CSI 3150 Project Phase 2: To-Do List

Logan Howard

https://github.com/fris16ky/my-app

Table of Contents

Problem Statement:

The to-do list app is a mobile application that allows users to write down and keep track of their tasks in a linear, list-based format. Users can create, delete, and mark reminders as completed. Once all reminders are deleted, a wholesome image pops up congratulating the user on completing and removing all of their reminders. A to do list is highly practical, from creating shopping lists to writing down your daily tasks, it boasts a variety of uses. It truly can be used by any and everyone uniquely, whilst being simple enough for even young kids to use!

System Requirements:

This application takes care of the basic need of 'writing something down before you can forget it'. The simplistic and easy access application allows you to instantly pull up the app and write down any reminders that you think you will ever need. There is no time aspect nor completion order requirement, so you can truly go at your own pace. With a simple click of the button, you can delete any reminder as well, although there is no confirmation prompt, so be careful! Additionally, if you decide to cancel your reminder while in the prompt to set one your progress will be saved.

However, this application is rather basic; what you see is what you get. This application excels in simplicity, you can set, delete, and check off reminders. There are no other quirks or additions, so this will not be an 'entertaining' app per se. Think of it as an efficient app, like how you would use Notes on iPhone, or Google Keep for Pixel users. This app does not provide any back-end support, meaning no user accounts with reminders tied to login information. This app also does not have any 'importance degree', the layout of the list makes every reminder seem equally important, which can be detrimental in certain scenarios.

The To-Do List app's scope is endless. It can be used to keep track of grocery items, a diary, homework log, work log, or even to track bets placed. There are no limits such as time or even amount, the screen will 'stretch' boundlessly with however many reminders the user decides to input.

Conceptual Design and Functional Requirements:

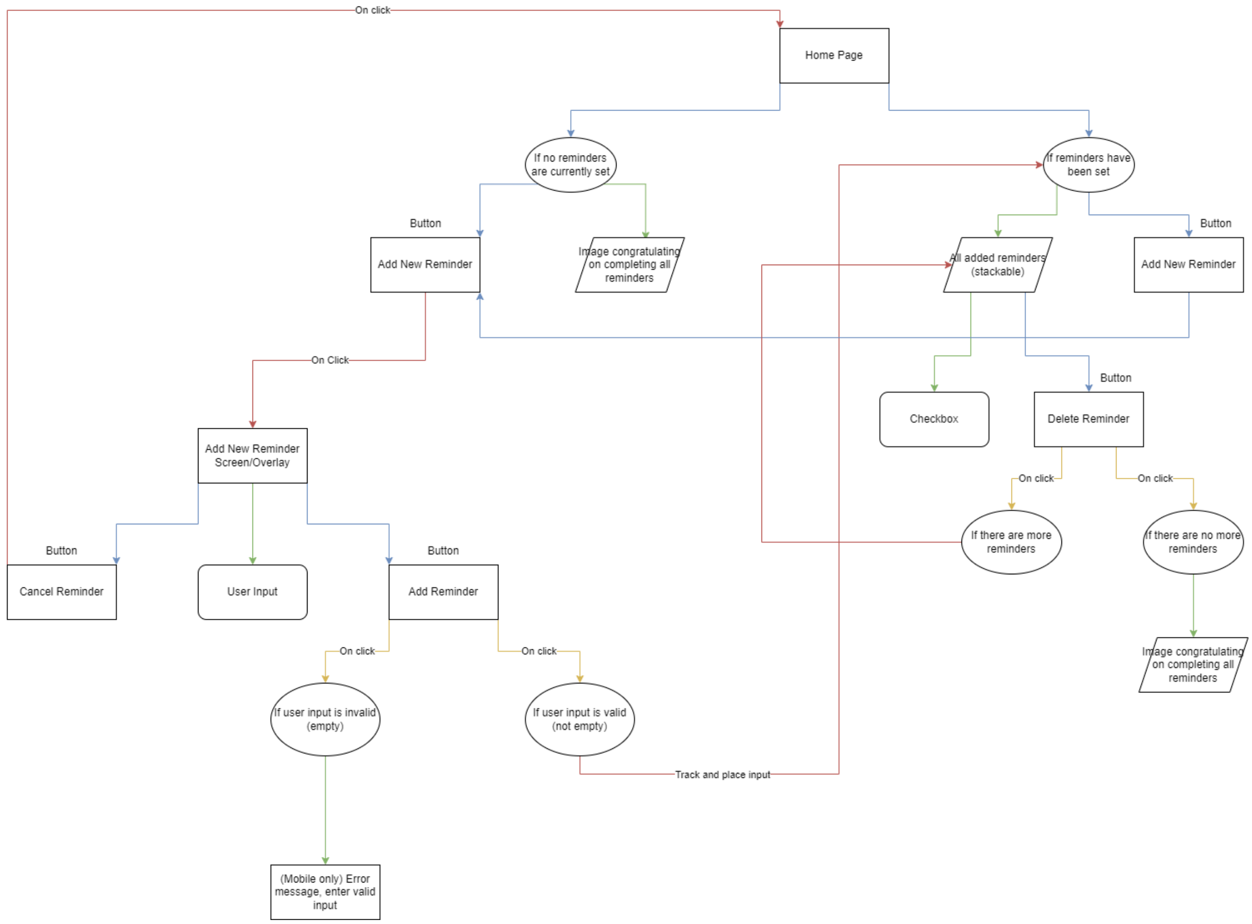
The functionalities that this app supports are the following:

- The user can click "Add New Reminder" to begin the process.
- The user can manually input their reminder into the input text box.
- The user can click cancel to stop the process, which will save whatever text they had put into the box for the next time they want to set a reminder.
- The user can click Add Reminder to display the reminder back on the home page.

