CSCI-5530

Computer Science Capstone

June 2025

Ashraf Saad

CariBeats

Michaella A. Zias

mz02156@georgiasouthern.edu

michachu25@gmail.com

1. Background

CariBeats is an Android and Web music streaming application focused on Caribbean genres, including soca, reggae, dancehall, kompa, reggaeton, and more. Still, it can be used for any kind of music. The app was designed to enhance the listener's experience by intelligently queuing songs by tempo (BPM), genre, and mood, avoiding abrupt transitions that are common in most streaming apps.

This project addresses a major gap in existing media players. They typically shuffle or play songs in playlist order, without any beat or tempo logic. CariBeats offers an enhanced experience using BPM-aware playlist clustering, mood-adaptive themes, and integration with the Spotify Web API for real music previews. (Still a work in progress)

---

2. Evolution of the Project

- **Initial Stage (Intro to Software Engineering)**: Originally a simple project in Eclipse with a manually created, very minimal dataset and no API integrations.
- **Second Stage (Handheld/Ubiquitous Computing)**: Introduced Jamendo API, enabling basic streaming and keyword search.
- **Current Stage (CariBeats)**:
    - Developed as a full Android Studio app
    - Integrated Spotify Web API for 30-second preview playback. (Still a work in progress)
    - Uses a Kaggle dataset for offline features (no audio, metadata only)
    - Implements machine learning (BPM-based clustering) for playlist flow
    - Expanded with an HTML/JS website (GitHub Pages) for lightweight web access and toggle-based search

---

3. Purpose

To provide an intelligent music player that curates playlists based on tempo, genre, and mood. It delivers a seamless and enhanced Caribbean music listening experience.

---

4. Scope

- **Android App** (primary platform)
- **Web App** (HTML/JS GitHub site)
- **Spotify API Integration** for music playback (Still a work in progress)

---

5. Technical Architecture

Piece	Platform	Purpose
Android App	Android Studio	Core mobile experience with genre filtering, playlist creation
HTML/JS Website	GitHub Pages	Lightweight web access with toggle-based filters
Spotify Web API	Android & Web	Actual music playback (30s previews) with Play, Pause, Skip, Back (Still a work in progress)

---

6. Functional Requirements

6.1 User Interface

- **Main Screen** — App title and description, as well as the main menu
- **Genre Selection Screen** — Dropdown spinner for genre, filters, and displays song list
- **Song List** — Displays BPM, genre, song title, artist, and mood of song
- **Now Playing Screen**
    - Play, Pause, Skip, Back buttons
    - Mood label
- **Playlist Screen**

    - Save playlists
    - Rename and manage playlists

- **Stats Screen**
    - Displays user listening insights and allows the user to reset insights

- **Settings Screen**
    - Theme toggle (Dark/Light Mode)

6.2 Genre Selection

FR1.1: User selects a genre from the dropdown (reggae, soca, etc.)
FR1.2: System displays matching songs from the local dataset

6.3 Song Playback

FR2.1: Clicking song opens playback activity **FR2.2**: Auto-queue based on BPM ±5 tolerance
FR2.3: Play/Pause/Next/Previous controls

6.4 Mood Clustering

FR3.1: K-Means BPM clustering into:

- Chill (slow)
- Moderate (mid)
- Hype (fast)
  FR3.2: UI theme reflects song's mood cluster

6.5 Playlist Management

FR4.1: Users can create/save custom playlists
FR4.2: Playlists persist between sessions

