Initial Architecture Document

Team Number: 24

Team Members

1. Andrew Hughes
2. Eric Zhuo
3. Bayley Duong
4. Preston Chanta
5. Will Hecht

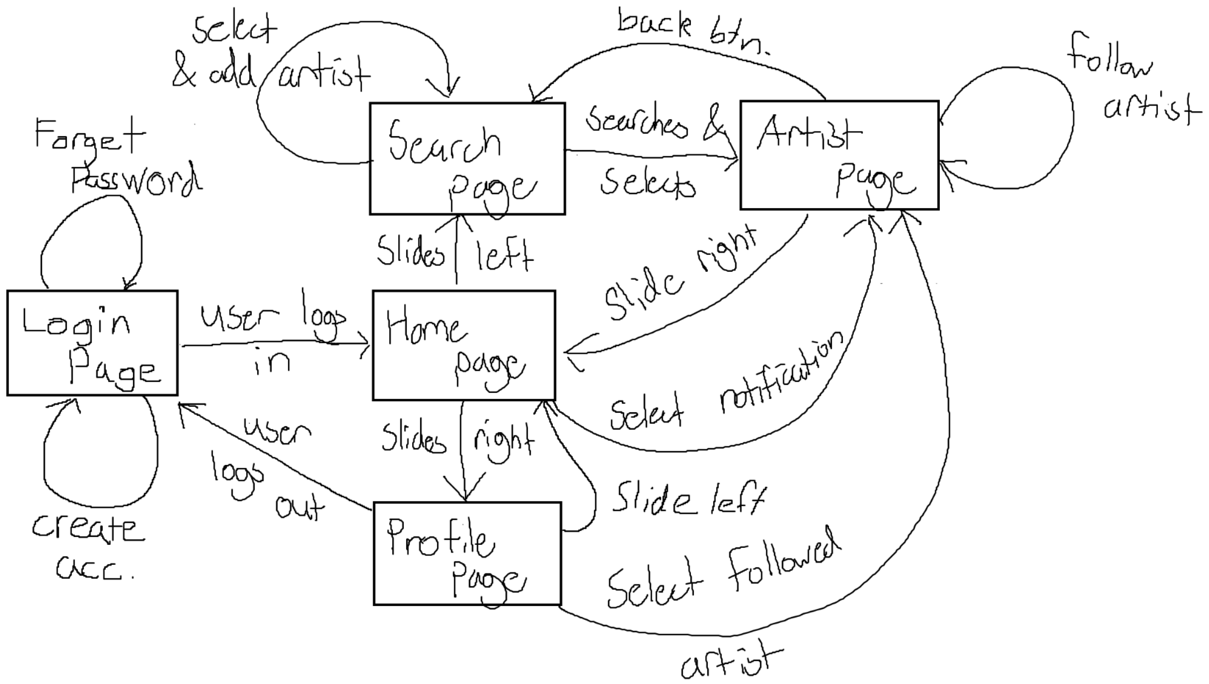
Project Name: Timely Tracker

Project Synopsis

An app that tracks artists and their recent activities.

Architecture

User Interface Flow Diagram



The *User Interface Flow Diagram* is a flowchart on what and how the user traverses the app.

Following the *User Interface Flow Diagram*, when the user opens the app for the first time they will be met with the *Login Page.* If they have decided to stay logged in they will start at *Home Page.*

The *Login Page* is self-explanatory. Here they have the options of creating an account, recovering an account (*Forget Password)*, and logging in. Once the user has decided to log in they will be greeted by the home page.

The *Home Page*, which could look similar to what is shown below, is where the user will be receiving most of their information from the app. This will be described in further detail in the *UI Diagram* Section. From here the user has the options of swiping left or right (the method of swapping pages could be changed later) to go to the associated pages. They can also select a notification to go to the associated artist's page.

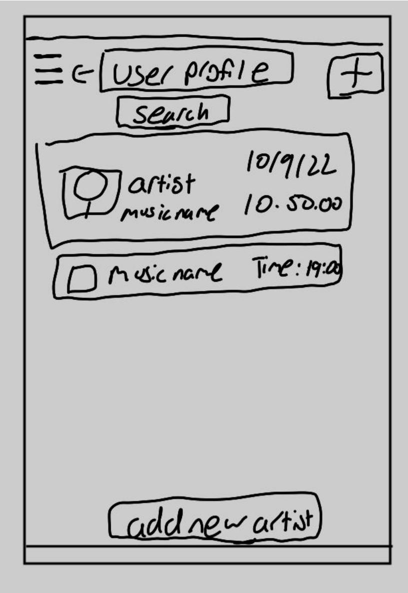
If they were to swipe right from the *Home Page*, they would be taken to their *Profile Page.* This page contains information about the user and some customization. From here they could log out or select a user-followed artist.

If they were to swipe left from the *Home Page*, they would be taken to the *Search Page.* This page would contain the ability to search and add artists to the user's followed list. They can also select an artist to visit their page.

If the user reaches the *Artist Page* through multiple ways, they will have the ability to look through artist feeds and activities to see what they have been doing recently. The user can follow them and things alike. From this page, they can return back to the *Search Page* by pressing a back button (which could be changed).

**The user can always return to the Home Page by swiping accordingly back to the "center".*

UI Diagram



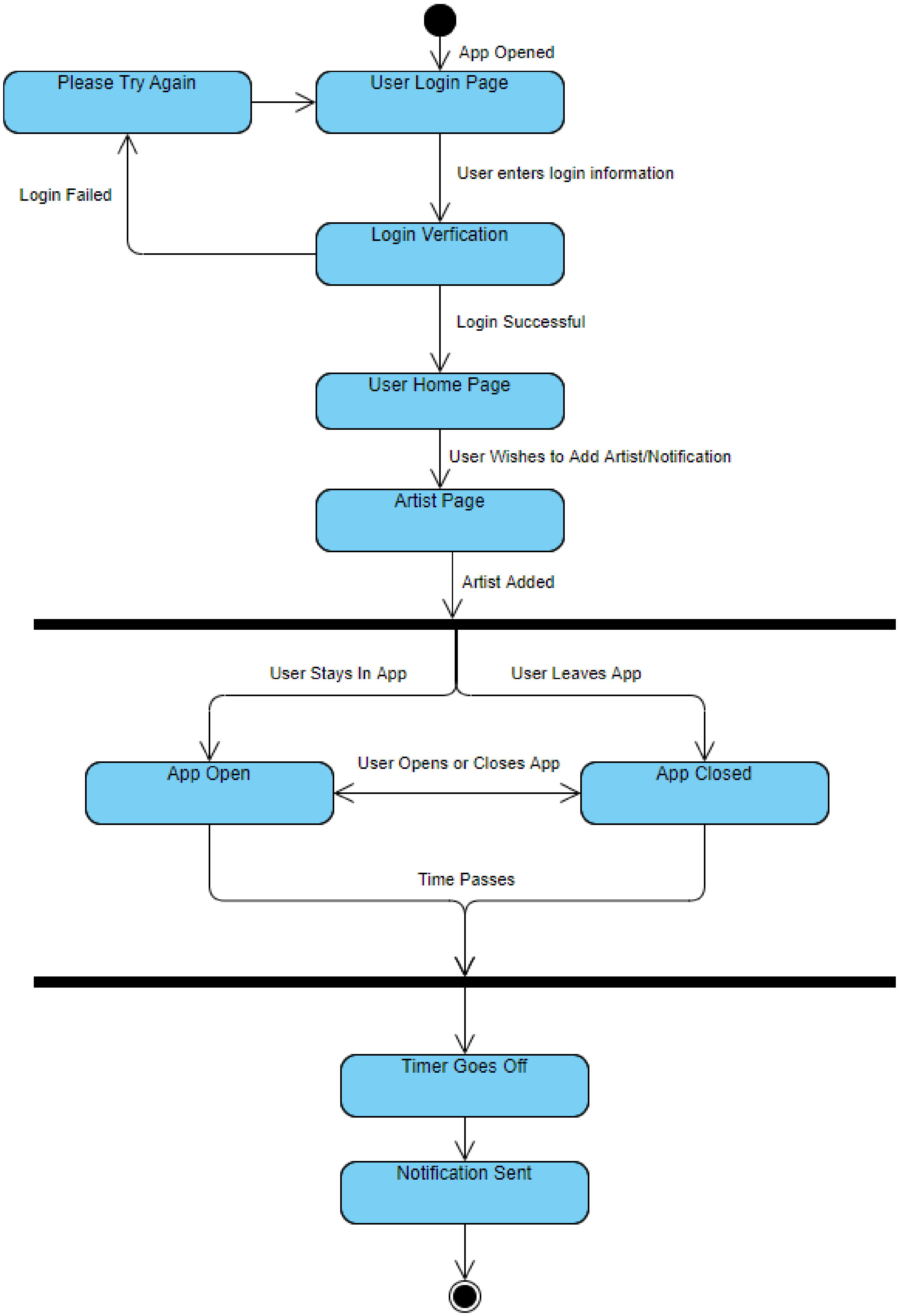
The UI design for our mobile app is styled up a "blocky" look to match the appearaace of other mobile apps on the IPhone. The user profile can be accessed by swiping right on the homepage of the app to be able to access the information that the user has inputted into the app. The information being what is their favorite artist, what kind of music they listen to, or what is the countdown of what music they like is releasing.

The rough homepage design that is shown above, showcases the fundamental features that the app will have. Being able to access the user settings either on the right or left side of the homepage and being able to search for specific artists or music that they want to keep track of. The tracking feature that is the main focus of our app is shown on the right side of the block with the artist information. It would count down to a specific date or time to be able to notify the user of when something is releasing. Making sure that they do not miss what they are interested in. To be able to assist in tracking, at the top there is a searching feature that would assist the user in finding what music or albums to be able to track. Utilizing the database of Spotify we would be integrating it into our app to be able to accurately find what music or artist to track and count down for the user.

There are also quality of life features for the user to be able to navigate around the homepage more quickly, like a “add new artist” button at the bottom to be able to quickly add a new artist to their homepage if they wish to. There is also an added button at the top right to be able to add in new music if the user wishes to so that they can track what music they might be interested in. Another feature on the homepage would just be being able to manually add in a tracker if it does not exist in the Spotify database, it would be located with the plus button on the top right of the homepage.

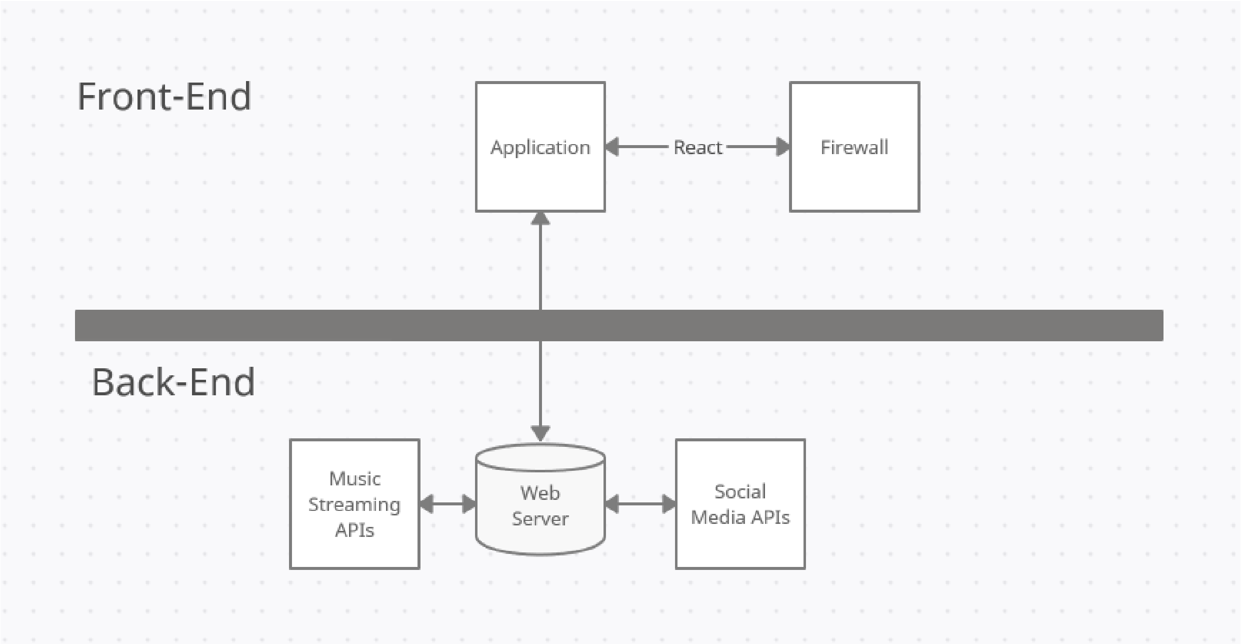
Instead of not being able to add in music or artist because the Spotify API was not able to locate a specific music or artist, the user is able to input the information with specific parameters to be able to add in an artist or music name to be able to countdown. They are able to adjust the date or time to countdown so they are notified at the right time and date to help them remember what music is releasing at that specific time frame.

State Diagram


App Opened
Please Try Again
User Login Page
User enters login information
Login Failed
Login Verfication
Login Successful
User Home Page
User Wishes to Add Artist/Notification
Artist Page
Artist Added
User Stays In App
User Leaves App
App Open
User Opens or Closes App
App Closed
Time Passes
Timer Goes Off
Notification Sent

Above is a basic description of the states the app goes through in order for a user to login to the app, add a timer for an artist they like, and then be notified when that timer goes off. When the app is first opened, the user would be greeted with a login page. The user would then enter their login information, and if successful would be brought to their homepage. If the login information is incorrect, the app would take them back to the login page. From the homepage, a user would go to an artist page to add their desired notification. Once the notification is added, the user would be able to stay in the app or leave the app closed until the timer they set ends. Once the desired amount of time passes, the timer would go off, which would cause a notification to be sent to the user. During this time, any number of timers could be added, deleted, or edited and still not affect existing timers.

Technology Stack Diagram



The above diagram shows how the front-end and back-end of our application are connected. In the front-end, we have an application that the user will be able to see and interact with in a way that is described in our state diagram. The application will also be connected to a firewall via react-native, since we will be implementing the app using it we will be able to make the application available to Android and iOS products without having to create two of the same application. The firewall will be able to protect our application from malicious software and make sure that it is safe for our users to use. Below the front-end is the back-end, the section our users will not see. In it, we have a web server that keeps the application going and contains all the contents that will be needed for the application, like the code and the database for the user's account. Along with the web server, we also will be using social media and music streaming services APIs.