

# Project Documentation

## Project Collaboration

Once we did the project proposal that was due on Monday, our group lost communication and maybe motivation. Alas, I did not have that stop me. I took it in my best interest to complete the project myself. So my project was completed by Akash Dansinghani. It was somewhat of a difficult task but was able to be completed. If I had to rate myself, I would rate myself between a 8-9 since I started kind of late because of somethings that transpired but I did what I could with the time I had.

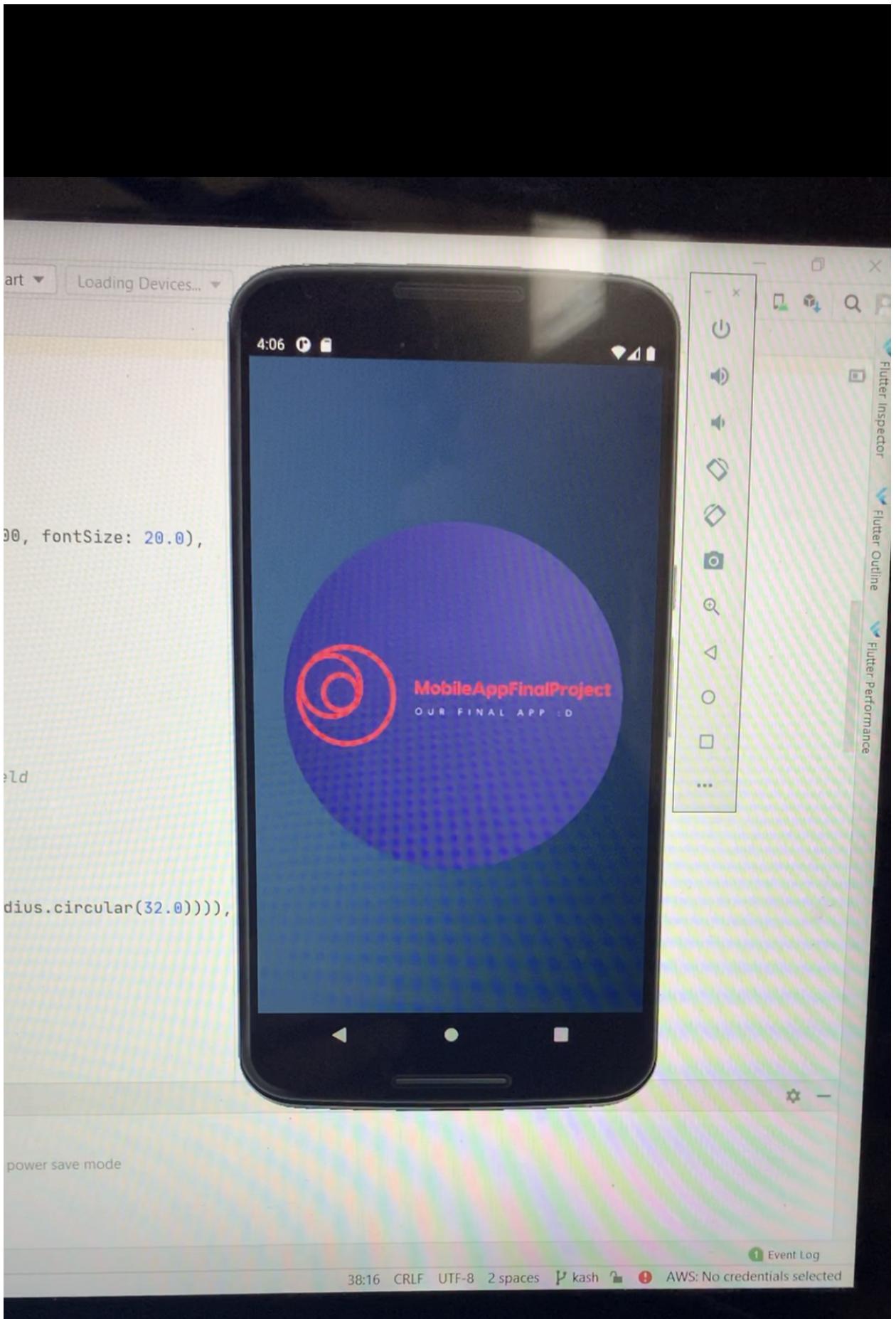
## Project Summary

Our group is creating an Inventory Management app in React Native. The goal of the project is to have an application where a business could track and manage their inventory efficiently and have everything conveniently located in one place. An administrator user will be put in place who can manage the inventory and manipulate items when needed. There will be users without the same administrative rights which are used for employees to track inventory. The purpose of this is for security and to make sure the employee is not able to fabricate or falsify the inventory. The plan is to get rid of manually tracking inventory and being able to track it all through an application. We will have a feature where you can scan each barcode and it will fill in the information of the product using the UPC barcode.

