Nakra Ath-Ly, Christopher Lopez

Bo Sheng

CS 443

21 December 2021

Mobile Project

1.Project Statement:

Our objective for this project was to construct a mobile app that can track bets for gamblers who want to show their net gains or their betting history/career. The app will have a signup or a login for the user. If it's a new account it should make a person sign up and once they sign up they can login and see their history. While having an account you are able to see your bets and put in bets. It’s hard to keep track of your wins and losses in a betting season and some people wonder if they are positive or negative in terms of earnings for the season. This will help the average customer proceed throughout the season with educated decision making. We decided to develop this app because both of us gamble and wondered if we could create any easy way to track our own betting/gambling history. Personally, we don’t use excel spreadsheets because it requires a lot of manual input which can be a hassle to deal with. It is a good way to see if you're winning or losing in the long run with minimum effort. We bet on professional NBA and NFL games so the app helps keep our winnings recorded properly. I believe other gamblers who face the same issue we face when it comes to tracking their winnings will appreciate this app and see it can solve a lot of their problems. This application is supposed to perform for the user with minimum input. They can enter in a title for their bet, the amount they wagered, and the amount

won or lost, and we calculate their total ROI (Return on Investment) and total profit or loss throughout their betting career. We don’t believe there's another app like this on the market except gambling sites that see your cash out and stuff. This seems to make the user add all their bets so they can have it in any way they want. If the user bets in real life or online, this app can keep track of all the bets they put in. We thought that if this was an official app it would be either free and they have to do a monthly payment or it would cost a few dollars to purchase the app and there would be no more purchases afterwards. Our main goal is to make gamblers and betters feel comfortable and know their net gains of gambling. It can save people from quitting or even help them keep going. People will be willing to quit if they see they have lost way more than they have won.

2.Application Design

For our application, we decided to use React native. We decided to use React native over the native approach (Kotlin/Java for Android or Swift/Objective-C for iOS) because it allows us to develop our application for both platforms simultaneously. This saves us the trouble of having to learn two different languages, and maintain two different codebases, if we wanted to launch our app on both iOS and Android. Our app has several pages. We have an initial landing page, which users see if they are not currently logged in, and it has two buttons, which users can use to either login or signup. See figure 1.

Figure 1. Landing page with Sign Up & Login buttons

When pressing either button, it takes you to a separate screen. For logging in, it takes you to a page where it allows you to enter your email and password, and then login to your account,

as shown in Figure 2. For signup, it allows you to do the same thing, but then registers you with the provided information, and automatically signs you in, as shown in Figure 3.

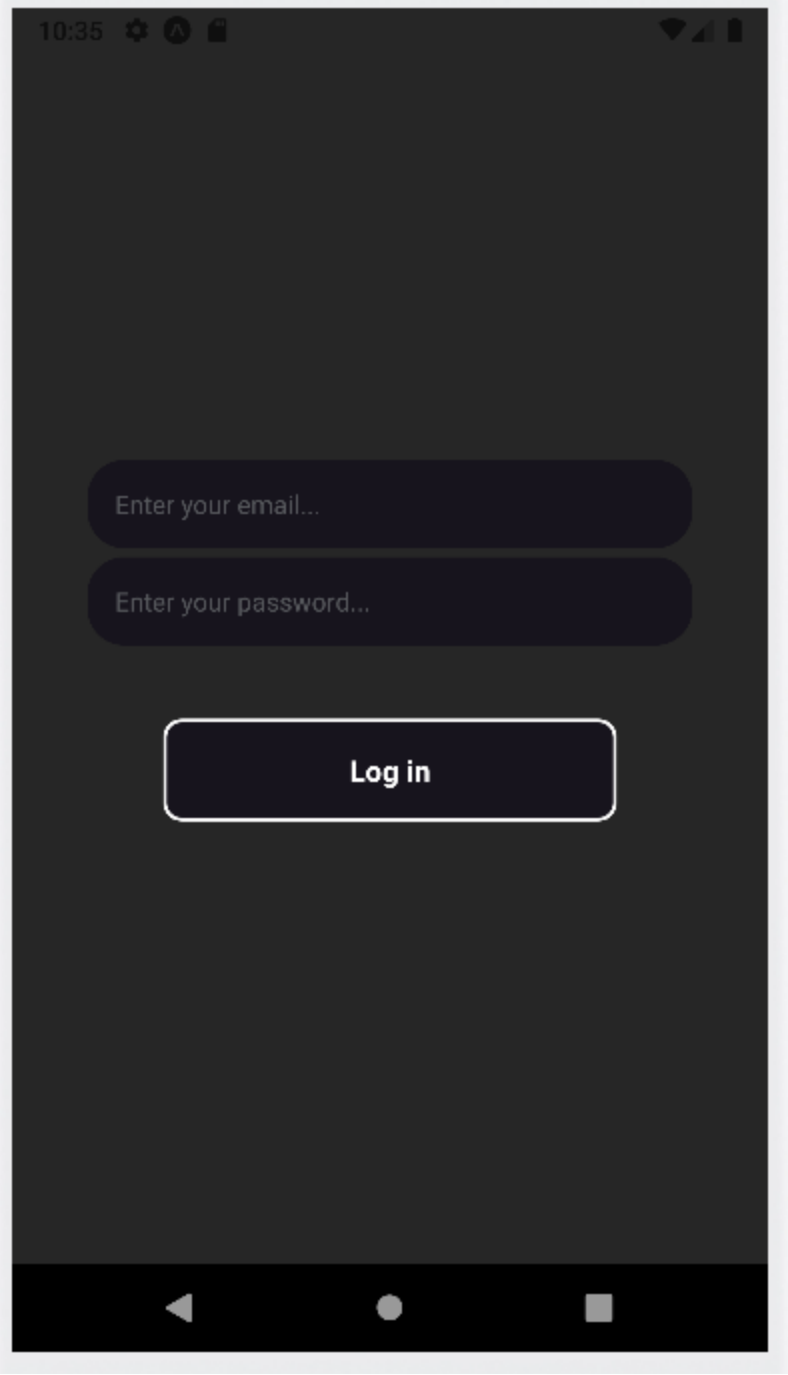

Figure 2: Login screen

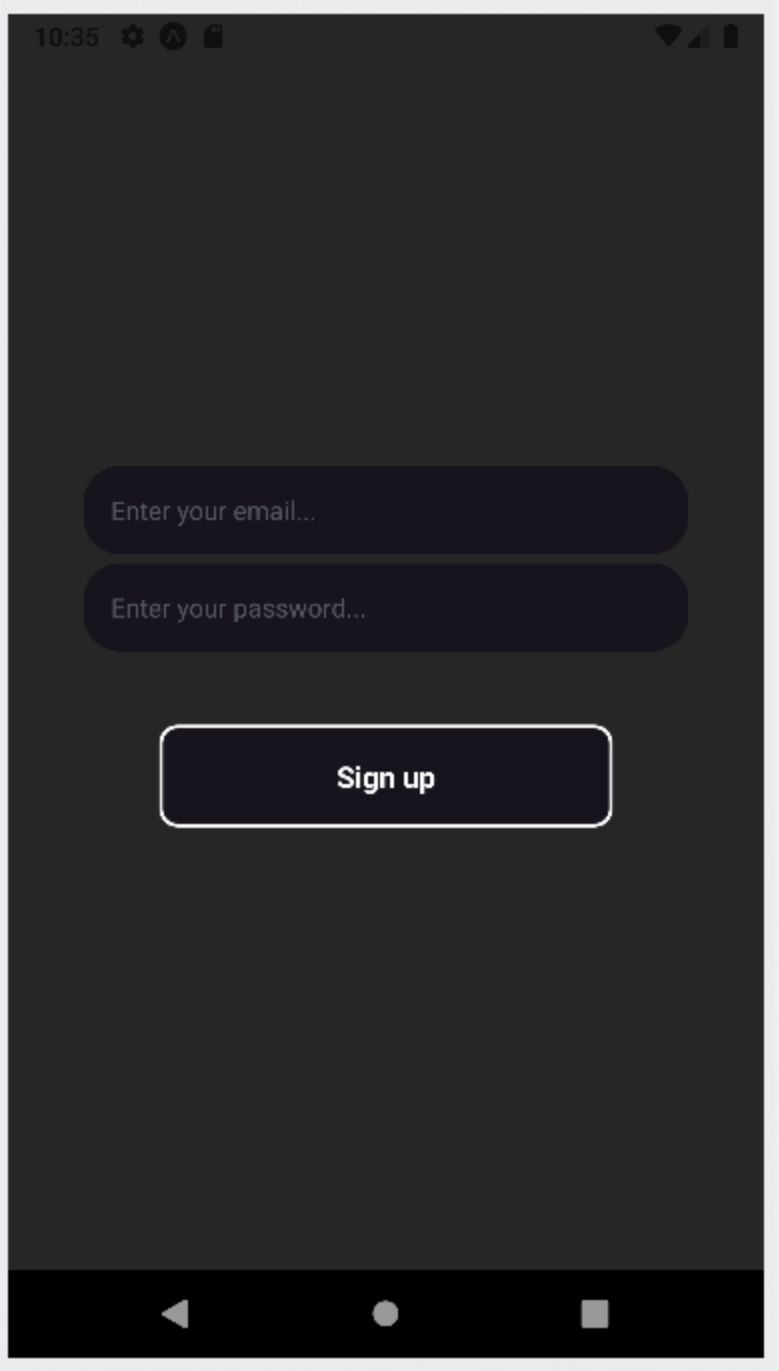

Figure 3: Sign up screen

For our user authentication, we decided to use firebase. We chose firebase because it makes it extremely easy to handle user authentication, and requires very little effort from us to get started. After logging in, you are directed to the home page. On this page, we show you your

career betting states. We show the total amount of money you've wagered, the total profit / loss, and your total ROI. There are also two buttons, one for creating new bets to track, and another to signout. Below these buttons, we display the bets you've already entered into the app. If the user has not created any bets, we don't show anything below the buttons. See Figure 4.

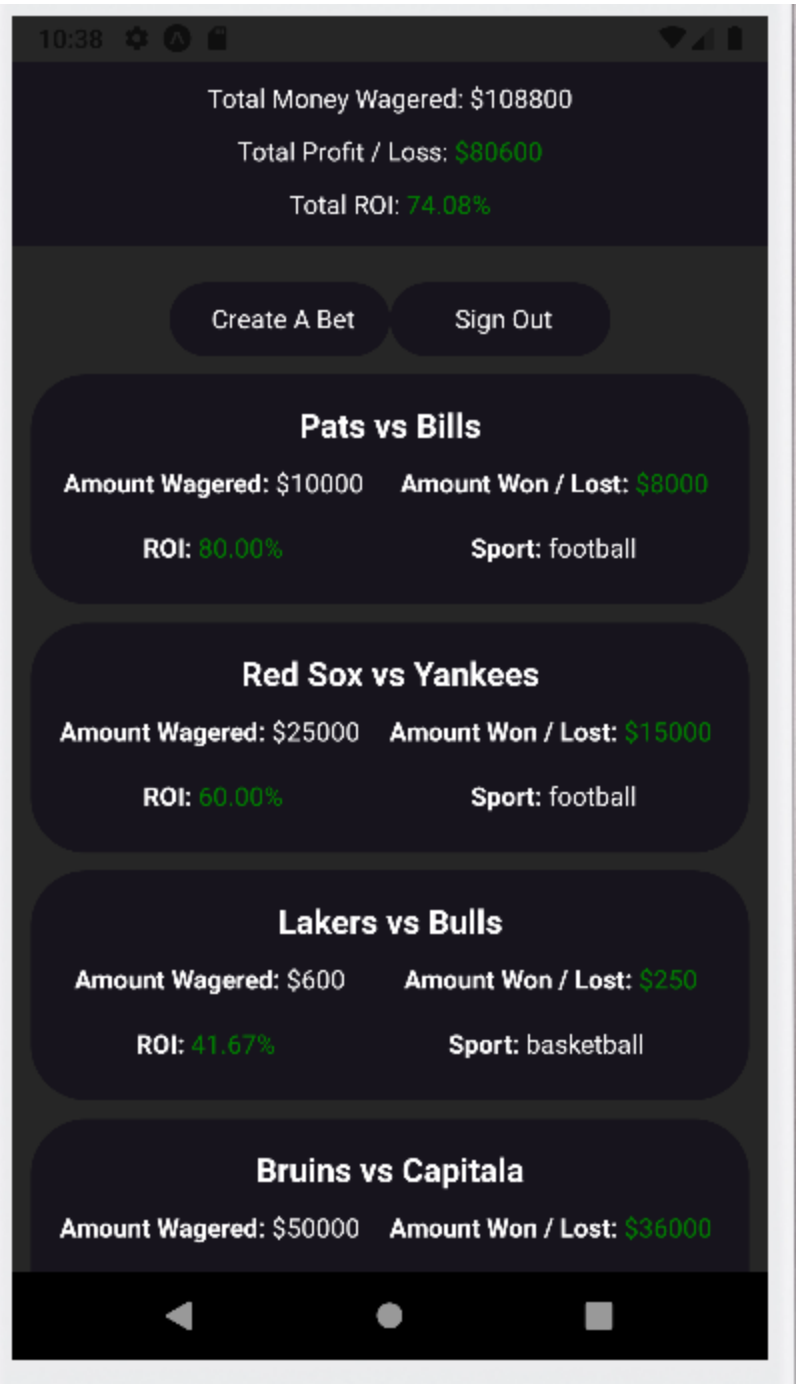

Figure 4: Home page showing user stats & bets

When clicking on the create a bet button, you are taken to the create a bet screen. From here, you are able to enter in the title of your bet, the amount wagered, the amount won or lost,

and the sport. We calculate the ROI automatically after you enter in your information. You can then click the submit button, and this saves this bet to the database. For our database, we decided to use firestore. Similar to firebase, it was very easy to set up because the documentation for firstore is excellent, and because of this requires far less setup time than traditional databases. It is also a NoSQL database, allowing for easy access and manipulation.

After clicking submit, it takes you back to the home page, and displays the new bet you just added, and updates the total values. If you want to edit a bet, you can simply click on the card on the home screen, and it takes you to the edit bet page. This page is essentially the same as the create bet page, the only difference is that the values are pre-populated with the values corresponding to the bet you're editing. When you're done, you can click the sign out button to sign out of the app, which will take you back to the landing page.

3. Application Implementation and Evaluation

The code for our app is organized into two main folders. There is the screens folder, which holds all of the screens, or pages, such as the landing page, home page, create a bet page, etc. We then have the components page, which holds the only reusable component currently in our app, which is the BetDisplay card that shows entered bets on the home page. As previously mentioned, we used react native for our project, so the code is written entirely in Javascript. For state management, we used React hooks and functional components, so state is managed throughout using the useState and useEffect hooks. For more information on hooks, you can visit the React documentation here: https://reactjs.org/docs/hooks-intro.html

For testing, given that we used React native, we were able to test our app on both Android and iOS devices. We used both Android & iOS emulators, along with actual Android and iOS phones. For functionality and performance, we spent several days clicking through the app and making sure the primary features worked as intended, such as inputting & editing bets, having all of a users bets load upon login, etc. For performance, Expo, which is the platform we used for our react native development, it has performance testing built in to their platform, which allows us to monitor the UI FPS, number of dropped frames, and stutters. For our app, throughout all our testing, we maintained a steady 60 FPS. The app however does not always work correctly as intended. On certain screens, the header on the homepage that shows the betting stats is sometimes obstructed by the top navbar on the phone. This seems to be a bug with the built-in react native components affecting our code for some reason, because when wrapping our code in a component that is meant to accommodate for this, we still have this bug, so we were not able to fix this. The only other bug is that, occasionally on the first load when loading the home screen, the total profit / loss, roi, etc. is not calculated, despite the fact that the wagers are being populated. This is another weird bug with the react-native hooks, since through testing and building the equivalent code for react web, the code should be correct, but it's not always working correctly on mobile devices, so this is not a bug that we were able to fix. Apart from those, all of our code works as intended.

4.References

React Hooks

Expo

React Native

Firebase Documentation

5. Experience and Thoughts

Nakra:

We wanted to get an API for sports but we were not able to maintain one, so we had to do something different. I feel for the class it can be more interactive, there's a lot of slides which can be overwhelming for students.

Chris:

For me, the main things I learned through this project was really just react native and firebase in general. I had worked with react on the web before, but using react native was a whole different beast. I also got to learn a lot about firebase & firestore, as I had not used them before this project. When we originally thought of this, we wanted to use APIs to pull in live betting odds so that users could then see potential sports they'd want to wager on, but unfortunately, upon further inspection, the APIs had not been maintained in quite some time and were no longer functioning as intended. We also had a few bugs as described in section 3 that we were unable to fix, but they were mainly weird issues with some react native components as opposed to large errors in our own code.