

GeoMelody

Know your music, and where it's streamed

Functional Specification

Group members:

Christopher Cussen: 19424146

Ali Fouad Mazbouh: 18372503

Date 11/11/2022

Table of Contents

1. Introduction

- ❖ Overview
- ❖ Business Context
- ❖ Glossary

2. General Description

- ❖ System Functions
- ❖ User Characteristics and Objectives
- ❖ Operational Scenarios
- ❖ Constraints

3. Functional Requirements

- ❖ Defining the PWA's requirements

4. System Architecture

- ❖ Graphical User Interface
- ❖ System Backend

5. High-level Design

- ❖ Object Diagram
- ❖ Context Diagram
- ❖ Use Case Diagram
- ❖ Class Diagram

6. Preliminary Schedule

7. Appendices

1. Introduction

1.1 Overview

For our project we decided to create a progressive web application which will take audio input through a mobile phone and will then use an existing API like Aud-D to identify the inputted song, record where and when the user inputted the audio into the PWA, and display information on both the song and the artist. We also plan to implement neural network technologies to reduce any unwanted noise in the recording to improve the system's ability to identify a song playing in a noisy environment.

While similar applications do exist, like Shazam for example, this project will be a progressive web application which not only identifies the song for you, but will be able to store every song you identify in a database accompanied by a timestamp of when you recorded the audio and where. This will make it easier for the user to remember which song they heard when they try to find the song later on. As well as this, the web app will recommend similar artists for the user to listen to based on the song that was recorded.

Our reasoning for implementing this service as a progressive web application over a mobile application is that we want the service to be as easily accessible as possible. If a user is at a music festival, for example, and hears a song that they like but do not know they will only have a very finite amount of time to identify the song before it ends and their chance is gone. PWA's require no download and so if a user wants to identify a song but hasn't got the app, the likelihood that they succeed in this is greater if the service is available instantly on the web. Also, PWA's implement progressive improvement principles and so the service will work for any user regardless of their preferred browser. So whether it's an Android phone or IOS, the service will be instantly available.

1.2 Business Context

1.3 Glossary

- ❖ **Progressive Web Application (PWA)** → An application designed to utilise modern Web capabilities to provide users with a better experience than a traditional native application. Essentially a hybrid between a mobile application and a regular web page that runs inside a web browser and can be viewed as a website anchored by CSS, HTML, or other web-related standards and have the same look and feel as typical native apps.
- ❖ **Graphical User Interface (GUI)** → The front end of a computer application that the user interacts with and is meant to create a user-friendly environment so the application is simpler to use.

- ❖ **Application programming interface (API)** → A mechanism that allows for two pieces of software to communicate and interact with each other. The client (application user) sends a request to a server and receives a response via an API.

2. General Description

Our PWA aims to allow users to identify and record songs that they are unfamiliar with and to enable them to broaden their music taste and knowledge in a quick and easy way.

2.1 System Functions

1. Create Account / Login

In order for the user of the service to be able to save the information of what songs they record, they will need to have an account so that their history can be pushed to a cloud database. The app will have the capability of remembering a user's login information to make the process of logging in and recording a new song as quick as possible.

2. Search Song

The page of the PWA that allows users to record the song. The GUI for this will be very simple and straightforward as to make the process of recording the desired song as time efficient as possible.

3. View recorded songs

This page will display the history of the user's previous searches, as well as a timestamp of when it was recorded and the location of where it was recorded. This will make it easier for the user to find the song they want among those that they have already recorded. Once they find the desired song, they can click into it and see all of this information, as well as more music by the same artist and other music recommendations.

2.2 User Characteristics and Objectives

The goal of this project is to create a location-based music recognition service that is reliable, quick and easy to use. Given the purpose of this project, making the app's GUI simple to minimise time consumption is crucial. There's no point creating an app that involves a series of time consuming steps as it would be useless in fulfilling the user's immediate needs. Therefore, a simple, user-friendly and straightforward process will be implemented to avoid this.

The app is also meant to be able to help the user in broadening their music taste and knowledge. This will be done by displaying additional information on the artist who's song was identified like their most popular songs.

