# Section 1 - Project Description

## 1.1 Project

Cadence

## 1.2 Description

Cadence is a dating platform focused on matching people based on their musical preferences and hobbies. We’re hoping that by providing users with a platform where they can search for potential partners based on a common interest we can match users with other users that are in their area that they have high compatibility with.

## 1.3 Revision History

| **Date** | **Comment** | **Author** |
| --- | --- | --- |
| 10/27/23 | UML Diagrams added to Section 5 and Section 6 | Daniel Nunez |
| 10/31/23 | Use Case Diagram updated with feedback from presentation | Daniel Nunez |
|  |  |  |
|  |  |  |
|  |  |  |

**Contents**

[Section 1 - Project Description](#_1fob9te)

[1.1 Project](#_3znysh7)

[1.2 Description](#_2et92p0)

[1.3 Revision History](#_tyjcwt)

[Section 2 - Overview](#_1t3h5sf)

[2.1 Purpose](#_4d34og8)

[2.2 Scope](#_2s8eyo1)

[2.3 Requirements](#_17dp8vu)

[2.3.1 Estimates](#_3rdcrjn)

[2.3.2 Traceability Matrix](#_26in1rg)

[Section 3 - System Architecture](#_lnxbz9)

[Section 4 - Data Dictionary](#_35nkun2)

[Section 5 - Software Domain Design](#_1ksv4uv)

[5.1 Software Application Domain Chart](#_44sinio)

[5.2 Software Application Domain](#_2jxsxqh)

[5.2.1 Domain X](#_147n2zr)

[5.2.1.1 Component Y of Domain X](#_3o7alnk)

[5.2.1.1.1 Task Z of Component Y1 of Domain X](#_23ckvvd)

[Section 6 – Data Design](#_z337ya)

[6.1 Persistent/Static Data](#_3j2qqm3)

[6.1.1 Dataset](#_1y810tw)

[6.1.2 Static Data](#_4i7ojhp)

[6.1.3 Persisted data](#_2xcytpi)

[6.2 Transient/Dynamic Data](#_1ci93xb)

[6.3 External Interface Data](#_3whwml4)

[6.4 Transformation of Data](#_ihv636)

[Section 7 - User Interface Design](#_2bn6wsx)

[7.1 User Interface Design Overview](#_qsh70q)

[7.2 User Interface Navigation Flow](#_3as4poj)

[7.3 Use Cases / User Function Description](#_1pxezwc)

[Section 8 - Other Interfaces](#_49x2ik5)

[8.1 Interface X](#_2p2csry)

[Section 9 - Extra Design Features / Outstanding Issues](#_32hioqz)

[Section 10 – References](#_1hmsyys)

[Section 11 – Glossary](#_41mghml)

# Section 2 - Overview

## 2.1 Purpose

The purpose of Cadence is to create a platform where music lovers can seek out other like-minded individuals with a passion for music. The goal of seeking out other like-minded individuals is much like other dating platforms where they can meet people and make a deep connection with each other. Cadence aims to make the process of finding compatible partners easier by matching users based on a shared interest, chiefly amongst these is music.

## 2.2 Scope

Cadence’s scope includes a functioning platform hosted on a website where users can register new accounts and log in to existing accounts which allows them to update their personal data and preferences. Using user’s preferences Cadence will search for other users with similar preferences and match them based on their similarity.

## 2.3 Requirements

The team is using estimates regarding some of our requirements for the project. Chiefly among these is man hours, SRS Requirements, and money. Since Cadence isn’t being funded from within, Cadence’s cost will only be based on total man hours working on the project.

