**Table of Content**

[**1. Introduction**](#_8kx8m6i6gm6q) **2**

[**2. Project Requirements**](#_i0mfv23vu30t) **2**

[2.1 Functional Requirements](#_ilpx22dzjnad) 2

[**3. System Functionality**](#_9g9hrb2qjr2u) **2**

[3.1 User Authentication & Profile Management](#_amp72p53s2bu) 2

[3.2 Music Search with Filters](#_54exbng9r0va) 3

[3.3 Save for Later Functionality](#_x6ijgskisoaj) 3

[3.4 Administrative Controls](#_ddlkk4ox560e) 3

[**4. Activity Diagram**](#_ym3bo9m2afe5) **4**

[**5. Conclusion**](#_et2jnu6wnbeq) **5**

# **1. Introduction**

The document establishes requirements to develop an interactive web application that uses Django as its framework. The application functions as a music recommender tool with search capabilities and supports different devices ranging from mobile phones to tablets. The application supports efficient searching capabilities for users together with personalized content delivery through secure user interaction features.

# **2. Project Requirements**

## **2.1 Functional Requirements**

* **User Authentication:**

The system must provide all users with reliable access for both entering the system and departing from it.

* **User Profile:**

Every account holder maintains their own adjustable profile which stays accessible from login until logout.

* **Music Search Feature:**

Users can perform robust searches with these three search filters enabled in the system.

* Music genre
* Date range
* Artist
* Album
* Record label
* **Save for Later:**

Music selections previously saved can be accessed by users when they need to recall them.

* **Administrative Controls:**

Within this system structure administrators can create new music information while removing old entries and performing information updates.

# **3. System Functionality**

## **3.1 User Authentication & Profile Management**

* **Login/Logout:**

The system requires users to provide authentication credentials to access the platform. The system presents the user profile after completing the authentication process.

* **Profile Management:**

Users can both edit their personal metadata and manage their saved music collection to maintain current profile access.

## **3.2 Music Search with Filters**

* **Search Module:**

Users achieve music search results through query entry and filter application which includes genre, date range, artist, album and record label. The search result precision increases through this method.

## **3.3 Save for Later Functionality**

* **Saving Music:**

Users can save music through the search results by applying an "add to later" option. Users can easily find saved music because this feature promotes better engagement with the platform.

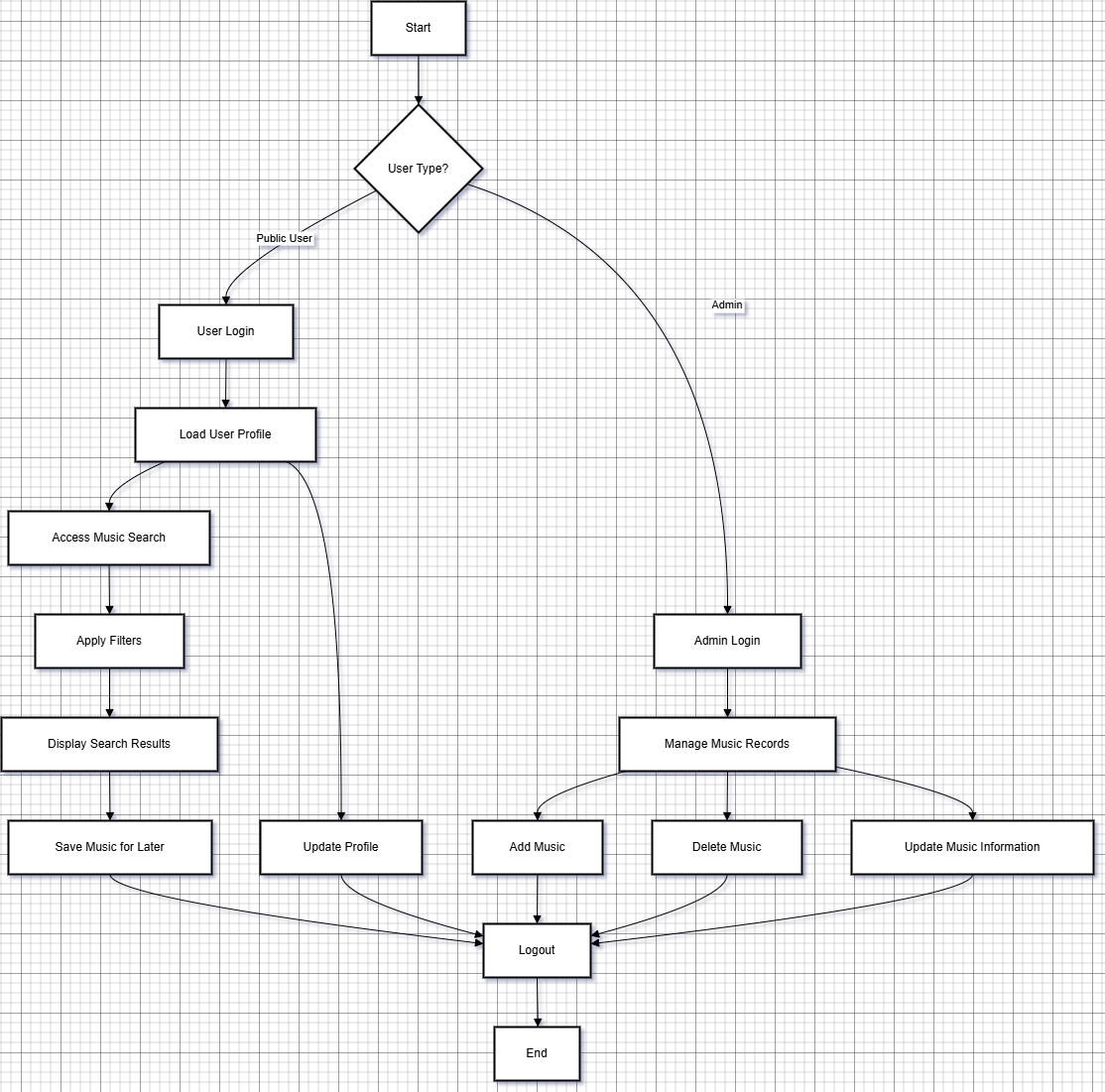
## **3.4 Administrative Controls**

* **Music Management:**

The functions which allow admin users to manage the music database exist only for their exclusive use. This includes:

* Adding new music entries.
* Deleting obsolete records.
* People who manage the database can update artist name records along with album titles as well as record label information.

# **4. Activity Diagram**

****

This diagram demonstrates:

* Public Users should perform the sequence of steps that includes logging in followed by accessing their profile to update information while they can search for music using filters before selecting music and logging out.
* Administrators login to access a function for music record administration which includes addition and deletion and update operations until they exit.

# **5. Conclusion**

The document specifies all functional system features needed to build the interactive music recommendation application along with its essential requirements. The application achieves a dynamic user-friendly experience because it includes safe user authentication with profile customization features along with advanced search tools and detailed administrative monitoring systems. An activity diagram included in the document effectively illustrates the operational process for public users as well as administrative personnel which leads to a well-built system structure.