What makes this web app stand out from other similar ones like it is that each song identified by it is accompanied by a time-stamp of when and where the song was recorded. The purpose of this is to enable users to correctly place each song in their

search history with the time and place that they identified it as to not get mixed up with which song is which.

2.3 Operational Scenarios

Scenario 1: *User login*

1. The user opens the PWA in their preferred browser.
2. The GUI clearly displays a login button and a create account button
3. The user either does a quick login or creates their account. (login information includes name and email address).
4. If user has an existing account already, their login information can be saved for a quick login and they are ready to start recording the song
5. If the user does not have an account, they click create account and input their details. After this they will be able to use the PWA

Scenario 2: *User identifies unknown song*

1. Once logged in, the user clicks the record button and the app will begin recording the inputted audio.
2. Once the app successfully recognises what song is playing it will display the name of the song as well as the artist. If the user wants more information they can click into the artist and be able to see the rest of their discography.
3. The information of this song identification is saved in a database

Scenario 3: *User searches for previously identified song*

1. On the home page of the PWA, The user has the option to either record a song or to check their search history for previously identified songs.
2. User clicks into his/her search history.
3. Here, each song that the User had identified previously using the app is displayed in chronological order, each accompanied by a timestamp of when and where the user identified the song.
4. The user can either click into one of these songs or navigate back to the home page using the back button

2.4 Constraints

Time Constraint → As we only have a very finite amount of time to develop and launch this progressive web app, we are in turn limiting ourselves and the project's high success that it could potentially achieve. We do plan however to spot where we can improve upon our project as its development is undertaken.

New technologies constraint → As both members of this project are new to developing progressive web applications, and somewhat new to Javascript, the developmental stage of this project may take longer than we initially anticipated.

3. Functional Requirements

Recognition of Audio snippets, as the correct song and recording of the location in which the recognition takes place.

Description - The microphone on the device is used to record the audio snippet which is then processed on the backend and identifies the song being played. It then grabs the current location of the device and saves it with the song in question.

Criticality - This requirement is the most important within the system as it is the backbone of our project and we require the recording of the audio and the location of the device using our product to be accurate.

Technical Issues - Possible struggles in loud environments and areas that may cause issues recording audio to be tested and accounted for. There should be very few instances of inaccurate location unless gps spoofing is in play as gps location services are very adequate nowadays.

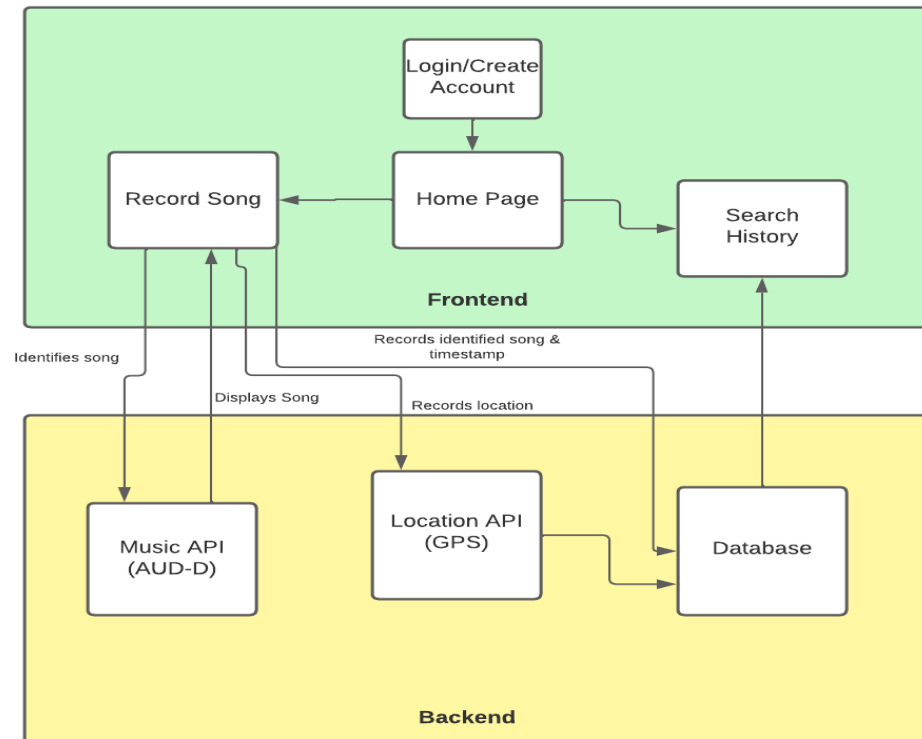
Dependencies with other requirements - Android mobile device with functioning microphone and location abilities to be able to use our product to the fullest extent of it's intended use case.