### 2.3.1 Estimates

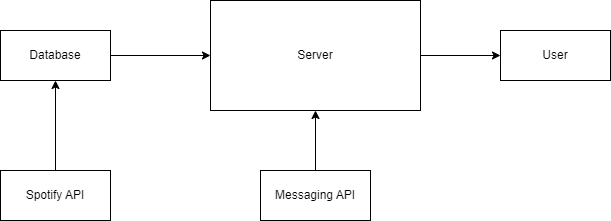
| **#** | **Description** | **Hrs. Est.** |
| --- | --- | --- |
| 1 | Website Design | 10 |
| 2 | Website Setup and Integration | 30 |
| 3 | Database | 10 |
| 4 | Messaging System | 15 |
| 5 | Matchmaking Algorithm | 20 |
| 6 | Spotify Implementation | 25 |
| 7 | Server Setup | 20 |
| 8 | Testing | 10 |
| 9 | Documentation | 10 |
|  | **TOTAL**: | 150 |

2.3.2 Traceability Matrix

| **SRS Requirement** | **SDD Module** |
| --- | --- |
| Req 2 | 7.1 |
| Req 3.1 | 7.2 |
| Req 3.2 | 5.1 |
| Req 3.3 | 8.1 |
| Req 3.4 | 5.2 |

# Section 3 - System Architecture

The system architecture of Cadence will run primarily in Javascript using Node.JS. The frontend of the platform will be using HTML and CSS for the formatting and some React to link it with the backend. Furthermore, MongoDB will be used for the database that holds user information and other necessary information including information sent from Spotify . Cadence will also utilize the Spotify API to obtain data from users that elect to link their Spotify accounts to ensure greater accuracy for matchmaking.



# Section 4 - Data Dictionary

| **Users Table** |
| --- |

| **Field** | **Notes** | **Type** |
| --- | --- | --- |
| ID | Unique identifier for User | DECIMAL |
| NAME | The name of the User | VARCHAR |
| EMAIL | E-mail of the User used for verification | VARCHAR |
| LOCATION | The location of the User, used for matchmaking | VARCHAR |
| AGE | The age of the User | DECIMAL |
| GENDER | The gender of the User | VARCHAR |
| SEXUALITY | The sexuality of the User, used for matchmaking | VARCHAR |
| GENRE | The User’s favorite genre of music | VARCHAR |
| ARTIST | The User’s favorite musical artist | VARCHAR |
| MATCH ID | The ID of Users the User has matched with | DECIMAL |

| **Genre Table** |
| --- |

| **Field** | **Notes** | **Type** |
| --- | --- | --- |
| NAME | The name of the genre | VARCHAR |
| PARENT | The name of the genre this genre is a sub-genre of (if any) | VARCHAR |

| **Artist Table** |
| --- |

| **Field** | **Notes** | **Type** |
| --- | --- | --- |
| NAME | The name of the artist | VARCHAR |
| GENRE | The name of the genre t | VARCHAR |