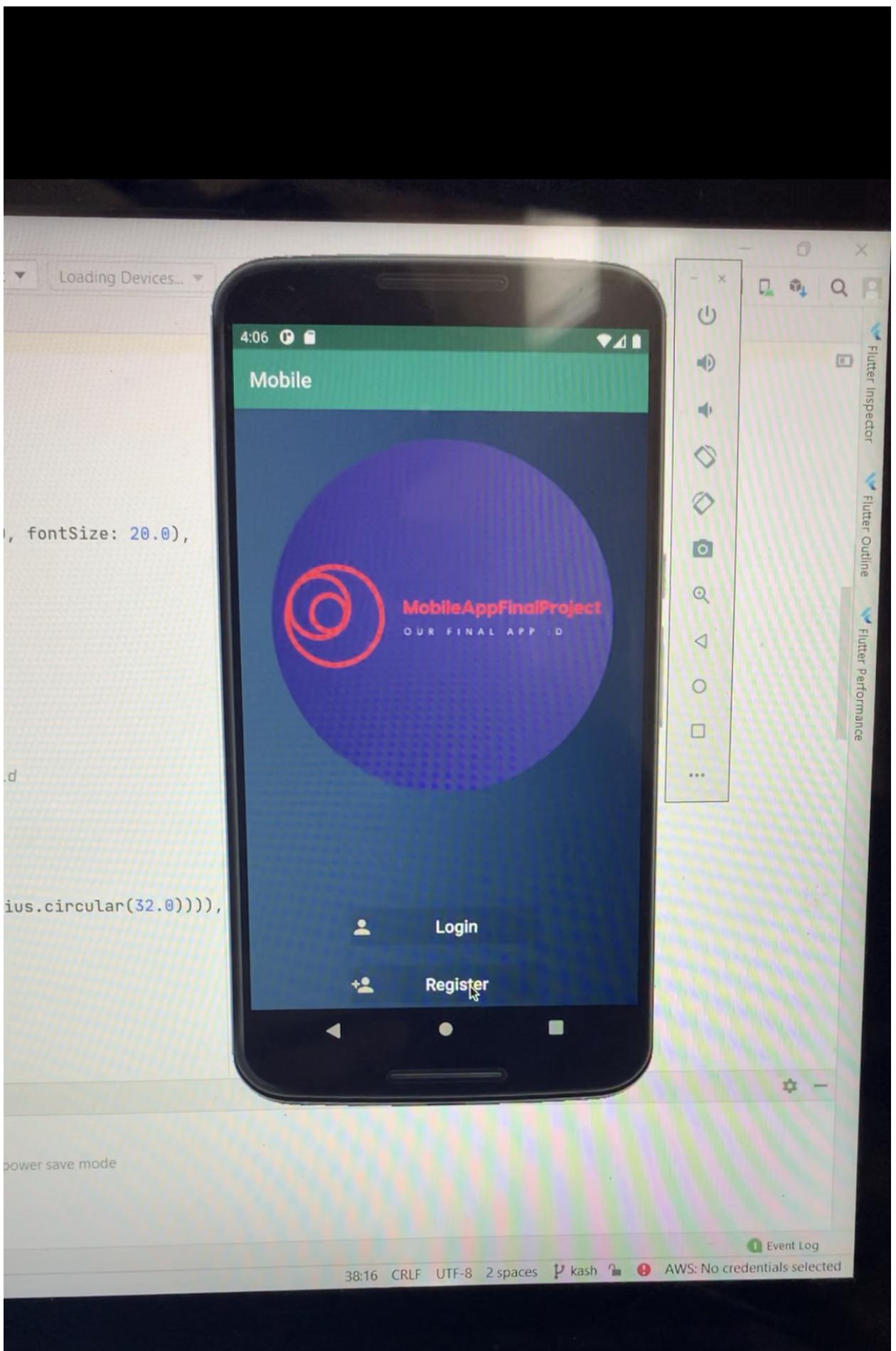
## Number and Name of unique screens in Project

We will use a total of five screens. Each of the screens are listed below:

