Host-It

Created By:
Michael Anthony Cabrera
Nida Hameed
Anthony LiFonti
Linette Maliakal

Table of Contents

List of Project Participants **3**

Anthony LiFonti 3

Nida Hameed 3

Linette Maliakal 3

Michael Anthony Cabrera 3

Abstract of the Project **5**

Project Narrative **6**

Body of Research **9**

Conducted Research 9

Results 10

React Native Research 15

Design Considerations and Specifications **16**

Mockup 18

Low-Fidelity Mockups 18

High-Fidelity Mock-ups 20

Summary of Research **23**

Research Survey 23

Design Choices and Mockups 23

Final Considerations 23

Appendix **25**

List of Project Participants

Anthony LiFonti

Roles: Developer (JavaScript, React Native), Design
Contributions:

- Developed the app navigation structure using React Navigation, include tab menu and nav stacks
- Integrated Firebase and designed app's dataflow
- Created Home Page's Card solution with Nida
- Designed High-Fidelity Mockups with Adobe XD
- Developed the Event Page and its layout
- Worked on Landing pages - minor layout and functionality additions
- Lead team's usage of GitHub with GitFlow style structure

Nida Hameed

Roles: Developer (JavaScript, React Native), Design
Contributions:

- Worked on Events List page and Landing page of application
- Discussed and provided feedback for design implementations and user experience
- Created Low-Fidelity Mock-ups
- Worked with group to maintain updates throughout with Trello
- Created Home Page's Card solution with Anthony

Linette Maliakal

Roles: Developer (Javascript, React Native), Design
Contributions:

- Worked on the Contacts page of the application
- Discussed and provided feedback for Design implementations and mockups
- Provided extra resources to learn React Native
- Testing of code with Travis CLI

Michael Anthony Cabrera

Roles: Developer (Javascript, React Native) Quality assurance
Contributions:

- Conducted research by creating survey and distributing among peers

- Developed some of the list functionality
- Debugging React-Native build
- Compiled documentation of group progress into final report

Abstract of the Project

We intend to develop an event-planning mobile application in which people can communicate plans for events and hash out ideas. This application would also serve as a direct alternative to Facebook's event planning functions. We previously worked on an application of the same intent using Cordova and had a Minimum Viable Product (MVP), but, in this independent study, we would like to further our work on it and this time use React Native.

Scope:
We aim to implement a "post-it style" list. This ties in with the branding of the application, "Host-It," in that we would style the to-do lists and task lists akin to a series of post it notes. The application will also be designed so that a user can log-in as well as customize and save events. This includes scheduling, sharing, and the coordinating of events. Time permitting, we will add categories and making these events available to other users that have accounts within the application. This would allow users to be invited to events, and grant them the ability to RSVP to events.

Target audience:
Our primary target audience is college students who are looking to schedule on-campus events for clubs and organizations. However, the application will be appropriate for most audiences. To achieve this, we aim to make the application accessible and simple. This will be achieved with a sleek design and user interface, as well as the integration of the sticky notes. This deliberate design choice will cement the simplicity of the application, as well as its purpose. During the research phase, we mainly distributed our survey among peers, thereby collecting data on our primary target audience.

Underlying technologies to be researched:
We will research React Native in order to develop this application. We do not have prior experience with the framework, so this study will offer us an opportunity to learn and gain experience in React Native. Seeing as our previous MVP was made in Cordova, this study will require a complete rewrite of code. We will implement Firebase data storage, Google sign-in and account creation with Email, and Google Maps API. We will also be using Adobe XD for our prototypes, and we will be using wireframes for the planning process.

Project Narrative

Our project originally started out with our idea of creating an Event Planning Mobile Application geared towards a younger college-age audience. Inspiration came from our usage of Facebook's Event feature on their web and mobile applications and not gaining the most out of what we would like for Event planning. The goal of making an application was to, in a way, "revamp" the Facebook Event feature. We wanted the application to make it easier to for people to communicate, plan their events, and hash out their ideas. Our initial application, developed in Spring 2019, was built with Cordova as the mobile development framework.

Over the summer, we decided as a group to take the application further and develop it using React Native as the mobile framework. We thought it would be useful for us to learn and make the flow of the application better. We wanted to implement a "post-it style" list, to may it a playful sort of engaging application. We also wanted the application to be something the user could login to and personalize their own events that can be saved and shared with other users. Some of the features we wanted to make available are the following:

- Scheduling
- Sharing
- Coordinating events
- Guest List organized by attendance (RSVPs)

We also envision using the following technologies to build our application (aside from React Native):

- Firebase (for data storage and O-Auth Sign-In through Google)
- Google Maps API
- Sketch or Adobe XD for initial prototype

After establishing our initial plan of what we envisioned the application to do, we went in depth and analyzed what pages would be necessary to create and how to go about designing them for the best user-flow for a mobile application. Our Home Screen would have the purpose of listing events that the logged in user is attending. There would be a hamburger menu that, when expanded, shows a search option to sift through the events, a link to the Contact List page, and a link to a page that lists the events the logged in user is hosting. The Contact List page shows all the contacts the logged in user has added from within the application, or any of the persons they may have interacted with through the application. The Event Contact List page is for all those invited to a specific event, organizing the persons into categories by who is a "Going," "Not Going," or "Invited" and the "Hosts." This page is accessible through the specified Event page.

The Event page shows all the details of an event and includes the interactive “Post-It” notes. It also includes a checklist of assignable items and a chat.

At this point in time, we were also working on deciding our color scheme and fonts to be used. We used resources such as coolors.co and font-pairing websites such as fontpair.co to help choose fonts that worked well together. We also realized we wanted to implement a feature where when an event is generated and a personal cover image is not picked by the host, then a default image would generate and populate the event based on the text in the title. For example, “Dinner with Friends” would include a generic dinner graphic. This was inspired by what Google does with their Calendar application.

We spent much time working on specific aspects of our UI. The Hi-Fi mockups included in this document are very close to what we aspire to have for our UX/UI. We altered the accent color various times to see what would work best with the application and the default darker theme of it. We based this research on what we had seen as a common theme for apps that use the dark theme.

We then moved forward to the development stage of the application. After setting up each of our individual environments, we started working towards having a minimum viable product. We made branches based on our features to get done in order to avoid conflicts with merging and divide up the work. The “landing” branch includes the pages that the app opens with if the user is not signed in (the Welcome screen, sign up, and reset password parts). The “home screen” branch has the screens and components related to the homepage, which shows the list of the user’s events. We also implemented react-navigation for movement between the pages and considered libraries such as Redux for state management.

