Taylor Yoeuth

COMP 4630

Professor Haim Levkowitz

November 29, 2021

Term Paper: HouseBuddy

Project Proposal:

My partner and I have come up with an idea to make it easier for new homeowners do their shopping for their newly purchased house. With our idea, we wanted to create a way to take out the tedious task of searching for items for their home and having 12 different tabs for one item. We have created a way for users to create a list, and within that list it will specify what kind of item the user would want. Within that, it will show the user some possible items from Amazon for the user to purchase. If the user pleases so, they then will be able to check off that box saying that "purchased" the item.

The reason why we wanted to target newly homeowners was because we are graduating soon, and hopefully down the road five+ years from now we will be on our way purchasing a home ourselves. What better way to create an application if you truly believe you are going to use it yourself? We wanted a way for users to be able to sign in to save their lists, and we did so by using Firebase.

Implementation:

Firebase was a good way to implement on having users creating a profile, and to record their data. Users will then be given the option to sign up or to sign in with their google account. This way it was easy for users to just use their google account instead of creating a new login and a new password. This was the first time my partner and I had used firebase and how to implement it towards our project. Code was simple with documentation but first we had to understand why it was doing what it wanted to do.

Once we had that setup, we then and created a story board on how we wanted our application to look like. We pulled up paint onto our computer and started sketching some ideas. We needed a logo to represent our application just like a mobile application on the apple store and google play store. We decided to create a logo using a free logo maker online that incorporated a house hence why our application name is HouseBuddy.

Once we created our logo, we needed some inspiration on how we wanted our application to look like and to make it unique but also modern. We then drew up a mock design in paint and agreed that we needed five different tabs for users to click on: Home, Favorite, Browse, Profile, and Settings. Before users reached home, they would be prompted to sign into their account which will then be registered into our Firebase which is visible to the admins only (my partner and I). Then right after they have successfully done so, they will be prompted to our Home tab which they will be able to see the "rooms" which can be clicked on and then users will be able to

add in items to their specific rooms. Rooms included can be the master bedroom, office, guest room, bathroom, etc. Once the users enter their item, they will click a button which will allow the item to be added onto a list. Then that item will be able to see on where to purchase the item from Amazon. We did not have a curated list of items, so we went with necessities first since we do not have a lot of time to make this a full-blown application.

In our current stage of the application, we can add items onto the list but are working on how to implement the way to showcase Amazon onto the application. We will then allow users to click onto the item and be brought to Amazon to purchase the item themselves. We hope to figure out on how to do this with the given time remaining in the semester.

Our Amazon shopping or shopping feature is our main highlight for this mobile application, and hence why we wanted to create this project. We wanted to minimize the task of shopping for household items with having a curated shopping list for our users in a modern way. Without having this implemented, we would just have a mobile application that is no more than a to-do list.

We also implemented different tabs within the Home General tab which allows users to have different lists. Let's say you own multiple properties; we wanted a way to distinguish between different properties to make it a multi-use mobile application. We looked at it in a business aspect as well and saw that this would allow more users to use our mobile application than any other application out there.

Next, we have our Favorite tab. As of right now we do not have anything implemented for this tab, however we wanted users to have a way to see all their favorite items they had on their lists to make it easier to jump to. The Browse tab is not implemented yet as well, however we planned on having it to have a browser for users to explore items using the internet. We do not know if we have enough time to implement this feature but if we continue this mobile application in the future, we plan on adding this feature.

Our Profile tab shows the user their profile which contains their email address used to register for the account and allows users to sign out of the account as well. We planned on having a full blown profile page where the user can upload their picture and have it saved so the user can see their profile but with time constraints that is not on our priority list. We also thought of having the idea of having some type of forum for members who talk about their favorited items and others can see their honest reviews about it, but we can save this for future implementation.

Finally, we created a settings tab which will be the support section for our application. This can be where users can personally send us an email to give us a question or file a "support ticket". The user will enter their name, their email address, their message, and it will automatically bring them to their desired email account which will be sent to HouseBuddy's email where we can view the support ticket. We believe this is important because with all mobile applications, there tends to be some sort of problem, and users should be allowed to tell us about their problems. This was intended to be made if we wanted to deploy our mobile application through the apple store or the google play store.

