MusFinder: Intelligent Music Recommendation System

Final Report

1. Members and Contributions

Haydon Yang (hyang57@u.rochester.edu)

- Application Implementation
- Drawing of Prototype and Storyboard

Nancy Choi (nchoi4@u.rochester.edu)

- Storyboard
- Use Case 1
- Evaluation of Working Prototype

Shouyi Lin (slin50@u.rochester.edu)

- Implementation method
- Conclusion & Future Work

Wenxing Wang (wwang92@u.rochester.edu)

- Use Cases
- User interface description

2. Introduction

MusFinder is an intelligent music recommendation system designed to integrate with existing music applications, enhancing users' listening experience by providing more personalized and context-aware music suggestions.

2.1 Addressed Problem and Importance

Most music applications generate recommendation lists based on users' recent activity and listening history. While these recommendations often align with users' general taste, they lack contextual adaptability for specific situational needs. For instance, imagine a user who has just completed a difficult exam and is driving home at sunset. The fading sunlight on the lake evokes a relaxed, reflective mood, but their music app continues recommending songs based solely on their usual preferences—such as trap music—without considering the moment's emotional context. Users need a more intuitive and flexible way to communicate their current mood, environment, or situation to receive more relevant and personalized music suggestions.

2.2 Limitation of Existing Products

a. **History-Based Recommendation:** Most music applications generate recommendations based on users' past listening history and liked songs, aiming to suggest tracks that align with their established preferences. While this approach can be effective for reinforcing familiar tastes, it lacks real-time customization and fails to account for users' changing moods or situational needs. These platforms do not offer a direct way for users to request specific types of recommendations beyond generic playlists or broad genre selections. As a result, the recommendations often become predictable and repetitive, limiting musical exploration and failing to introduce users to fresh, diverse content that might better suit their evolving preferences.

b. **User-Created Playlists:** Music applications enable users to create and share playlists based on specific themes or moods, offering a degree of personalization. However, searching for suitable playlists using keywords can be unreliable, often yielding inconsistent quality. There is no guarantee that a playlist has been carefully curated with high quality or truly aligns with a user's intended mood or preference.

c. **Autoplay:** Music application can recommend songs with a similar style to the one the user is currently playing. The music recommendation is very limited to the playing style of the current one, which is not flexible to match other needs.

2.3 Proposed Solution

The solution is designed to be embedded in existing music apps. Instead of relying solely on past habits, the new functions in the app should allow users to describe what they need in the moment and receive suitable music recommendations. Typing a sentence like “I want relaxing music for a sunset drive” or “I want to feel like a secret agent while walking” should generate recommendations that fit the scene. Beyond text, the app could accept images, videos, or short audio clips. A user could snap a picture of the ocean, and the app would suggest songs that match the vibe. User feedback should go beyond simple likes and dislikes. If a song has the right feel but the lyrics don’t fit, the user should be able to say, “I love the melody, but please give me more like this with better lyrics.” If they love the intro but not the chorus, they should be able to ask for songs with a similar opening style. The more detailed the feedback, the better the recommendations. This solution makes music recommendations feel less like a static algorithm and more like an interactive, real-time experience tailored to what the user actually wants at the moment.

3. User Requirement Gathering and Analysis

3.1 Interview

This study explores how the environment affects music preferences to improve context-aware recommendations. Participants (18–30) chose music for different settings , discussing mood, activity, and discovery habits. Insights will refine personalized music suggestions.

3.1.1 Target Participants

a. **Group 1:** 18–22 (undergraduates at UR)
b. **Group 2:** 23–30 (young professionals in Rochester)

3.1.2 Interview Steps

a. **Environmental Scenarios**

   Present various settings or weather scenarios (e.g., sunny day by the river, rainy commute, busy city street, quiet café).

b. **Music Preferences**

   Have participants pick their top three preferred song types/genres for each scenario and explain how the environment influenced these choices.

c. **Contextual Discussion**

   Ask follow‑up questions about factors like mood, activity level, and whether location or weather impacts their music discovery interest.

d. **Data Collection & Analysis**

   Compare preferences within and between groups, noting how different environmental conditions alter music selection.

Use insights to refine your app’s environment‑aware recommendation features.

3.1.3 Interview Questions

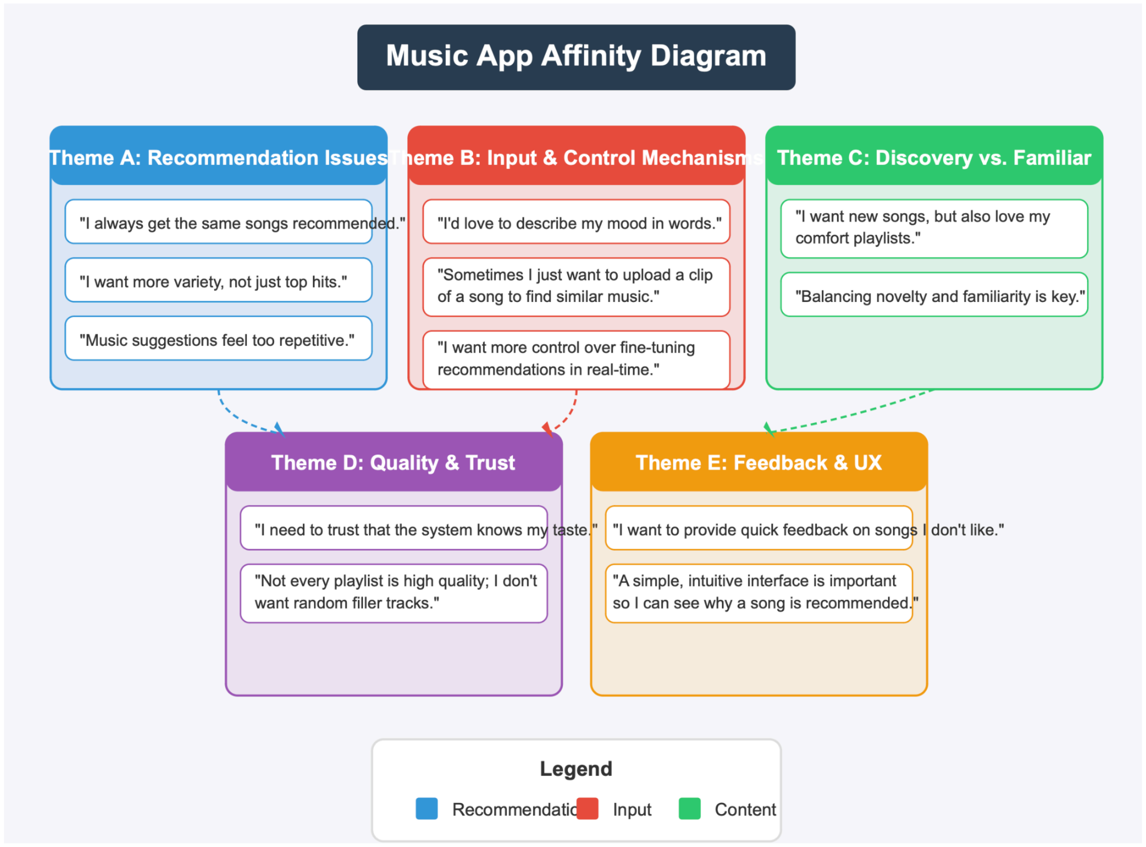
a. **Environmental Influence**
   - How does your physical environment (e.g., weather, location) affect your music choices?
   - Can you name your top three genres for different scenarios (e.g., rainy commute, quiet café)?

