

# Abstract

- DailyDrop is an iOS application that enables users to share a song each day in response to a prompt. The application combines a minimal frontend design in SwiftUI with a Firebase backend for real-time storage and retrieval of user submissions. Spotify's public Web API is integrated for song metadata. The project demonstrates principles in mobile application development, API integration, data persistence, and user session handling.

## Vision Statement

- Today, people constantly share personal media such as photos, videos, and messages with friends/community. However, this is not done for music. Most people listen to music on a mobile app using one of the many streaming services, Spotify being the largest. Daily Drop aims to add a social layer to daily music consumption by facilitating the consistent flow of music between friends who are part of a community (College Campus). The main purpose of this app is to harness music's power to bring people together to strengthen local community and friendships, as well as expose users to new music.

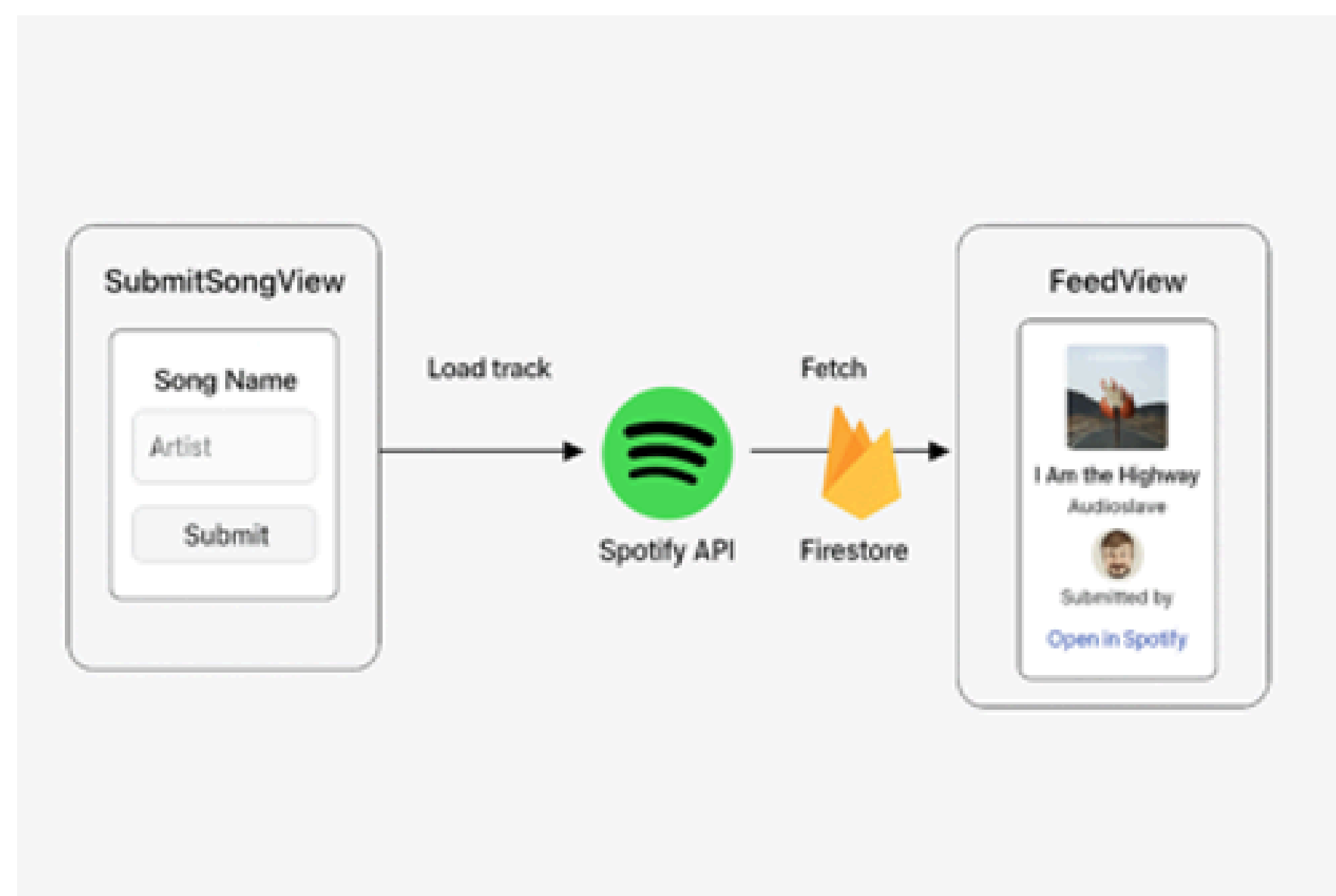
