CSD301 - Software Engineering

CloudTunes

Project Report

Prakhar Bhasin - 1810110167
Apoorva Nautiyal - 1810110040
Kirtik Singh - 1810110109

Shiv Nadar University

INDEX

Problem Statement

1. Motivation

Dropping music files in your free Google Drive storage and being able to access it anywhere with unlimited cross-device access. Magical, isn't it?

2. Project Scope

Our project's aim is to deliver a music streaming service that can play multimedia files from commonly used cloud services, like Google Drive. The service we aim to create is planned to enhance the music experience for a wide audience.

3. High-Level Requirements

The essentials of our project cover the following for an optimum music experience for our users:

1. Cross-platform and Cross-device availability of a User's music without the usage of external storage media.
2. Minimizing hassles with a seamless backup of the user's music library without having to spend time switching hard drives, SD cards, etc.
3. Incurring minimal monetary expenses for experiencing their favorite music, without subscribing to expensive streaming applications like Apple Music, Spotify, JioSaavn, Gaana, etc.
4. Secure, one-click integration of a user's personal Google Account with our platform.
5. A fresh visual experience for your favorite music.

These features appeal to the masses and a global, free solution is quintessential in today's age of digitization.

4. Impact and Realisation

As compared to offline storage mediums, Cloud Storage is inexpensive as well as much more efficient in terms of sorting, manageability, access, and backing up. It is also more

secure and convenient compared to other forms of storage. It also adds enhanced security and convenience to more traditional forms of storage.

The service we aim to deliver will take all the music files directly from your cloud storage, and will let you see them all in one place, just like any other music player.

It's thousands of songs, with you, anywhere, anytime.

If not a dream, then what is it to have such fast, reliable, and cost-effective access to music files with a state of the art visual experience.

SOFTWARE REQUIREMENT SPECIFICATION

Table of Contents

1. Introduction

Over the last twenty years, people have witnessed a wave of innovations in how music is performed and consumed. The need for a music service in today's day and age is not only to provide the user's favorite music but to provide easier and widespread access to the music.

The service we aim to deliver will take all the music files directly from your cloud storage, and will let you see them all in one place, just like any other music player.

It's thousands of songs, with you, anywhere, anytime.

1.1 Purpose

Our project aims to deliver a music streaming service that can play multimedia files from commonly used cloud services, like Google Drive. The service we aim to create is planned to enhance the music experience for a wide audience with the help of appealing visuals and ease of access. We also aim to provide a personalized experience to each user with a recommendation system catering to the user's tastes.

1.2 Scope

The essentials of our project cover the following for optimum music experience for our users:

- Cross-platform and Cross-device availability of a User's music without the usage of external storage media.
- Minimizing hassles with a seamless backup of the user's music library without having to spend time switching hard drives, SD cards, etc.
- Incurring minimal monetary expenses for experiencing their favorite music, without subscribing to expensive streaming applications like
- Apple Music, Spotify, JioSaavn, Gaana, etc. Secure, one-click integration of a user's personal Google Account with our platform.
- Personalized recommendations for every user.
- A fresh visual experience for your favorite music.

1.3 Definitions, Acronyms, and Abbreviations

API	Application Programming Interface
CBMS	Cloud Based Music Service
APK	Android Package

1.4 Development Information

- **Prakhar Bhasin**
  - Coder
  - Designer
- **Kirtik Singh**
  - Coder
  - Tester
- **Apoorva Nautiyal**
  - Designer
  - Tester

2. Overall Description

2.1 Product Perspective

Cloud computing is gradually revolutionizing the music industry and the way digital music is being consumed. Our cloud based music streaming aims to reduce the dependency on buying and downloading (digital download) songs over the internet and instead gives the user access to them via the cloud in the form of streaming service. This also brings new services that help users to create playlists and receive recommendations for songs and bands according to their music interests.

Every cloud service has two major components: a front end and a back end. The front end is the medium of interaction for the end-user, the way the service is accessed.

Behind the scenes is the back end of the operation. Here's where the various components aggregate to form a service to provide the required functionalities.

2.2 Product Functions

Secure User Authentication and Login: This function logs the user into the app securely using their Google Account. It is the stepping stone of our app's functionality.

Synchronization with Cloud Service: Audio files stored on the User's cloud services are automatically synced with the app for them to listen to their music anytime, anywhere. This is done through the Google Drive API.

View Recommendations: Each user will be provided with a unique experience of the service through personalized recommendations based on their listening habits.

Provide Feedback: The users must be able to read reviews and provide feedback. This feedback is crucial for the maintainability as well as improvement of the service.

2.3 User Characteristics

The user characteristics are:

- Direct users: These are the users who download the application via the Google Play Store, or the APK file, and install it on their phone. They would be accessing their respective Google Accounts via the inbuilt app login and accessing music files stored in Google Drive.

- Admin: The Admin panel in our CBMS is the Google Cloud Platform panel where quantitative and qualitative details of the API accesses are displayed.