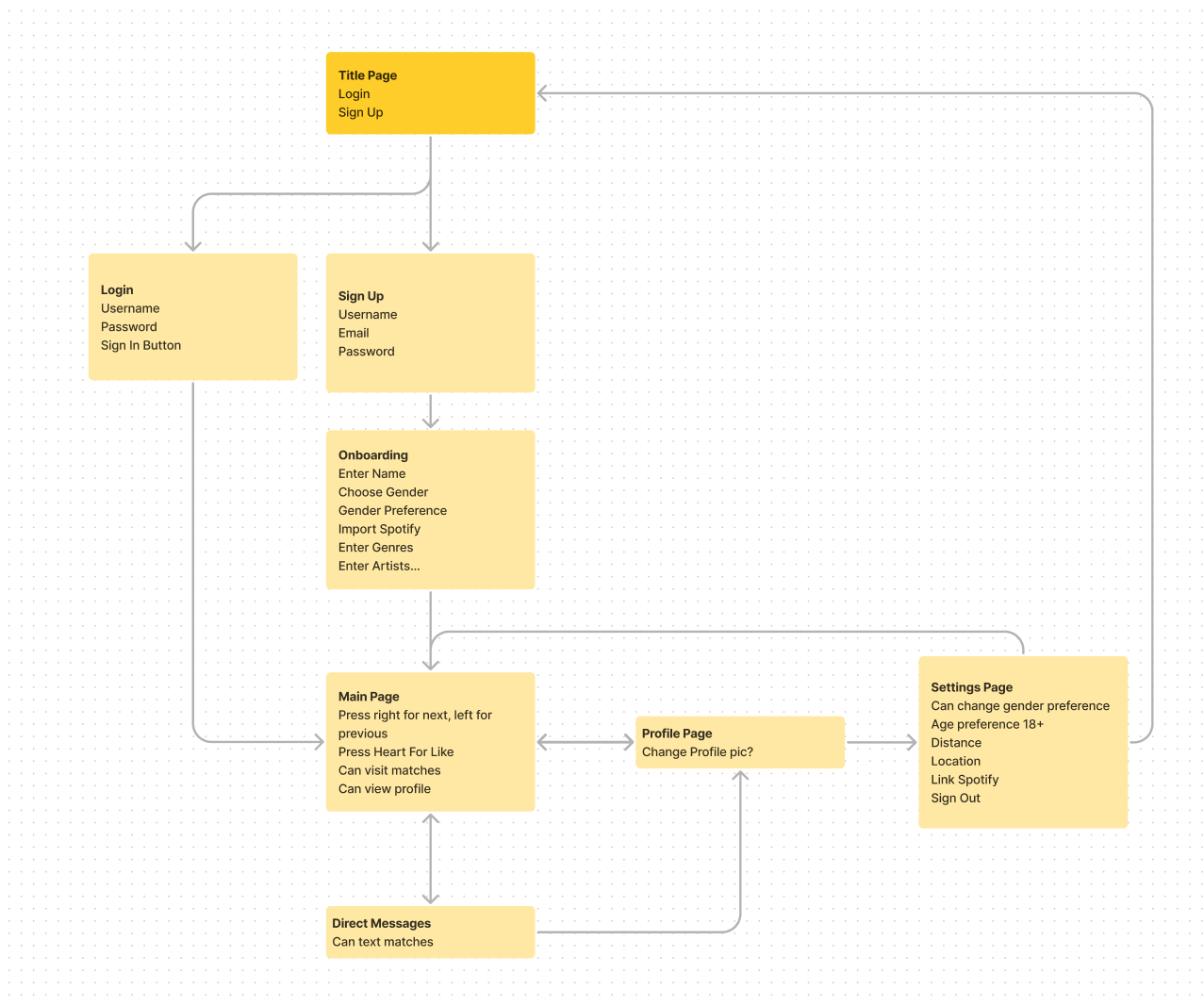
# Section 5 - Software Domain Design

## 5.1 Software Application Domain Chart

Cadence’s software application domain consists of a website that will communicate with a database containing user information. The user information will be used by a matchmaking algorithm to determine possible matches with other users. Users will also be connected to a messaging system so that they can communicate with each other after matching.

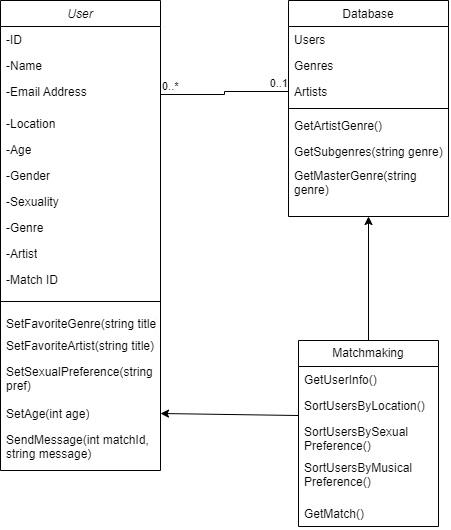
## 5.2 Software Application Domain

The website will allow users to navigate between pages to access their profile, direct messages, and the matchmaking service. The database the website is connected to contains information on users sexual and musical preferences as well as their location to determine potential matches with other users based on their information.



# Section 6 – Data Design

Most of the data stored in the database is related to their user and is thus stored in the User table. Other data that is used includes a hierarchy for genres that groups sub-genres as children of the genre they are derived from. Additionally, a table is used to group a musical artist with the genre they perform in to allow for matchmaking to weigh artists in the same genre.



