CSC-433 - Software Engineering

Utica College: Spring 2020

Professor: Dr.Kim

Deliverable #3: Functional/Design Specifications

Due Date: Thursday, April 9th 2020

Team Possible

Project Name: Bump-it

Team Members:

Project Leader: Todd Hrim

Project Manager: Alberto Garcia

Lowest-Level Programmer: Franklin Nuth

Table of Contents

System Objectives

Splash Page

The Splash Page will be the first page the user will see once they open up the web application for the first time. This page will consist of the company's logo, mission statement and greeting message. Once the user is ready to use the application they can easily continue by clicking the “Get Started” button to continue. With a simplistic nature the user will easily be able to navigate to the Login and page and so on. All pages following will follow this design layout consistently.

Sign in Page

The Sign In page will be the first page the user will encounter. If the user doesn't have an account yet they will be guided to the Sign Up page. The Sign In page will have two fields; E-mail Address and Password, which will be input fields. The user will be able to use their E-mail address along with their Password for authentication. The user's E-mail address will be their username they will be able to Sign In and get access to the application with. If the user enters the wrong username or password they will be prompted with an error message until they enter their correct credentials.

Sign up Page

If the user is a first time user they will have to Sign Up in order to get access to the application. The Sign Up page will have 3 input fields; Name, E-mail and Password. If the database recognizes the E-mail address has already been registered into the database the user will be promoted with an informative message clarifying the E-mail has already been registered. Otherwise, if the user enters all of their information correctly they will be granted an account to use the application and are able Sign In.

Genre Selection Page

The Genre Selection page will allow the user to browse through a variety of different music. The user will be able to select from 4 different Genres of music; Disco, Hip-hop, Pop, and Country.There will be 3 actions on the Genre page that will allow the user to navigate and trigger an action for the use of the application.The first two actions are located on the page via navigation bar; Forward, Backwards. The third action on the page allows the user to add a song to the Music Queue page. This third action is an iterative action, assigned to each song. This page will lead the user to continue to the Music Queue page.

Music Queue Page

The Main Queue Page is what the user sees once they’ve added their song selections. The user will be able to see all of the songs that have been added by each user in the room and vote accordingly. There will be 4 actions that will trigger events; Forward, Backwards, Like, Dislike. The Like and Dislike actions will allow the users in the room to vote for which songs they would like to hear play. The songs with the most

likes will surface to the top in order of Likes and then get played. The song objects on the page will feature song information such as; Artist, Song Title, Likes, Dislikes and Album Art. Once the current track that is getting played finishes, the next highest liked song will get played in the queue. This is a music playlist that is dynamic and real time, therefore users will all be updated concurrently.

Music Player Page

A music player will be integrated into the application. The music player will allow media files such as high quality .mp3's to be played. Once the song with the highest Likes reaches the top of the queue the song will be played using this media player.

Recommendations Ad Design

The user will get Recommendations from the application. The user will be prompted by a banner style ad that will recommend the user to listen to a song based upon the users liked music. These ads are triggered within 30 minutes of app use. Thus, this allows the app to collect enough information about the users most liked music. It is assumed that the user will like at least 5-10 songs within this time range, as well as listen to a total of 5 songs.

Major Requirements

Front End Development:

- HTML
- CSS
- Bootstrap
- Vue JS

Back End Development:

- Python
- User Database
- Music Database
- Redis
- Daphne

Server Major Requirements:

- Django REST Framework

Design Constraints and Limitations

One constraint or limitation found in the web application is that the web application will only be able to run music data for the application, not actual live streamed music. This is because implementing music data is more feasible than live streaming music with the amount of time left for the project. The team will still aim to develop the rest of the application as intended for deployment.

Function Descriptions

Use Case Templates

Use Case #1: Logging In

When logging in, the user should only be able to see two fields: Username and Password. Once the user inputs their information correctly, they should seamlessly be directed to the main portion of the web application. They can then move on to join or create rooms as they wish.

