Mobile Devices (CSCI 4100U) Midterm Prototype Check

PropertyPal: A Tenant Management App for LandLord Properties

Group Members:

- Zainulabuddin Syed, Harsh Patel, Ashwin Gnanaseelan, Georges Anthony Gutierrez R.

Function Requirements Implementation:

Multiple Screens/Navigation

Throughout the App, there are many screens and navigation use cases. These screens and options will be listed in this document showcasing the features PropertyPal has. In terms of general navigation, the user will use the navigation bar found at the bottom of the app as seen in Figure 0.

Figure 0:

Dialogues and Pickers

Implementations of Dialogues and Pickers in our application are found within the Properties Tab where it will prompt the user to add a property that they would like to manage and keep track of. Here they will be asked to enter their information about the property. Figure 1 is the filled form. Figure 2 displays a picker in the form of a calendar to allow the user to pick the start date of the tenancy, once clicked it will return the date and year. Figure 3 is also a picker that allows the user to pick the duration in months of the tenancy. Figure 4 is a dialogue that prompts the user if they want to log out of the application. This can be found in the settings tab.

Figure 1:

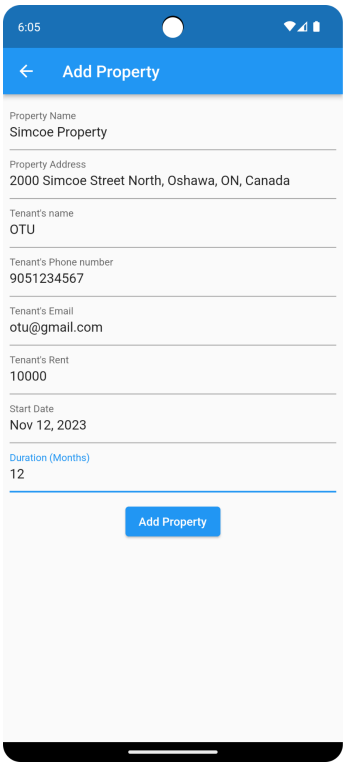


Figure 2:

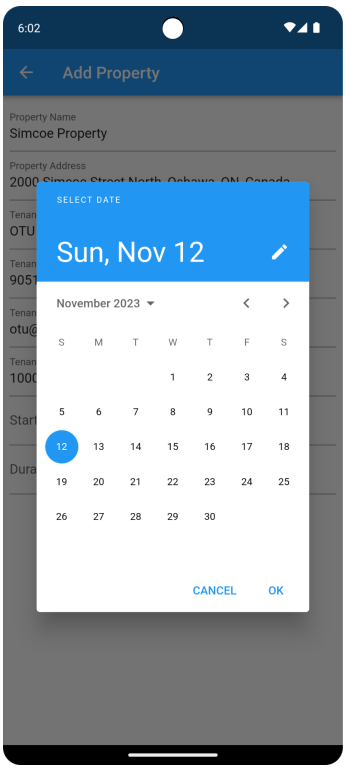


Figure 3:

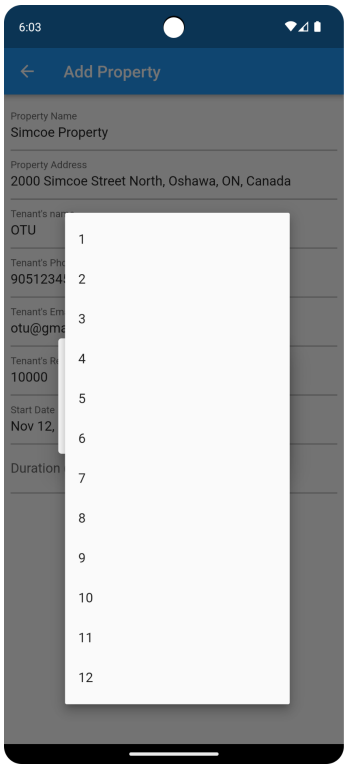
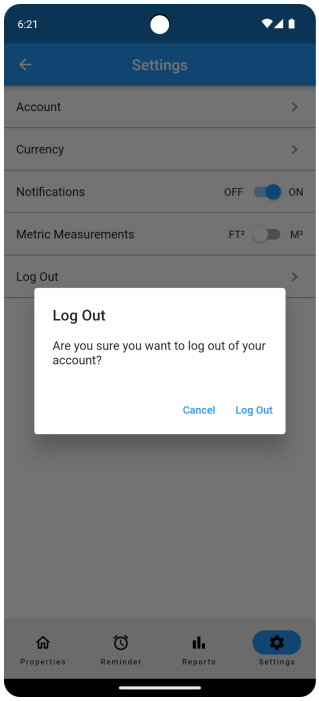


Figure 4:



Snack Bars

The implementation of snack bars in our application. Figure 5 showcases the sign-up page. Figure 6 the home screen, displays a welcoming snack bar message: "Welcome to PropertyPal". In Figure 7, there's a snack bar that appears after successfully changing the account password.