6.6 Search

FR5.1: Search by title or artist

FR5.2: Real-time dynamic filtering

6.7 Statistics

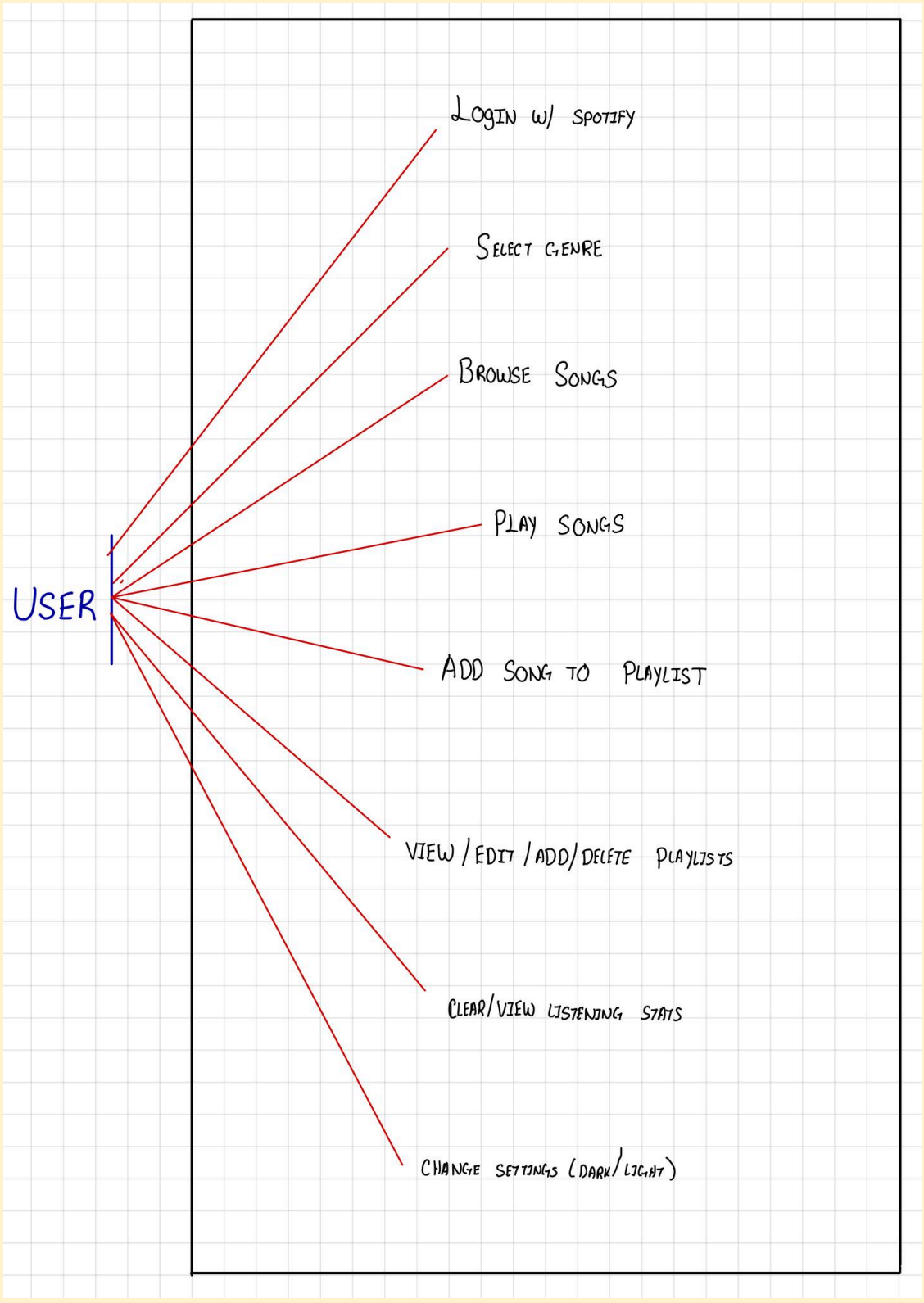
FR6.1: Display listening stats (by mood/genre)
FR6.2: Stats update as user interacts