## 6.1 Persistent/Static Data

The persistent data in our data model includes the user’s ID stored in the user table and the data in the genre and artist table. The genre table is used to maintain a hierarchy between genres and sub-genres.

### 6.1.1 Dataset

The genre and artists tables both consist of persistent and static data and are used with each other to determine how closely related different genres and artists are and by extension is used with the user table to determine how compatible different users are based on their genre and artist preferences.

### 6.1.2 Static Data

The genre and artist tables contain static data because a sub-genre cannot change which genre it is derived from and artists are unlikely to perform in a different genre. Additionally, The ID of the user in the user table is not meant to be changed as it is required to differentiate individual users when every other field in the table can be changed or can overlap with other users.

### 6.1.3 Persisted data

All the data listed in this section is persisted data as all the data will need to persist between any periods when the platform goes down.

## 6.2 Transient/Dynamic Data

Most of the user’s data is dynamic data as we want the user to be able to update their information and preferences as they need to.

### 6.2.1 Dataset

The user table consists almost entirely of dynamic data and is used with itself to match users based on their information and preferences.

### 6.2.2 Dynamic Data

In the user table, many of the data fields are dynamic so that the user can change their preferences and information whenever they need to.

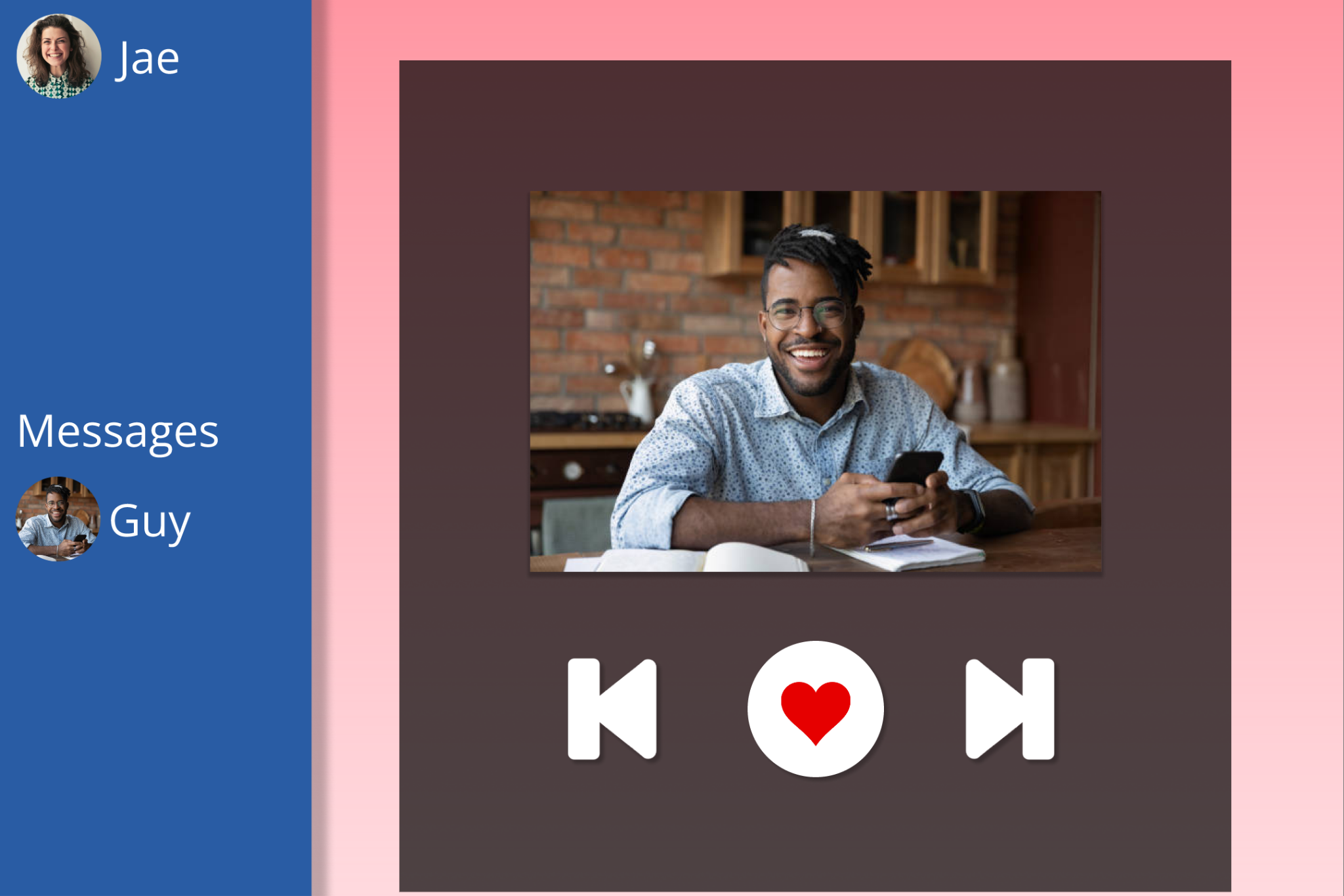
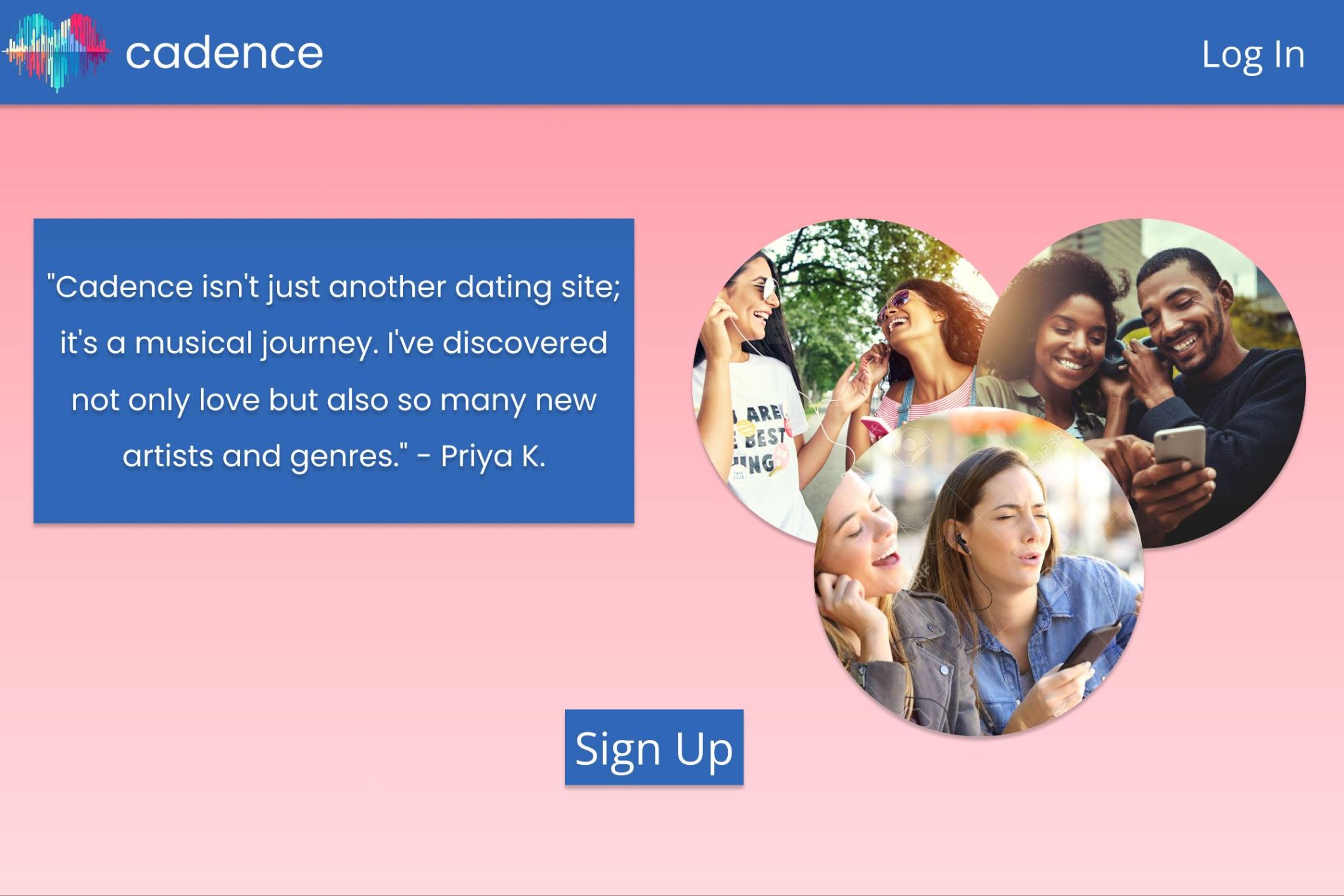
## 6.3 External Interface Data