Our workflow was planned out to be as follows:
Branches would be merged into the dev branch for the development stage of combining what we individually have been working on. A staging branch would be created so we could test the application before pushing the updated commits to be live. The master branch would be the live branch where a functionally working version of the application would always exist, a never broken version.

For the debug distribution, we'll be aiming to have all of the mocked-up functionalities available for testing, including WIP features. Any optimizations will be done to the release build. We plan on having all of the determined MVP features in the release build while continuing development to see what else we can add. Visual Studio has features that will help us manage these different configurations. Also we are looking into implementing the Travis CLI for easier tests and debugging.

Our project currently has a drawer menu for navigation and two navigation stacks to streamline the experience for users depending on their sign-in status. The Home page is becoming more alike the mockup's design, with new cards and monthly sections implemented with SectionList. Additionally, the cards now navigate to the event page with a passed event key prop for smooth database reference. Speaking of, the application is now connected to our database on Firebase and can retrieve remote data. The Event Page has received layout/visual improvements and Guest List page has been worked on as well.

By the end of the week, we are working to have an application with the following TODOs completed:

- Services/Firebase:
    - Implement saving of events through new event page
- Add support for Notes and Todos
- App structure/navigation
- Finalize Event Page UI design
- Improve sign-in experience/'landing pages”
- Implement "event cover" images
    - auto-generated “event covers” based on type of event if user does not use their own image (ex. a dinner event would show a graphic of a dinner setup on the card)
- post-it notes for notes to be left on the events as messages or tasks

Body of Research

Conducted Research

Using Google Forms, we conducted a brief study in order to help develop scope and determine the demand for a product like Host-It.

We used the following consent message to verify consent for research participation. This also ensured safe research practices. This message appeared at the beginning of the survey, conducted through google forms, and was followed by a checkbox for participants to fill if they consented to the study.

> Project Title: Host-It
> Researchers: Anthony LiFonti, Nida Hameed, Linette Maliakal, Michael Anthony Cabrera
> Faculty Sponsor: Nicholas Hayward.
>
> Introduction:
> You are being asked to take part in a research study being conducted by Anthony LiFonti, Nida Hameed, Linette Maliakal, Michael Anthony Cabrera for an independent research study under the supervision of Nicholas Hayward in the Department of Computer Science at Loyola University of Chicago.
>
> Purpose:
> The purpose of this study is to figure out how people use and benefit from the Facebook Events function, as well as how to develop an alternative to it.
>
> Procedures:
> If you agree to be in the study, you will be asked to: answer survey questions regarding your opinion regarding influencers and their regards to your spending habits. The survey will be recorded and reviewed.
>
> Risks/Benefits:
> There are no foreseeable risks involved in participating in this research beyond those experienced in everyday life.
>
> Confidentiality:

These surveys will be recorded and reviewed. The surveys will be stored in a password protected computer.

Voluntary Participation:
Participation in this study is voluntary. If you do not want to be in this study, you do not have to participate. Even if you decide to participate, you are free not to answer any question or to withdraw from participation at any time without penalty.

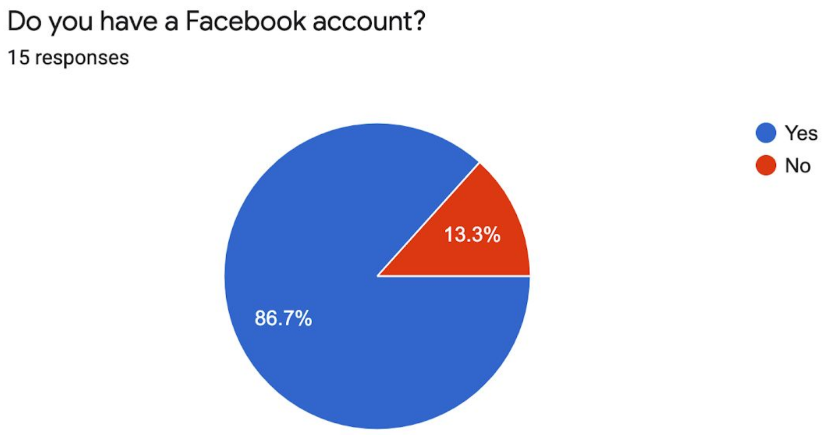
Contacts and Questions:
If you have questions about this research study, please feel free to contact Michael Anthony Cabrera at 312 358 8485 or mcabrera2@luc.edu.

Statement of Consent:
Your response below indicates that you have read the information provided above, have had an opportunity to ask questions, and agree to participate in this research study.

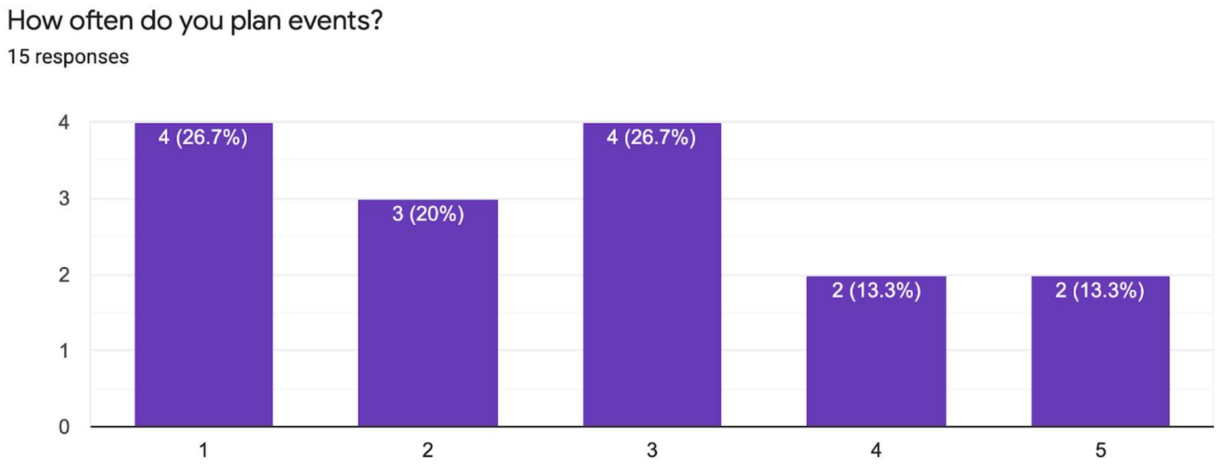
Results

In order to structure the study we conducted, we asked questions using a 1-5 scale, with qualifiers that differ depending on the question.