Use Case #2: Signing In

When signing in, the user should be able to see four fields: First Name, Last Name, Username, and Password. The Password field will also be accompanied by the Confirm Password field, which allows the user to input their password twice. The user will be slightly inconvenienced by the Confirm Password field, but it helps protect the database from having accidental information, and the user cannot make a mistake when they input it into the database. When the user fails to correctly replicate their password or have information from another user, they will be denied and prompted to try again on the same screen. Once they are successful, they will be redirected to the main portion of the application.

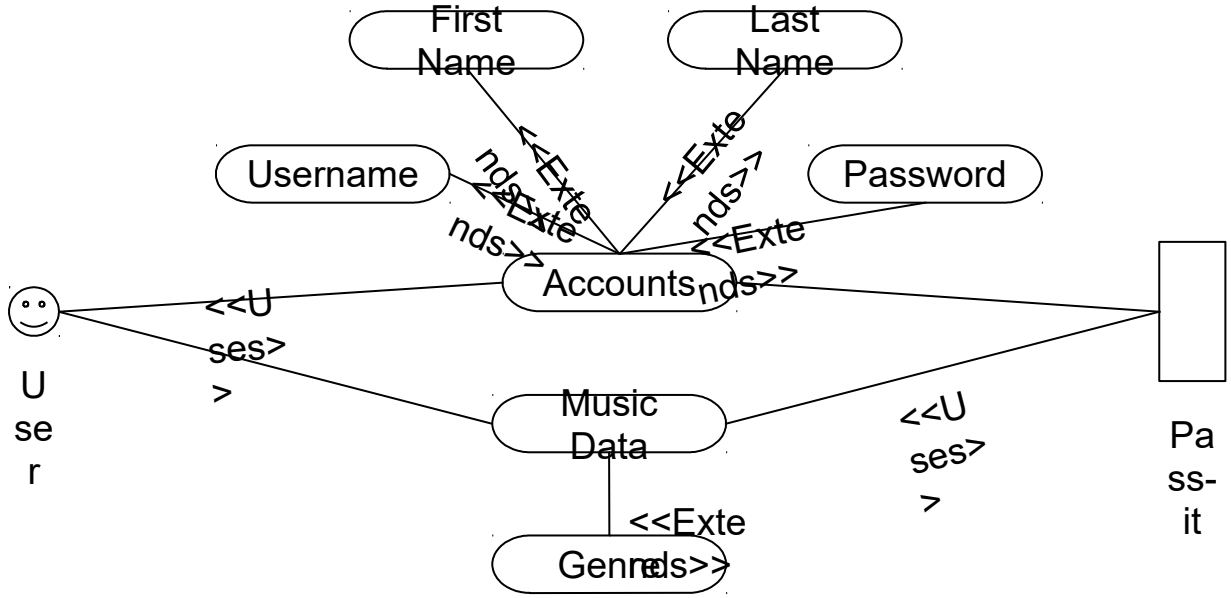
Use Case #3: Voting

When a user joins a room, they should be able to see a playlist of songs in a queue. Next to each song in the playlist, the user should be able to see two numbers divided by a front slash. The number on the left represents likes, and the number on the right represents dislikes. The songs in the playlist will be adjusted according to how the songs are liked and disliked. As soon as voting is over, the playlist will be sorted from most popular to least popular in descending order. All the users will then automatically be moved to the Music Data Player page.

