Final Report

Section 1:

Linnea Dierksheide

- Backend, Rest API
- Recommendation system

Melody Lee

- Frontend

Christina Ng

- Frontend

Sreya Nalla

- Backend, Rest API
- Recommendation system

Section 2:

Bar Voyage is a mobile application that informs and guides users through an optimized fun night out by providing tailored recommendations and real-time, crowdsourced images. Bar Voyage's main values are through its recommendations, real-time content, location, and user ratings. Through these 4 values, Bar Voyage allows users to 1) discover new places and optimize their time on a night out and 2) promote knowledge sharing and community through crowdsourced images/videos. This way, when a user, or voyager, is looking for their next bar destination, they aren't left wondering where they're off to next. Our app eliminates the common disappointing scenario of aimlessly hopping from one bar to the next, unsure if the next will be any better. By giving users the power to view live content of a bar from other users, everyone is empowered to share their own experiences and contribute to the Bar Voyage community.

The app has a simple workflow. A user can sign up for an account and provide simple demographic data such as their gender and age, which is used in our recommendation system. Once logged in, a user can select multiple categories depending on their preferences for their night. Some options include karaoke, dancing, dive bar, and sports among others. After selecting their preferences, the user can interact with a map screen of the different bars that meet their preferences and are nearby. They can also scroll through a list of recommendations that Bar Voyage provides taking into account their personal preferences, past reviews, and other users on our app. After deciding which spot they want to go to, the user is presented with Part 1 of our review system, where they rate the place, mark adjectives to describe the current bar, and if there's a cover or long lines. Once they're at their next spot, they'll be asked to upload content at the previous bar, ultimately growing the Bar Voyage community.

Section 3:

bar-voyage.github.io/website

- An About page with a description of our project
- A Contributors page with our team members' pictures, bios, and social links
- A Source Code page that links to our GitHub organization containing our code repositories
- A Home page with the project presentation video and commercial video
- Link to the final report

Section 4:

Frontend:

- Expo Vector Icons, React Native Async Storage, React Navigation Stack, React Google Maps, React Native Geolocation Services, React Native Get Location, React Native Image Picker, React Native Web Maps, Firebase, Axios, React Hooks, React Stately (checkbox, collections, combobox, list, slider, tabs, toggle)

Backend:

- AWS SDK, Content Type, Express, HTTP Errors, Mime, Multer, MySQL, Parse URL, Query String, UUID, XML2JS, various buffer and xml libraries

Section 5:

A user of bar voyage interacts with a frontend web app, written in React Native. This frontend communicates with a backend hosted on an AWS EC2 instance. Some information it gets and passes from the backend includes but is not limited to the recommendations and the current user. The backend system interacts with three other components: our content storage, database, and our recommendation engine. Our content storage is hosted on AWS S3. This is where our user's uploaded content gets hosted on and where the content gets pulled from for each bar. Our database uses AWS RDS. The database contains information about both users and bars. This database interacts with our recommendation engine which uses AWS Lambda.

Our recommendation system works differently for new users and existing users. For new users, we take a SQL-Based Recommendation approach where recommendations come based on preliminary user data (age group, gender, location), and nightly user preferences. For existing users, we use memory based, user-item collaborative filtering. Recommendations are made similar to new users, where we use both preliminary user data and their nightly user preferences

but we also take into account similar users and their ratings. So for example, if Kate is similar to Ashley in that they both enjoyed Mad Hatter and CUT Above, and Kate has been to L'Annexe, we will recommend L'Annexe to Ashley.