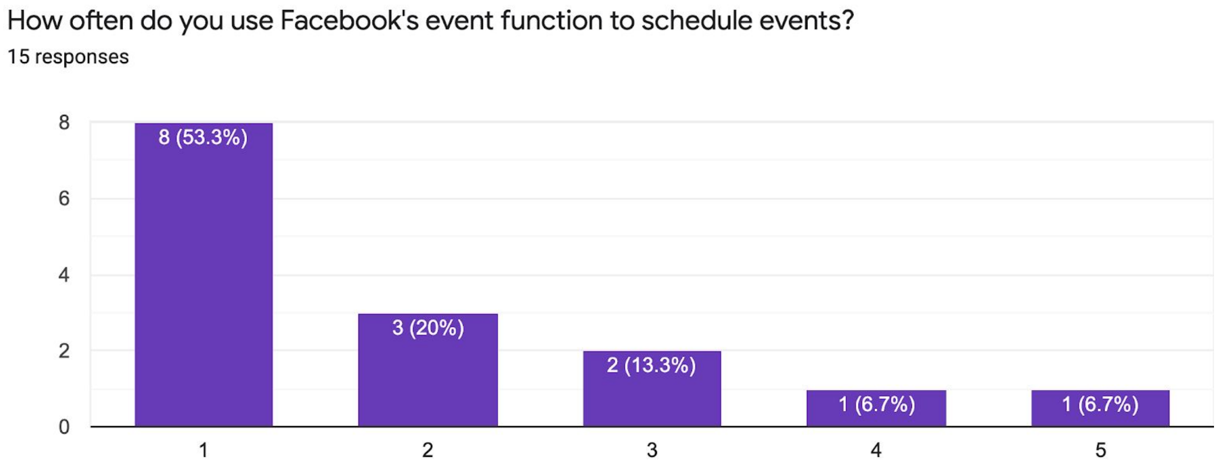
Chart 1



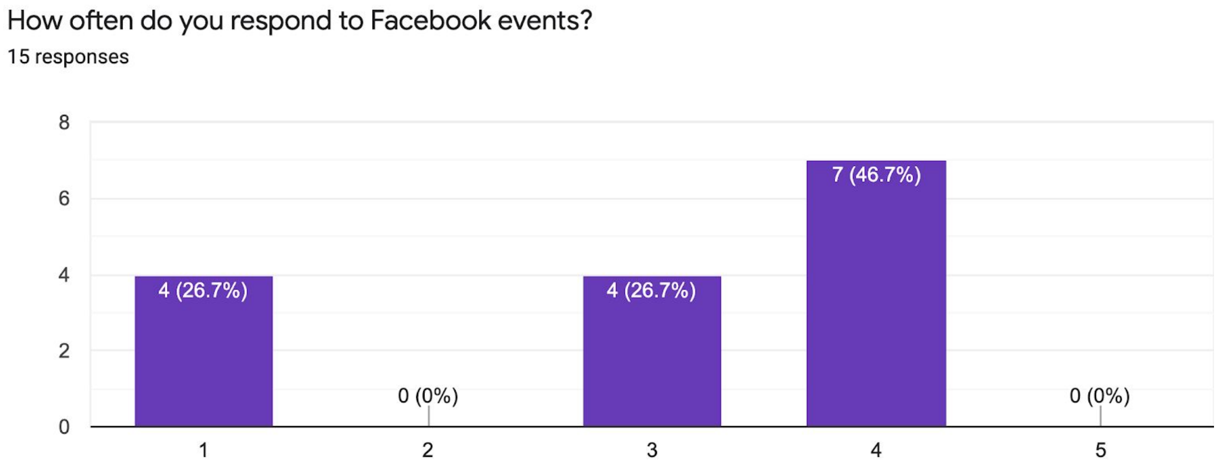
Graph 1 - 1 being not very often, and 5 being very often.



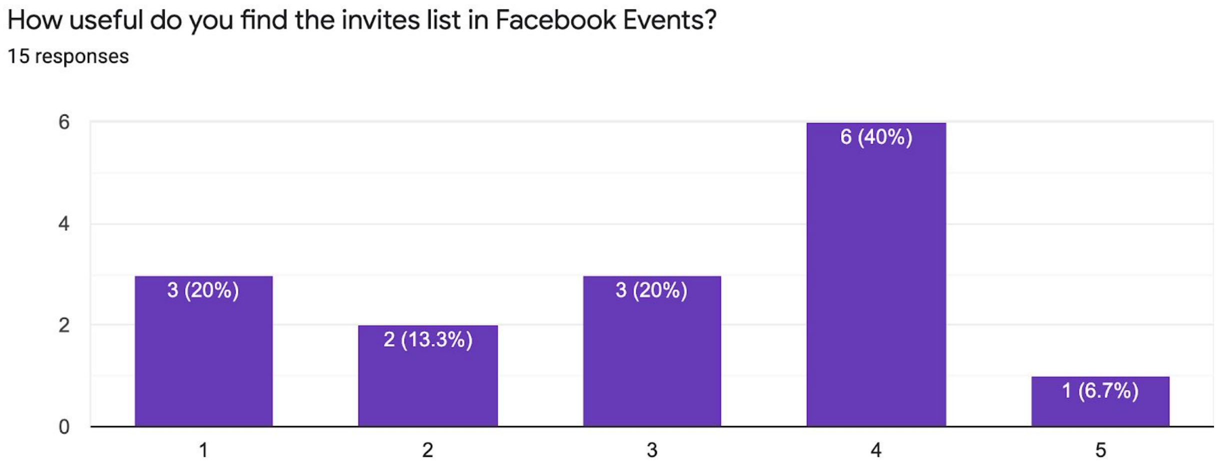
Graph 2 - 1 being not very often, and 5 being very often.



