**Quick PC**

*PC Building Assistance Mobile Application*

****

**Tryout Records Documentation**

CECS 491A T/TH 12:30 PM

# **Table of Contents**

[**Table of Contents**](#_id9gyep3b8ui) **1**

[**Try Out Records**](#_ojmnhpk0q2eo) **2**

[Git/Github](#_33dm7413409) 2

[User Interface](#_lj21fisxuvtc) 3

[Firebase](#_fg6gim8hg23u) 4

# 

# 

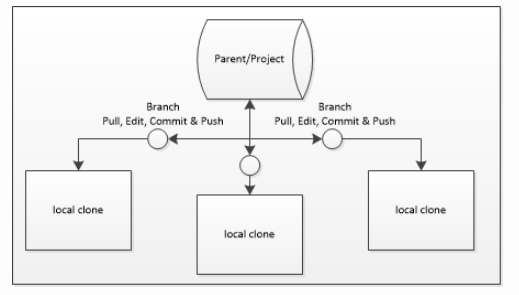
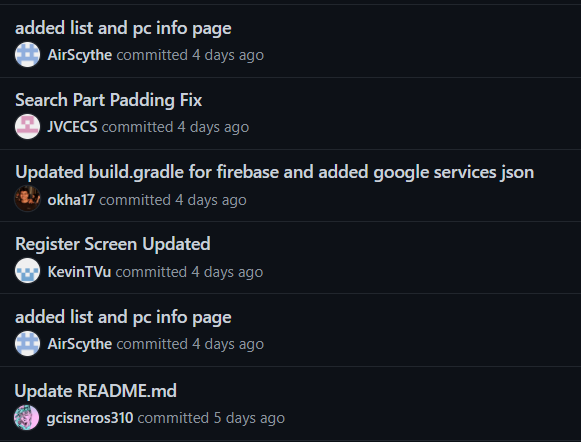
# 

# 

# **Try Out Records**

## Git/Github

* In order to maintain consistency throughout our code and documentation we needed to integrate Git into our Android Studio project. After installing Git, we created a repository on the Github platform that allowed our group to collaborate together. Initially we faced problems pushing/pulling information to/from our Github repository. This resulted in duplicate data and accidental pushes. We overcame these problems by researching Github collaboration models and enforcing high Git quality standards. Git/Github is the foundation for consistent workflow in our project so we focused 4 hours towards research.

