

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:

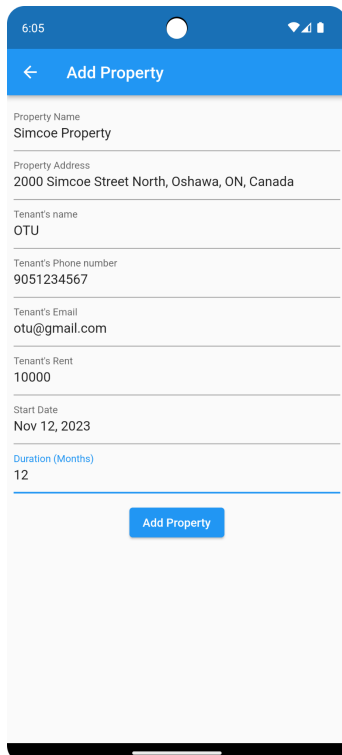
A mobile app screen titled 'Add Property' with a blue header bar. The form contains the following fields: 'Property Name' (Simcoe Property), 'Property Address' (2000 Simcoe Street North, Oshawa, ON, Canada), 'Tenant's name' (OTU), 'Tenant's Phone number' (9051234567), 'Tenant's Email' (otu@gmail.com), 'Tenant's Rent' (10000), 'Start Date' (Nov 12, 2023), and 'Duration (Months)' (12). A blue 'Add Property' button is at the bottom.

Figure 2:

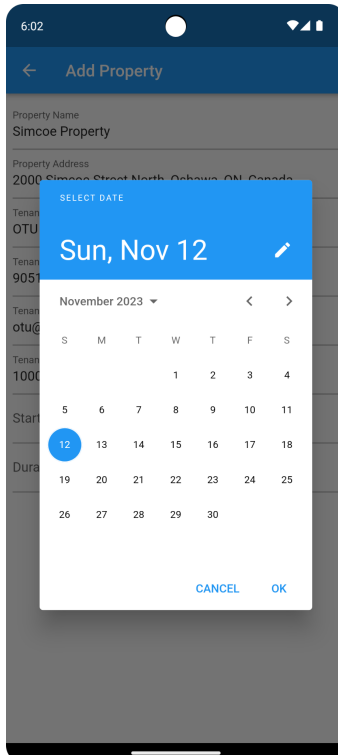
A mobile app screen titled 'Add Property' with a blue header bar. A date picker overlay is visible, showing 'Sun, Nov 12' and a calendar for November 2023. The calendar has days 1 through 30. The date '12' is selected. There are 'CANCEL' and 'OK' buttons at the bottom of the picker.

Figure 3:

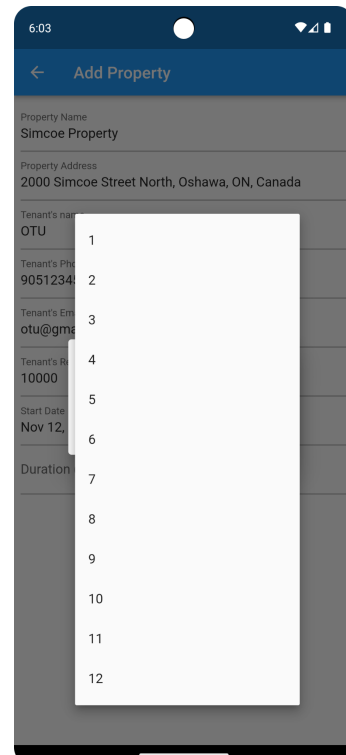
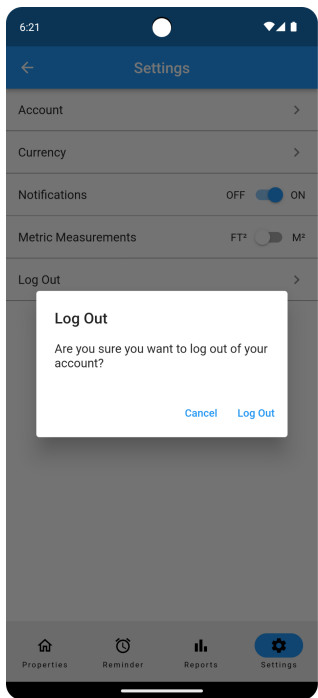
A mobile app screen titled 'Add Property' with a blue header bar. A duration picker overlay is visible, showing a list of numbers from 1 to 12. The number '12' is selected.