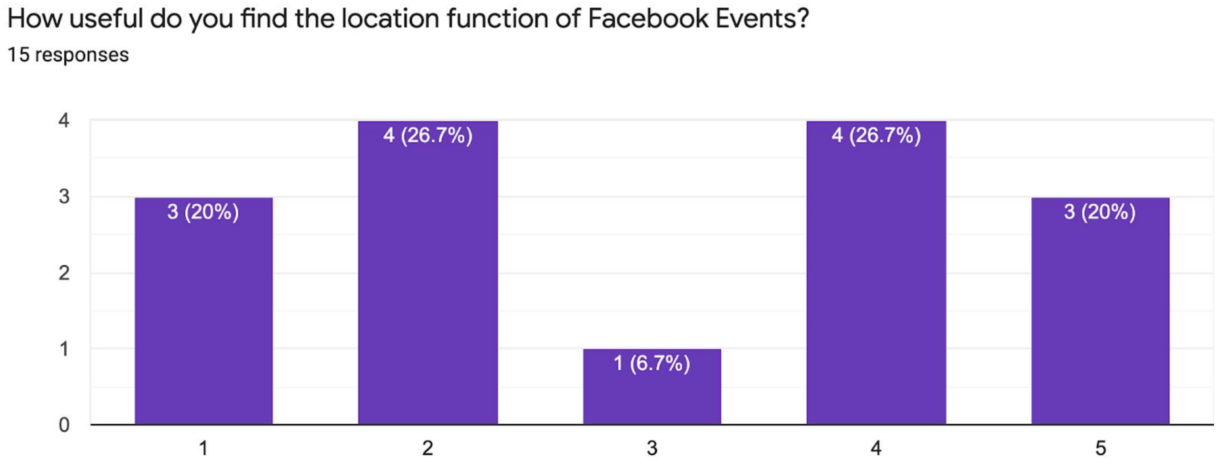
Graph 3 1 being not very often, and 5 being very often.



Graph 4 1 being not very useful, and 5 being very useful.



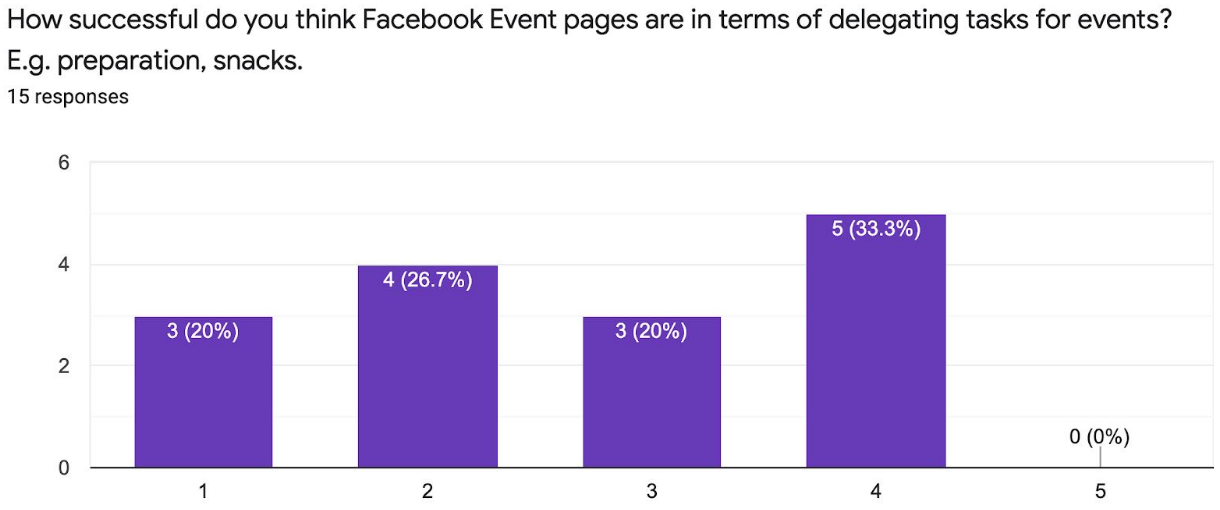
Graph 5 - 1 being not very useful, and 5 being very useful.



How useful do you find the location function of Facebook Events?

15 responses

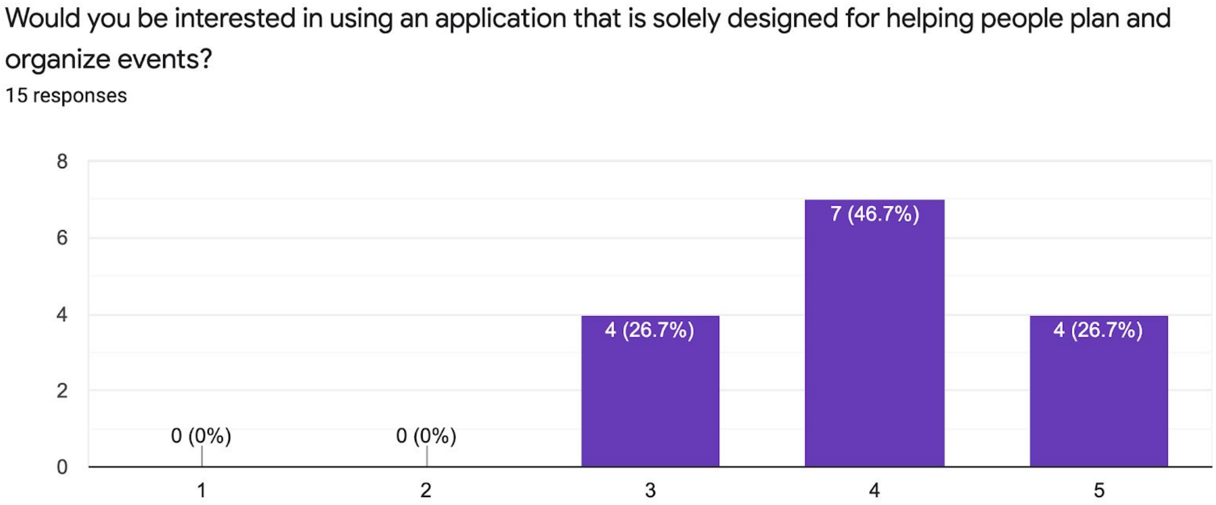
Graph 6 - 1 being not very successful, and 5 being very successful.



How successful do you think Facebook Event pages are in terms of delegating tasks for events? E.g. preparation, snacks.

15 responses

Graph 7 - 1 being not very interested, and 5 being very interested.



Would you be interested in using an application that is solely designed for helping people plan and organize events?

15 responses

React Native Research

We decided on using React Native to develop our application since it is commonly used in the industry and we wanted to gain experience with it. We developed our old prototype of Host-It using Cordova and React Native was completely new to all of us.

One of our primary resources was the variety of React Native notes provided by Dr. Hayward. Most aspects of the app's development (Getting started, learning state/props, navigation, etc) were covered by this resource. Whenever we needed further information, our other primary resource was the official React Native documentation. The guides and API doc helped us learn the finer details of React Native development. For example, if we needed to know all of the possible Props that FlatList supports, the official API was the best resource for that.

Additionally, the React Navigation docs helped us in a similar fashion. This library was a central part of our application's development so we frequented this resource often.

Another resource used during our research for React Native in general was the online EDX course -- taken over the summer by Linette and given to us by a previous Graduate Student.

Our project also required the use of a remote database, so we had to research with regards to React Native's best database options. After considering the most popular choices, provided by Dr. Hayward and our own brief personal research, we decide to use Firebase. This choice made was the most appropriate for our application for a few reasons; notably, Firebase's realtime database, powerful free tier, and our previous experience with it made Firebase the best choice.

However, using Firebase with React Native was not without its compromises. Integrating this service with React Native is a common hurdle for many developers. One common solution is to use a library known as React Native Firebase. We considered using this library, but wanted to make sure that any gained ease-of-integration was not outweighed by the added complexity and abstraction. Ultimately, we decided to use the stock Web SDK supplied by Firebase. There is a helpful chart on React Native Firebase's site that compares the features supported with the two different approaches.
(https://rnfirebase.io/docs/v5.x.x/getting-started#Supported-Firebase-Features). Our project was covered by the Web SDK features so we decided to go with that solution.

Design Considerations and Specifications

Through this mobile application, we wanted to create a mobile application that made a common task of planning easier for people. We wanted to create an application in which a user could plan and organize events all in one place without the hassle of being users of other platorns or connected to other things. The goal of this mobile application is to aid in event-planning for users of all ages.

A feature we looked really closely at was Facebook Events, where only Facebook users are able to create and view events. This application removes the barrier of being a Facebook user for those who simply need and event-planning feature. We conducted a survey to see how many people utilized Facebook events and would consider moving it off of the platform to a event-planning specific applications. The people surveyed indicated that they would be very interested in using a event-planning specific application, which is directly aligned with the purpose of this application.

Originally this project was created in Cordova and had a working interface, but with the decisions to rebuild it in React Native, we also decided to give the interface an update. We took our old mockups and re-evaluated them to build a more concise application that created a better user experience for the users. We added better navigation through the use of a side menu, also referred to as a "hamburger menu" that shows up if the user taps on the side instead of a navigation bar on the bottom. Using the side drawer allows us to maximize screen space for app content. Additionally, we also tried to keep all elements three taps or less away for a more concise experience for the application user.

The application begins with a landing page where they can either login or sign up. After logging in, the user is directed to the Events List page will list all the events that are currently on their calendar. Instead of having some sort of welcome page or something similar, we made the decision to have the Events List page be the page users are directed to after login. We wanted to add to the conciseness to the application and decided that removing a filler page of some sort in between would be the best way to create a positive and simplified user experience. This is in direct contrast to the Facebook Events feature that requires users to locate the tabs and then further navigate to specific events within the Events page to find specific events or create events. We made the creation of a new event very easily accessible by including an add button at the bottom corner of the screen.

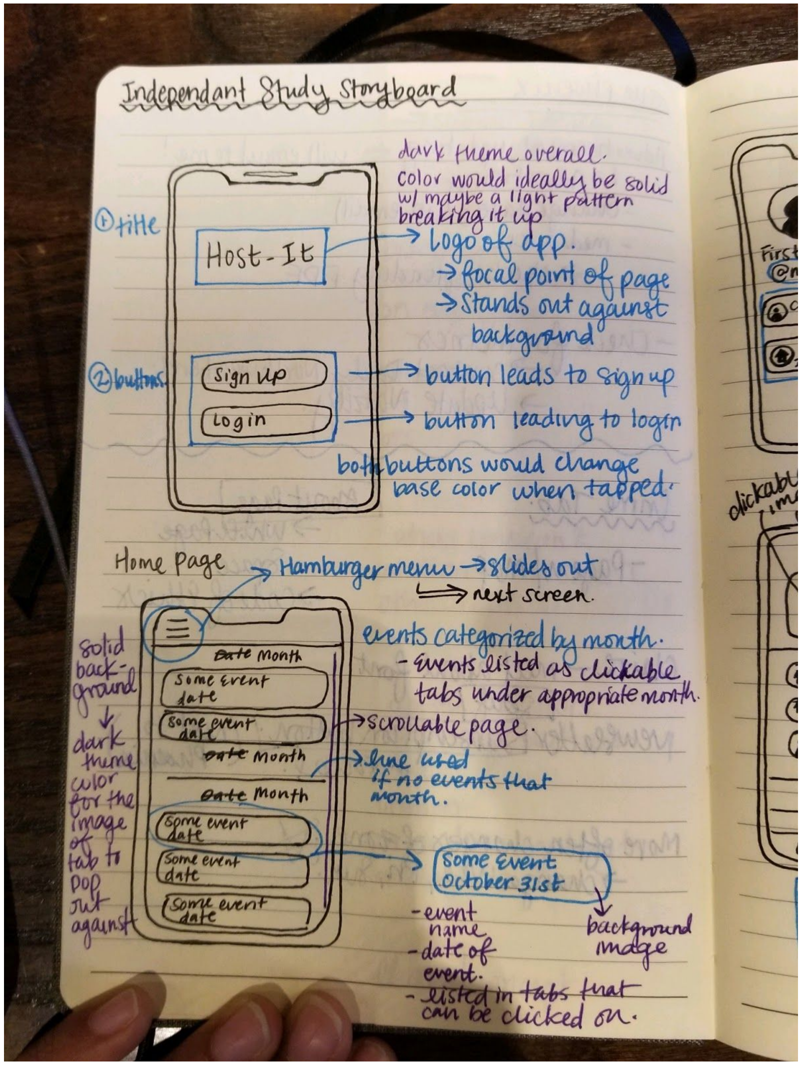
We decided to use a dark theme for our application because based on our research we found that there were several benefits to having a dark theme for the sake of the user. The dark mode aesthetic is becoming increasingly popular, but it has several advantages besides just the visuals. It is also better for eye strain and reducing eye fatigue. Furthermore, it is more accessible for those with visual impairments, migraines, and visual disorders. Dark themes can also be better for battery life of a device. Furthermore, we used resources such as coolors.co and font-pairing websites such as fontpair.co to make more informed decisions on colors and fonts.

