Code Documentation for Real Estate Investment Project Mobile

Karan Lnu

CS-682

1 Introduction

This document provides a comprehensive guide to the codebase of our Real Estate Investment project. The project is developed using React Native for mobile applications and is designed to provide a robust and user-friendly platform for real estate investors. It allows users to manage their investments, calculate potential returns, and make informed decisions.

The project leverages various technologies such as Expo for handling vector icons, and React Navigation for managing screen navigation. The codebase is organized into several key components, each responsible for a specific aspect of the application. These components include user profile management, property details, investment calculations, and various utility components.

2 Technologies and Packages Used

The project is built using React Native, a popular framework for building mobile applications using JavaScript and React. It also utilizes a variety of additional packages to extend functionality and improve the user experience:

- @gorhom/bottom-sheet: Used to create a bottom sheet component, a common UI element in mobile applications.
- @react-navigation/bottom-tabs, @react-navigation/native, and @react-navigation/stack: Part of React Navigation, used for routing and navigation within the app.
- class-transformer: Provides utilities for transforming plain JavaScript objects to class objects and vice versa.
- expo, expo-constants, expo-file-system, and expo-status-bar: Part of the Expo SDK, which provides a set of tools and services for building, deploying, and quickly iterating on iOS, Android, and web apps.
- react and react-native: Core packages for building the app.

- react-native-a11y-slider and react-native-slider: Provide slider components for user input.
- react-native-elements: Provides a set of reusable UI components.
- react-native-gesture-handler and react-native-reanimated: Provide more fluid, low-level control over gestures and animations in the app.
- react-native-google-places-autocomplete: Provides a text input component that auto-suggests Google Places as the user types.
- react-native-maps: Used to display maps.
- react-native-safe-area-context: Provides a way to manage safe areas in the app, which are areas of the screen not covered by system UI (like the notch on an iPhone).
- react-native-webview: Provides a component for displaying web content.
- reanimated-bottom-sheet and rn-sliding-up-panel: Provide components for creating sliding up panels, another common UI element in mobile apps.

These packages together provide a robust set of tools for building a feature-rich mobile application.

3 Components Detail

3.1 Google Maps Components

1. SearchBar: This component implements a search bar using the Google Places Autocomplete API. It renders a search bar with autocomplete functionality for places using the Google Places Autocomplete API.
2. PropertiesMap(onSearchPress, onMapPress, type): This is the main function of the component. It renders a map view with markers representing properties and handles marker press events to navigate to property details.

3.2 Properties Data

1. PropertyList(): The PropertyList component component in the provided code is a crucial part of a real estate application. It is responsible for displaying a list of properties fetched from an API. Each property in the list is represented by a `PropertyCard` component, which shows essential details about the property.
2. The details displayed for each property include the price of the property and its cash flow, among other primary details. These details provide a quick overview of the property's financial aspects, allowing users to make informed decisions about their real estate investments.

3. In addition to displaying property details, the `PropertyList` component also provides a search functionality. This is implemented using the `SearchBar` component, which uses the Google Places Autocomplete API to suggest locations as the user types.

4. By selecting a location from the search suggestions, users can filter the list of properties to only include those in the selected location. This feature enhances the user experience by making it easier to find properties in specific locations.

5. Furthermore, the `PropertyList` component allows users to view more detailed information about a property. This is achieved by navigating to the `PropertyDetails` screen when a `PropertyCard` is clicked. This screen would display more in-depth information about the property, such as its full description, images, and possibly more detailed financial metrics.

6. PropertyDetails(route, navigation): This component is responsible for displaying detailed information about a specific property. The details are fetched from an API using the property's ID.

3.3 Calculator

EditCalculator: This component is used to edit various properties of a calculator, such as property info, purchase info, income info, financing info, and operating expenses.

The module consists of several classes, each contributing to a different aspect of real estate investment analysis:

- **BuyRentHold**: The primary class orchestrating the investment analysis process.
- **CashRequirements**: Manages calculations related to the cash requirements for property investments.
- **Financing**: Deals with aspects of property financing, such as mortgage calculations.
- **OperatingExpenses**: Computes the operating expenses associated with owning a property.
- **PropertyInfo**: Holds information about the property being analyzed.
- **PurchaseInfo**: Contains details regarding the purchase aspects of the property.
- **IncomeInfo**: Manages income-related data for the property.

3.4 User Profile

UserProfile: This component displays a user profile with options such as About, Glossary, and Settings, and a logout button.

4 Models

1. Location Model: The Location class in JavaScript is designed to encapsulate a geographical location. It includes properties for latitude, longitude, and a name.

2. User Model: The Property class in JavaScript represents a property listing with various attributes. It can be used to create instances representing individual property listings, and these instances can then be manipulated or displayed as needed in your application.

5 Conclusion

In conclusion, the Real Estate Investment project is a comprehensive mobile application built using React Native and a variety of other technologies. It provides a robust platform for real estate investors to manage their investments, calculate potential returns, and make informed decisions. The application is organized into several key components, each responsible for a specific aspect of the application, and uses a number of models to represent data consistently.

This document provides a detailed overview of the project's codebase, including the technologies and packages used, the structure and functionality of the components, and the models used to represent data. It serves as a guide for understanding the codebase and can be used as a reference when developing or maintaining the application.

The project demonstrates the power and flexibility of React Native and the associated technologies in building complex, feature-rich mobile applications. It stands as a testament to the capabilities of these technologies in the field of real estate investment and beyond.