MusFinder : Intelligent Music Recommendation System

Team Members : Haydon Yang, Nancy Choi, Wenxing Wang, Shouyi Lin

Introduction

Project Overview:

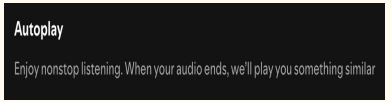
- Developed an intelligent music recommendation system combining friendly interface and LLM feature
- Implemented real-time interaction features to enhance the music discovery experience

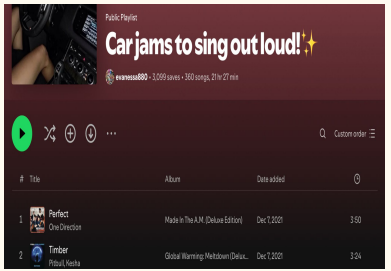
Motivation:

- Traditional music recommendation systems often lack flexibility and personalization
- Our goal: Create a smarter system that adapts to user input and provides dynamic recommendations

Key Features:

- Real-time playlist updates and feedback handling
- Integration of large language model (LLM) API for intelligent search and suggestions
- Integrated with existing music application





Use Case 1

Description:

Users can upload a photo or enter keywords to find music matching that mood or vibe.

Rationale:

Helps users convey an atmosphere visually or verbally when genre filters aren't enough.

Source:

People seeking quick, mood-focused music discovery

Fit Criterion:

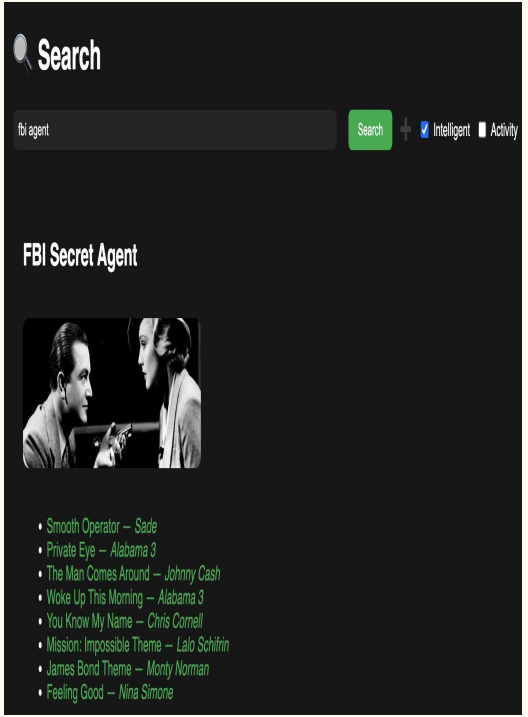
A photo or keywords yield recommendations that match the intended atmosphere.

Customer Satisfaction:

4 if results reliably capture the user's desired vibe.

Customer Dissatisfaction:

3 if visual or keyword interpretation is off, forcing manual searches.



Use Case 2

Description:
The system automatically recommends music based on the user's real-time environment (location, weather, time of day).

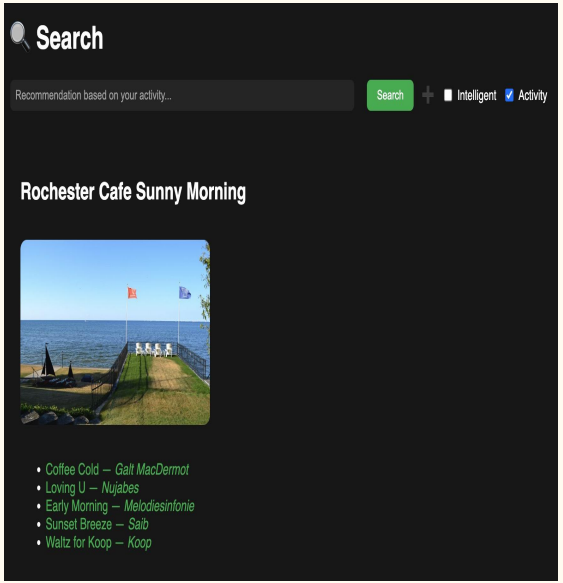
Rationale:
Saves user effort by matching music to changing conditions without manual searching.

