Software Engineering: SoundBook

# Problem Description

At *Generic Music Streaming Company (GMSC)*, user downloads are pivotal for the growth of the company. However, finding ways to make users want one’s application instead of the competition’s is often difficult without using *monetary incentives*. This is often a task for the marketing, financial, and software developer teams. The reason why this is a difficult task is that it is the age of social media and people are quick to dismiss trends without *constant reinforcement*. This has a big impact on the finances of a company as failed marketing campaigns can cost the company thousands, if not millions if not implemented efficiently.

## Constraints of the Problem

### Internal Constraints

Internal constraints include financial constraints, informational requirements, and human resources. These are constraints because funds for the development of this project need to be allocated, the source and format of the information being presented to the user need to be researched, and the organization will need to find people specialized in social media marketing, which is where HR comes into play.

### External Constraints

External constraints include the customers of the organization, competitors of the organization, and the state of the industry. These are constraints because the customers rule whether GMSC is a success or a flop, meaning the needs of a customer need to be researched. The competitors of the organization need to be researched to gain knowledge about what they are doing to engage users. Finally, which social media is trendy at the moment, who are the competitors the organization needs to be aware of, and the relevant trends are factors that need to be known about the state of the industry.

# Proposed Solution

In this age of social media, user-generated content is the key to marketing any app or service. To gain millions of organic user engagement the organization must focus on encouraging the app’s users to share GMSC. Allowing the users to share content organically is a great strategy to do this. The solution being proposed implements user engagement by creating a trend that will encourage users to share the brand through social media with the use of a web page. The name of this web page will be *SoundBook*, this webpage will collect user data (music trends and use) from the application and present it visually to the user in an *easy-to-digest* way. This will not only cause users to share it through social media organically but also cause users from competitor companies to want to use GMSC’s application with the use of FOMO or *fear of missing out*.

## Scope of the Solution

### Objective

User engagement and downloads to be organically increased as well as users switching from the competitors’ platform to the organization’s platform are the main objectives of this project.

### Main Components of the Solution

As of now, the data scraper needed to implement the information, the building of a webpage capable of presenting this information visually, and finally, a database to store the user's information are the main components.

### Principal Users

The customers that already use the application as well as users of the competitor organizations trying out GMSC’s application will be principal users of this website.

### Organizational Areas Impacted

Marketing making sure everything works well with the user, finance making sure funds are allocated to develop this project, and the software developing team implementing this solution are all organizational areas impacted.

### How will User Participation be Involved?

Logging into the website with GMSC credentials and retrieving the data once processed is how user participation will be involved.

# Technical Feasibility

## Availability of Hardware

Hardware is readily available as the user will only need a mobile device/computer to engage.

## Availability of Software Required

The Software required to implement this solution will be built by the software engineers and the data being collected comes from our music streaming platform.

## Availability of Knowledge and Expertise

People with web development skills in the software development field are common, therefore this will not be an issue. Likewise, this is the case with social media marketers.

## Availability of Other Required Technologies

At this moment, other technologies outside of what the company already has are not being considered for the project.

## Acknowledgments

This paper was written with the aid of Isaac Fernandes, who provided his insight into the software requirements.

# Formal Requirements

## Problem Synopsis

At *Generic Music Streaming Company (GMSC)*, user downloads are pivotal for the company’s growth. However, finding ways to make users choose SoundBook over the competition is difficult without using *monetary incentives*. Achieving organic growth is difficult because it is the age of social media, and people quickly dismiss trends without *constant reinforcement*. This has a significant impact on the finances of a company, as failed marketing campaigns can cost the company thousands, if not millions.

## Intended Audience

This document is intended for developers, testers, and project managers as a guide for the product. Additionally, project stakeholders in other departments may find this document helpful and informative. It is highly recommended to read the paper thoroughly to get an overview of the product.

## Intended Use

The purpose of this document is to define the end-user interactions that the program should support and all functional and non-functional application requirements. Once completed, this document will specify what must be done for the application to be completed.

## Scope

The developed product will allow users to share insights regarding their listening habits, organically expanding the company’s brand reach. The proposed solution implements user engagement by creating a trend that will encourage users to share the brand through social media with a web page. The name of this web page will be *SoundBook*; this webpage will collect user data (music trends and use) from the application and present it visually to the user in an *easy-to-digest* way. This will not only cause users to share it through social media organically but also cause users from competitor companies to want to use GMSC’s application with FOMO or *fear of missing out*.

# System Overview

## ER Diagram

Diagram

Description automatically generated

Figure 5.1: ER Diagram

This diagram visualizes the database that will be used in the scope of this project.

## Wireframe

Graphical user interface, text, application

Description automatically generated

Figure 5.2: Front page of the website

This wireframe visualizes the desired design for the front page of the website. This design may be changed considering the general idea proposed in figure 2.2.

Graphical user interface, text

Description automatically generated with medium confidence

Figure 5.3: Application wireframe

This wireframe visualizes the desired design for the web app once the user logs in. This design may be changed considering the general idea proposed in figure 2.3.