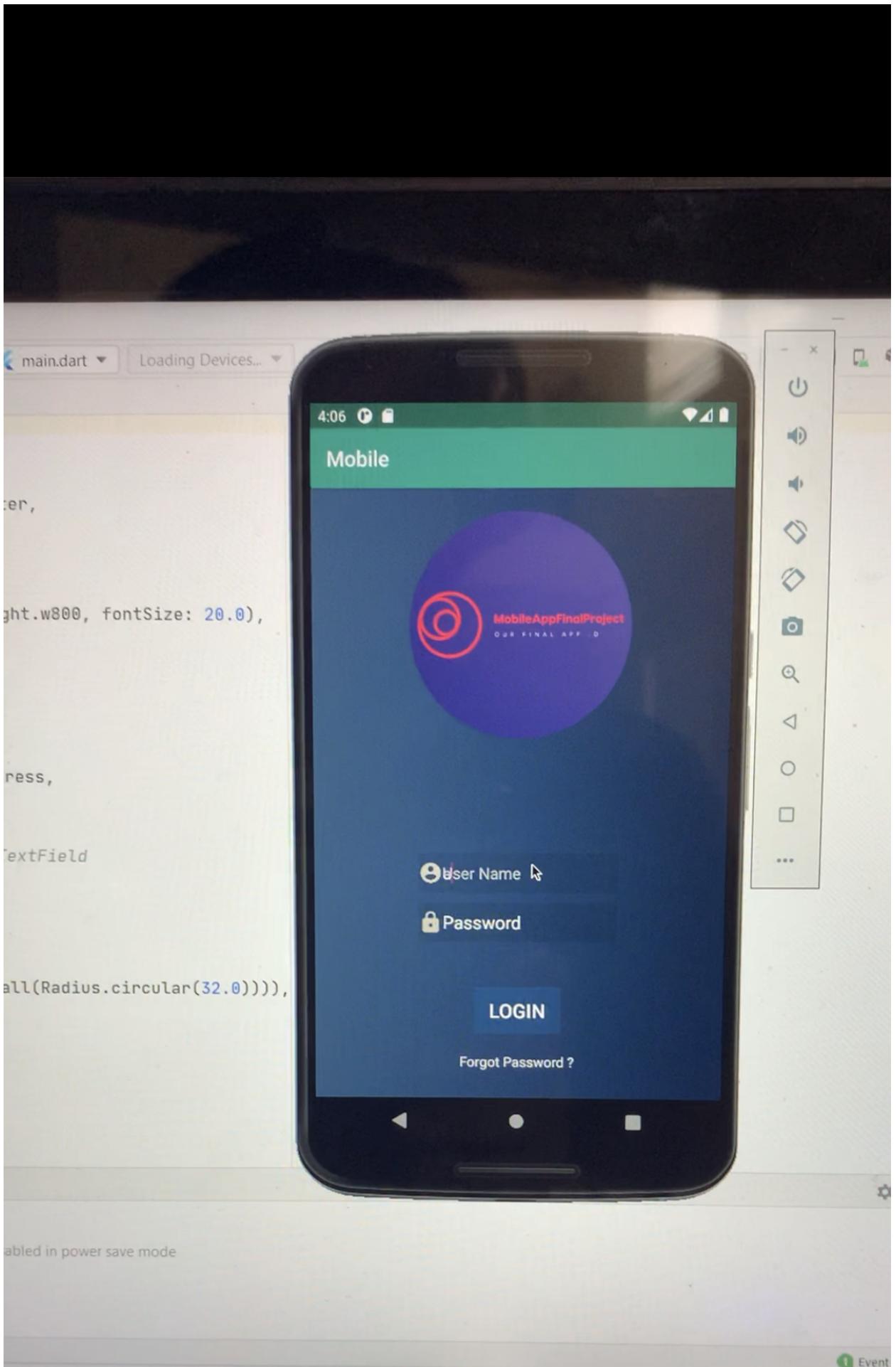
- Splash Page



- Home Page

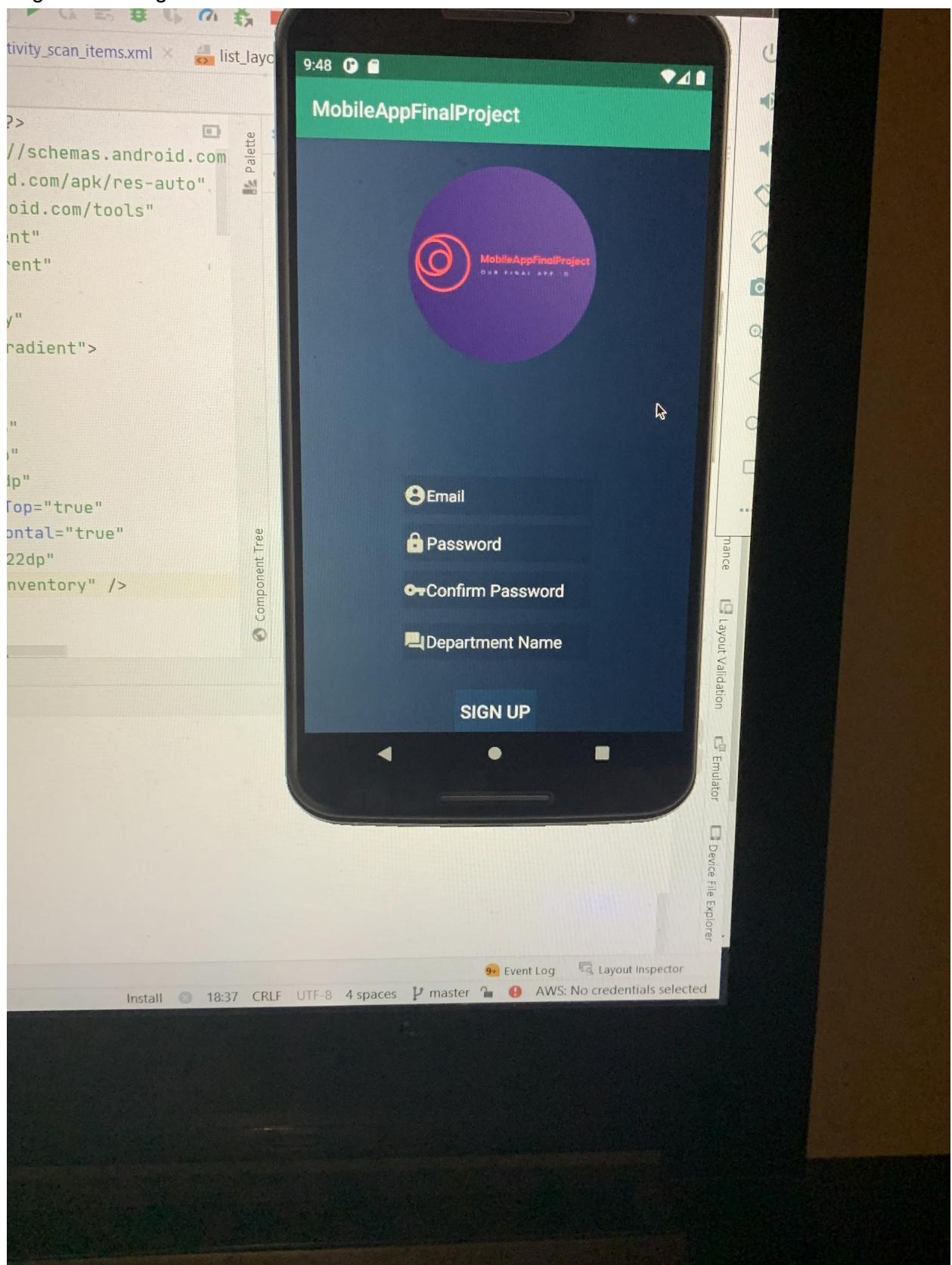


- Login

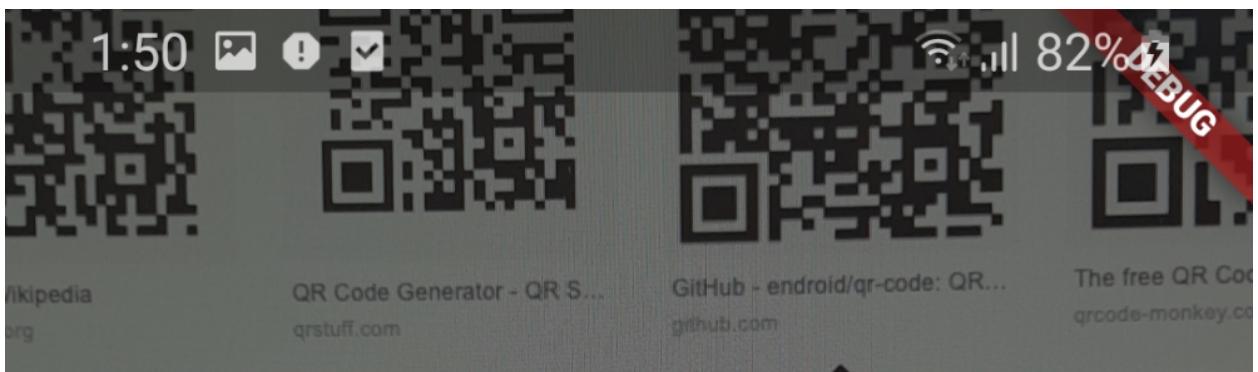


abled in power save mode

- Registration Page

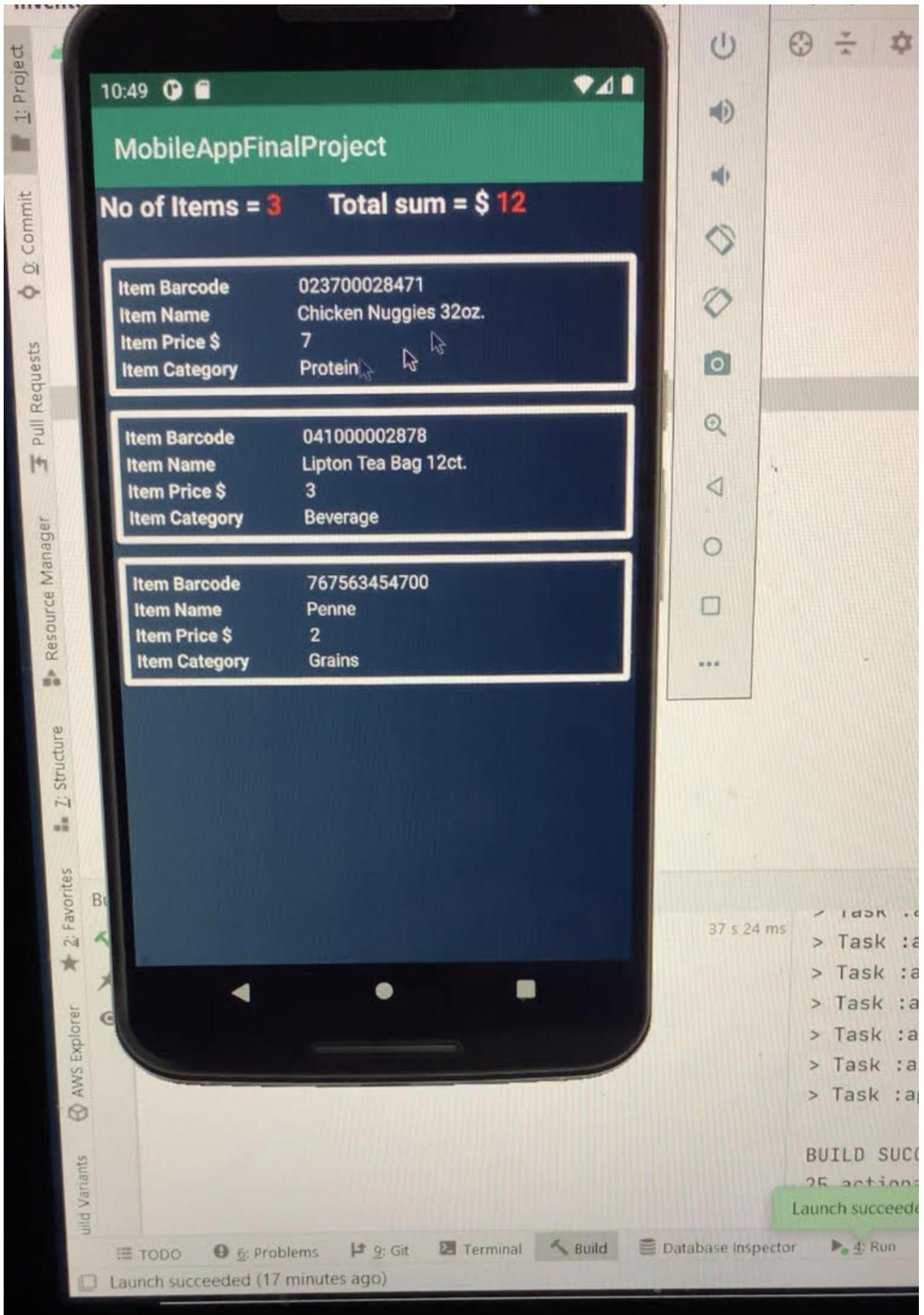


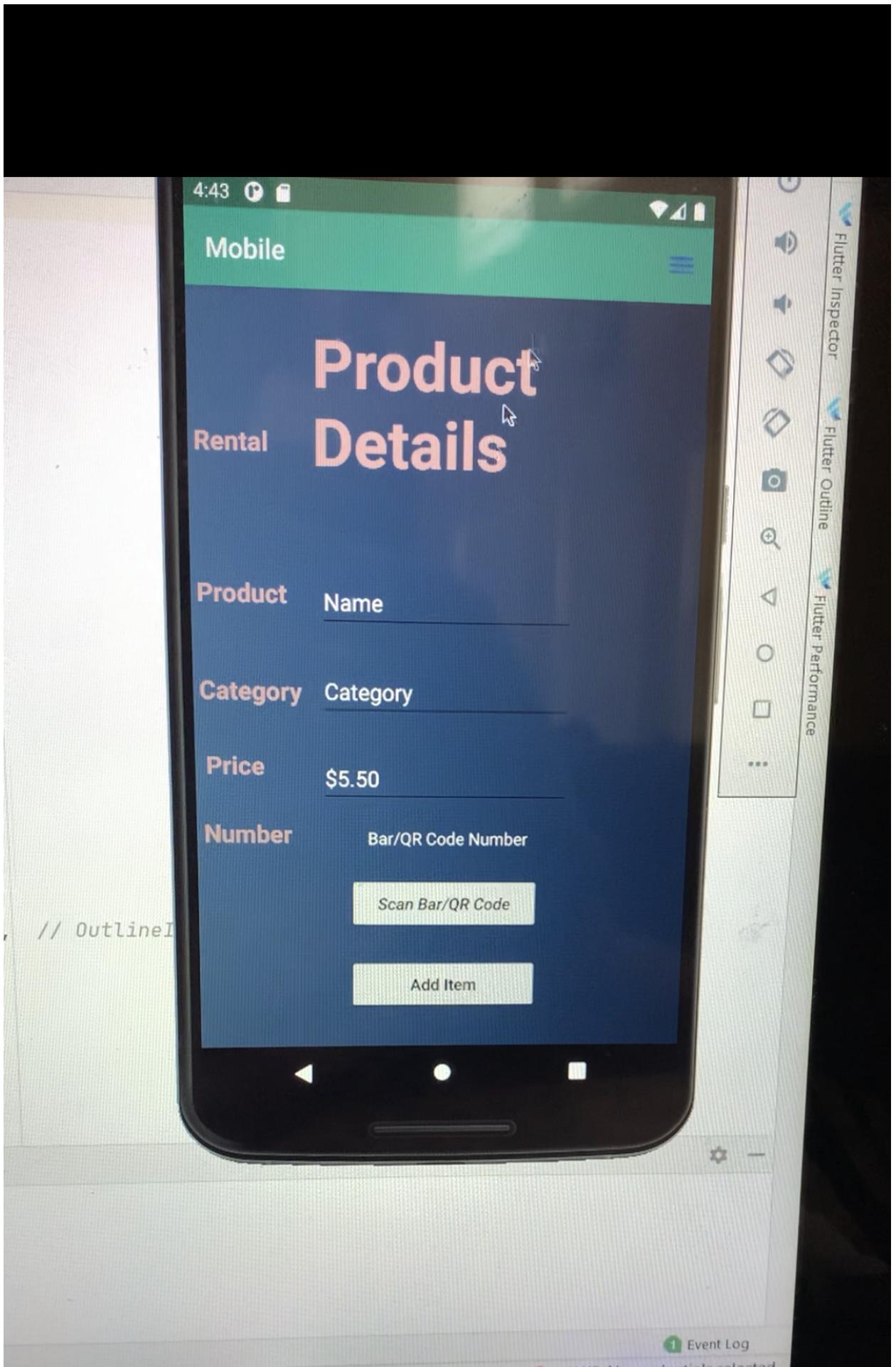
- Barcode Scanner



SAMSUNG

- Inventory/Admin

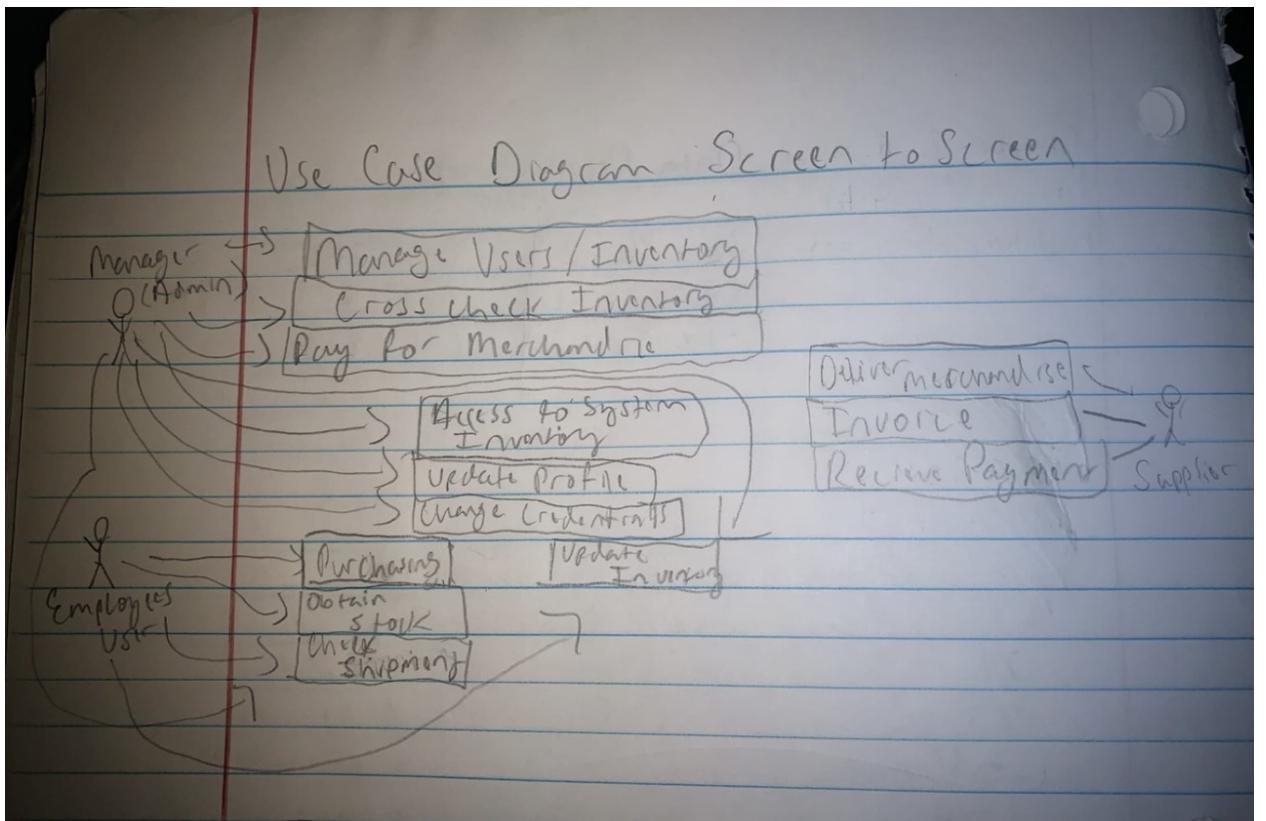




- Settings - Logout Option

## Use for transitioning from screen to screen

- Slide Menu - We will have a slide menu which is