b. **Music Preferences & Current Solutions**
   - Do you rely on autoplay, daily recommendations, or playlists? Why or why not?
   - How well do current platforms adapt to your real-time mood or setting?

c. **Future Expectations**
   - Would you use a system that takes text, images, or videos as input for recommendations?
   - How useful would real-time feedback be in improving music suggestions?

3.2 User Analysis

3.2.1 Affinity Diagram


Music App Affinity Diagram
Theme A: Recommendation Issues
"I always get the same songs recommended."
"I want more variety, not just top hits."
"Music suggestions feel too repetitive."
Theme B: Input & Control Mechanisms
"I'd love to describe my mood in words."
"Sometimes I just want to upload a clip of a song to find similar music."
"I want more control over fine-tuning recommendations in real-time."
Theme C: Discovery vs. Familiar
"I want new songs, but also love my comfort playlists."
"Balancing novelty and familiarity is key."
Theme D: Quality & Trust
"I need to trust that the system knows my taste."
"Not every playlist is high quality; I don't want random filler tracks."
Theme E: Feedback & UX
"I want to provide quick feedback on songs I don't like."
"A simple, intuitive interface is important so I can see why a song is recommended."
Legend
Recommendation
Input
Content

3.2.2 Personas

a. **Persona 1: Alex**

Alex is a 21-year-old college student living near the Genesee River, currently pursuing a Bachelor's degree. Curious and open-minded—yet sometimes indecisive—Alex loves exploring new music but often ends up listening to the same hip hop playlists. He walks by the river to clear his head, studies at cafes, and hangs out with friends, so he wants music that fits each of these different atmospheres. Traditional recommendation engines keep pushing him the same artists and tracks, leaving Alex frustrated and eager to branch out into jazz, R&B, and other genres. He's heard about a new music recommendation app that personalizes playlists based on mood and location, and he's excited to see if it can introduce him to unfamiliar sounds in a seamless way. However, Alex worries about how complicated the app might be to set up—he wants quick, intuitive features without having to fill out endless surveys or navigate a steep learning curve.

b. **Persona 2: Dio**

Dio is a freelance graphic designer with a Bachelor's degree in design who juggles multiple projects while moving around for meetings, co-working sessions, and coffee breaks. He needs music that can adapt to his shifting energy levels—something chill during focused work, upbeat when he's traveling, and fresh enough to avoid the repetitive playlists that quickly bore him. Dio is also passionate about exploring global cultures through music and wants an easy way to discover new artists from around the world. He often wonders if there's a feature that can tailor recommendations to his location or daily schedule. While he's excited by the idea of a smart recommendation engine, he's wary of services that rely on shallow or stale data; he wants personalized, dynamic suggestions that truly keep pace with his on-the-go lifestyle.

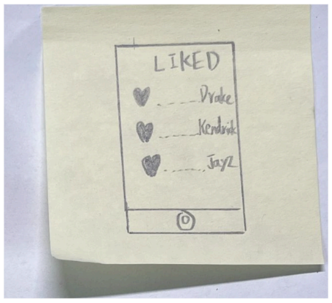
4. Low-fidelity Prototyping

4.1 Storyboard

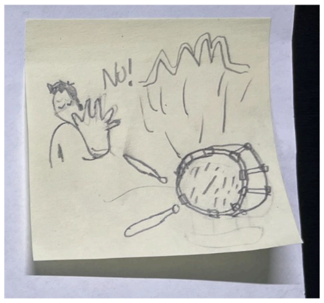
Alex, a 21 year-old college student, is on a walk during sunset along the Genesee River. As Alex typically listens to hip-hop, he can't find songs that match the atmosphere. Typical music recommendation systems primarily focus on a user's past listening habits and interactions, limiting their ability to explore new genres or discover music that aligns with different moods, activities, and environments. Moreover, none of these assure quality and are often quite fixated on styles.

Alex wants to explore new music that fits any atmosphere (in this case, a relaxed ambiance). He discovers a music recommendation system with features where he can find songs that match his surroundings/desires! These features consist of user input (text, images, video, etc), identify location/activity level + sensors, and allow feedback. Alex uses the text input feature (‘*recommend me songs that fit the vibes of a walk during sunset on a nice summer day, preferably R&B/jazz music’*), and a music recommendation playlist is created for him! At last, Alex is happy he finally found songs to his liking.

Alex, a 21 year-old college student, is on a walk during sunset along the Genesee River.

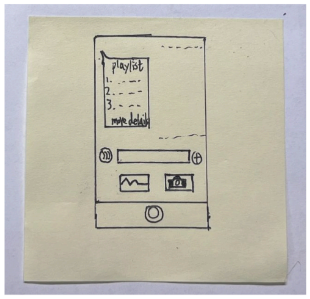


As he typically listens to hip-hop, he can’t find songs that match the atmosphere.