6.8 Spotify API Integration (Optional)

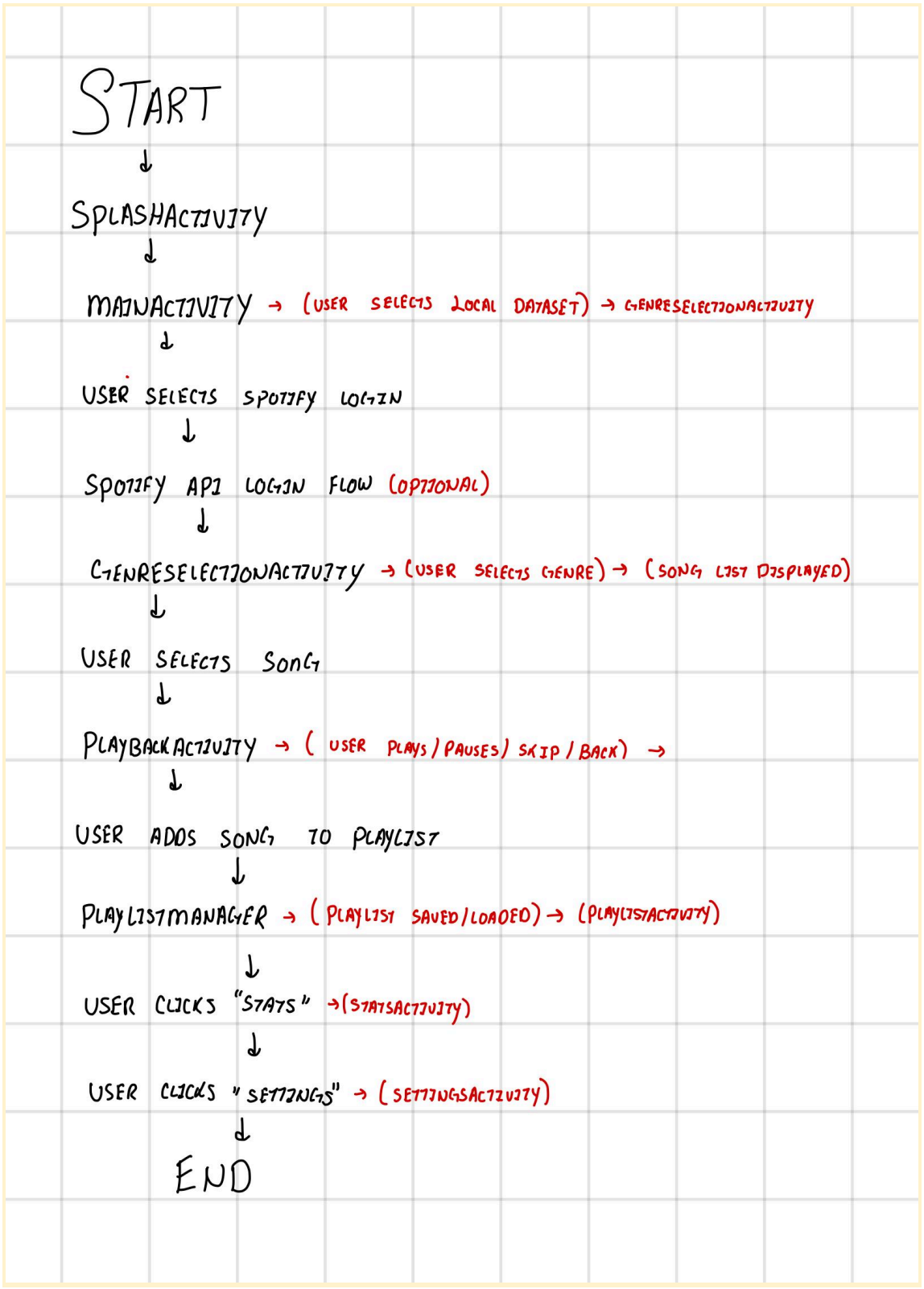
FR7.1: Spotify user authentication

FR7.2: Stream full tracks (if available)

Use Case Diagram



Work Flow Diagram



---

7. Non-Functional Requirements

- Responsive UI (dark/light mode)
- Graceful handling of missing BPM/audio data
- Offline functionality with preloaded dataset
- Smooth transitions, low-latency playback
- Toast/snackbar feedback on user actions
- Support Android API level 24+
- Responsive design for all screen sizes
- Web app compatible with modern browsers

