

# Spotify Playlist Generator

Team: **Evan Easton and Adam Hayes**

## **SUMMARY**

### **Work Done:**

We've set up our functionality for creating an account and logging in as a created user, while handling cases of only 1 account per email, and validating login credentials.

**Adam:** User interface functionality, validating login credentials, MongoDB connection

**Evan:** Create user accounts, RouteController structure, HTTP request interactions

### **Changes/Issues:**

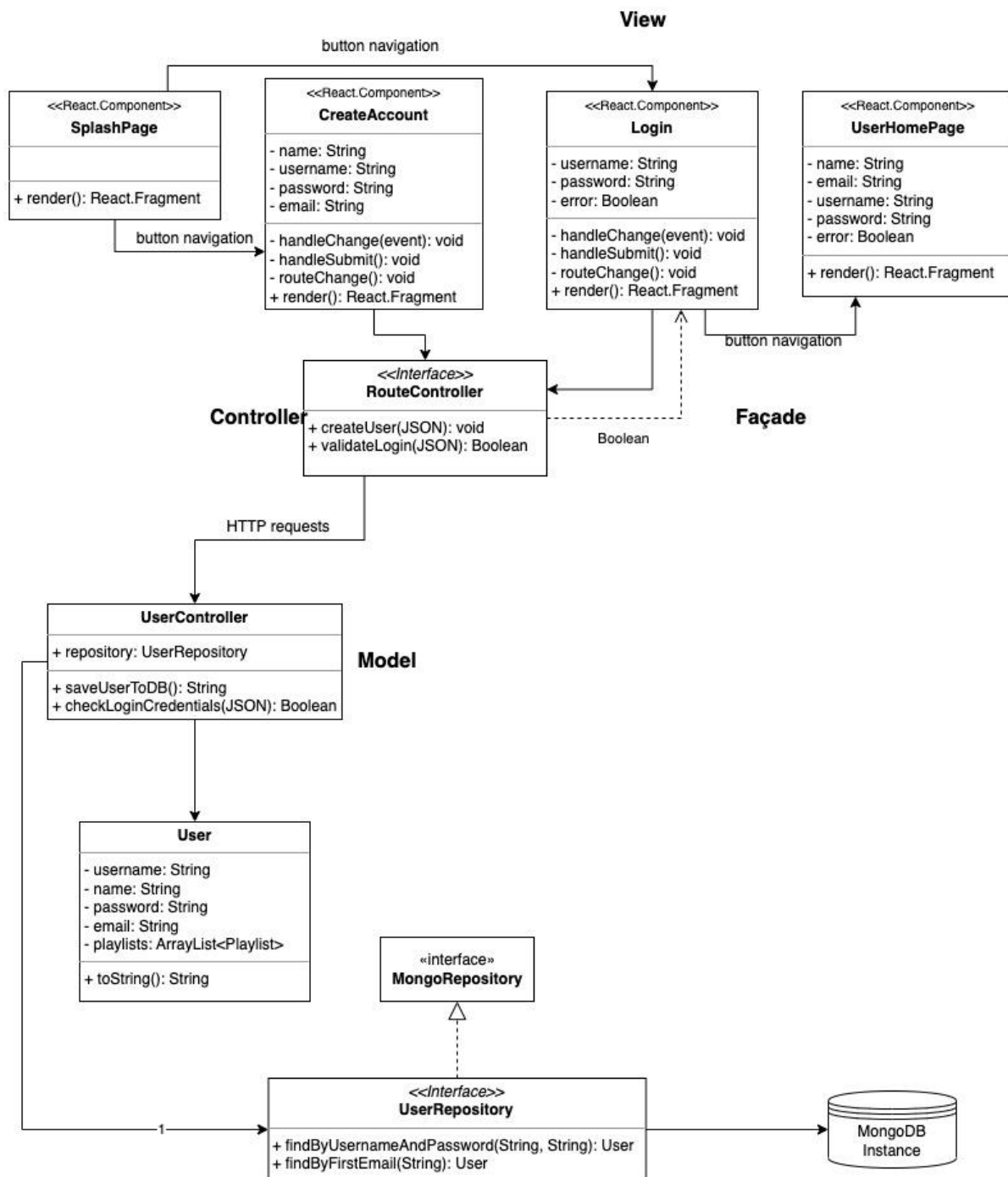
There have been a few things changed from our initial design. Originally, we thought that the RouteController would be an object that would be referenced by the React framework of the front-end, but we realized that we only needed the RouteController to be an interface of methods that the React front-end can utilize to make calls to the back-end. We also had a bit of difficulty figuring out how to properly have the RouteController make requests to the DB and return the proper information. We also removed the DatabaseCommunication object and replaced it with a Spring Boot/Crud repository that does the work for us so we don't have to make intricate database connections. Spring Boot enables us to use lots of dependency injection that made our lives a lot easier when dealing with the MongoDB database.

### **Patterns:**

MVC: we've set up our project with a Model-View-Controller. Our View is the React Component pages which communicate directly with our Route Controller interface (Controller), the Route Controller interface sends HTTP requests which our backend is currently handling with UserController (Model).

Façade: our RouteController is an interface which handles all communication between the frontend and backend

## CLASS DIAGRAM



## **NEXT ITERATION**

Going forward, we need to implement the functionality for a user to generate a playlist, and save it to their account.

TODO (frontend):

1. User Home Page w/ generate playlist option
2. Search for song functionality
  - a. Search bar
  - b. Display list of songs from backend
  - c. Handle user selecting song to add to playlist generation module
  - d. Send selected songs to backend
  - e. Retrieve playlist generated from backend and display
3. Display users saved playlists on their homepage

TODO (backend):

1. Server needs to reach out to Spotify API for information from the user
2. Save playlists to user objects
3. Strategy algorithms for selecting a related song based on user chosen song
4. Implement builder pattern to build playlist instances
  - a. Return playlists back to front end to give to React

Final Product:

Website where user can create an account and login, then on their homepage they can view previously generated playlists and generate new playlists. To generate a new playlist, the user will search for 1-5 songs that they like, then based on those songs a playlist will be generated for them and displayed.