Technical Document: Stocks App

Tron Baraku – Developer

This technical document describes the main functionalities of the "Stocks" application and provides details about how these features were implemented. The developed application provides users with real-time information about crypto currencies.

1. API Implementation

The application receives the data regarding the currencies through an external API. To receive this data, HTTP requests are made through the Volley library. There are two types of responses received from the API depending on the URL of the request:

- https://app-vpigadas.herokuapp.com/api/stocks/
- https://app-vpigadas.herokuapp.com/api/stocks/{currencyID}

The first one provides a general list of the currencies while the second one provides additional details about the specified currency. The data received from the first API is manually parsed by extracting the relevant data for each currency (id, name, symbol, price, and percentage). The extracted data is used to create *CurrencyEntity* objects, which are then used to display a list of all the currencies.

When a user clicks on one of the currencies displayed in the main activity, a request is made to get additional details about that currency. In this case, the GSON library has been used to automatically map the JSON responses to Java objects.

2. Offline mode

The application provides an offline mode, which is facilitated by the SQLite database created using the Room library. The application automatically stores and updates the data about the currencies. The application checks the connection status of the device using the *ConnectivityManager* system service to determine whether to load the data from the database or make a new HTTP request. There are three entities:

- *CurrencyEntity* – Represents the general data received from the API (/stocks/).
- *CurrencyDetailsEntity* – Represents the detailed data received from the API (/stocks/{currencyID}).
- *CurrencyFavoriteEntity* – Represents the currencies that the user has favorited.

3. Favorites

A switch has been implemented in the *CurrencyDetailsActivity*, which allows the user to mark a currency as favorite. The initial state of this switch is determined by examining the database to check if the user has already marked the currency as favorite. An action listener has been attached to this switch to track the actions of the user (add or remove a currency from the favorites). The IDs of the currencies that the user marks as favorited are added to the *CurrencyFavoriteEntity* table. The filtering functionality, which is used

to display only the user's favorite currencies in the *CurrencyFavoritesActivity*, is achieved by performing an inner join between the *CurrencyFavoriteEntity* and *CurrencyEntity* tables. As a result, only the necessary data regarding the user's favorited currencies is queried and displayed.