Once the user has added a reminder to the Home Page:

- The user can view their set reminders in order.
- The user can use the Checkbox to denote whether they've finished a reminder.
- The user can click Delete to remove any reminder they choose.

Below is the .png form of my Flowchart Diagram. With only one additional screen, the flowchart is relatively simple, and a lot of the different aspects link back to one another. For the if statement check for adding a reminder, the "Track and place input" line is to be seen as an output, which then takes you to the "Reminders have been set" part of the home screen.


On click
Home Page
If no reminders are currently set
If reminders have been set
Button
Add New Reminder
Image congratulating on completing all reminders
All added reminders (stackable)
Button
Add New Reminder
On Click
Checkbox
Button
Delete Reminder
Add New Reminder Screen/Overlay
On click
On click
If there are more reminders
If there are no more reminders
Button
Cancel Reminder
User Input
Button
Add Reminder
Image congratulating on completing all reminders
On click
On click
If user input is invalid (empty)
If user input is valid (not empty)
Track and place input
(Mobile only) Error message, enter valid input

Technology used: React Native/Expo.

Future Scope:

My initial expectations for this project were to adapt the Uber-Clone-App we developed in class into a Tesla-themed app, with cheaper prices and differing navigation tools, such as to send you to SpaceX or Twitter. However, that and my following project (a weather app) fell apart, the latter due to an outdated API that I tried to utilize. Regarding this specific application, my initial expectations were to create a reminder tracking app that users could intuitively understand and navigate with ease. I wanted to simplify and dumb-down my web application from Phase 1.

From this expectation, I believe I accomplished everything I wanted to achieve. The mobile application is simple, yet effective. It is user friendly and provides every need that I think a consumer would require. In theory, I would've liked to add more to the app, whether it be something fun like the Puppy Slideshow from phase 1, or another application in one, I do not know.

As for future improvements, I have a few ideas. I would like to scale the Checkbox and Delete button, as when long (1+ line) reminders are inputted, they seem to stay at the top. Additionally, I wanted to figure out the different checks and styling that exist for mobile and web apps. Similar to how we used the hamburger menu and detected how big of a screen a user is using and adjusted. This could aid in an app that looks good in both web and mobile, as I had to make a few edits (from "Delete Reminder" to "Delete") due to problems I faced developing the app. I envisioned a Check All and Delete All button too, to make the app easier for the users. Moreover, a more unique or appealing design to attract more users could be added in the future, moving away from the basic form that it is in. Similarly, one final thing I'd like to add would be animations; for example, changing what it looks like when the user clicks on the CheckBox - from a check to a party popper? Or possibly a celebration when the user checks all boxes, or something of the like.

Challenges Faced:

In addition to inexperience, I dealt with a few technological issues. I would follow our examples in class and install and import react-native elements, but at times they would randomly break. For example, I ran into a fair amount of Chokidar error codes, and vulnerabilities that npm audit fix --force couldn't solve. Additionally, despite having sound and correct code, sometimes nothing would appear at all when I ran the yarn web. As I mentioned earlier, I also went through 3 projects as well, with this being my third. The weather API that I was planning on using, metaweather, had shut down, which I did not realize until I had created and brainstormed my project ideas already.

Overall, I experienced a lot of errors that came from me not fully understanding react-native in its entirety. As I began developing different apps, my knowledge improved drastically, but early on I was extremely frustrated and felt cornered. One final challenge I faced was the Checkboxes. My internet was out when we went through them in class, and I wasn't able to catch up on the lecture due to other projects and studying. The tutorial I based my app on used the old CheckBox import from react-native, which no longer worked. As I tried to devise a solution, I found out that it was replaced by importing from @react-native-community/Checkbox, so I tried and failed to implement that. I finally remembered that there was a lecture/GitHub file in which we went through checkboxes, and I realized that we needed to import from expo since I was developing an expo app.

Estimate of effort:

I started brainstorming and thinking of ideas starting around mid-to-late November.

Throughout all of November (let's say 20th-30th), I put in around 15-20 hours. I brainstormed different ideas and created around 10 different expo apps as I switched from working with my desktop to my laptop.

In the first week of December, I put in around 5 hours, finalizing my ideas, and creating now-deleted GitHub repositories and expo apps.

In the second week of December (up until the due date), I put in around 20-30 hours, a lot of it researching different apps and a lot of my error codes. A few additional hours came from making this project report and the presentation + demo video.