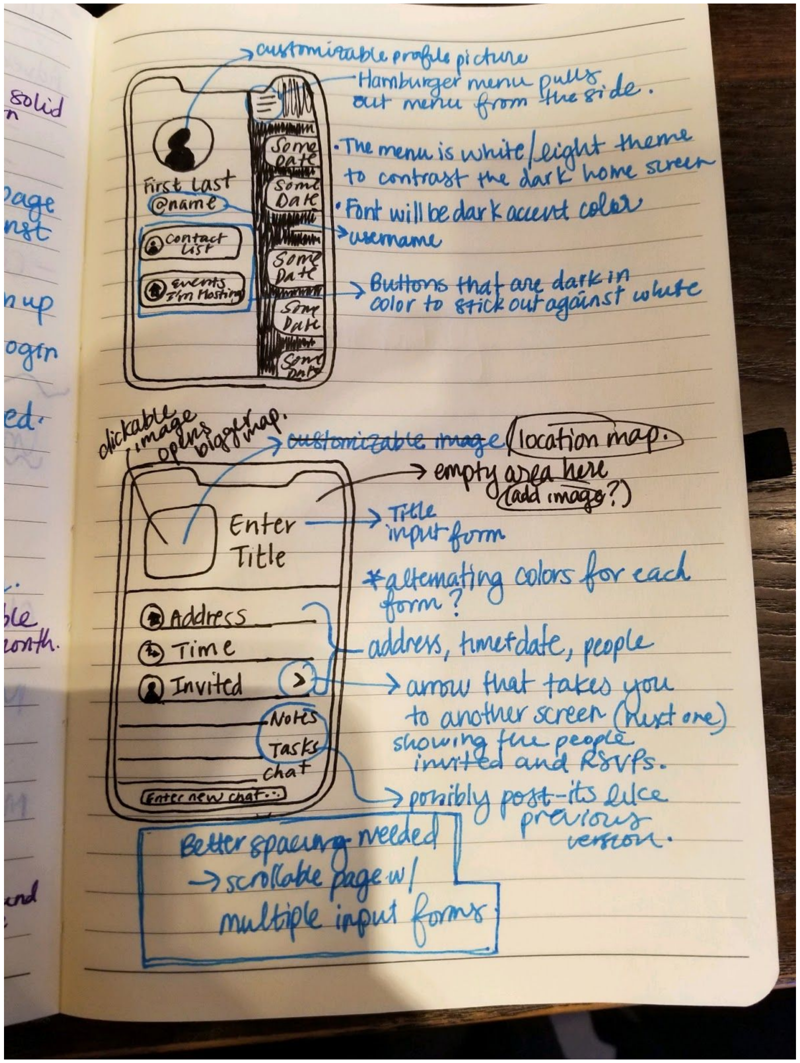
Mockup

We created several mockups throughout our entire process to help our aid our process. We began with taking our former work re-evaluating what could be done differently. Based on this we created low-fidelity mockups that reflected the features we wanted to add and the layout of our screens. Then, we created high-fidelity mock-ups to reflect our final decisions.

Low-Fidelity Mockups

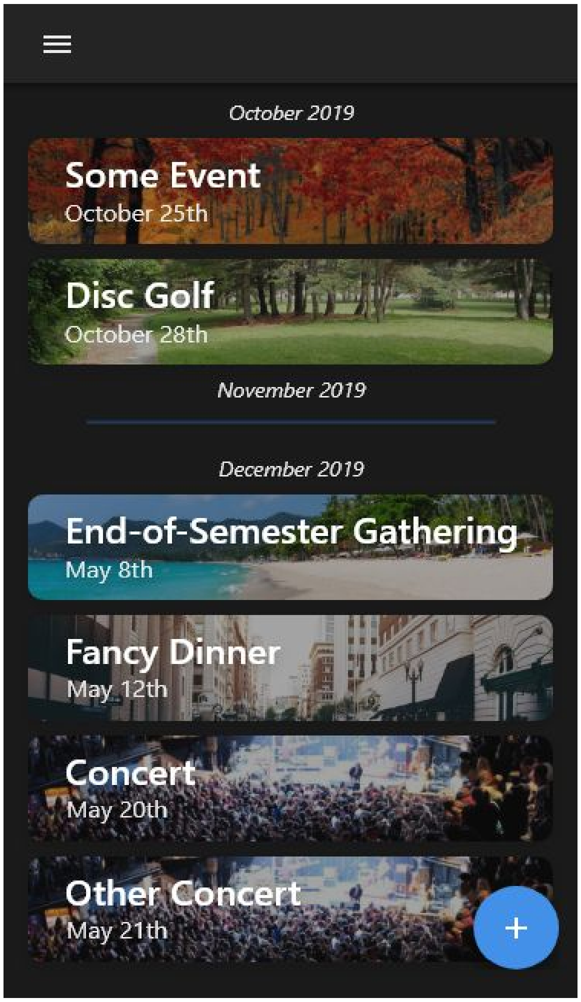
The low-fidelity mockups were done by hand and list out our thought process in the application design. It illustrates all the things we took into account when creating the interface to create a positive user experience. The low-fidelity mockups helped us streamline our ideas and thoughts before jumping into the high-fidelity mockups without much of a plan.