Framework/ Libraries:

At first if you look at all our submissions you can see that we tried making the project through android studio. We learned java and the mechanics on how to use XML. After making what we envisioned our mobile application to look like, we did not enjoy using android studio and went to a more efficient approach and used React Native. React Native allowed us to use a web browser for people who do not have an android, but also it also allowed us to not always deploying through an emulator to see what we were doing. It still is employed on android through a QR code, so we thought this was perfectly fine.

We are learning a new language and working with a workable user interface for the first time and learned as we go. We would like our mobile application to do a lot more than what we currently have, but just do not have the time to learn the skillset but for future implementation we have high hopes for our application.

Libraries are important in a mobile application, and we chose ones that we thought would serve the best for our application. Our most important library is having Expo. Expo allows us to initialize and serve our mobile application on Expo Go to open it with a QR code on either IOS or Android. Without this library, there would be no way to deploy our mobile application, but have it just run through a web browser which is not what this class is for.

Another library we included was react-native-tab-view. This allowed the tabs we have in our mobile application. We used react-native-gesture-handler to make it smooth when users need to pinch text, and rotation of the mobile application. We used react-native-safe-area-context to have our card borders appropriately to make the mobile application not have any issues with border sizing and padding.

For our seamless transition between pages, we included a library named react-navigation/native and react-navigation/stack. These two libraries allowed the screens to be switched on top of each other like a “stack”. Then bringing it back and forth without having any input lag on the device.

We used touchable opacity which is used to control clicking for the user. If the user is clicking on items, it makes it be able to look like its being clicked on. We used an onPress function to handle all items being clicked on, and with this it allowed certain things to be clicked on to bring it to the right place it is supposed to go to. It minimizes using functions but utilizes our onPress we are using.

Lastly, we used not a library, but we implemented Modal from the react-native library. With touchable Opacity, after the textbox fades in, we needed a way for it to fade out seamlessly when users enter their item. It acts like a popup window and having it visible when users click on the room and having them be able to change the room name and then having it not visible when they click the “save”, which when then show them the homepage with the list and their changed room name.

Problems We Encountered:

Some of the problems we have encountered so far was trying to have a smooth transition between pages on android studio. With this problem, it gave us more of a reason to switch to React Native. React Native was something more similar to use and it just felt more dated to use if we wanted to use something for our future jobs. With this switch we had to learn how to use react native in order to transfer all of our code that was implemented with android studio.

Another problem we encountered was on our homepage where we had to use cards to hold our room names and used a flatlist in order to show all the cards. Once a user clicked a card, it would then bring them to that card's page and enter in their items for that room. However, when we clicked the card, it was not navigating to the new page we wanted to bring them to. We spent hours trying to figure out how to navigate between pages in react native however, with all our efforts it still was not working. That's when we decided to have the list already shown for the user and they can just add onto their list easily from there. This allowed a more modern approach but a simple approach.

We followed a tutorial on YouTube that taught us how to use functions to call specific items using useState. For flatlist, it allowed us to call each of our flatlist cards in order, so the users can see the names of the rooms they have. The flatlist then called data, which was our card names like master bedroom, office, etc. which is set in an array. Then it will then render the item by calling the id of each of the data in the array, which then showed the users the room names. The problem we had was to make it seamless when users clicked on the cards in the flatlist. This problem was with our touchable opacity.

When we first tried to implement firebase, it didn't allow us to sign in but rather forced it to the be entered in right away which gave us a white screen. At first, we did not know what the problem was, but then we soon realized we were routing it wrong. We scrapped that code and created new code from the start to back track our mistake and figured out a more modern and cleaner look we wanted to do it.

Resources used as links:

https://reactnative.dev/docs/touchableopacity

https://docs.expo.dev/get-started/installation/

https://reactnavigation.org/docs/tab-based-navigation/

https://reactnative.dev/docs/navigation

https://reactnavigation.org/docs/stack-navigator/

https://reactnative.dev/docs/modal

https://www.youtube.com/watch?v=iMCM1NceGJY&t=347s

https://www.freelogodesign.org/

https://rnfirebase.io/