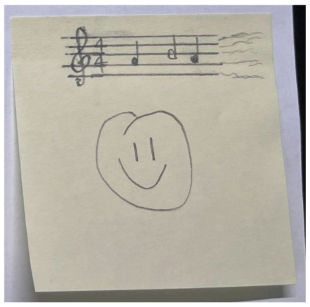


Alex wants to explore new music that fits any atmosphere (in this case, a relaxed ambiance)

JAZZ
TEXT
PHOTO
LOCATION
He discovers a music recommendation system with features where he can find songs can match his surroundings/desires!



User input (text, images, video, etc), identify location/activity level + sensors, recommend music based on location, feedback


Alex is happy :)

4.2 User Flow

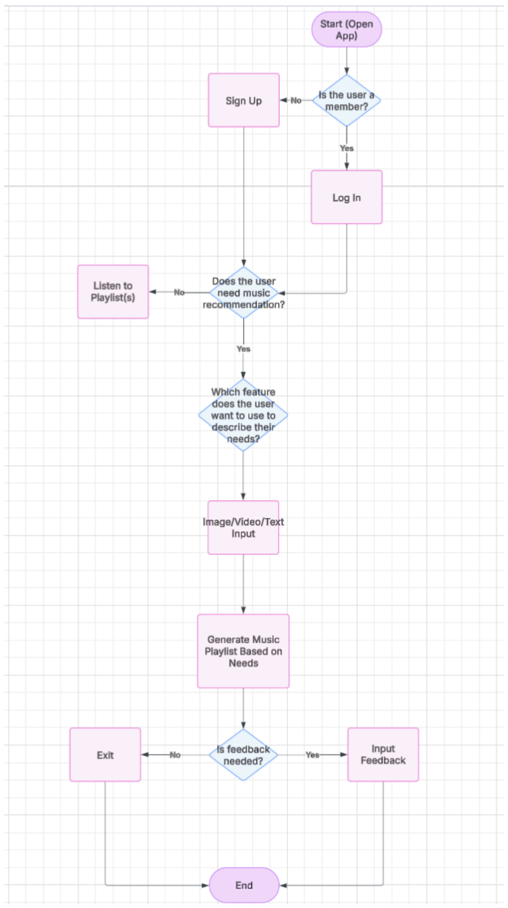
4.2.1 Use Case 1

Users can upload a photo or enter keywords to find music matching that mood or vibe, helping users convey an atmosphere visually or verbally when genre filters aren't enough. This is perfect for those who seek quick, mood‑focused music discovery. Based on the user input, a photo or keyword(s) yields recommendations that match the intended atmosphere. Customer satisfaction would be 4 if results reliably capture the user's desired vibe. Customer dissatisfaction would be 3 if visual or keyword interpretation is off, resulting in forced manual searches.

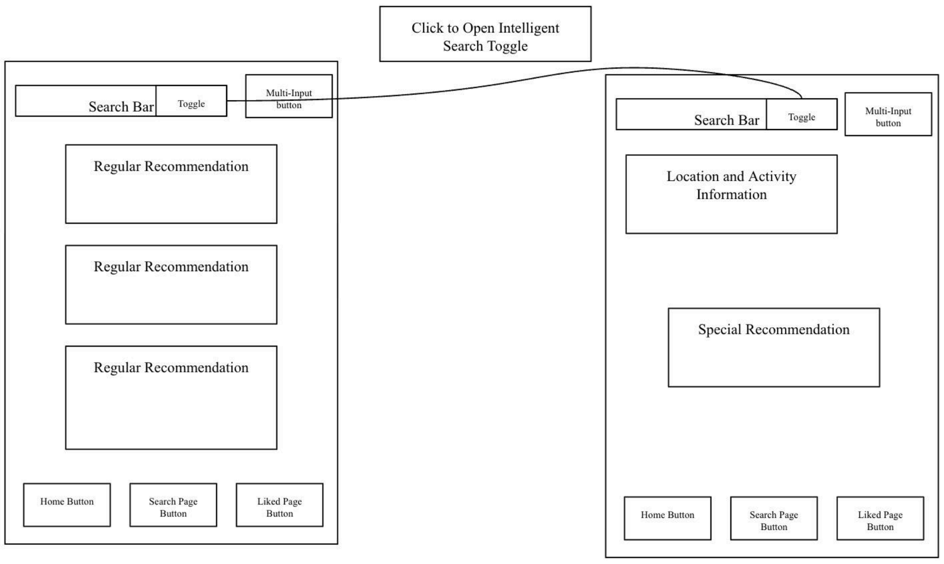
As for the User Flow Diagram, we first begin by opening the app and signing up/logging in. If the user does not need music recommendations, they can simply listen to existing playlists. If music recommendations are needed, then the user can choose which feature they would like to utilize to describe their needs: in this case, image or text input. Then, a music playlist will be generated accordingly.

For the Wireframe/Wireflow, upon opening the app, there is a search bar at the top that users can toggle; this allows users to input image, video, or text. This feature is similar to AI chatbots i.e ChatGPT, Claude, in the sense that users are able to easily and quickly describe their needs and receive results, providing an efficient and effective experience. Lastly, a music recommendation playlist is curated specifically for the user!