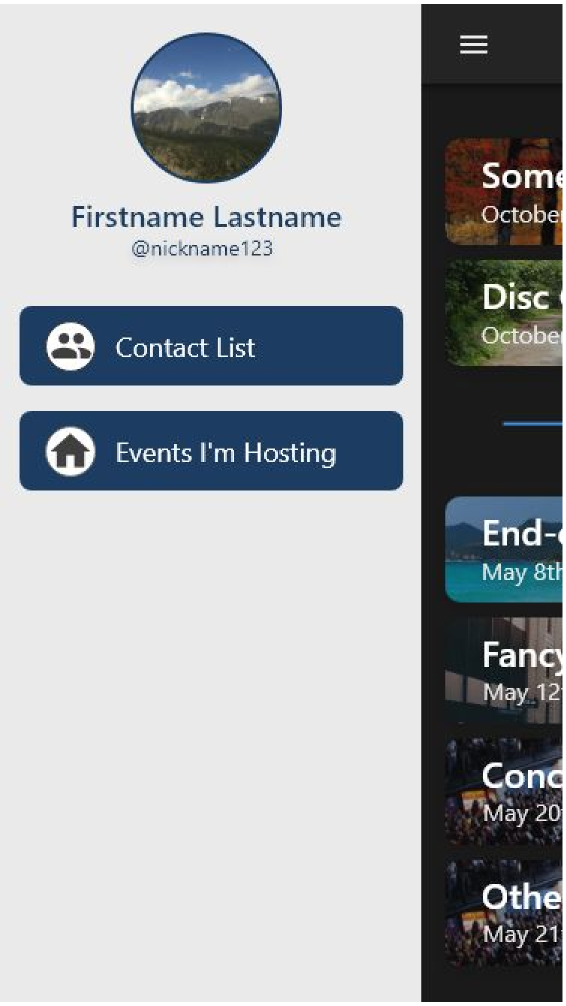

Independant Study Storyboard
dark theme overall.
color would ideally be solid w/ maybe a light pattern breaking it up
① title
Host-It
→ logo of app.
→ focal point of page
→ Stands out against background
② buttons.
Sign up
Login
→ button leads to sign up
→ button leading to login
both buttons would change base color when tapped.
Home Page
→ Hamburger menu → slides out
⟹ next screen.
~~Date~~ Month
Some event date
Some event date
~~Date~~ Month
~~Date~~ Month
Some event date
Some event date
Some event date
events categorized by month.
- events listed as clickable tabs under appropriate month.
→ scrollable page.
→ line used if no events that month.
Some Event October 31st
solid back-ground ↓ dark theme color for the image of tab to pop out against
- event name
- date of event.
- listed in tabs that can be clicked on.
background image

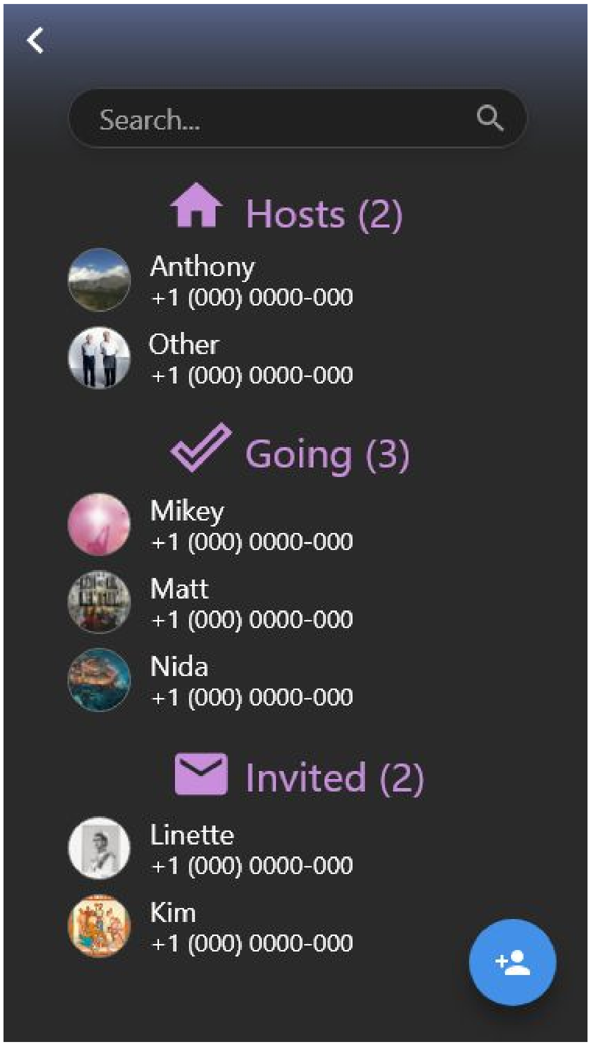

→ customizable profile picture
- Hamburger menu pulls out menu from the side.
First Last
@name
Contact List
events I'm Hosting
Some Date
Some Date
Some Date
Some Date
Some Date
· The menu is white/light theme to contrast the dark home screen
· Font will be dark accent color
→ username
→ Buttons that are dark in color to stick out against white
clickable image opens bigger map.
~~customizable image~~ (location map.)
→ empty area here (add image?)
Enter Title
→ Title input form
*alternating colors for each form?
Address
Time
Invited
Notes
Tasks
Chat
Enter new chat...
address, time/date, people
→ arrow that takes you to another screen (next one) showing the people invited and RSVPs.
→ possibly post-its like previous version.
Better spacing needed
→ scrollable page w/ multiple input forms

High-Fidelity Mock-ups

The high-fidelity mockups are a reflection of the design choices we made for this application. They illustrate the final product we hoped to achieve. These mockups were created using Adobe XD.







Search...

Hosts (2)

Anthony
+1 (000) 0000-000

Other
+1 (000) 0000-000

Going (3)

Mikey
+1 (000) 0000-000

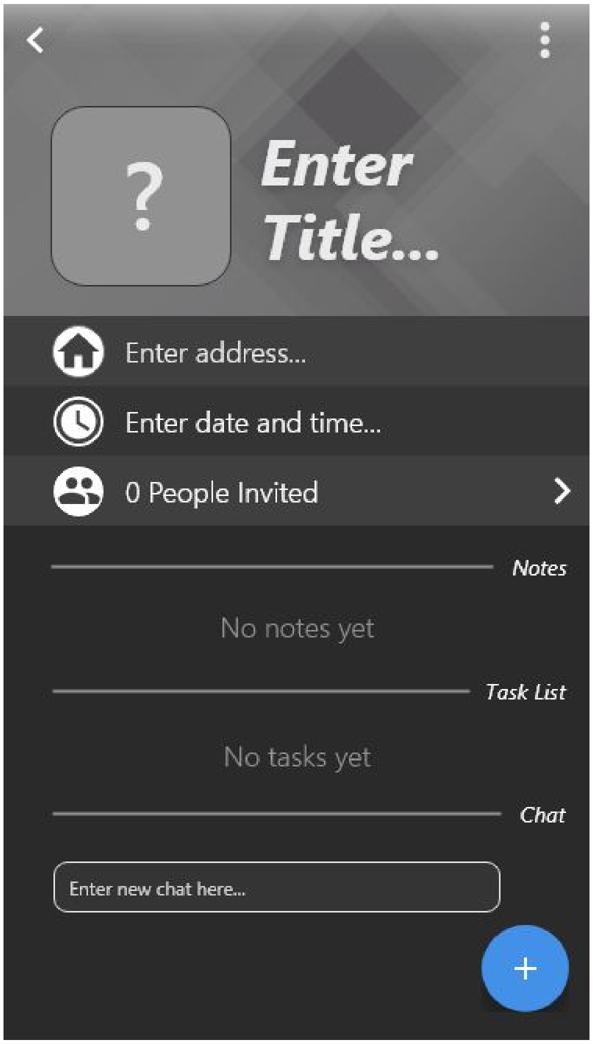
Matt
+1 (000) 0000-000

Nida
+1 (000) 0000-000

Invited (2)

Linette
+1 (000) 0000-000

Kim
+1 (000) 0000-000



Enter Title...

