U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

My touristic wallet

Practical Assignment #2

Mobile Computing 2022/23

Group 7
Alberto José Ribeiro da Cunha - up201906325
Joana Teixeira Mesquita - up201907878
João Bernardo Costa - up201907355

Features:

The app developed contains the following features:

- Allows users to store money divided by currencies
- Users can also remove any currency they like
- Convert all the money they have into a currency
- Show graphics representing how much each currency impacts the total amount

Architecture:

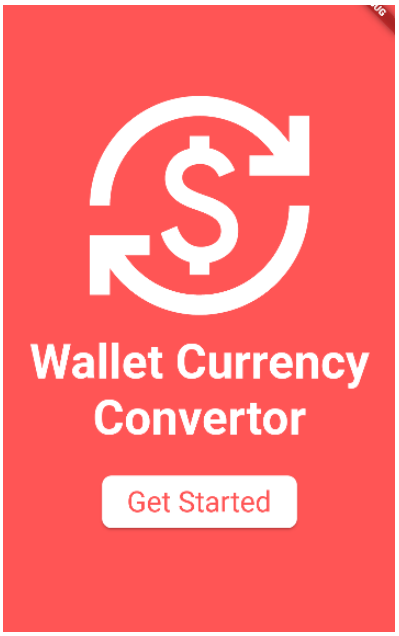
The project is composed of a simple Flutter application which makes https REST requests to the Fixer API from APILayer (https://apilayer.com/marketplace/fixer-api) to obtain the exchange rates of currencies.

Information about the amounts and conversion rates are stored locally using the Shared Preferences flutter package.

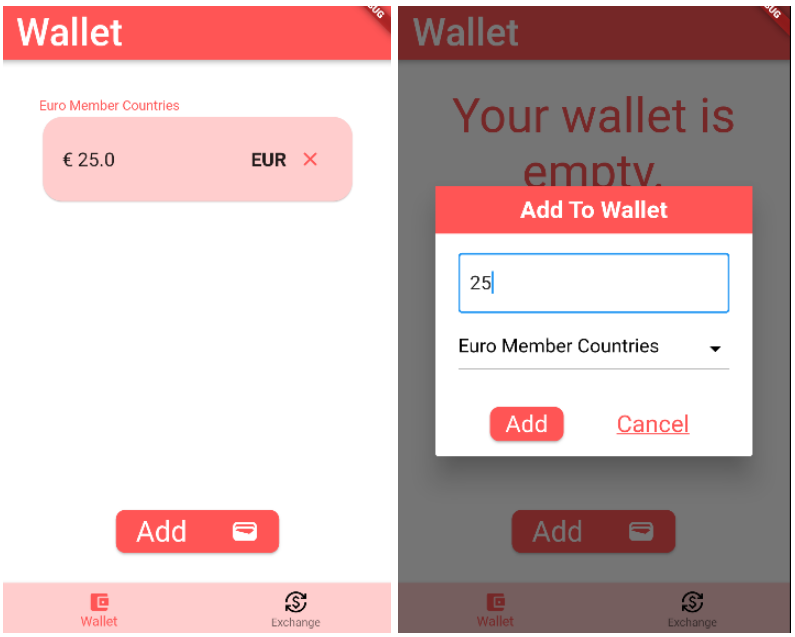
Interface:

The application has 3 screens: the home page, the wallet page and the exchange page.

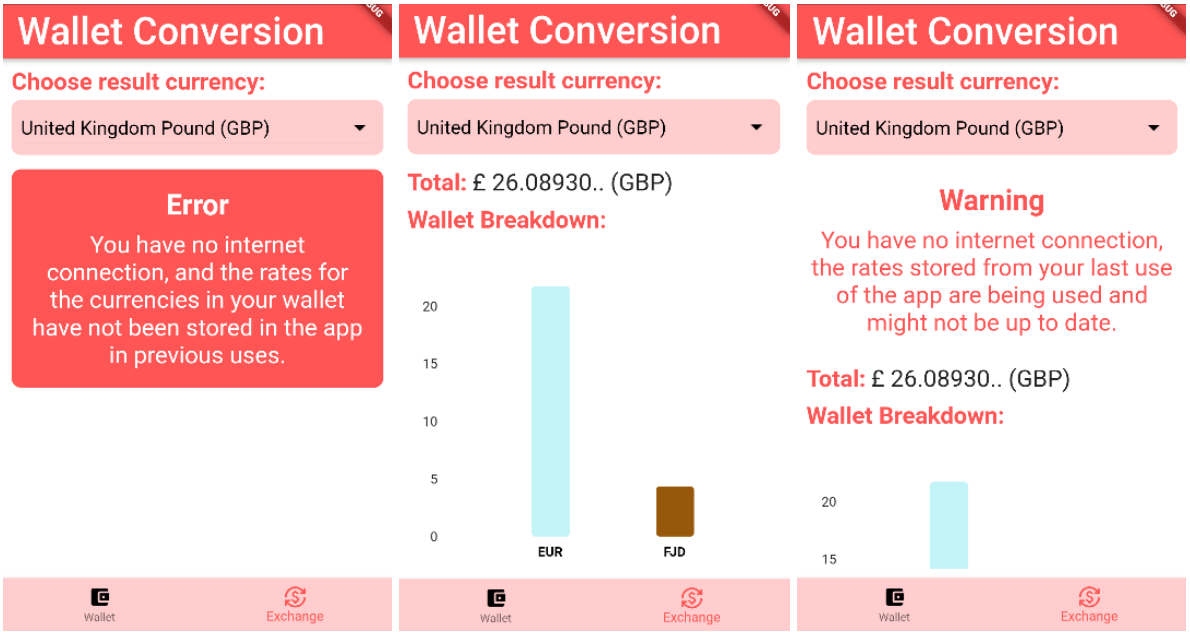
The home page is a simple starting page with a button to access the rest of the application.