## User persona

Graphical user interface

Description automatically generated

Figure 5.4: User Persona

The target audience for this project is ‘Alex.’ Alex belongs to Gen Z, a generation that the use of social media has characterized. One of the main contributors to the success of this project is the fear of missing out. FOMO will aid in sharing the platform because it will drive people like Alex to share insights into their music through social media. The target social media platforms are Twitter, Instagram, Snapchat, and TikTok. These platforms are easy to use and are widely used. The format of the SoundBook needs to be easily shared with these target platforms.

## System Components

The user interacts with the web application that uses a data-scraperto gather information and a database to store that information.

The web application will be designed according to figures 2.2 and 2.3. The ‘inside’ of the SoundBook will contain the data gathered from the music streaming platform and showcase the most listened to songs of each genre and a playlist to go with it.

The data-scraper will be needed to gather the information with the permission of the user via the streaming platform’s API. The scraper is required to visualize the data in the Soundbook and analyze patterns and music similarities.

The database will store information regarding songs, albums, artists, playlists, genres, and users. This database must keep the information gathered, aid the process of analyzing and building the SoundBook for future uses, and collect information regarding listening habits. The structure of this database can be seen in figure 2.1.

## System Features and Requirements

### Functional and non-functional requirements

|  |  |  |
| --- | --- | --- |
| **Req. Num.** | **Priority** | **Description** |
| **R001** | **1** | The user login page requires GMSC credentials (username/email and password). |
| **R002** | **1** | Upon signing into a new device, prompt the streaming service account authentication. |
| **R003** | **1** | Web scraper that gathers listening habits and other information on the user. |
| **R004** | **1** | Database that stores data gathered by web scraper. |
| **R005** | **1** | Every song has a share option and can be sent via a link or third-party app. |
| **R006** | **1** | An attractive front page to sign in and go to the application. |
| **R007** | **1** | Ability to search for songs/artists/genres. |
| **R013** | **1** | Intuitive UI centered around the SoundBook, easy navigation without having to press the back button in the browser. |
| **R016** | **1** | Error validation if email input is invalid. |
| **R017** | **1** | Error validation if the username is already in the database. |
| **R018** | **1** | Verification that email exists. |
| **R019** | **1** | Use streaming service API to play songs in SoundBook. |
| **R020** | **1** | Shareable to social media platforms. |
| **R022** | **1** | Security middleware for password encryption. |
| **R008** | **2** | Function to play music selected. |
| **R009** | **2** | Playlist functionality (add, view, share) |
| **R010** | **2** | Artist functionality (favorite, view, share, play top songs) |
| **R011** | **2** | Album functionality (save, view, share) |
| **R012** | **2** | Ability to add other users and view their saved songs. |
| **R014** | **2** | Suggest new music to users based on their listening habits. |
| **R015** | **2** | Allow users to submit their music suggestions. |
| **R021** | **2** | Deploy to a cloud platform such as Heroku. |
| **R023** | **2** | WebApp will be able to be used from any device with internet access. |

Figure 5.5: Functional and non-functional requirements.

### External Interface requirements

|  |  |  |
| --- | --- | --- |
| **Req. Num.** | **Priority** | **Description** |
| **R001** | **1** | An attractive front page to sign in and go to the application. |
| **R002** | **1** | An attractive, shareable design for the soundbook |
| **R003** | **1** | Buttons to play music |
| **R004** | **1** | Playlist naming and making |
| **R005** | **1** | SoundBook will be easy to navigate from any device |
| **R006** | **2** | SoundBook will have options for different ranges of time in which to display the insights. |

Figure 5.6: Functional and non-functional requirements.

Graphical user interface, application

Description automatically generated

Figure 5.7: Interface functionality

A picture containing graphical user interface

Description automatically generated

Figure 5.8: Interface functionality 2

## System constraints

### Assumptions

Assumptions are that users have access to reliable internet service and have an account with either streaming service.

### Dependencies

This project depends on the user having a registered user on either platform and actively using it, and the user has some social media platform.

### Design constraints

The design constraints are: the project must adhere to the terms of service of the streaming platforms being used. The hosting service has file structure limitations, forcing the web application to be designed according to the limitations.

# Software Design

## Process-Oriented Flowchart

Diagram

Description automatically generated

Figure 6.1: Process-Oriented Flowchart

This flowchart showcases the top-level overview of how information flows through the web application. The web application will use a database to store user data and an API respective to each music streaming service to collect the data.

## Information Topology Chart

Table

Description automatically generated

Figure 6.2: ITC

This information topology chart gives information about the database management system and related subsystems. This provides an insight into the inner workings of the web application. It is recommended to read this along with the OSD.

## Data Flow Diagram

Diagram

Description automatically generated

Figure 6.3: DFD Level One

Level one of the DFD illustrates a top-level overview of how information flows through the application, from the subsystems to the processes shown in figure 3.1.