4.2.1.1 User Flow Diagram



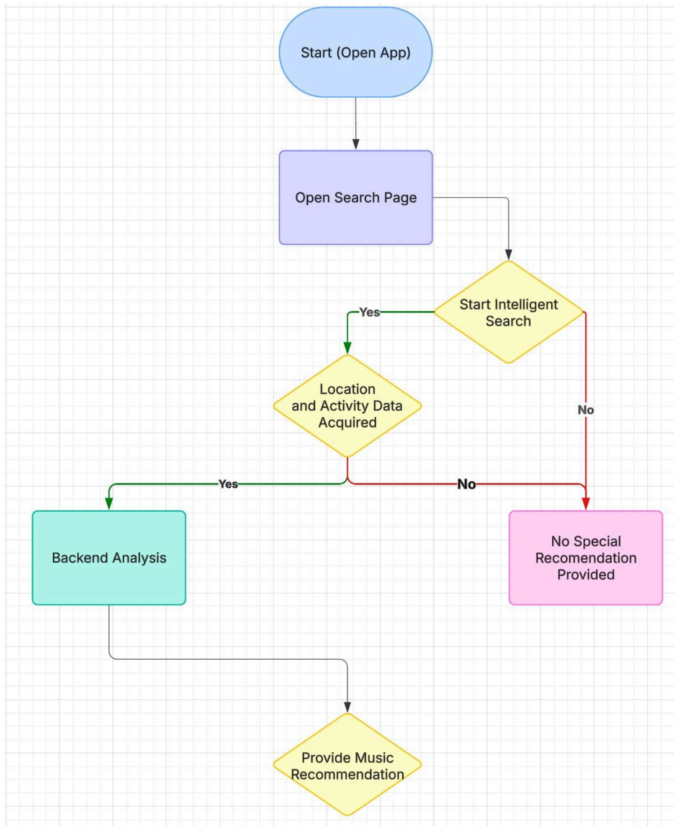
4.2.1.2 Wireframe + Wireflow



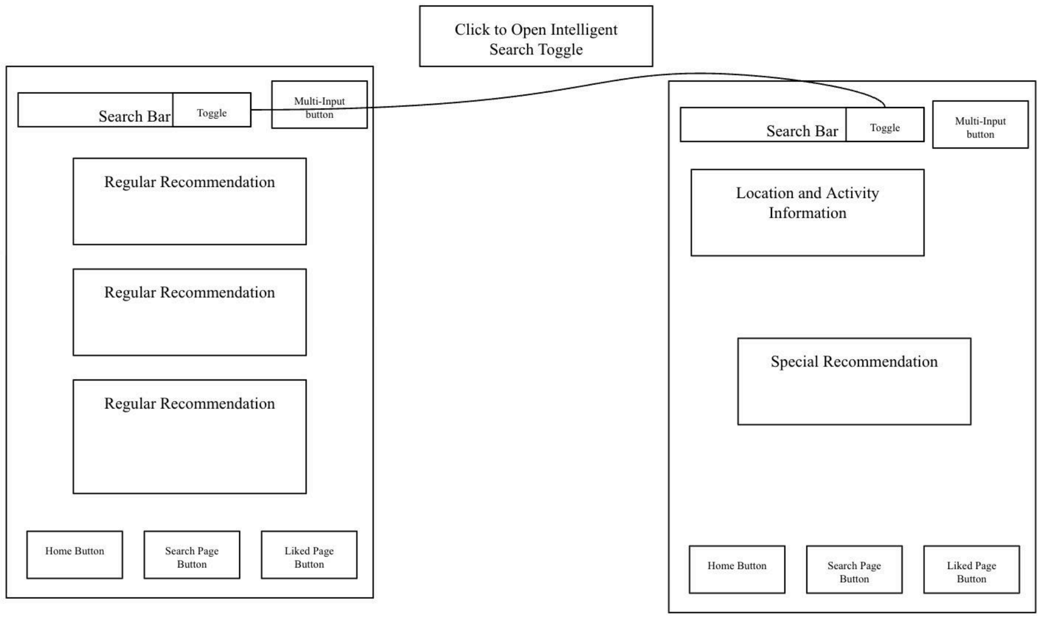
4.2.2 Use Case 2

Users can receive automatic music recommendations tailored to their location and activity, such as heart rate (BPM), whether they are walking or sitting, and their movement speed. This ensures the music aligns with their current pace and environment for a more immersive listening experience.

4.2.2.1 User Flow Diagram


Start (Open App)
Open Search Page
Start Intelligent Search
Yes
No
Location and Activity Data Acquired
Yes
No
Backend Analysis
No Special Recomendation Provided
Provide Music Recommendation