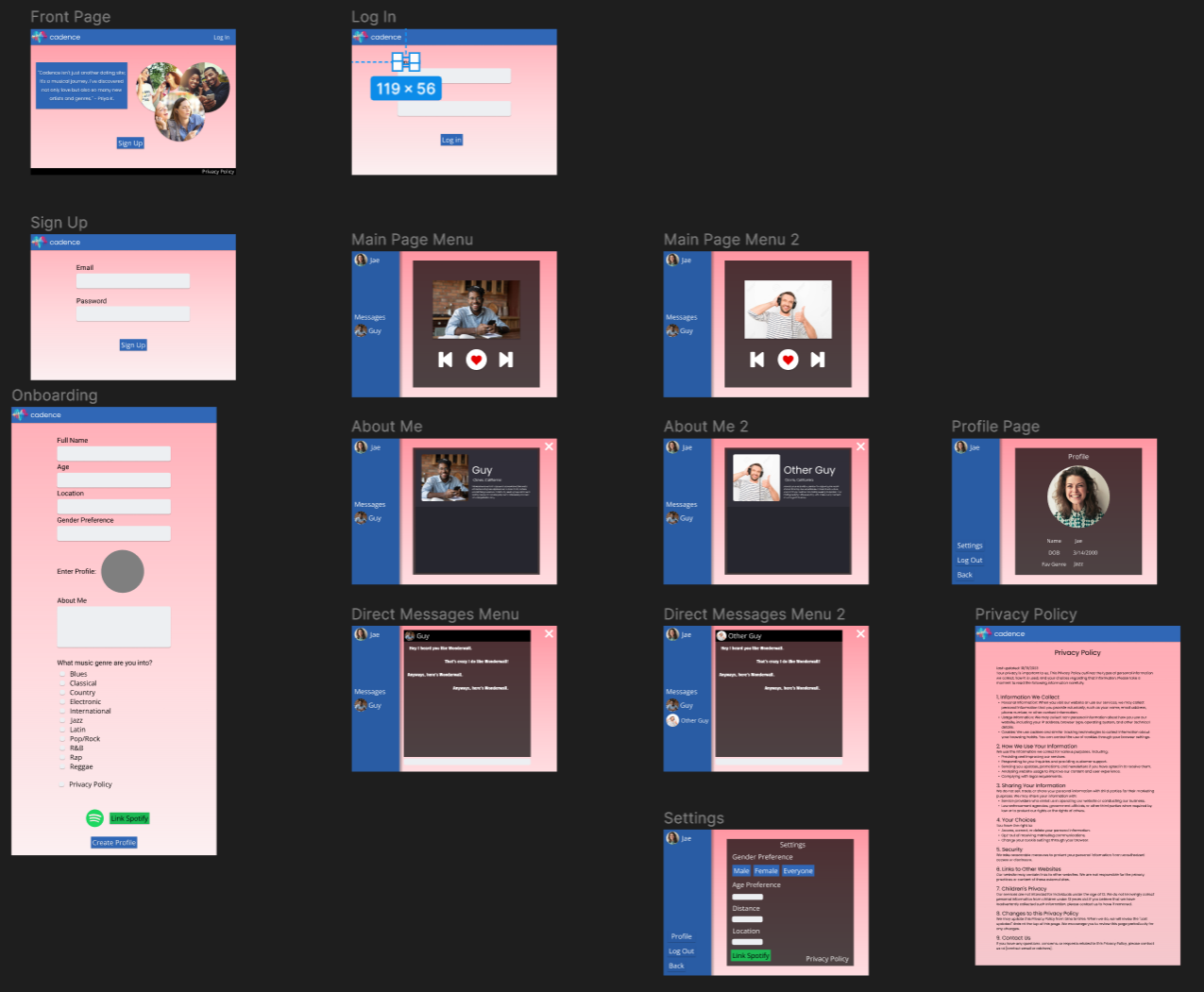
Data is pulled from the Spotify API if users choose to link their Spotify account to update the genre and artist fields in the user table. Additionally, data may be pulled from the Spotify API to fill in the genre field of the artist table.