Diagram

Description automatically generated

Figure 6.4: DFD Level Two

Level two of the DFD illustrates how the Data Management and SoundBook subsystems work together to aid processes 1, 2, and 3. Process 4 outputs the data, charts, and songs of the finished SoundBook.

Chart, box and whisker chart

Description automatically generated

Figure 6.5: DFD Level Two: Electric Boogaloo

This second illustration of DFD level two illustrates how the Login Management subsystem processes data through process 3. Process 3 has a significant distinction between admins and regular users, as admins have more privileges than normal users.

## Use Case Diagram

Diagram

Description automatically generated

Figure 6.6: UCD

The USC illustrates the interactions between the users, admins, and system.

## Use Case Scenario

Graphical user interface

Description automatically generated

Figure 6.7: UCS 001 and 002

Graphical user interface

Description automatically generated with low confidence

Figure 6.8: UCS 003 and 004

A screenshot of a computer

Description automatically generated with medium confidence

Figure 6.9: UCS 005 and 006

Graphical user interface, application

Description automatically generated

Figure 6.10: UCS 007 and 008

Graphical user interface, application

Description automatically generated

Figure 6.11: UCS 009 and 010

Table

Description automatically generated

Figure 6.12: UCS 011

The use case scenarios above illustrate the different interactions users, admins, and systems have inside the app. The users/admins can share the soundbook, trigger soundbook creation, login, and authorize API access. The System loads and updates the database, requests, and processes data.

## Object Relationship Diagram

Diagram

Description automatically generated

Figure 6.13: ORD

This ORD gives a top-level overview of the different objects needed for the web application. This ORD shows that the user can have many profiles and soundbooks. Each soundbook may have many models, and the models depend on the API’s data/

## Object Structure Diagram

Graphical user interface, application

Description automatically generated

Figure 6.14: OSD

The OSD shows what methods and variables the objects would have in this program. The user profile would store personal information and be used to gather data from users. The soundbook would keep the name of each soundbook, creation date, who it belongs to, and the number of songs in the ‘book.’ The Soundbook object would also be able to be deleted and shared. The model object would be a constructor for the different charts. The API object would gather information from the API once the user authorizes its use. Additionally, this object would pass the data to the models to create and request authorization in case it is not already approved.

# Decision Tree and Table

## Decision Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Conditions** | **1** | **2** | **3** | **4** | **5** |
| Has linked account | Y | Y | N | N | N |
| Click on Spotify | Y | N | N | Y | N |
| Click on Apple Music | N | Y | Y | N | N |
| Click on other services | N | N | N | N | Y |
| **Actions** |  |  |  |  |  |
| Allow SoundBook creation | X | X |  |  |  |
| Ask to link to Spotify. |  |  |  | **X** |  |
| Ask to link to Apple Music. |  |  | X |  |  |
| Prompt “Sorry for the inconvenience." |  |  |  |  | X |

Figure 7.1: Decision Table

I chose this scenario for this decision table because it fits the table better. Each condition is unique and independent and would be difficult to represent in a decision tree, as the conditions are not shared. The table also helped me realize ambiguity in my logic; I had written "Has account" instead of "Has linked account," and if the user has no account, it would not matter if any of the services were clicked.

## Decision Tree

Diagram

Description automatically generated

Figure 7.2: Decision Tree

I chose to use a decision tree for this scenario because the processes are checked in stages, and the actions are related. This decision tree helps visualize a pivotal part of the SoundBook. Without this process, the users would receive an incomplete SoundBook, which would harm the brand of the application and the purpose of the SoundBook.

# Project Planning

## Tabular list of tasks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Activity** | **Predecessor** | **Description** | **Estimated Time (days)** | **ES** | **EF** |
| A | - | Design Database | 5 | 0 | 5 |
| B | A | Design Website Architecture | 5 | 5 | 10 |
| C | A | Design Visualizer Skeleton | 7 | 5 | 12 |
| D | A | Create artwork for the website | 6 | 5 | 11 |
| E | B,C,D | Implement Website | 10 | 12 | 21 |
| F | E | Implement API | 5 | 21 | 26 |
| G | E | Implement working UI | 10 | 21 | 31 |
| H | G,F | Test Website | 5 | 31 | 36 |
| I | G | Select Hosting Service | 3 | 31 | 34 |
| J | I | Upload Website | 2 | 34 | 36 |
| K | J | Test online functionalities | 2 | 36 | 38 |
| L | K | Deploy | 2 | 38 | 40 |

Figure 8.1: Task List

This is the list of tasks that need to be finished in order for the project to be deemed as complete.

## PERT/CPM

Diagram

Description automatically generated

Figure 8.2: PERT/CPM

This is the PERT chart for the tasks that need to be completed. In this figure it can be seen that the critical path is A, C, E, G, I, J, K, L.

## GANTT

Chart

Description automatically generated

Figure 8.3: GANTT Chart

In figure 3.1 the GANTT chart for the project can be found. If the project were to be started August 8th, the finish date would be October 12th.