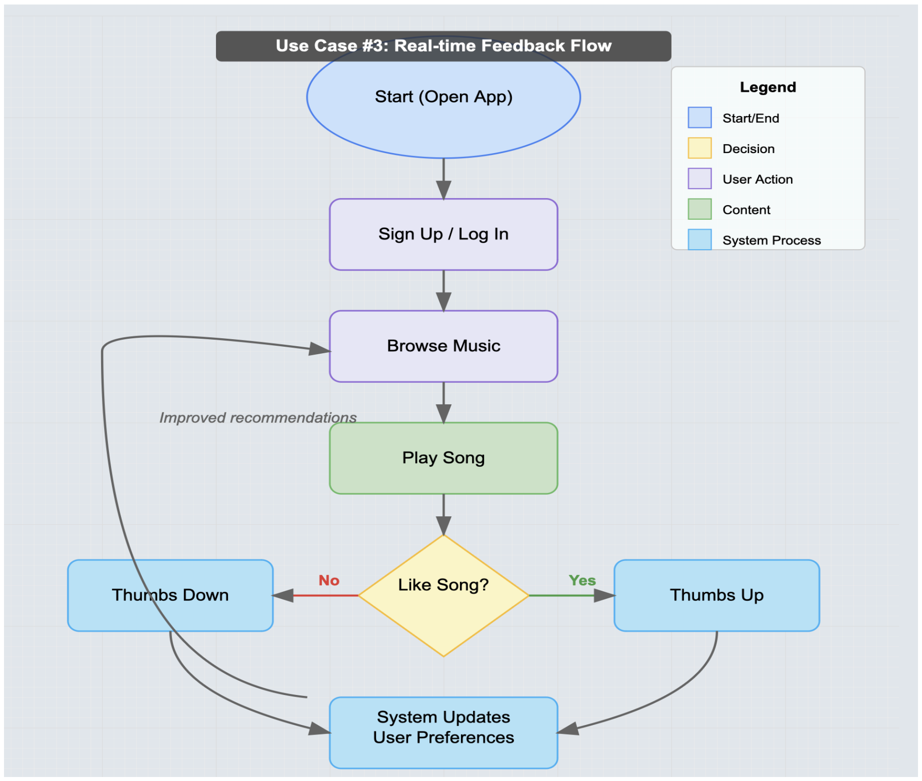
4.2.2.2 Wireframe + Wireflow



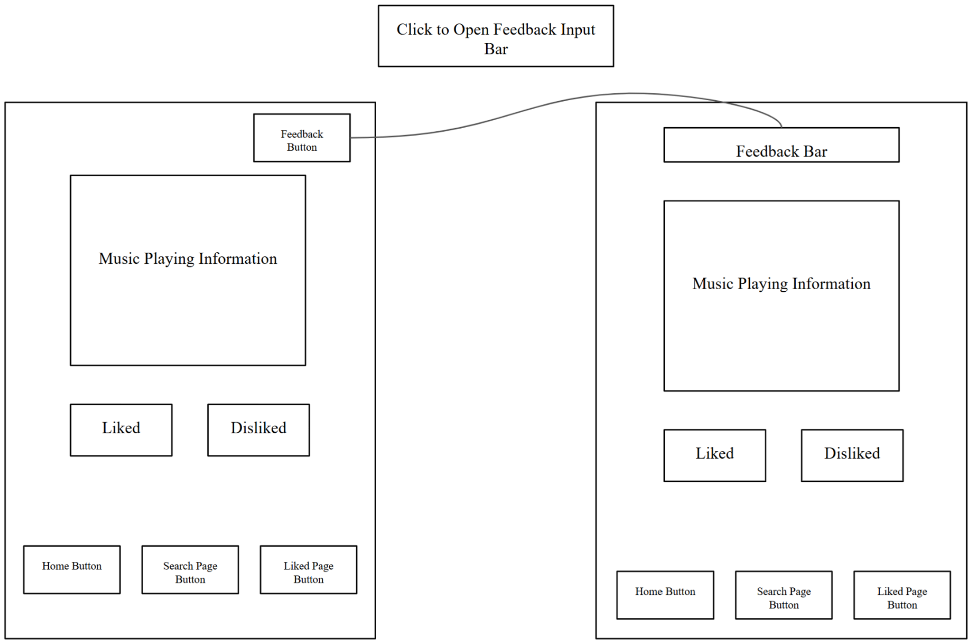
4.2.3 Use Case 3

The system enables users to provide instant thumbs up/down feedback on songs, immediately updating their preference profile to improve future recommendations. This minimal-effort interaction helps personalize music suggestions more effectively, addressing the needs of users who want control over how the system adapts to their tastes. Success is measured by noticeable improvements in recommendation quality after feedback, with high user satisfaction (4/5) when the system delivers better suggestions and moderate dissatisfaction (3/5) when feedback appears ineffective.

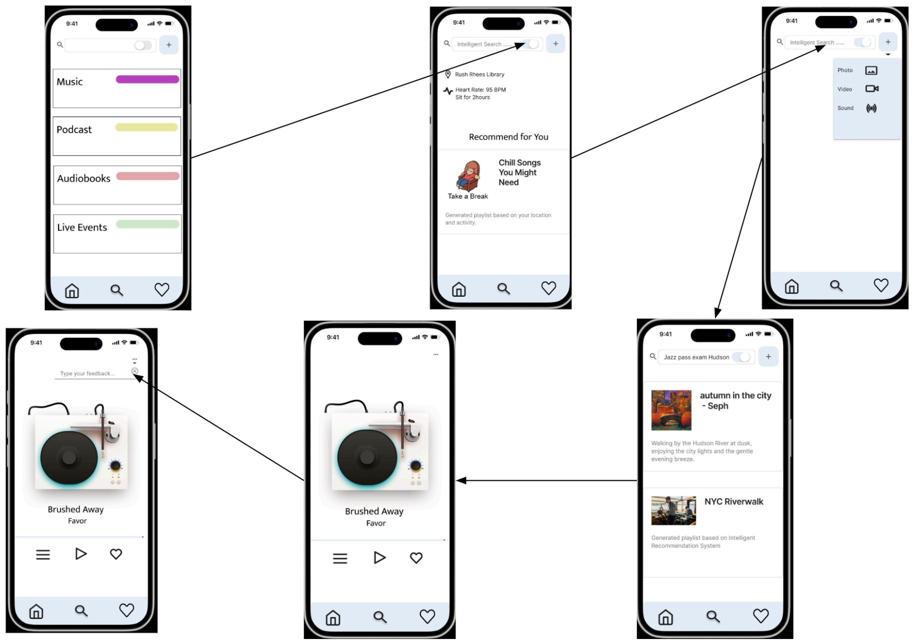
4.2.3.1 User Flow Diagram



4.2.3.2 Wireframe + Wireflow



4.3 Prototyping


9:41
Music
Podcast
Audiobooks
Live Events
Intelligent Search ......
Rush Rhees Library
Heart Rate: 95 BPM
Sit for 2hours
Recommend for You
Chill Songs You Might Need
Take a Break
Generated playlist based on your location and activity.
Photo
Video
Sound
Jazz pass exam Hudson
autumn in the city - Seph
Walking by the Hudson River at dusk, enjoying the city lights and the gentle evening breeze.
NYC Riverwalk
Generated playlist based on Intelligent Recommendation System
Brushed Away
Favor
Type your feedback...

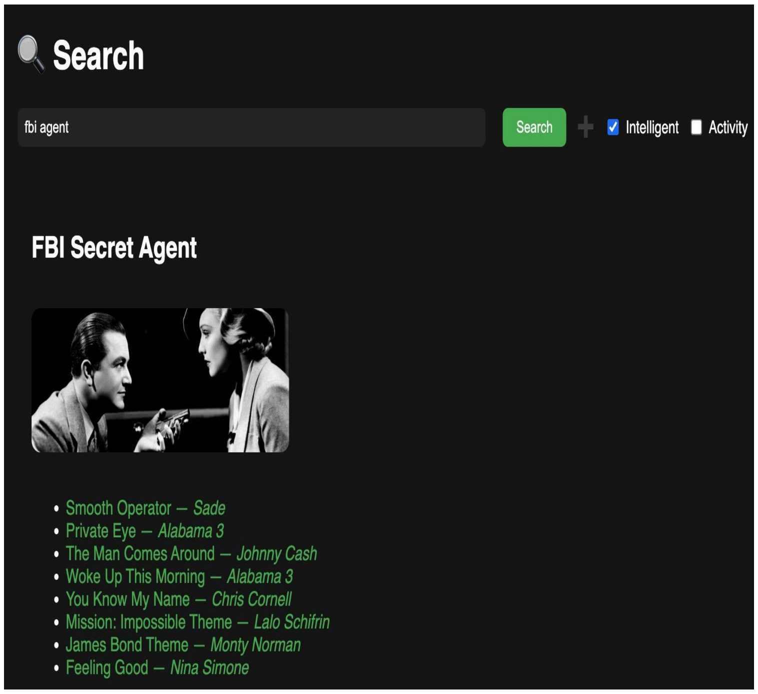
5. Working Prototype and Implementation

5.1 Use Case 1: Photo/Keyword-Based Music Recommendation

5.1.1 User Interface Description

The interface features a clean, dark-themed search page where users can type in keywords or upload an image to initiate a music recommendation. The top portion contains a search bar with a clearly labeled button, allowing for intuitive input. Once a query is submitted, the system returns a curated list of songs that match the intended mood or vibe derived from the keyword or visual input. Results are displayed in a vertical list, with artist names and song titles shown in a readable font and highlighted in different colors for clarity. An image panel may appear to enhance the connection between the visual cue and the results.