2.4 Front End Description

Our CBMS will provide users with a beautiful user interface based on Google's revolutionary Material Design. Users would be able to manage all their music stored on the cloud. The library would be filtered based on Song Title, Album, and Artists. We will also provide a powerful recommendation system that would allow users to find new music they might like, thereby diversifying their tastes.

2.5 Back End Description

The Back End of our CBMS is essential for the efficient working of all the functions of our app. All the APIs utilized would be clubbed together to function as a whole. The Back end would mainly take care of two aspects - Handling Secure User Logins through Google and Generating Unique Recommendations for every user.

2.6 Software Interfaces

Front End Development - Flutter

Back End Development - FireBase/Python

3. System Features

3.1 Secure Login through Google Cloud Platform

The application is developed, giving utmost priority to user security and data privacy. Thus, the login system implemented is OAuth, which is one of the finest and most user-friendly methods for securely accessing user data. OAuth gives the users complete information about the data they are providing to the application, even having an option to invoke the access whenever they want.

3.2 Unlimited Access to Music on Google Drive Platform

Google Drive is universally accessible by anyone having a Google Account, which in turn comes with Google Drive storage. The software application will provide users to directly access their cloud storage, and the music files stored in it.

3.3 Uninterrupted Music Playback

With the advent of music streaming services, come persistent ads and interruptions. To make sure users are free from all these distractions and annoyances, our CBMS requires no paid subscription and ships with zero ads.

3.4 Recommendation System

The recommendation system is a sub-component of our CBMS. It provides users songs which they may like, based on the songs that they previously listened to. Every logged-in user should have access to the recommender system. The system will go through the songs that the user previously listened to, then according to that information, it should provide songs to the user. An API for the same will be constructed to make HTTP requests to access large databases of music records to provide these recommendations.

3.5 Efficient Usage of Device Storage

High-quality audio files take up a lot of storage on our mobile devices. In many cases, it leads to a shortage of storage space. Our CBMS fetches all of your music libraries from Google Drive which minimizes usage of storage. We only take up a small portion of the device's storage to cache the metadata of files for faster app performance.

4. Performance Requirements

4.1 Hardware Requirements

- The software requires an Android smartphone with a working Internet connection.

4.2 Software Requirements

- An Android App providing users access to our CBMS.
- Users should have an existing Google account.
- Users should have music files stored in their Google Drive storage.

4.3 Design and Implementation Constraints

- The app can only run on the Android Operating System due to the lack of a MacOS based machine for iOS development and testing.
- The app is unable to run on the web due to the instability of the development language on web-based platforms.
- The overall app accesses would be limited due to a lack of funding for the project.

5. Non-Functional Requirements

Our CBMS will follow all the criterions for an industry level software including Functionality, Reliability, Usability, Efficiency, Maintainability and Portability in order to provide users with a high quality experience.

5.1 Software Quality Attributes

Availability: The Cloud service account of the User must be available at all times.

Correctness: The user should log in with the correct credentials.

Usability: The app is easy to use and the UI is designed to facilitate the user's experience. The application is designed to keep the end-user in mind so that it is intuitive to use and easy to learn.

5.2 Security Attributes

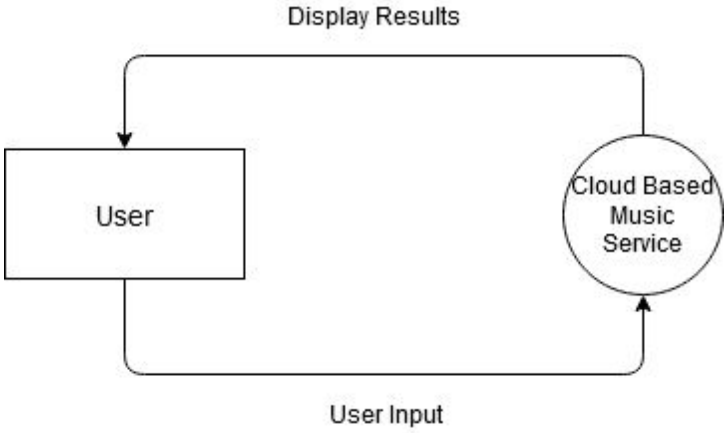
The application itself as well as the Back End platform doesn't store any of the User's data or login credentials. Processing of User data is done via secure Google APIs and industry level techniques of oAuth are used for User login.

6. Conclusion

This Software Requirement Specification document is prepared to give requirement details of the project, "**CloudTunes**". Firstly, definitions of the problem and the general description of the service are given, along with details of the development team. System Features are thoroughly stated in the succeeding section. Then, all the functional, non-functional, and interface requirements are stated in a detailed manner. This document will help constitute a basis for the design and development of the service to be developed.

DATA FLOW DIAGRAMS

Level 0 Diagram (Context Level Diagram)



Level 1 Diagram