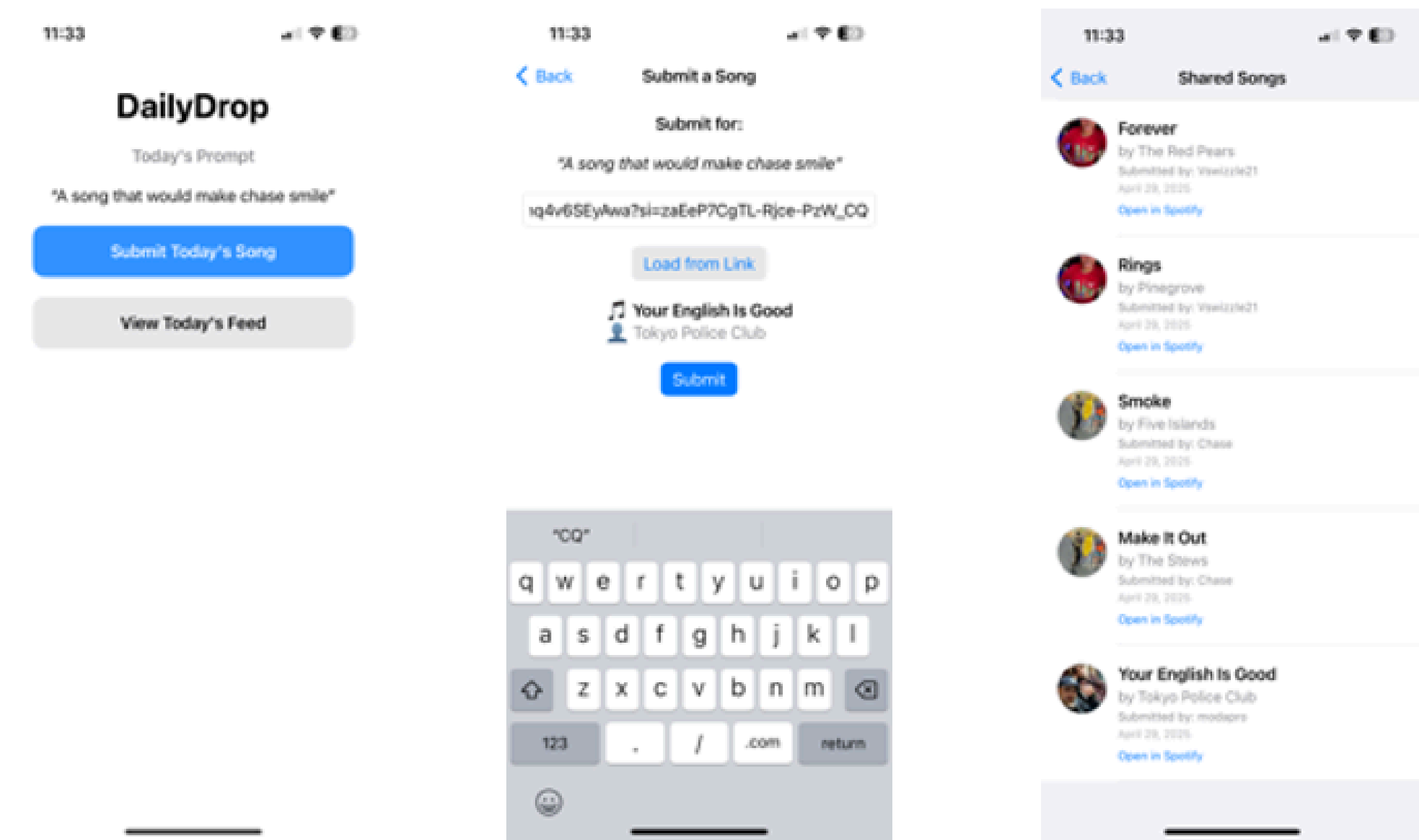
## App Architecture

- Frontend: SwiftUI-based interface using MVVM pattern
- Backend: Firebase Firestore (NoSQL) for scalable cloud data storage
- Authentication: Firebase Auth for anonymous user identity
- External API: Spotify Web API for track metadata
- Local Storage: UserDefaults for profile persistence

# Daily Drop

Mohammad Zaidi

Trinity College Computer Science Senior Capstone Project



## Technical Challenges

- Reliable parsing of Spotify links
- Async UI updates and Firebase listeners
- Image data compression for Firestore constraints
- Time-sensitive daily prompt fetching
- Device testing without TestFlight

## Key Features

- Daily song prompt retrieved from Firestore
- Song submission via Spotify URL with metadata autofill
- Profile creation (username + optional avatar)
- Real-time submission feed filtered by prompt
- One-tap deep link to open track in Spotify

## Result

- Stable iOS deployment with multiple-user testing
- Real-time CRUD integration with Firebase
- Functional Spotify integration with metadata display
- Hosted on GitHub and ready for TestFlight