Section 6:

Linnea: I learned how to create a Node JS API application, furthered my knowledge of React, and how to create and integrate AWS functions like EC2, S3, and Lambda, into a project. When starting, I was pretty unfamiliar with Node JS, but we did a lot of research and iteration to figure out how to get a simple API endpoint running, and then with how to structure our code with best practices. In the process of figuring out how to deploy our application, I learned a lot as well. Setting up the EC2 was somewhat familiar, but I gained experience with that. I took on the bulk of the responsibility with the S3 bucket, and learned a lot in figuring how to integrate that into our application, from getting the files, processing them in our backend, and then storing them in the S3. Where I would have done things differently would be to start working on the Lambda and recommendation algorithm earlier and to have figured out a better approach that was simpler and easier to implement, and then to build on it later. This would have made it easier to integrate recommendations into our database format as well, which was a challenge we faced at the end. Also, had we done this earlier, we could have gotten better test data for training our model.

Melody: I learned how to set up a React Native app and structure the codebase to optimize for code reuse and to take advantage of React's component-based style. Next time I would develop using a mobile device or emulator early on, since the support for Expo web view is much more limited. For example, instead of being able to use a popular and well-supported library called react-native-maps, we had to use an offshoot web version which didn't include as much functionality. I also would have given myself more time to work on features and bug fixes before demos to minimize last-minute panic troubleshooting.

Christina: Working on this project, I learned a lot about front-end in both prototyping and React. When we first started doing mockup designs, I was able to play around with Figma and while working on the project, I was able to work with a lot of new libraries in React and learn some new concepts! If I had to redo this project, I would have stuck to using a phone simulator or Expo's emulator as this would have made it easier to work with certain components of our app such as the map. I would also create more detailed plans for what needed to be on each page (components, layouts). It would have also helped if we created skeleton sketches of each page earlier and slowly built functionality over it. This would have allowed us more flexibility and time to implement other features we didn't get a chance to work on.

Sreya: Prior to beginning this project, I had minimal knowledge in many of the components that led to our final product. I hadn't before had to map out a system overview from scratch or really weigh the pros and cons of a tool that we were planning on using. Much of my time spent on this project felt uncomfortable – the feeling of the unknown scared me. However, looking back, I realize how much I learned. I learned how to create a Node JS Rest API with POST and GET requests. I learned a bit about React Native. Most importantly, I was able to expand on my existing AWS skill set. After having worked with AWS over the summer during my internship, I was much more familiar than a few of my teammates, but I still had much more to learn. I became a lot more familiar with integration AWS into Node JS and Python and also better understood the importance of Lambdas and the different database options that were available. Aside from the technical skills that this project allowed me to learn, I was able to sharpen some of my organizational skills. Throughout this process, I learned how to better split up tasks among myself and my team, play to each other's strengths and plan well in advance. Our team used Trello boards, Google Drive and met at a timely manner to achieve everything we were able to present. If we had the chance to do this project again, I think I would tell myself to be a bit less hesitant. I struggled with getting started on the recommendation system because I was afraid of the unknown – I kept putting it off until the very end when I was forced to get started on it. I wish I had been more confident and willing to take on this challenge earlier on so we could have delivered a more polished product. Having this part completed earlier in the project timeline would've allowed our group to better integrate components and improve latency within our project's recommendation system.

Section 7:

Our advice for future students would be to not get too frustrated while building the project and to enjoy learning new things! We suggest thinking about how to break down frontend code into reusable components as well as checking that AWS does not keep charging your credit card. Some ideas for follow-up projects are a social app to see which of your friends want to go out on particular nights and where, a generalized version of Bar Voyage that applies to all kinds of places (airports, tourist spots, parks, grocery stores, etc), and a swipe-based app for finding the best cocktails suited to your taste in your area.

Section 8:

Frontend setup instructions
Backend setup instructions

Section 9:

A user is able to create an account, log in, input their preferences, see local recommendations (from recommendation system) based on their location in both map and list views, view information and top adjectives about bars, check-in and submit ratings, upload photos, and see previous bars they have checked into. Map markers are accurately placed, however clicking map markers does not take you to the bar information page. Formatting/layout on mobile devices is inconsistent with web/desktop view. Retrieving a user's geo coordinates is not instantaneous and takes a few seconds, so submitting vibe preferences too quickly may result in errors.

Appendix I:

N/A

Appendix II:

Link to homework assignments