usually standard in apps where we will have the name of the user, inventory, Barcode Scanner (or we will just have that in the inventory section) and settings. I will plan on using a NavDrawer to complete this. The Navigator operation will help.
- For the transitions between pages or individual items in the inventory I will be using a page\_route and page\_transition for the actual effects like FadeTransition and SlideTransition. I may look into public libraries like flutter gems to make the app look better and have a nicer and finer touch.



## Use for each screen

- Login Page - plan on using email and password only. No need for Google Sign In or any other third party platform because everything will be done locally we will not need personal information of employees.

The screenshot shows the Firebase Authentication console for the 'Inventory-Management-App'. The left sidebar contains navigation icons for Home, Settings, Users, and other services like Firestore and Storage. The main header says 'Authentication' and has tabs for 'Users', 'Sign-in method', 'Templates', and 'Usage'. A search bar at the top allows searching by email address, phone number, or user UID. Below it is a table listing four users:

Identifier	Providers	Created	Signed In	User UID
employee3@yahoo.com	Email	Jul 28, 2021		tKrmRv5AeaeYHiXSoG4mGfQu111
employee2@aol.com	Email	Jul 28, 2021		pA1wso3eHNX1zNJ2NvyN92Kop...
employee1@gmail.com	Email	Jul 28, 2021		qsASfICxx9Njypj08rMs8zUIYr2
akashd123@aol.com	Email	Jul 28, 2021		SVONR7:Dl4ZGuDycH71e0VFQpZ...