Source:
Users seeking personalized, context-aware recommendations.

Fit Criterion:
Suggestions consistently align with environment and user preferences.

Customer Satisfaction:
4 if recommendations are reliably relevant and reduce user effort.

Customer Dissatisfaction:
3 if users must still manually curate music due to poor environment matching.



Use Case 3

Description:

The system collects quick user feedback (e.g., thumbs up/down) to refine future recommendations.

Rationale:
Real-time feedback personalized suggestions more effectively.

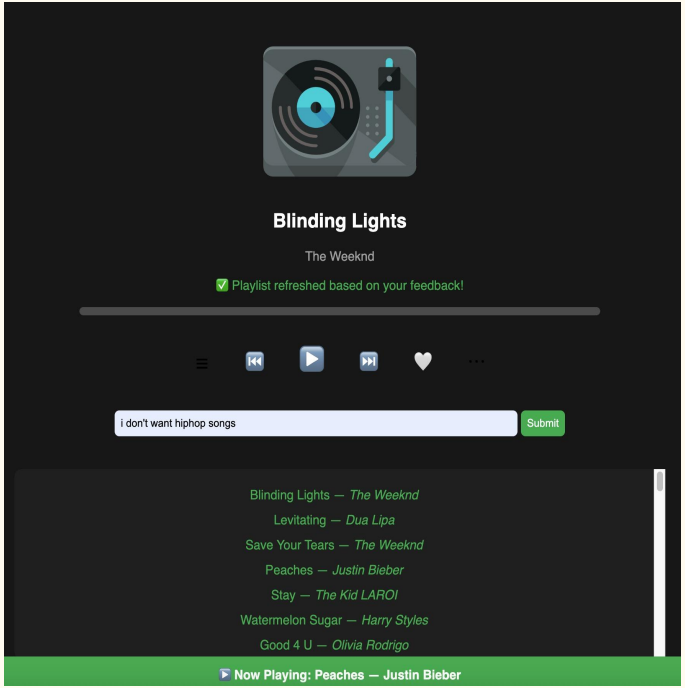
Source:
Users wanting control over how the system adapts to their tastes.

Fit Criterion:
Minimal-effort feedback leads to noticeably improved recommendations.

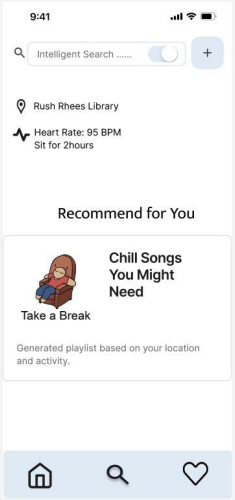
Customer Satisfaction:

4 if users see immediate positive changes in their music feed.

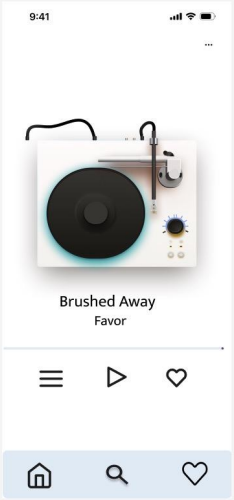
Customer Dissatisfaction:
3 if feedback doesn't significantly influence upcoming suggestions.