Figure 5:

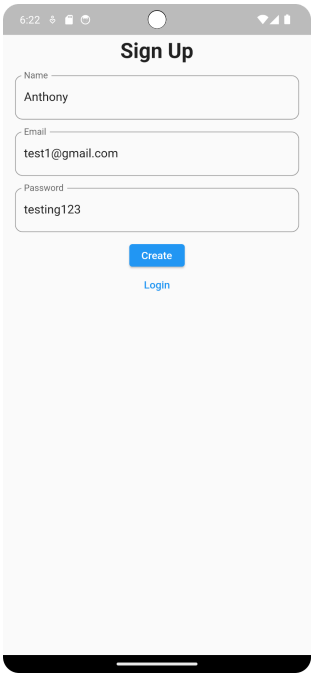


Figure 6:

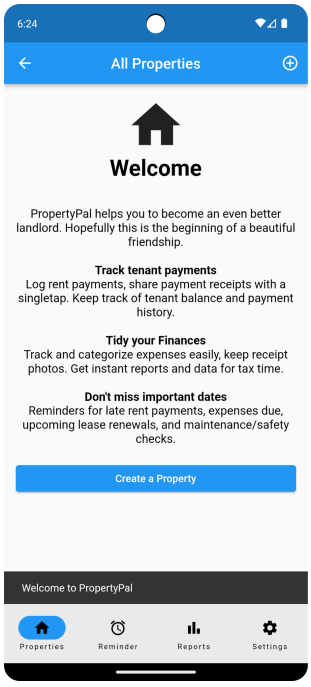
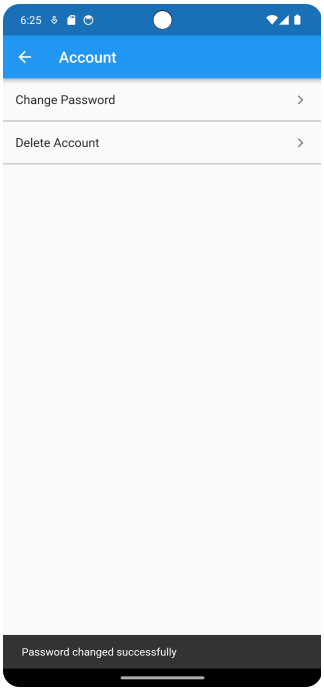


Figure 7:



Notifications

Implementation of Notifications can be seen when the user first adds the property. A notification will be sent saying that they have added a property as shown in Figure 8. In addition to that, we also provide a notification 5 days before the rent due date letting the landlord know that the tenant needs to pay their rent as shown in Figure 9.

Figure 8:

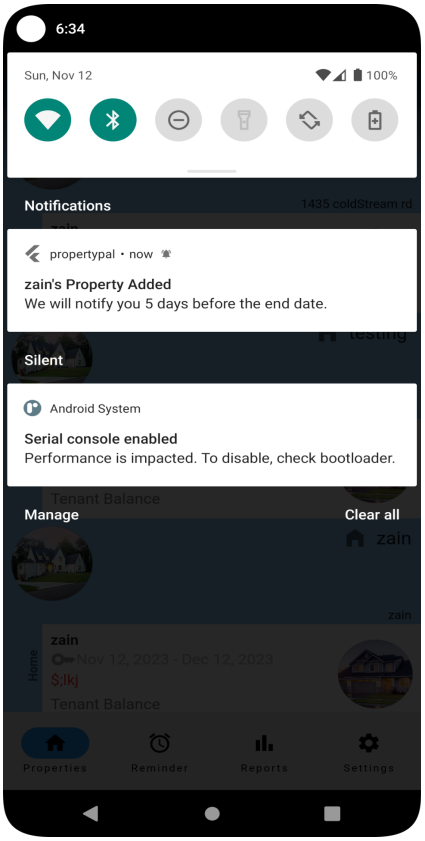
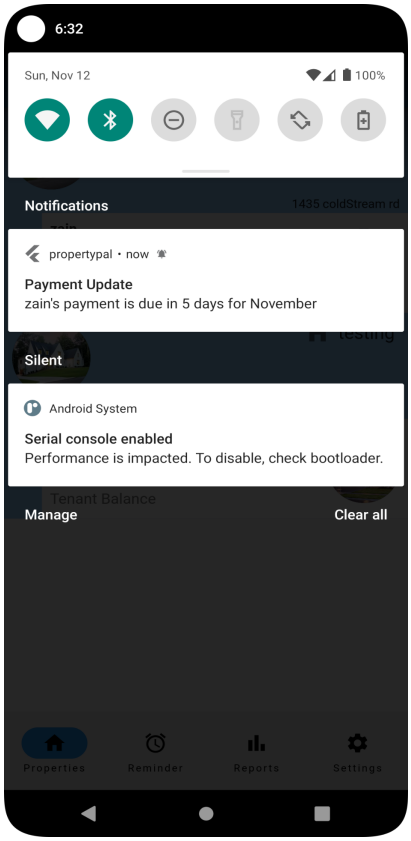


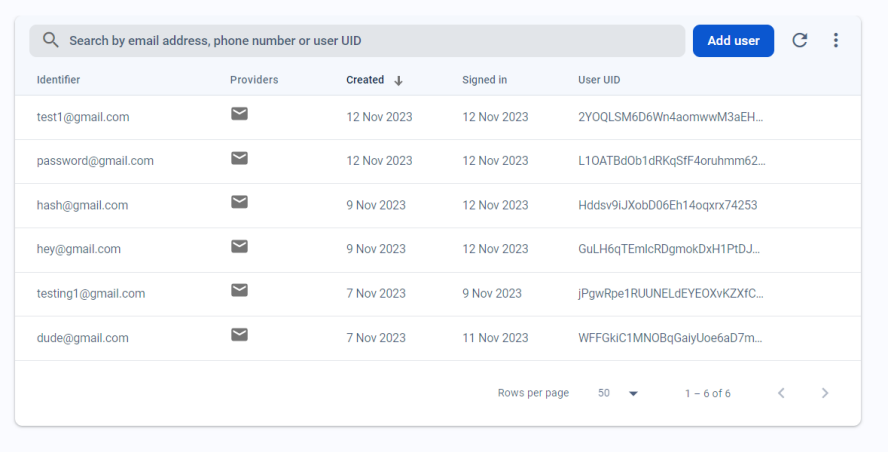
Figure 9:



Local storage (SQLite)

Implementation of Local Storage can be seen here where we used the Firebase authentication to set up our local storage where it saves the user credentials. While the actual implementation ensures the user is always logged in even when they close the application unless they specifically log out.

Figure 10:



Cloud storage (Firebase)

For the Implementation of Cloud Storage, we used Firebase. The information that the user inputs in Figure 1 is collected and stored in Firebase along with the login credentials to access the properties that the user has implemented as seen in Figures 11 and 12.

Figure 11:

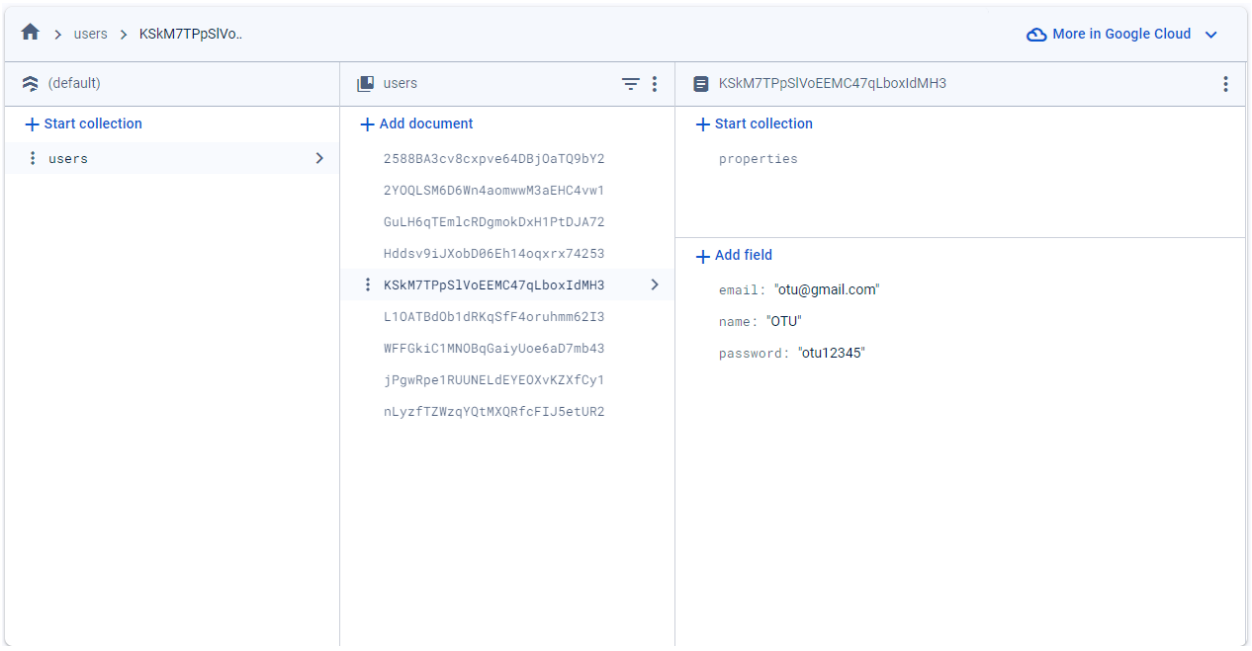
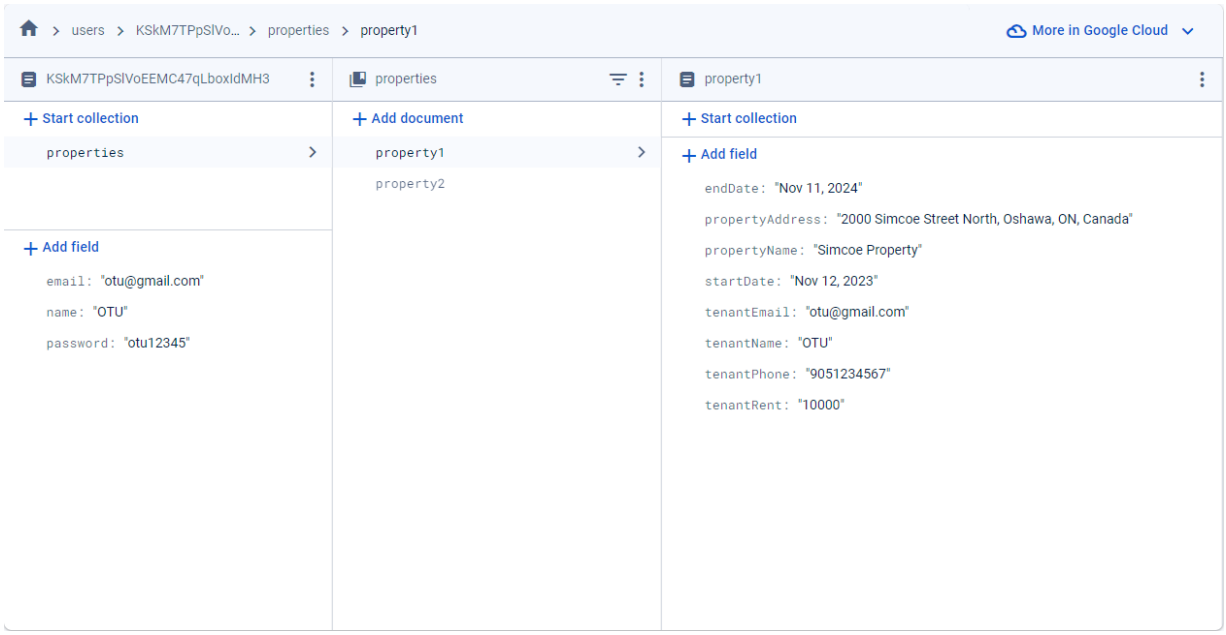


Figure 12:



HTTP Requests

The implementation of HTTP requests is featured in our Properties Tab. Within this tab, users are prompted to enter an address. We use HTTP requests to access the Google Maps API. Once the user begins to type their address the app will call the request and display five addresses that are related to the search query. It will allow the user to enter the right and accurate address as seen in Figure 13. The implementation of this can be found in the folder named google_maps_api and the file is named network_utility.dart.

Figure 13: