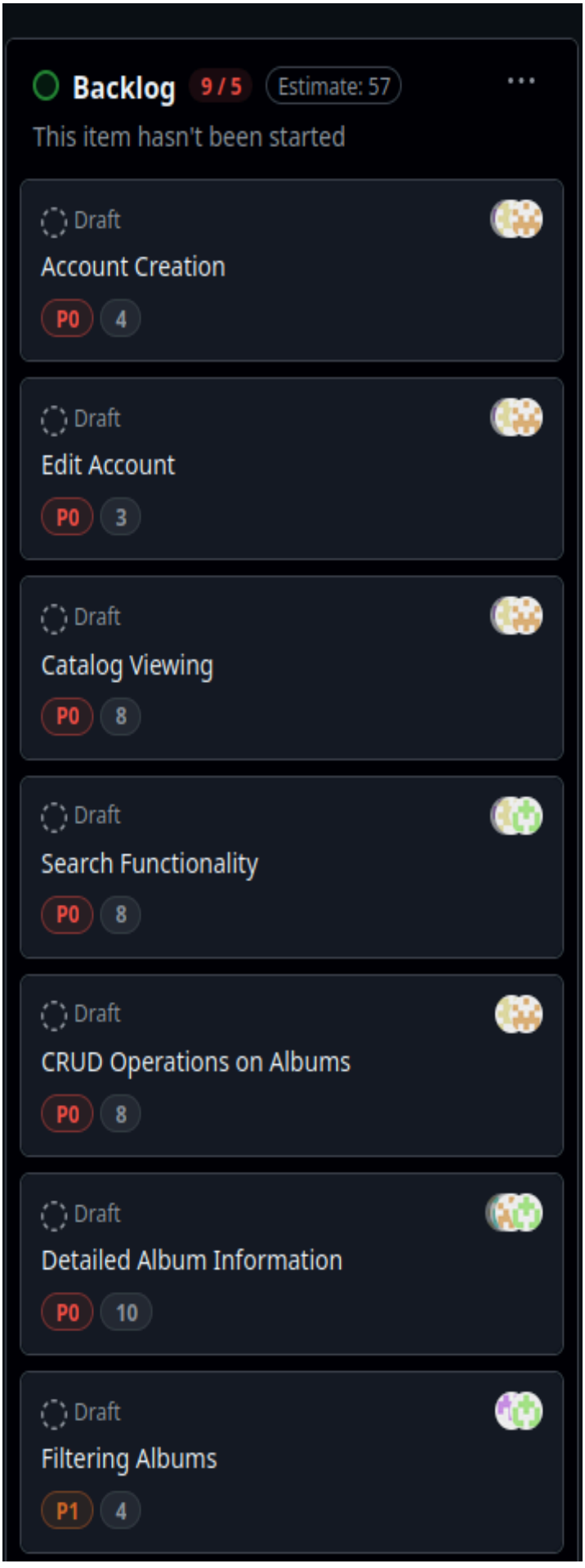
4.3 Development Roadmap (Three Iterations)

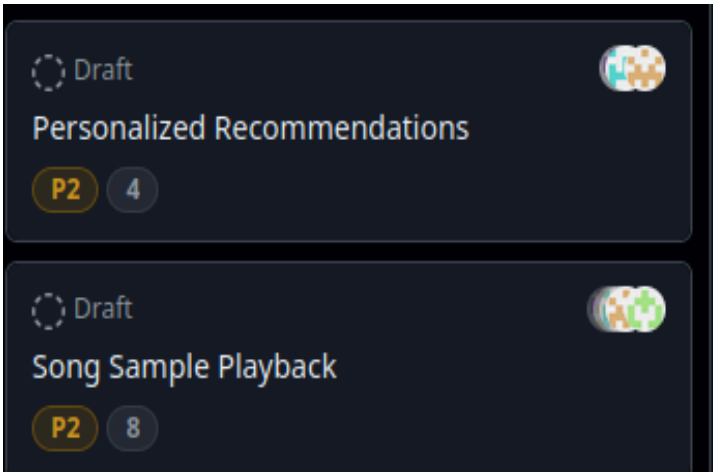
1. **Sprint 1 (Weeks 1–2)**
    - Implement account creation & login (front-end and back-end).

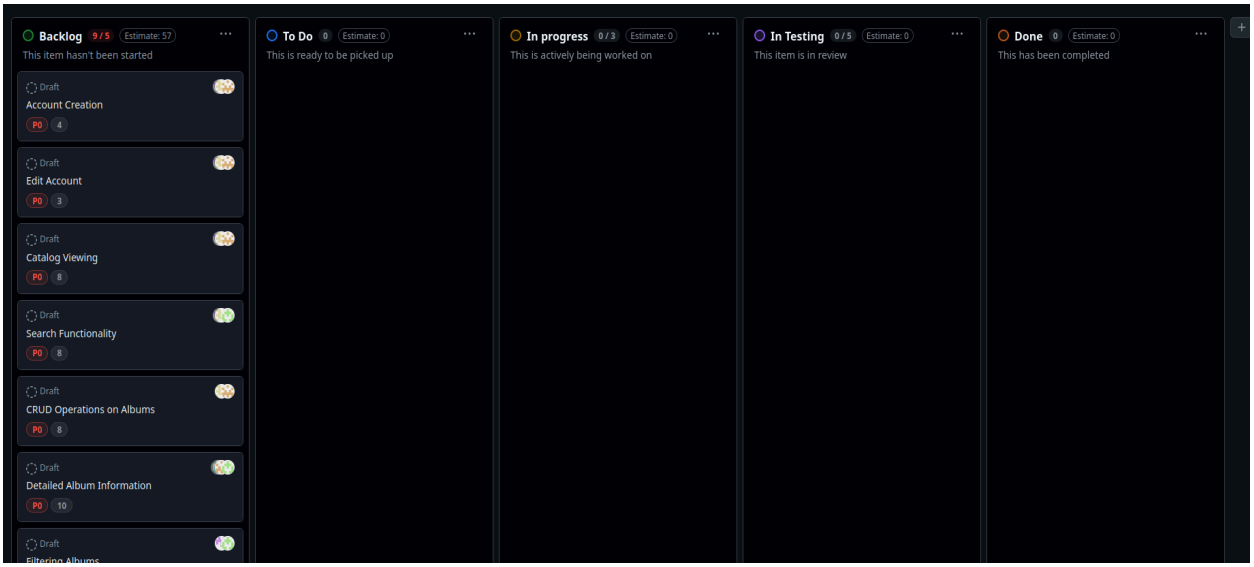
    - Build a basic catalog view (album model, UI).
    - Create search functionality (by title, artist, or genre).
2. **Sprint 2 (Weeks 3–4)**
    - Complete CRUD operations (add, edit, delete).
    - Implement album filtering (genre, year).
    - Expand album details (tracklist, release date, cover art).
3. **Sprint 3 (Weeks 5–6)**
    - (Optional) Personalized recommendations or favorites feature.
    - Integrate samples/pictures for albums.
    - Final refinements, bug fixes, and QA.

4.4 Agile Board Representation

- Will use **GitHub Boards** with “Backlog,” “In Progress,” “Review,” and “Done” columns.
- Each user story is a card with labels for priority and assigned developer(s).


Backlog
9 / 5
Estimate: 57
This item hasn't been started
Draft
Account Creation
P0
4
Draft
Edit Account
P0
3
Draft
Catalog Viewing
P0
8
Draft
Search Functionality
P0
8
Draft
CRUD Operations on Albums
P0
8
Draft
Detailed Album Information
P0
10
Draft
Filtering Albums
P1
4





---

5. Finalized Project Roadmap

5.1 Major Milestones and Deliverables

- **Week 1:** Project setup, user account skeleton, initial album data model.
- **Week 2:** Search feature working prototype, internal demo.
- **Week 3:** CRUD functionality complete.
- **Week 4:** Filtering and expanding album details.
- **Week 5:** Implementation of optional features (recommendations, favorites).
- **Week 6:** Final testing, documentation, and project handoff.

5.2 Key Risks and Potential Challenges

- **Data Completeness:** Incomplete external datasets could impact functionality.
- **Licensing Issues:** Must ensure no misuse of copyrighted materials in track previews.
- **Time Management:** Balancing nice-to-have features while ensuring core requirements are met.
- **Scope Creep:** Keeping focus on MVP despite tempting extra features.

5.3 Next Steps

- Finalize Kanban board with detailed tasks and priorities.
- Begin coding the front-end and back-end stubs.
- Schedule weekly team check-ins and code reviews.