The screenshot shows an example where the keyword **"FBI Agent"** is entered. The search returned tracks such as *"Smooth Operator"* by Sade and *"Mission: Impossible Theme"* by Lalo Schifrin, which align with a mysterious, agent-themed atmosphere. This visual demonstrates how the system interprets abstract concepts and moods and successfully matches them to appropriate music suggestions.



fbi agent

[✓ Intelligent] Activity

FBI Secret Agent

- Smooth Operator — *Sade*
- Private Eye — *Alabama 3*
- The Man Comes Around — *Johnny Cash*
- Woke Up This Morning — *Alabama 3*
- You Know My Name — *Chris Cornell*
- Mission: Impossible Theme — *Lalo Schifrin*
- James Bond Theme — *Monty Norman*
- Feeling Good — *Nina Simone*

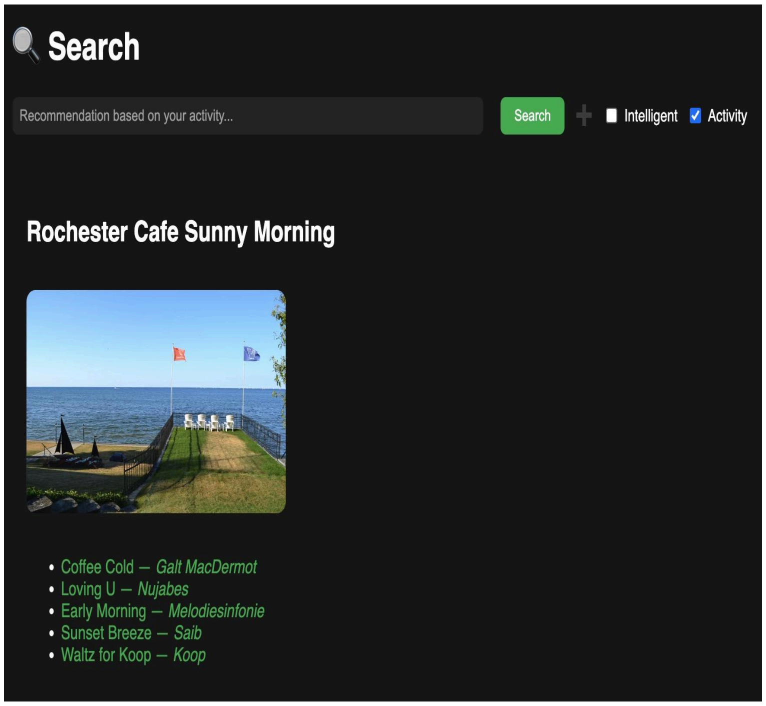
5.2 Use Case 2: Context-Aware Music Recommendation

5.2.1 User Interface Description

This interface displays a smart search tool that automatically adapts music recommendations based on real-time user context, such as location, weather, and time of day.

The top portion features a passive search input that reads, “Recommendation based on your activity...” with an optional manual search bar. Once the system detects environmental cues, a label appears summarizing the detected context (e.g., "Rochester Cafe Sunny Morning"), followed by a recommended playlist below. Each track is listed in a clean, readable format, offering a seamless listening experience without requiring manual input. Visual cues like a background image also help reflect the current atmosphere.

The screenshot shows the system identifying the user’s setting as “Rochester Cafe Sunny Morning.” In response, it suggests soft, mellow tracks such as *"Coffee Cold"* by Galt MacDermot and *"Wait for Koop"* by Koop—songs that complement a relaxing, sunny morning cafe vibe. This snapshot highlights how the app tailors recommendations to external conditions, promoting effortless, contextually relevant music discovery.



☐ Intelligent ✓ Activity

Rochester Cafe Sunny Morning

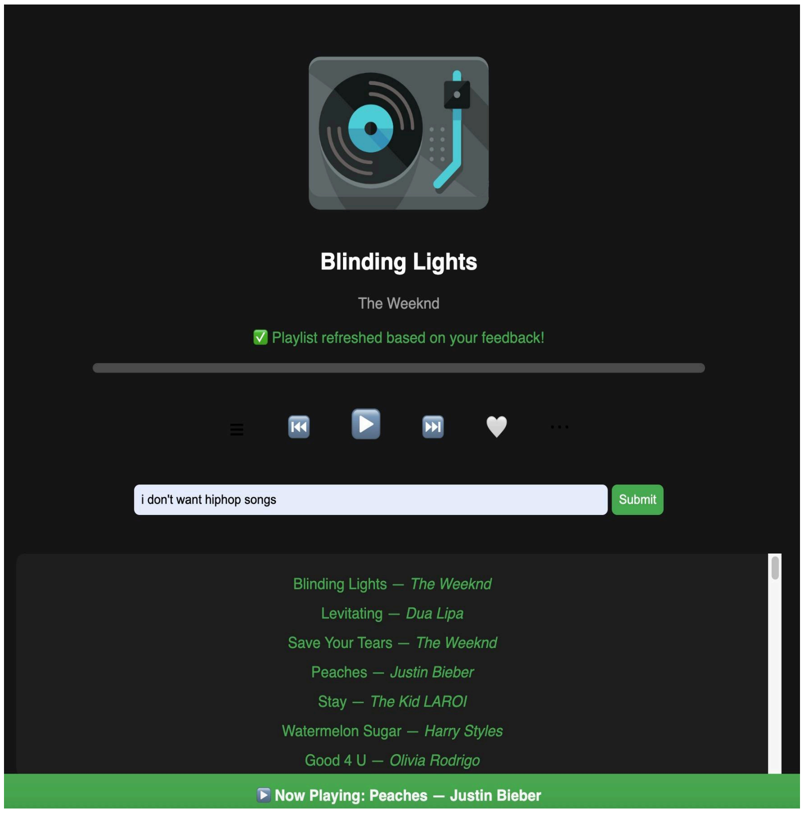
- Coffee Cold — *Galt MacDermot*
- Loving U — *Nujabes*
- Early Morning — *Melodiesinfonie*
- Sunset Breeze — *Saib*
- Waltz for Koop — *Koop*