Enter address...

Enter date and time...

0 People Invited

Notes

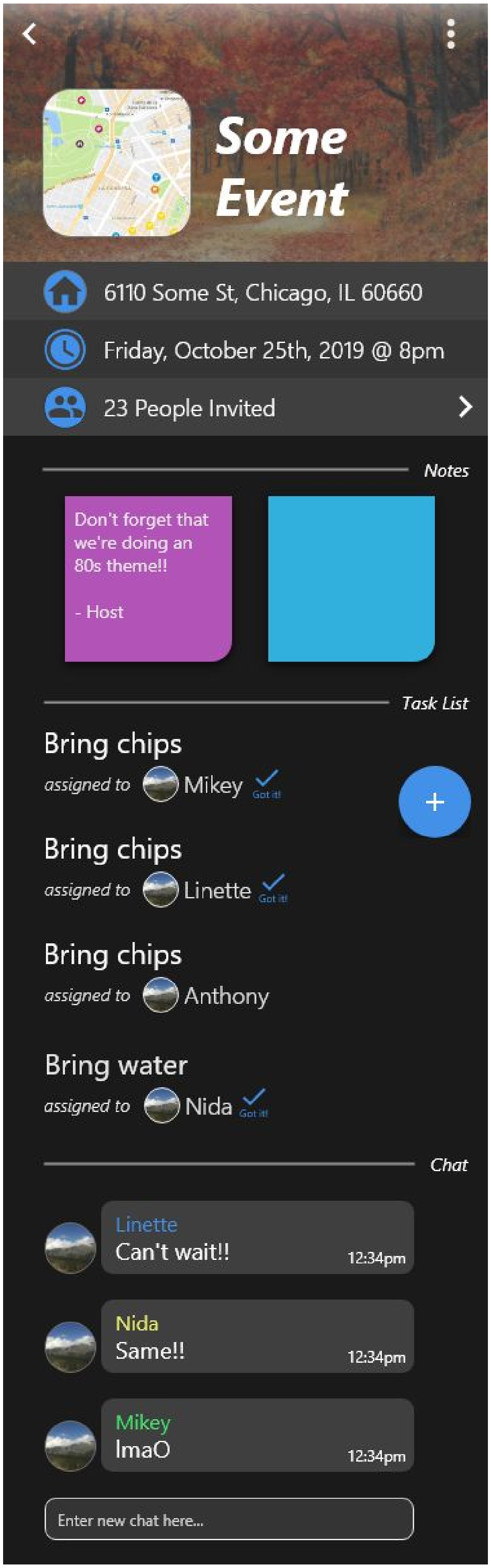
No notes yet

Task List

No tasks yet

Chat

Enter new chat here...



Some Event

6110 Some St, Chicago, IL 60660

Friday, October 25th, 2019 @ 8pm

23 People Invited

Notes

Don't forget that we're doing an 80s theme!!

- Host

Task List

Bring chips
assigned to Mikey ✓ Got it!

Bring chips
assigned to Linette ✓ Got it!

Bring chips
assigned to Anthony

Bring water
assigned to Nida ✓ Got it!

Chat

Linette
Can't wait!! 12:34pm

Nida
Same!! 12:34pm

Mikey
lmaO 12:34pm

Enter new chat here...

Summary of Research

Research Survey

From the research survey that we conducted, we saw that while, as noted in Chart 1, the majority of our target audience have Facebook accounts. However, although many of them plan or organize events, the majority of them tend to not use Facebook Event pages to do so (Graph 2). As noted by the responses we received in the last question that we posed to participants, our target audience would be interested in an application specifically dedicated to event planning as an alternative to Facebook Event pages (Graph 7).

With our questions about specific functions, we can see that participants enjoy Facebook's location system for events, but they find Facebook Event's available functions for delegating tasks is lacking. Therefore, with Host-It's focus on lists, in the form of post its, we would be able to remedy this issue.

Design Choices and Mockups

In our application design, we made several decisions to create a positive user experience. We wanted to simplify the navigation of more traditional event planning options like Facebook events. We researched best UI/UX practices to implement within our design and development. We ultimately made the decision to make changes such as implementing a drawer menu and dark theme to the application. Low-fidelity mockups were created to streamline the ideas we wanted to implement for this application and then we used Adobe XD to create our final high-fidelity mockups that illustrate our final goals.

Overall, we learned a lot from the design process and how to prioritize user needs to create an application that can set itself apart from the competition. We learned the importance of various levels of mock-ups in order to make more effective design choices and have a better development process.

Final Considerations

In conclusion, our app helps users plan and organize events by allowing them to schedule and share events with invite lists and to-do list functionalities. It was important to us to make the application user friendly, and to pare down functionality to make it pragmatic and accessible.

After conducting research, we were able to find the areas in which the application may succeed in comparison to Facebook's Event page implementation. We used React Native, and achieved the project by implementing modular design. We also implemented APIs such as the Google Maps API and Firebase data storage.

Appendix

Below are the resources we've used throughout the project timeline.

React Native

- https://facebook.github.io/react-native/docs/getting-started
- https://reactnavigation.org/docs/en/getting-started.html
- https://courses.edx.org/login?next=/courses/course-v1%3AHarvardX%2BCS50M%2BMobile/courseware/1c5f90b4bc6542bfb380da9fb4941147/3d6e95afbdf54261afdc6ded5d7f0526/
- https://rnfirebase.io/
- https://rnfirebase.io/docs/v5.x.x/getting-started#Supported-Firebase-Features
- https://medium.com/@rossbulat/introduction-to-react-navigation-and-navigators-in-react-native-3efcf7239a43

Design

- Coolors.co
- Fontpairs.co
- https://uxdesign.cc/the-dark-side-of-the-web-b5fb5cbb1b56