4. System Architecture

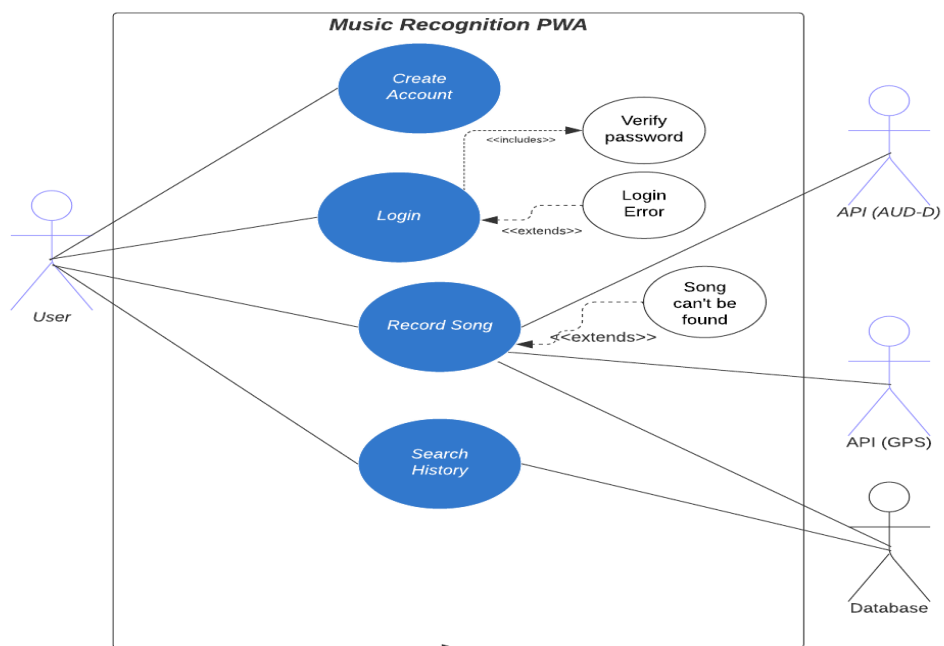
We will be using React to program our front end experience for the end user and we will use python for a backend and SQL for our database. We will use a music recognition api such as AudD or similar to detect the music we are recording through the microphone of the device and this will return the song we are looking for and as this happens we then use location tagging to get the location of the device at that moment and save it alongside the song in our database. For the location we will use Fused Location Provider which is provided for free by Google Play Services Location API's or another option is Android's own Location Manager API. Fused Location Provider is preferred however as it is better optimised in terms of device battery usage.

5. High-level Design

5.1 Object Diagram



5.2 UML Use Case Diagram



6. Preliminary Schedule

We used a Gantt Chart to produce a rough schedule of how we plan to complete the project and step by step parts of the project.

▼ To do

✓ Start the project

✓ Initial Set-up

✓ Writing of frontend and backend code

✓ Meet the supervisor

✓ First testing

✓ Improvement handling

✓ Final testing

✓ Submission

Add task...

▼ Doing

▼ Done

✓ Meet partner

✓ First meeting with supervisor

✓ Proposal Form submission

✓ Functional Spec

✓ Supervisor approval

7. Appendices

React: <https://reactnative.dev/>

Audd: <https://audd.io/>

Deciding how to get Location on Android:

<https://www.geeksforgeeks.org/how-to-get-current-location-in-android/#:~:text=There%20are%20two%20ways%20to,Google%20Play%20Services%20Location%20APIs>