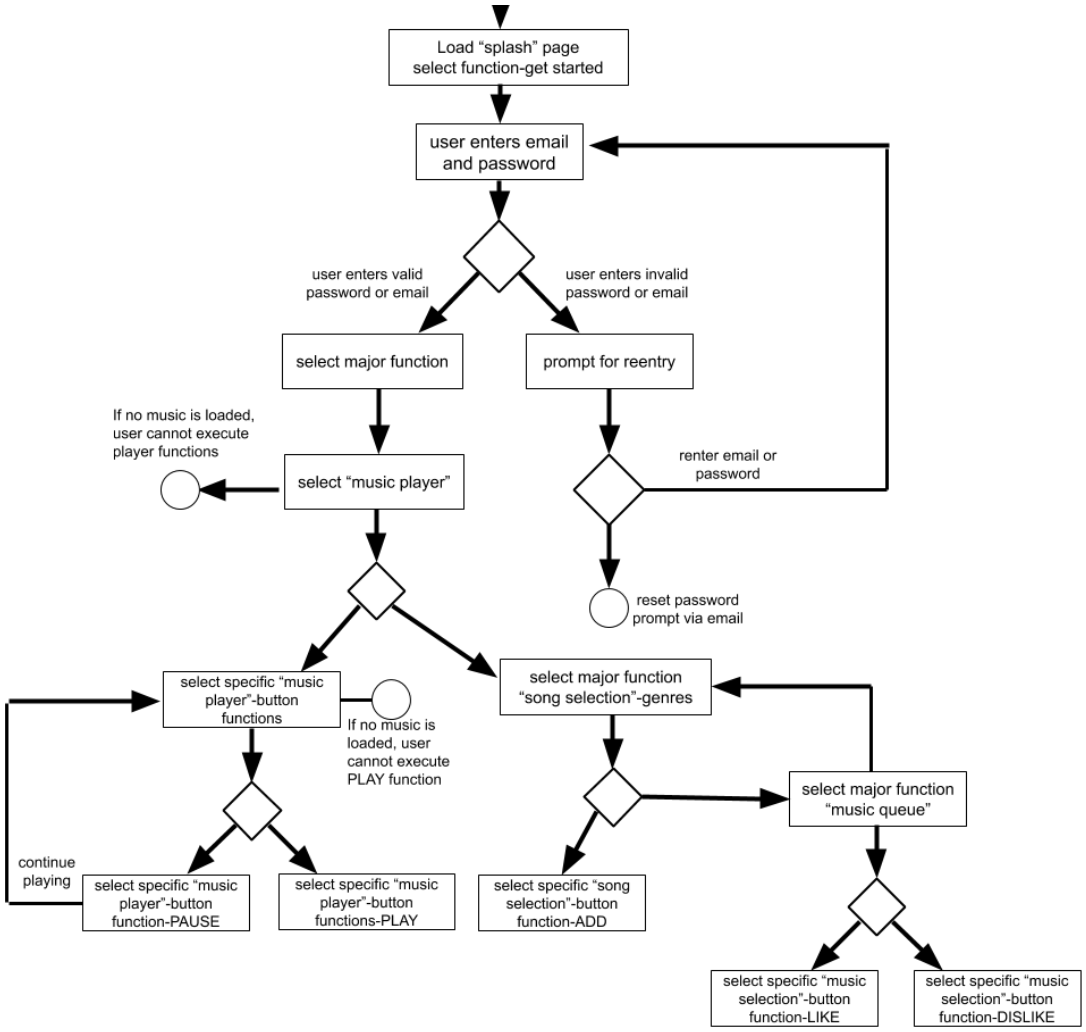
Use Case #4: Music Data Player

Once the users are moved into the Music Data Player, they are presented with a music player that should be reminiscent of common music players found on phones and computers by default. The users are then able to listen to the curated playlist. The user can then exit the playlist at any time or go back to the voting page to participate in another voting session.