**Tracking Change History With Github Github Collaboration Model**  


## 

## 

## 

## 

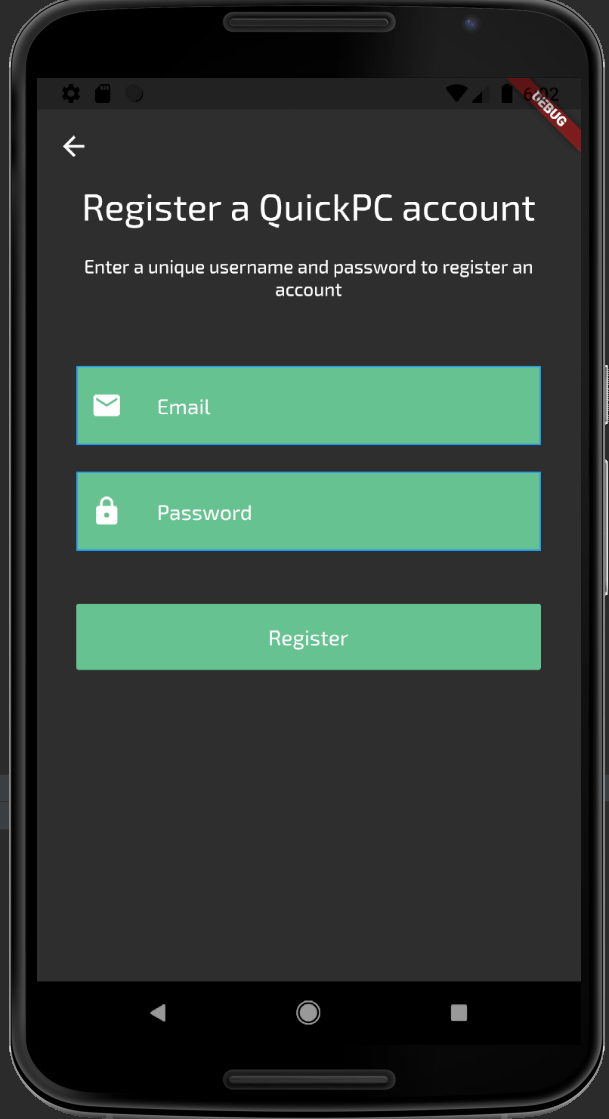
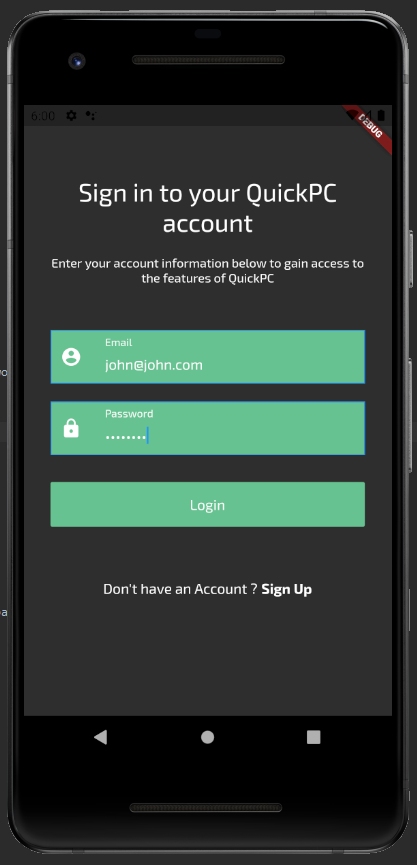
## 

## User Interface

**Home UI**

|  |  |
| --- | --- |
|  | * One of the things we wanted to prioritize was making sure that our UI was simple and easy to use. All of the group members had experience in using Android Studio but didn’t like how difficult it was to build a presentable UI through Kotlin. We decided to use Flutter using the Dart programming language to build our frontend and backend. One of the difficulties we found was creating a UI through Flutter. We managed to speed up the process by watching Flutter tutorials and documentation through YouTube. The process took around 5 hours to get a basic understanding on how Flutter works. * For the UI we explored the menu objects Flutter provided for implementation, such as buttons, forms, and animations. This would take 1 to 2 hours. * We also tested out pages formatting, seeing how far we could adjust margins. This would take less than an hour. * Ultimately we decided to focus on logging and our search feature. |