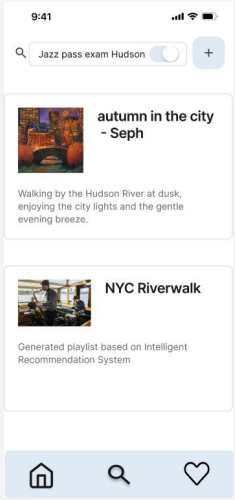
Activity



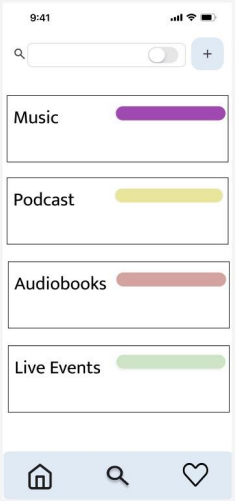
Play Page



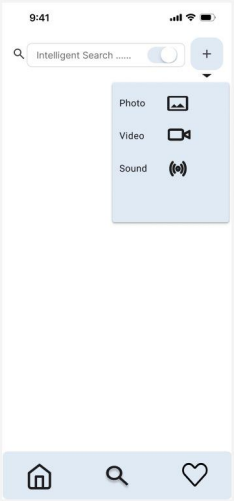
Recommendation



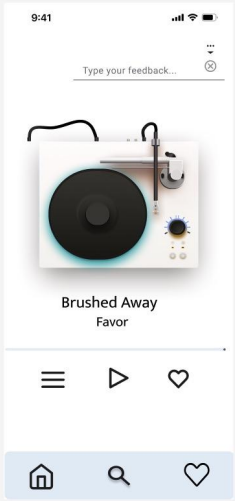
Search Page

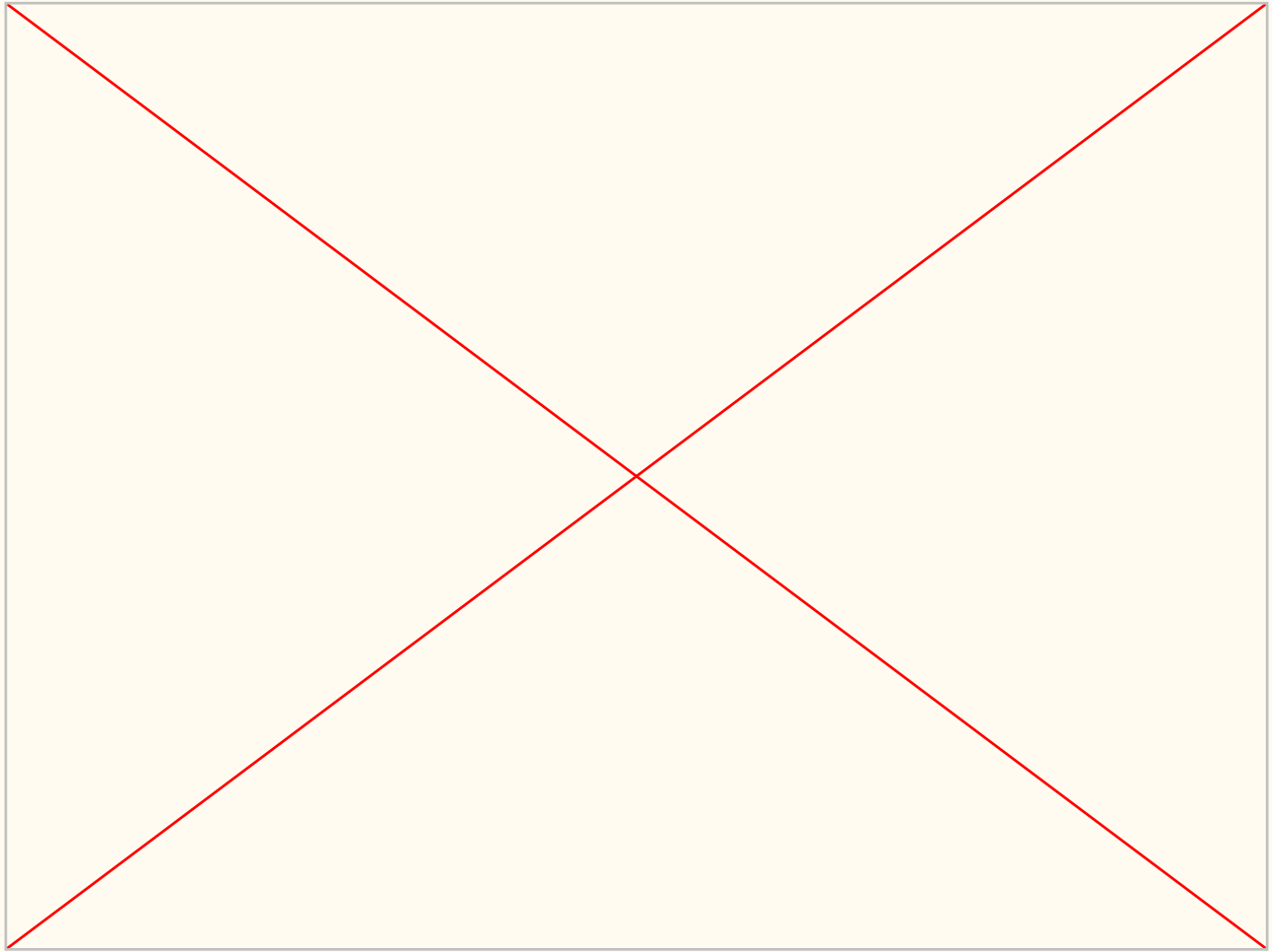


Meida Input



Play & Feedback





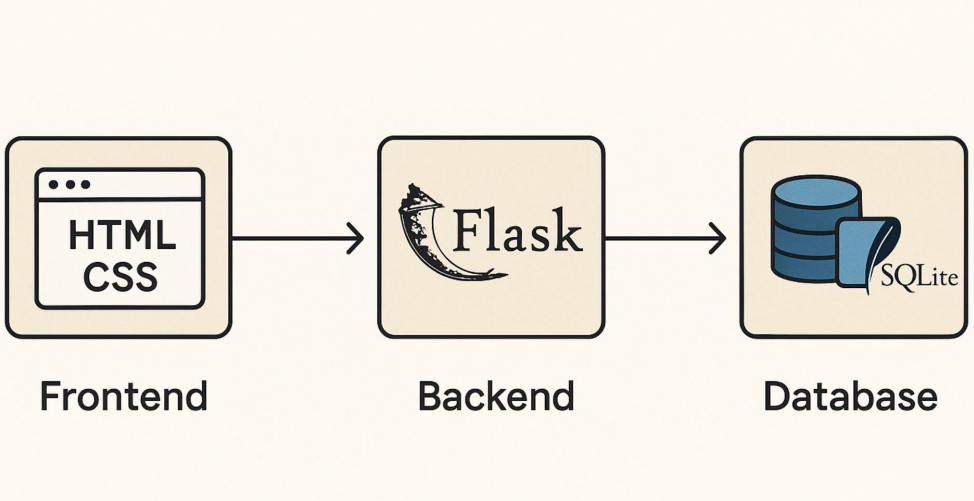
Implementation Method

Development Environment:

- Built with Python Flask framework
- Used SQLite database managed with SQLAlchemy
- Version control with Git

Frontend Development:

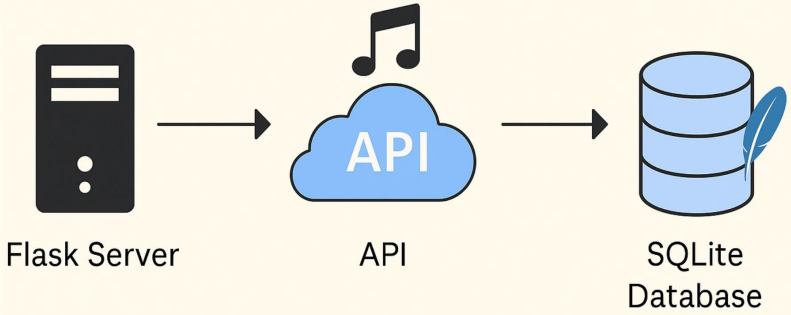
- Standard HTML + CSS, Flask routes
- Spotify-inspired style with fixed player bar
- AJAX handles pages' contents real-time updates



Implementation Method

Backend Development:

- Backend implemented using Flask
- SQLite used for relational database
- Integrated large language model (LLM) API for intelligent search and music recommendations
- Dynamic playlist generating and users' feedback on music recommendation fine-tuning



Evaluation

Goal: Evaluate the usability, effectiveness, and user satisfaction of MusFinder by focusing on three main features:

- Photo/keyword-based discovery
- Context-aware recommendations
- Real-time feedback learning

Key Questions:

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

Setup: 5 participants affiliated with University of Rochester; regular users of mobile apps and music streaming services i.e. Spotify or Apple Music.