At the bottom right of the table, there are buttons for 'Rows per page' (set to 50), '1 - 4 of 4', and navigation arrows.

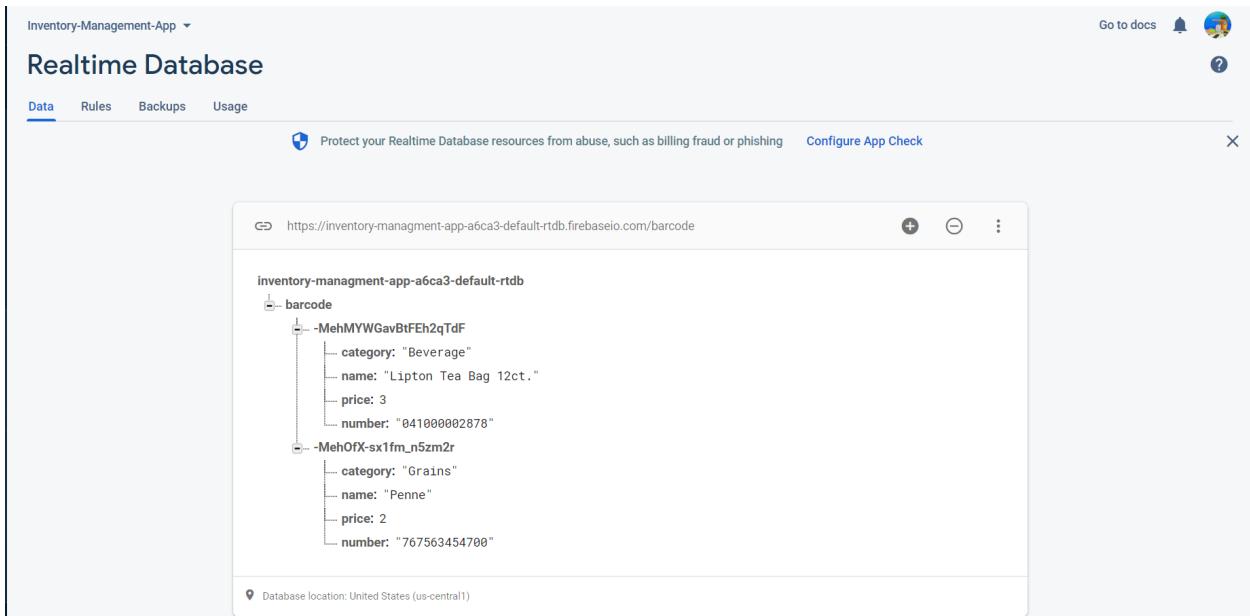
- Registration Page - Username which will be the first letter of first name, last name, and employee ID. Passwords will be based upon the user.
- Barcode Scanner - A barcode scanner will pull up so the user can scan the product and the ID will be pulled from the scanner. Using an API it should be able to scan any type of barcode.

The screenshot shows the Firebase Realtime Database console for the 'Inventory-Management-App'. The left sidebar has 'Data' selected. At the top, there's a warning about protecting against abuse and a link to 'Configure App Check'. The main area displays a hierarchical database structure under the path 'https://inventory-management-app-a6ca3-default.firebaseio.com/barcode':

```

inventory-management-app-a6ca3-default-rtdb
  └── barcode
      ├── -MehKqCyl2ZZIMicC9PP
      |   ├── category: "Protein"
      |   ├── name: "Chicken Nuggies 32oz."
      |   ├── price: 7
      |   └── number: "023700028471"
      ├── -MehMYWGavBtFEh2qTdF
      |   ├── category: "Beverage"
      |   ├── name: "Lipton Tea Bag 12ct."
      |   ├── price: 3
      |   └── number: "041000002878"
      └── -MehOfx-sx1fm_n5zm2r
          ├── category: "Grains"
          ├── name: "Penne"
          ├── price: 2
          └── number: "767563454700"
  
```

A note at the bottom indicates the 'Database location: United States (us-central1)'.