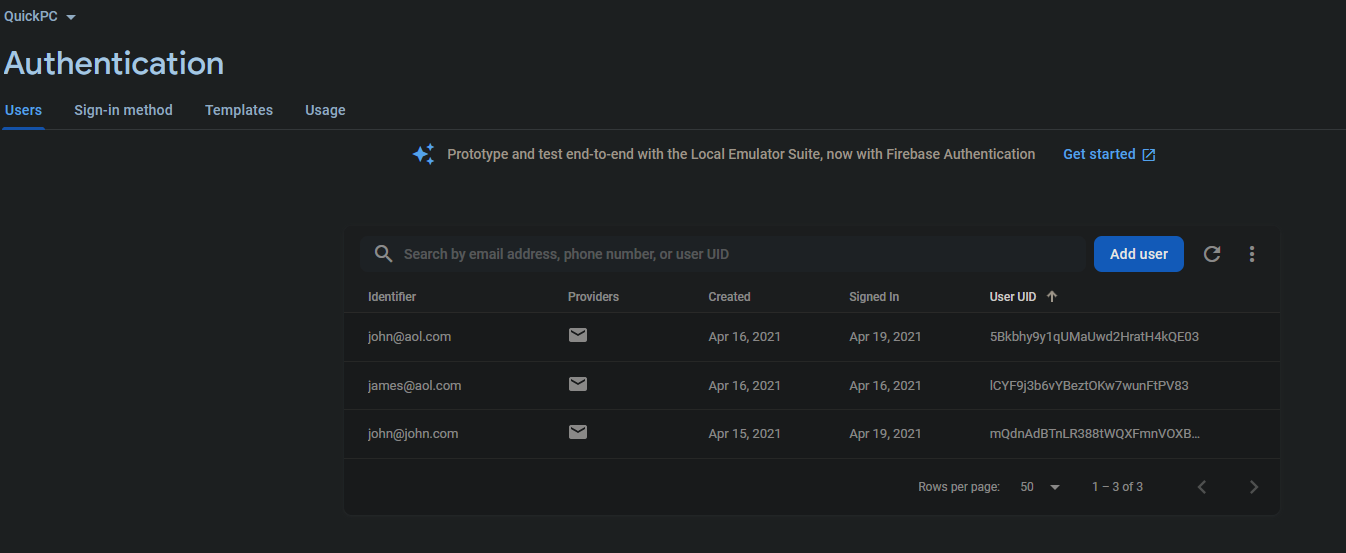
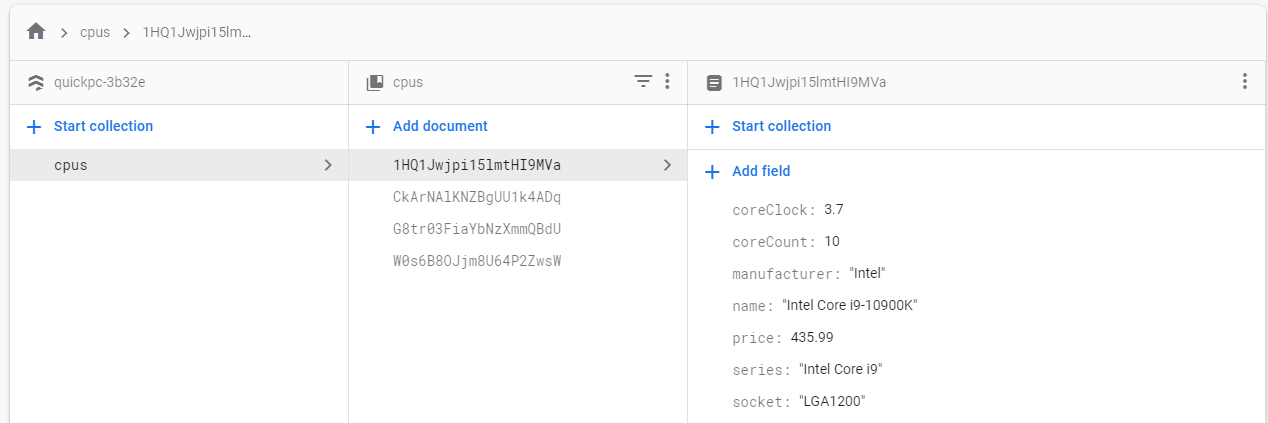
**Login UI** **Register UI**



## Firebase

* Another key component to making our application successfully was implementing a database to store user information and part information. A few of our team members were experienced in using a local database but since we wanted this application to be published through the Google Play store and Apple store we decided to expand our database to a cloud database known as Firebase.
* One problem we encountered was implementing Firebase and connecting our project to it. We spent around 2 hours figuring out how to authorize users and create valid accounts using a user’s email and password credentials.

**Firebase Authentication Database**



* For our search feature we needed to learn how Flutter can access and process data. We wanted the search feature to access a database containing info about PC components. We stored our database with test data for CPUs and tested the ways we can access it. Learning how to set up the project to access the database data took 2 or 3 hours.
* We wanted the app to list out the data to users. Passing database data between pages was a bit more complicated than expected. Figuring out how to list that data took about 1 hour.

**Search list UI** **Part Info Page UI**