# Section 7 - User Interface Design

## 7.1 User Interface Design Overview

The user interface is based on other dating apps with a simple design and flow to get you straight into matching with others. The main page is based on music apps where you can skip to the next person, go back to the last person (premium feature), and like the person who you want to try to match with.

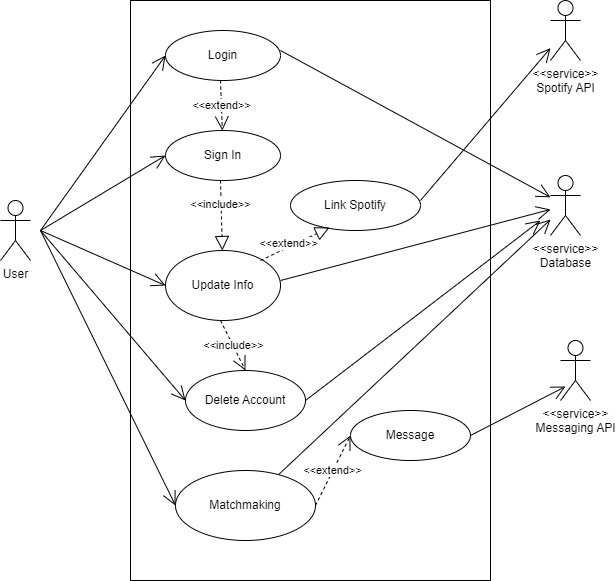




## 7.2 User Interface Navigation Flow

Users will first be directed to the welcome page which links to the log in and sign up pages. Both pages will have the user enter their credentials to access or create an account to access the main page. The sign up page navigates to the onboarding page where users will enter their information and preferences before taking them to the main page. The main page features the matchmaking system which will present other users profiles as well as a sidebar that shows your profile and matches. From the main page, you can click on the profile of the user the matchmaking system to view their about page or match with them to open a direct message with them. Users can also click on the profiles of the users they have previously matched with to open their direct messages with that user. Users can also click on their profile on the main page to navigate to their profile page. The profile page shows the users info and changes the sidebar to have buttons to navigate to settings, log out of their account, or to return to the main page. The settings page will allow users to change their information and allows them to link their account to Spotify. If the user logs out of their account, they will be returned to the welcome page.

## 7.3 Use Cases / User Function Description



# Section 8 - Other Interfaces

The Spotify API is an external interface that Cadence utilizes to update the users data and genre data.

## 8.1 Interface X

When the user links their Spotify account, Cadence will use the Spotify API to pull the users information on their most listened to artist and genre.

# Section 9 - Extra Design Features / Outstanding Issues

We tested Cadence's account creation, employing MongoDB and Postman to ensure the validity of three key criteria: a unique username, a unique email, and a secure password. If a user attempts to register with a username or email already in use, they receive an appropriate error message. Subsequently, during login, users need only provide their username and password, with any authentication errors prompting error messages. Our next phase involves comprehensive testing, integrating the backend and frontend to assess the entire website's functionality, encompassing messaging, matching, and Spotify integration. These additional features will be thoroughly tested to ensure their reliability and seamless performance within the Cadence platform.