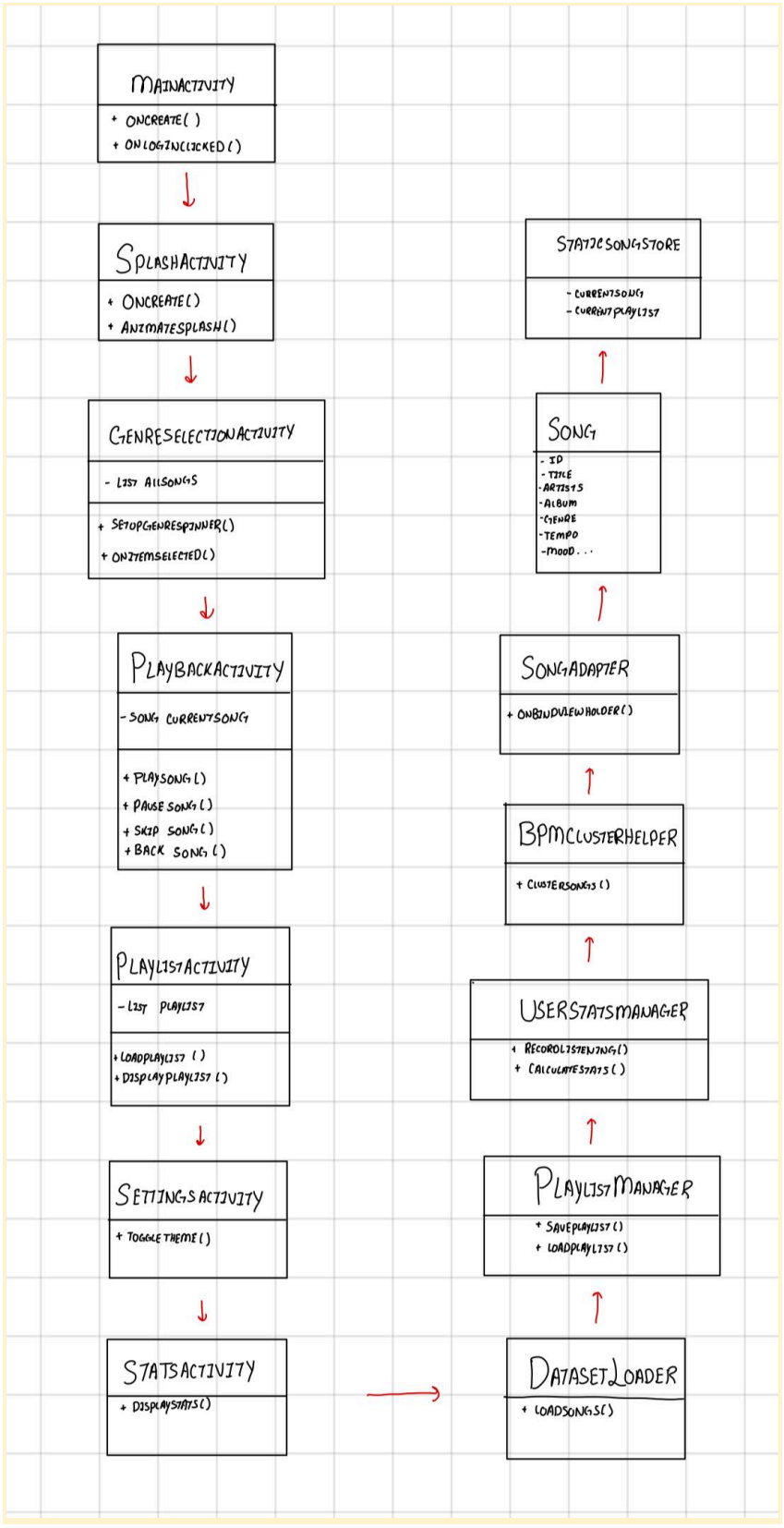
---

8. System Architecture — Android Modules

Module	Purpose
MainActivity	App entry point, description screen
SplashActivity	Animated splash screen
GenreSelectionActivity	Genre selection and song list
Song	Stores track data
SongAdapter	Binds song data to UI
PlaybackActivity	Playback controls
PlaylistActivity	Playlist management
PlaylistManager	Stores user playlist data
StatsActivity	User listening statistics

SettingsActivity	Theme settings
DatasetLoader	Loads the Kaggle dataset
StaticSongStore	Passes song data between activities
BPMClusterHelper	Clusters songs by BPM
UserStatsManager	Stores and computes stats

Workflow Diagram



---

9. Key Features

- Genre-based filtering
- BPM clustering (Chill, Moderate, Hype)
- Playlist creation and editing
- Mood-based UI themes
- User listening statistics and insights
- Web version with HTML/JS filters
- Spotify 30s preview with playback controls (Still a work in progress)

---

10. Cybersecurity Concerns

- Secure API key storage
- HTTPS for external API calls
- Internet permissions in the manifest are only for required features
- Secure handling of the audio stream URL
- No personal identifiable data collected

---

11. Challenges

- **Jamendo API Limitations** — BPM data is often missing or inaccurate
- **Spotify Web API Integration** — Complex OAuth and preview limitations
- **Modular Development** — Seamlessly combining Android app + Web app + Spotify API requires additional development effort

---

12. Future Enhancements

- Full Spotify authentication for user playlist access
- Full song playback beyond 30-second preview
- Smart mood-based playlist generation
- Hybrid genre suggestions based on user preferences
- Patent and commercialization roadmap
- Helps people create, edit, and perform straight from the app itself
- Analyze music on a more complex level, not just machine learning the tempo, and making the groups

---

13. Conclusion

CariBeats is an evolving project that bridges Android mobile development, machine learning, and web development, with a deep focus on Caribbean music culture. The app already delivers value through its intelligent playlisting and Spotify integration. With further refinements (OAuth, full playback, better UX), it has strong potential for future commercialization.

Work will continue beyond this semester to merge all components and pursue a possible patent and product launch, as this is in high demand.

---

14. References

Maharshi Pandya. (2022, October 22). *Spotify Tracks Dataset*. Kaggle. https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset

Spotify. (n.d.). *Spotify Web API Documentation*. https://developer.spotify.com/documentation/web-api/

Jamendo. (n.d.). *Jamendo API Documentation*. https://developer.jamendo.com/v3.0

Google. (n.d.). *Android Developers Guide*. https://developer.android.com/

GitHub. (n.d.). *GitHub Pages Documentation*. https://pages.github.com/

https://mz02156.github.io/CariBeats-Web/