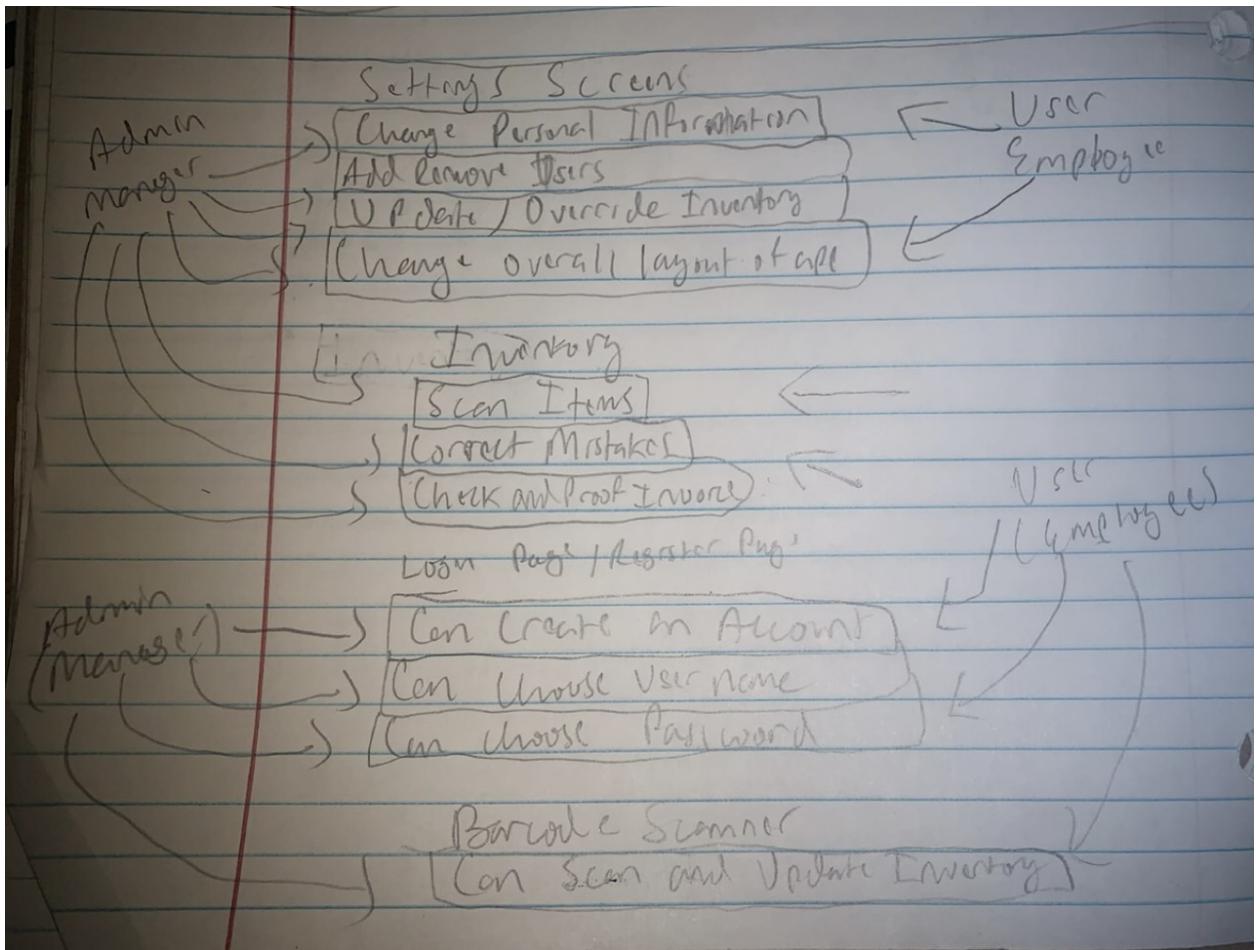


The screenshot shows the Firebase Realtime Database interface for the 'barcode' node. The database structure is as follows:

```
inventory-management-app-a6ca3-default-rtdb
  +-- barcode
      +-- -MehMYWGavBtFEh2qTdf
          |   +-- category: "Beverage"
          |   +-- name: "Lipton Tea Bag 12ct."
          |   +-- price: 3
          |   +-- number: "041000002878"
      +-- -MehOfX-sx1fm_n5zm2r
          |   +-- category: "Grains"
          |   +-- name: "Penne"
          |   +-- price: 2
          |   +-- number: "767563454700"
```

At the bottom of the interface, there is a note: "Database location: United States (us-central1)".

- Inventory Management for Users (Employees/Administrators) - All inventory will be stored in one place. We may create categories/departments for this but currently we will make sure we have a page for solely inventory. We are deciding to
- Settings - Change any information on the application like text size or password. I plan on having a couple other configurations for the administrator. Maybe add and remove users and manipulate the inventory.



## Proposed APIs you will integrate into your app

- Firebase Authentication
- Firebase Data Storage
- Firebase Database
- REST - specifically Zebra Savanna Barcode Intelligence which is based off of REST v2.0. This will be used to scan barcodes and input it into our database.
- Barcode Lookup API - this API has more support

## Summary of What is Done for the Project

- Designed the UI/UX in Figma.
- Connecting the Firebase API and Login Screen is completed.
- Users are setup
- Working on getting the database together but it is connected to Firebase Database

- We have currently planned out roles for each person and a timeline when we should have everything done
- We have a template put together for the application and research done

I understand that my grade is contingent on the amount of work I put in the project and my participation. If someone or multiple group members come forward and say I have not performed well enough to be considered doing collaborative effort, my grade will differ from the rest of the group.

## What got done in the project?

I ended up splitting from the group since we were not on the same page and completed the project by myself. I was able to create the logo, UI, Fonts, Backend, Splash Page, and Backend of the App. I was also able to connect the Firebase Authentication as well as the Realtime Database using a couple of sources from class.

## What was different than planned?

I ended up changing the design since this way seemed a lot easier. I also wasn't able to properly get themes working so I scratched the settings and I made the app as basic as possible, but focused on the UI/UX aspect of it.

## Issues with the assignment?

I had an issue implementing the individual aspect of my project. Since our app doesn't do rentals like car rental apps do, I had a hard time trying to figure out how to implement it. However, I was planning on doing it like the expiration of a product, but I ended up figuring out a way to implement it. I also had issues with my group, but I understand. This class was difficult for all of us so it makes sense.