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:

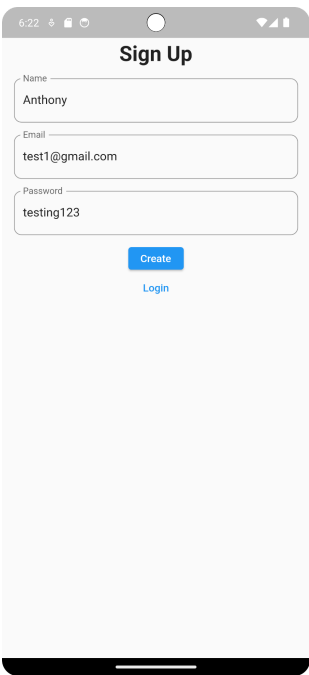


Figure 6:

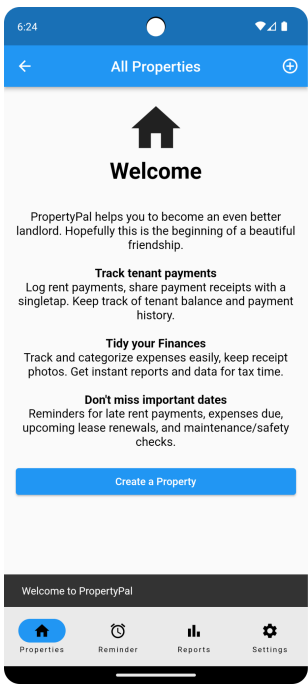
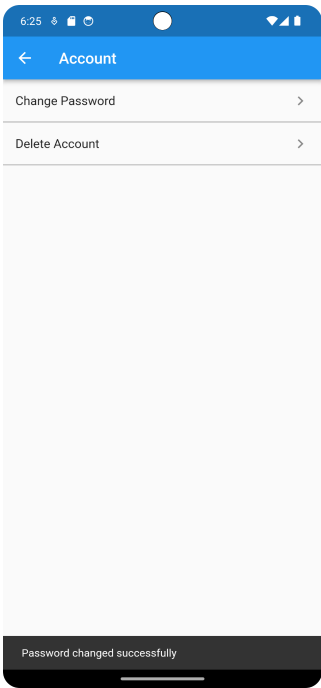


Figure 7:



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:

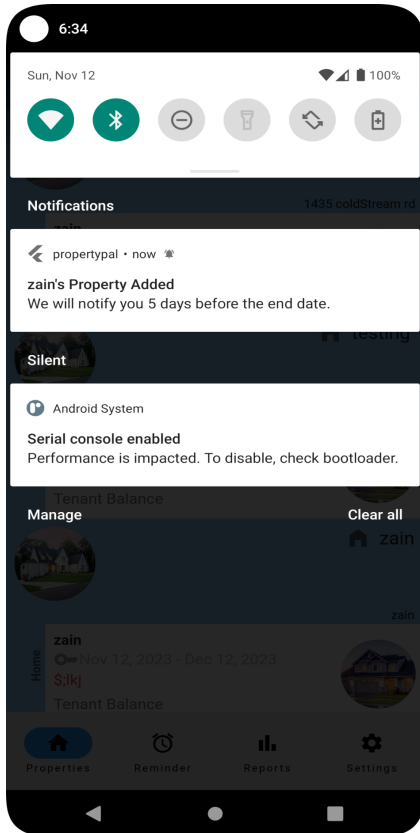
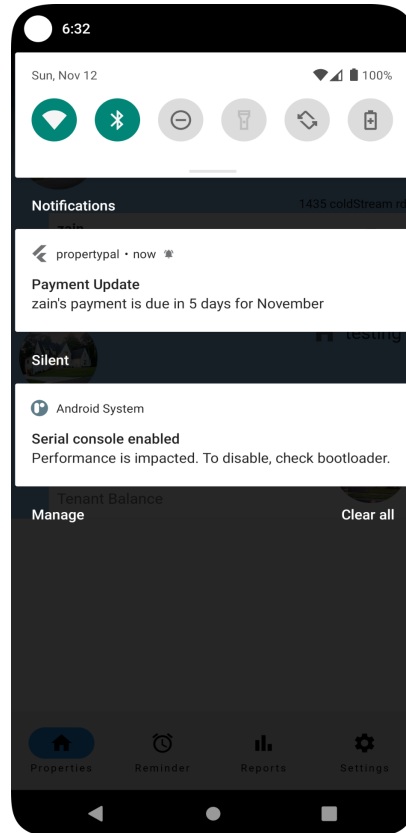