5.3 Use Case 3: Feedback-Based Music Refinement

5.3.1 User Interface Description

The interface for this feature allows users to interact with songs by giving real-time feedback through thumbs up/down icons or short text responses. Centered on the screen is the currently playing track, with the artist name and song title clearly visible. Below the song card, users can see a status update like "Playlist refreshed based on your feedback!" to confirm that their input has been processed. A feedback input box is available for text-based preferences (e.g., "I don't want hip-hop songs"), with a 'Submit' button to register the comment. The updated playlist then appears below, showcasing the system's adaptive recommendations. Key functions like pause/play and like/dislike icons are clearly accessible.

The screenshot shows a user playing *"Blinding Lights"* by The Weeknd. After submitting feedback ("I don't want hiphop songs"), the system updates the playlist, adding artists like Olivia Rodrigo and Harry Styles while reducing hip-hop tracks. The interface highlights the playlist update with a green notification, showing that the feedback was applied instantly—demonstrating effective personalization with minimal user effort.



Blinding Lights

The Weeknd

✅ Playlist refreshed based on your feedback!

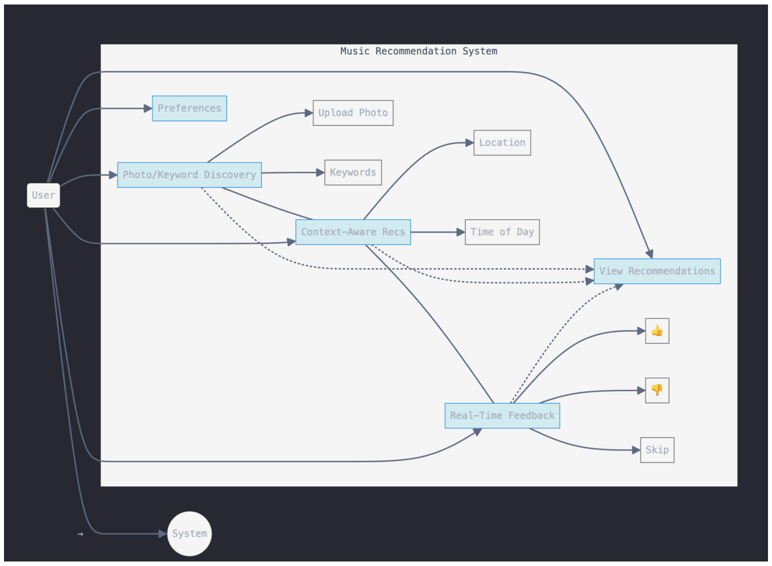
i don't want hiphop songs

Submit

Blinding Lights — *The Weeknd*
Levitating — *Dua Lipa*
Save Your Tears — *The Weeknd*
Peaches — *Justin Bieber*
Stay — *The Kid LAROI*
Watermelon Sugar — *Harry Styles*
Good 4 U — *Olivia Rodrigo*

▶️ Now Playing: Peaches — Justin Bieber

5.4 Use Case Diagram



6. Implementation Method

6.1 Development Environment

Our project was developed using Python 3.11 with the Flask web framework. We used Visual Studio Code (VS Code) as the primary code editor on macOS machines. The backend database used SQLite, managed through SQLAlchemy for ORM support. For version control, we used Git, with optional integration to GitHub for collaboration and backup.

6.2 Front-End Development

The front-end was implemented using standard HTML and CSS, designed to be mobile-responsive for better usability across devices. The layout followed a Spotify-inspired theme, featuring a fixed music player bar and custom CSS animations for interactive buttons. User input forms and playlist interactions were handled dynamically using AJAX, allowing smooth page updates without reloads. All routes were connected to Flask, enabling seamless front-to-back communication.

6.3 Back-End Development

The backend was developed using Flask to manage routing, server logic, and API integration. A Large Language Model (LLM) API was integrated to provide intelligent music recommendations based on user input. The SQLite database was used to store song metadata, preloaded with 20 sample songs for testing. We used SQLAlchemy to define and interact with the Song model. Backend routes handled intelligent search, playlist generation, and user feedback filtering. All data updates were performed dynamically, enabling real-time interactivity with the front-end.

7. Evaluation of Working Prototype

7.1 Goal & Key Questions

7.1.1 Goal

Evaluate the usability, effectiveness, and user satisfaction of MusFinder, an intelligent music recommendation system. The evaluation focuses on three main features:

a. Photo/keyword-based discovery
b. Context-aware recommendations
c. Real-time feedback learning

7.1.2 Key Questions

a. Can users successfully find music that matches their desired mood using image or text input?
b. Does the system adjust recommendations accurately based on real-time context (location, activity)?
c. Does user feedback meaningfully refine subsequent music suggestions?

7.2 Selection of Participants

a. **Actual number recruited:** 5 participants for the informal usability evaluation (conducted as a pilot study).
b. **User backgrounds:**

- **Age:** 18–30
- **Gender:** 3 male, 2 female

- **Computer Experience:** All were regular users of mobile apps and music streaming services like Spotify or Apple Music.
- **Professional domain:** University students (undergrad and grad) and young professionals.
- **Education:** College-level or higher.
- **Culture:** Primarily U.S.-based but some international diversity.
- **Motivation and Mood:** Generally motivated to discover better music for varying activities and moods.
- **Disabilities:** No participants reported disabilities that affected the test.

7.3 Development of the Evaluation Tasks

Each task included success metrics based on user satisfaction and system responsiveness. Participants were assigned one task per major use case.

7.3.1 Task 1

a. **Description:** Use the photo/keyword input feature.
b. **Prompt:** Upload a FBI-agent related photo or type "FBI agent vibe," and evaluate if the recommendations fit.

7.3.2 Task 2

a. **Description:** Experience context-aware recommendations.
b. **Prompt:** Walk outside (or simulate movement via location services) and check if song recommendations change appropriately.

7.3.3 Task 3

a. **Description:** Provide real-time feedback on a song.
b. **Prompt:** Give a thumbs-up or thumbs-down and observe whether the upcoming songs adjust accordingly.

7.4 Test Procedure

7.4.1 Pre-test (5 minutes):

Participants completed a background survey: age, music habits, preferred music apps, and expectations.

7.4.2 Test (15–20 minutes):

a. **Task 1:** Try uploading a photo or entering a keyword.
b. **Task 2:** Experience the location-based adjustment feature.
c. **Task 3:** Interact with the feedback system during listening.

7.4.3 Post-test (15 minutes):

a. **SUS (System Usability Scale) survey:** Standardized 1–5 scale on ease of use, intuitiveness, satisfaction.
b. **Semi-structured interview questions:**
    - "Did MusFinder understand your needs?"
    - "How intuitive was the feedback system?"
    - "What improvements would you like to see?"

7.5 Data Collection

7.5.1 Data Collected

a. **Task success rates** (whether users completed the tasks as intended)

b. **Task completion time**

c. **Error counts** (e.g., misinterpreted inputs, wrong recommendations)

d. **SUS scores** (1–5 scale)

e. **Interview feedback** (written notes)

f. **Behavioral observations** (hesitations, confusion, excitement)

Three aspects of usability:

g. **Effectiveness**

- % of users who successfully completed each task
- Self-reported match between input and recommendations (target 80%)

h. **Efficiency**

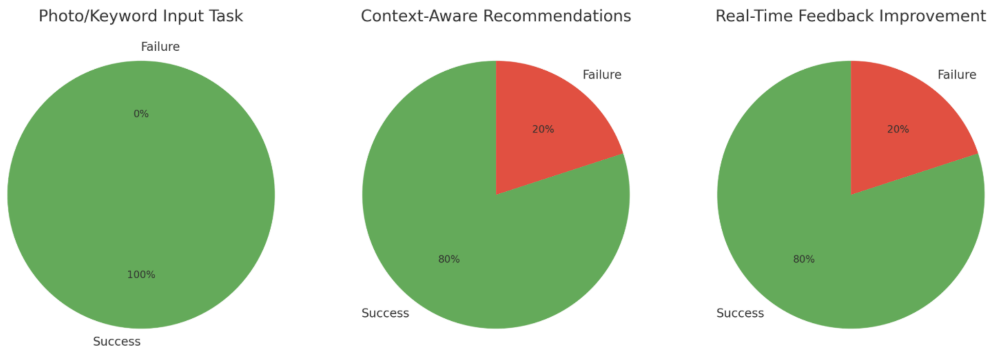
- Time taken to complete each task
- Number of errors (e.g., mis-clicks, needing help)

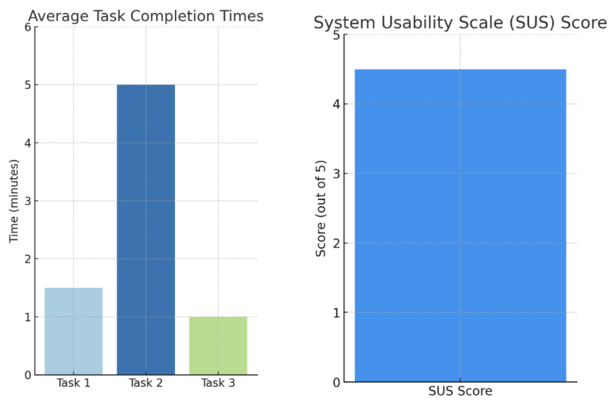
i. **Satisfaction**

- Average SUS score
- Qualitative feedback (positive or negative comments)

7.6 Evaluation Results

7.6.1 Quantitative Analysis





7.6.1.1 Task success rate

a. 100% completed the photo/keyword input task.

b. 80% experienced context-aware recommendations adjusting correctly.

c. 80% noticed an immediate improvement after giving real-time feedback.

7.6.1.2 Average task completion times

a. Task 1: ~1.5 minutes

b. Task 2: ~5 minutes (due to GPS simulation/setup)

c. Task 3: ~1 minute

7.6.1.3 Error Count

a. Minor: 1 user accidentally uploaded the wrong image format.

7.6.1.4 SUS score

a. Average: 4.5/5

b. Indicating strong usability and user satisfaction.

7.6.2 Qualitative Analysis

7.6.2.1 User Feedback Themes

a. "The system felt very easy to navigate, similar to Spotify."

b. "I loved that the music actually matched the picture I uploaded!"

c. "Real-time feedback worked well but could respond a little faster."

7.6.2.2 Observed Behavior Patterns

a. Users naturally explored the app without much instruction after the briefing.

b. Some hesitation on uploading images (format confusion), but quickly recovered.

c. Enthusiastic comments about expanding the music library for even more diversity.

8. Conclusion and Future Work

8.1 Summary

Our project successfully explored the development of an intelligent music recommendation system that leverages a large language model (LLM) API. The main goal—to enable responsive and personalized music suggestions based on user input—was achieved. Findings from our evaluation suggest the system's responsive layout and real-time feedback loops positively impacted user satisfaction, with users describing the interface as smooth and easy to use.

8.2 Limitations

One key limitation was the size and scope of our initial music dataset, which limited the diversity of recommendations. Additionally, while the integration of the LLM API worked well, prompt tuning remained a challenge, occasionally resulting in generic or ambiguous responses. Our evaluation was also limited to a small user sample, which may not reflect broader user needs or accessibility challenges.

8.3 Key Learnings

Through this project, we learned how responsive interaction design enhances perceived usability, especially when paired with real-time updates. We also gained practical experience in combining AI-driven APIs with user-facing features, and saw firsthand how personalization drives user engagement.

8.4 Future Work

Future improvements include refining the visual design for better clarity and accessibility, improving prompt engineering for more accurate recommendations, and conducting larger-scale user testing to uncover additional usability insights—particularly from a diverse set of users.