Development: Participants were assigned one task per major use case –

- Task 1: Use photo/keyword input feature
  - Prompt: Upload a sunset photo or type "relaxing music for a sunset drive," and evaluate if the recommendations fit.
- Task 2: Experience context-aware recommendations
  - Prompt: Walk outside (or simulate movement via location services) and check if song recommendations match appropriately.
- Task 3: Provide real-time feedback on a song
  - Prompt: Input feedback and observe whether the upcoming songs adjust accordingly.

Evaluation

Test Procedure:

Pre-test (5 minutes): Participants completed a background survey: age, music habits, preferred music apps, and expectations.

Test (15~20 minutes):

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

Post-test (15 minutes):

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

Data Collection:

What data were collected?

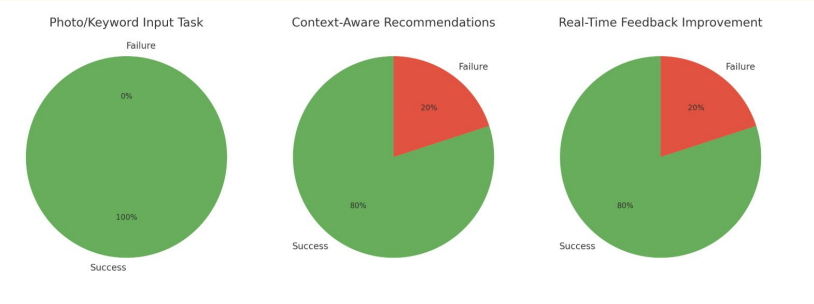
- Task success rates (whether users completed the tasks as intended)
- Task completion time
- Error counts (e.g., misinterpreted inputs, wrong recommendations)
- SUS scores (1~5 scale)
- Interview feedback (written notes)
- Behavioral observations (hesitations, confusion, excitement)

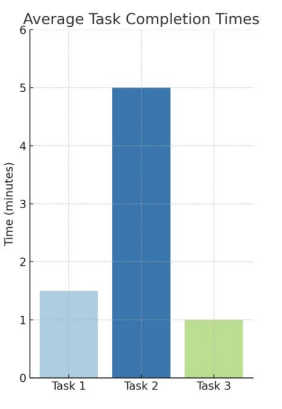
Three aspects of usability:

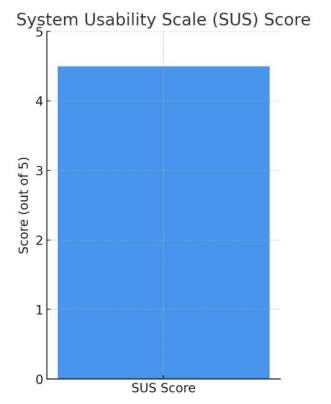
- Effectiveness
    - % of users who successfully completed each task
    - Self-reported match between input and recommendations (target 80%)
- Efficiency
    - Time taken to complete each task
    - Number of errors (e.g., mis-clicks, needing help)
- Satisfactory
    - Average SUS score
    - Qualitative feedback (positive or negative comments)

Evaluation Results

Quantitative Analysis:







Qualitative Analysis:

User Feedback Themes:

- "The system felt very easy to navigate, similar to Spotify."
- "I loved that the music actually matched the picture I uploaded!"
- "Real-time feedback worked well but could respond a little faster."

Observed Behavior Patterns:

- Users naturally explored the app without much instruction after the briefing.
- Some hesitation on uploading images (format confusion), but quickly recovered.
- Enthusiastic comments about expanding the music library for even more diversity.

Main Takeaways

Project Highlights:

- Built a possible intelligent music recommendation solution for the existing music application
- Leveraged large language model API for smart recommendations

HCI Insights:

- Responsive layout and real-time feedback made users feel in control and reduced interaction delays, leading to a smoother and more satisfying experience
- Personalized recommendations, powered by user input and language model interpretation, increased user engagement by aligning results more closely with individual preferences

Future Work:

- Improve the visual design of our pages and refine the prompts for the LLM as well as the feedback analysis.
- Conduct larger-scale user testing for deeper usability evaluation

Thank You!

Apr 28, 2025