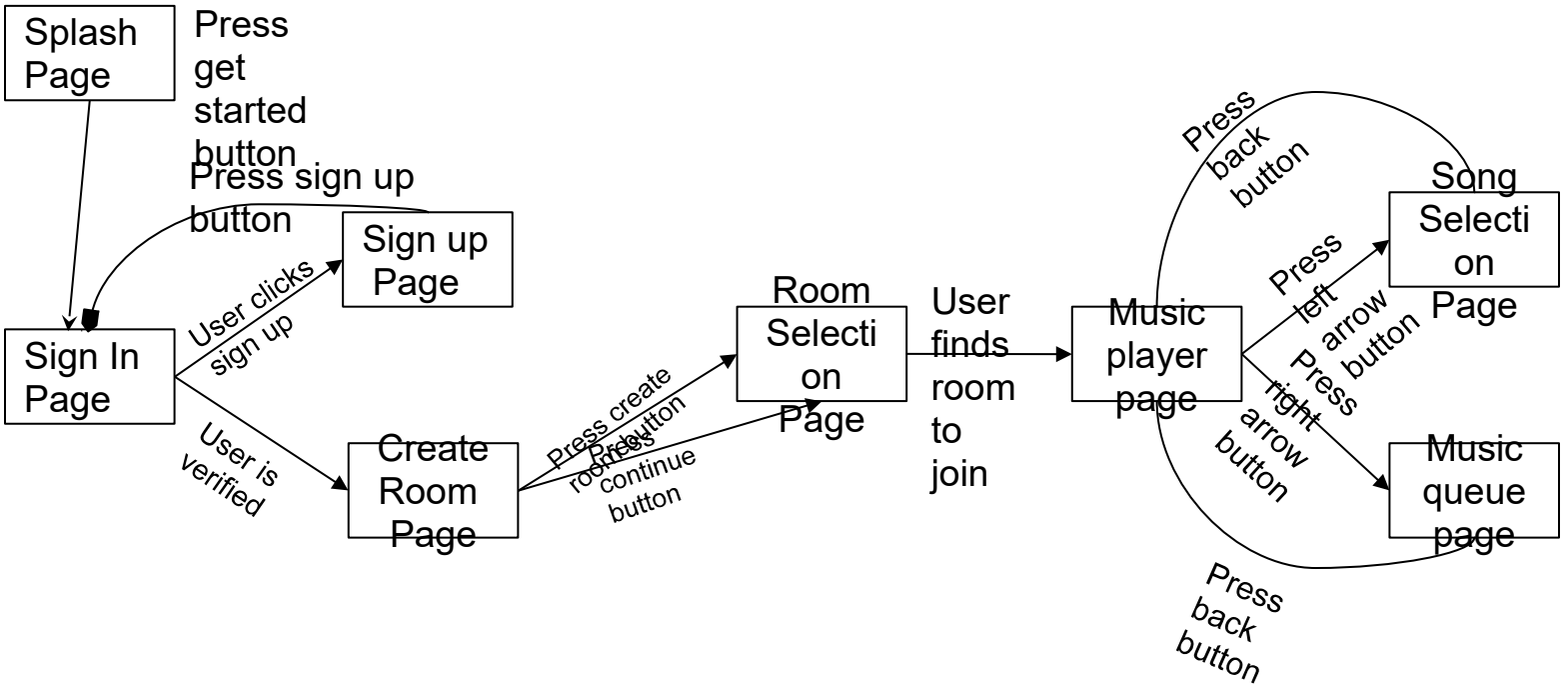
Use Case Diagram



Activity Diagram



State Diagram



States

- Splash page
- Sign in page
- Sign up page
- Create room page
- Room selection page
- Music player page
- Song selection page

(Splash page) ----press get started→ (sign in page)

(Sign In page) ----press sign up button---> (sign up page)
-----press log in button-----> (Create room page)

(Sign up page) ----press sign up button---> (sign in page)

(Create room page) -----create room button-----> (Room selection page)
------Continue Button-----> (Room selection page)

(Room selection page) ----- user presses room to join -----> (Music player)

(Music player) ----press left arrow button----> (song selection page)
----press right arrow button----> (Music queue page)

(Song Selection page) -----User adds song---> (Music player)

(Music queue page) ----Press back button---> (Music player)

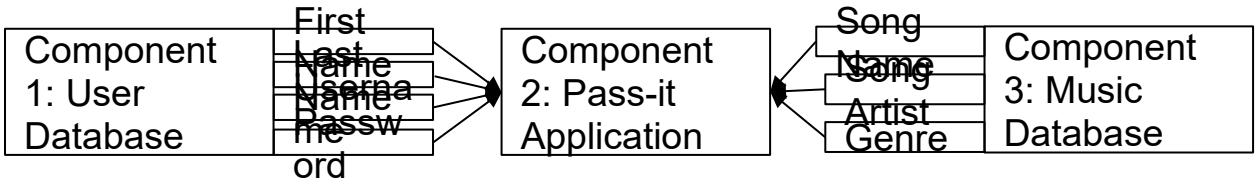
Data Design

[Insert ER Diagram here.]

User Interface Design

[Splash Page UI]
[Sign in Page UI]
[Sign up Page UI]
[Genre Selection UI]
[Music Queue UI]
[Music Player UI]

Component-Level Design



Data Tree Diagram

Comment Level Design

Data Tree Diagram