Figure 9:



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:

A screenshot of a web-based user management interface. At the top, there is a search bar with the placeholder text 'Search by email address, phone number or user UID' and an 'Add user' button. Below the search bar is a table with the following columns: Identifier, Providers, Created, Signed in, and User UID. The table contains six rows of user data. At the bottom right, there is a pagination control showing 'Rows per page' set to 50 and '1 - 6 of 6' items.

Identifier	Providers	Created	Signed in	User UID
test1@gmail.com	📧	12 Nov 2023	12 Nov 2023	2YOQLSM6D6Wn4aomwwM3aEH...
password@gmail.com	📧	12 Nov 2023	12 Nov 2023	L1OATBd0b1dRKqStF4oruhmm62...
hash@gmail.com	📧	9 Nov 2023	12 Nov 2023	Hddsv9iJXobD06Eh14oqxrx74253
hey@gmail.com	📧	9 Nov 2023	12 Nov 2023	GuLH6qTEmlcRDgmokDxH1PtDJ...
testing1@gmail.com	📧	7 Nov 2023	9 Nov 2023	JPgwRpe1RUUNELdEYEOXvKZXfC...
dude@gmail.com	📧	7 Nov 2023	11 Nov 2023	WFFGkiC1MNOBqGalyUoe6aD7m...

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:

🏠 > users > KSkM7TPpSIVo...			☁ More in Google Cloud ▼		
⌵ (default)	📁 users	⌵ ⋮	📄 KSkM7TPpSIVoEEMc47qLboxIdMH3	⋮	
+ Start collection	+ Add document		+ Start collection		
⋮ users >	2588BA3cv8cxpve64DBj0aTQ9bY2		properties		
	2Y0QLSM6D6Wn4aomwwM3aEHC4vw1				
	GuLH6qTEmlcRDgmokDxH1PtDJA72				
	Hddsv9iJXobD06Eh14oqxrx74253				
	⋮ KSkM7TPpSIVoEEMc47qLboxIdMH3 >		+ Add field		
	L10ATBdOb1dRKqSfF4oruhmm62I3		email: "otu@gmail.com"		
	WFFGk1C1MNOBqGaiyUoe6aD7mb43		name: "OTU"		
	jPgwrPe1RUUNELdEYEOXvKZXfCy1		password: "otu12345"		
	nLyzfTZWzqYQtMXQRfcFIJ5setUR2				

Figure 12:

🏠 > users > KSkM7TPpSIVo... > properties > property1						☁ More in Google Cloud ▼	
📄 KSkM7TPpSIVoEEMc47qLboxIdMH3	⋮	📁 properties	⌵ ⋮	📄 property1	⋮		
+ Start collection		+ Add document		+ Start collection			
properties >		property1 >		+ Add field			
		property2		endDate: "Nov 11, 2024"			
				propertyAddress: "2000 Simcoe Street North, Oshawa, ON, Canada"			
				propertyName: "Simcoe Property"			
				startDate: "Nov 12, 2023"			
				tenantEmail: "otu@gmail.com"			
				tenantName: "OTU"			
				tenantPhone: "9051234567"			
				tenantRent: "10000"			
+ Add field							
email: "otu@gmail.com"							
name: "OTU"							
password: "otu12345"							

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:

