

**Sebastian Zimmeck**  
**COMP 333: Software Engineering, Spring 2022**  
**Homework 4: Deployment (Heroku and Firebase) and Mobile (React Native)**

---

**Homeworks 2 – 4 will cover an ongoing app development project. A crucial part is to work together as group. It is recommended that you use GitHub to organize your work and keep track of the evolution of your codebase.**

**Check out the code examples, tutorials, and slides on the class website. They are intended to get you started and illustrate core ideas covered in the homeworks.**

**Problem 1 [30 Points]** You decide to deploy your Django backend to Heroku and your web frontend (from the previous homework; not the React Native frontend from below) to Firebase. You can deploy your code as described in the deployment tutorial on the class website.

The result of your deployment should be a functional app with the features you implemented in the previous homework. **Please note that you do not need to add missing features or fix bugs from the previous homework assuming that your app runs in a stable manner.** You can just deploy your app in the status of the previous homework. However, please describe in a readme the limitations of your app as compared to what was asked for in the previous homework. That way, we know what to expect. You can still get full points for this problem.

**You do not need to send us any code for this problem; just the URLs of your Firebase frontend and Django backend and a readme.**

**Problem 2 [70 Points]** You are getting to work on the mobile version of your app and decide on using React Native. If you are on macOS, please use the iOS simulator. Otherwise, please use the Android emulator.

Please submit your local frontend and local backend code and tell us how to run it in a readme. We want to run you React Native app completely local. **You do not need to deploy your React Native app. For both the React Native frontend and your Django backend, it is sufficient if it works locally.**

To get started with React Native you can use Expo (<https://docs.expo.io/>) and Create React App (<https://github.com/facebook/create-react-app>) as described in the React Native tutorial on the class website. In addition, if you want, you can also use React in your React Native app.

Note that in order for your React Native app to be able to interact with your Django backend *both* your React Native app as well as your Django backend must be running. You can open two terminal instances and run your frontend and backend at the same time.

While generally changes to your React Native apps are hot-loaded, this functionality may not always work, in which case you have to restart Expo and/or the emulator/simulator to have your most recent code changes reflected.

Your mobile app should have the same features as your web app, i.e., it is a full stack CRUD app with a mobile frontend that integrates with your Django backend the same way as your web frontend. As your web frontend, your mobile frontend is intended to allow a user to do the following via a graphical user interface:

- Create new songs in a database with title, artist, and rating
- Read a list of songs from the database with title, artist, and rating
- Update title, artist, and rating of existing songs in the database
- Delete songs (title, artist, and rating) from the database

The rating of a song should be based on a scale of 1 to 5.

A user should be only able to rate a particular song once. A user should be notified that they cannot rate a song twice if they try. A user should also be able to change their rating of a song.

Songs (title, artist, and rating) should be displayed for all users in the database. For example, if Sally entered "While we're young" and Don entered "Sweet Jane," both Sally and Don will see a song list containing "While we're young" and "Sweet Jane." The rating of a song that both Sally and Don see should be the average song rating of all your users.

You can assume that there are no two songs and no two artists with the same name. In other words, you do not need to check for duplicate songs or artists.

You can use an additional ID attribute that increments an integer in any of your database tables as a primary key, e.g., if it proves easier for deleting data consistently across tables. Use of such primary key is just one option. There are many possibilities to achieve consistency across your database.

In addition to React and Django you can use any library you like; for functionality, styling, fonts ... .

## Submission

Please upload all your files in one zip folder to Moodle. You only need to submit one solution per group. It does not matter which student's Moodle account your group is using. The homework is due on **4/29/2022, 10 am**.

- Include **both your React Native frontend as well as your Django backend code** in the submission.
- Only submit source files, including automatically generated source files. Do not include dependencies or automatically generated non-source files (e.g., .pyc files).
- Provide **clear instructions** in your readme on **how to install and run your app locally in a development environment. Provide the links --- Heroku and Firebase --- to your deployed app online**. Test that your app is running per your instructions. **It must be possible for us to recreate your app on our end by just following your instructions**. The easier you can make it for us to review your work, the better.

## **Grading**

**Please try to distribute the work evenly and indicate who did what in the readme(s) you provide. Unless you tell us otherwise, we assume that all students in a group contributed equally and, thus, the grade of each student will be equal as well. It is not possible to challenge a grade after grading based on one student having contributed more than another. If a student contributes less than their equal share, the student's grade will be reduced accordingly. Similarly, if a student contributes more than their equal share, the student's grade will be increased accordingly up to the maximum available grade.**