The wallet page shows the money in the user's wallet by currency and has a button that lets users add money to it, as well as a button for each currency to remove any existing currency.



The exchange page has a dropdown button to choose the currency the user wants to convert their money into. Furthermore when the wallet is converted graphs are shown detailing the percentages that each currency has on the conversion. This page also shows warnings for errors relating to connection issues.

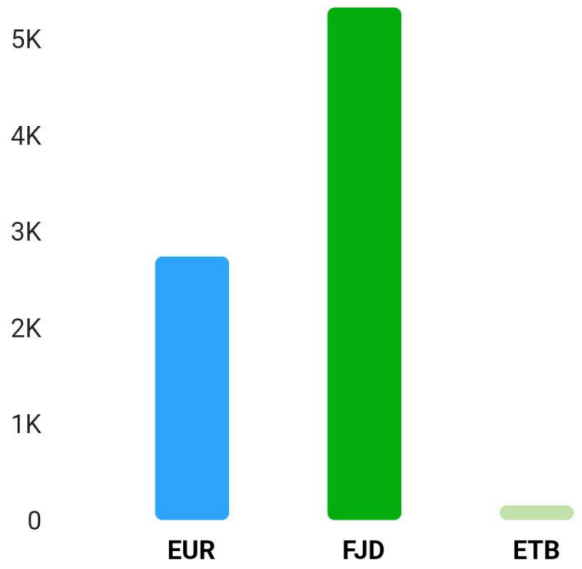


Graphics Generation

We created two types of graphs, a Bar Chart and a Pie Chart, with the fl_chart flutter package.

Bar Chart

The bar chart has a bar corresponding to each currency in the user's wallet, each with a different color and its height corresponds to the value of the currency after exchanged for the new one.



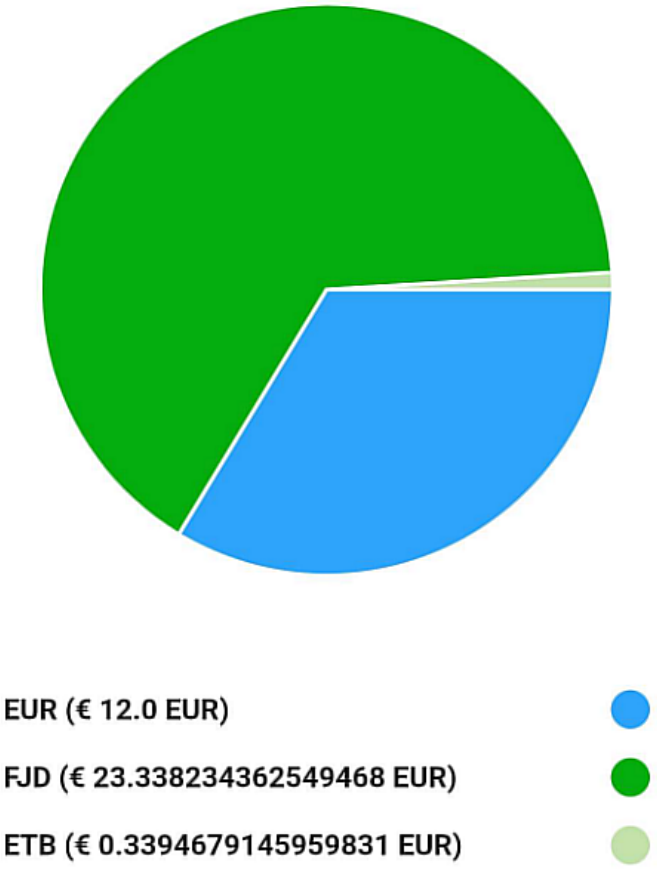
When pressing one of the bars it's possible to see the exact value of the currency after exchange.

To create this chart we created an **IndividualBar** class which stores the x, y and color of each bar and a **BarData** class which receives the lists with the currencies, amounts and colors of the bars and turns them into **IndividualBar** objects.

Finally, the Widget **BarGraphWidget** was created, receiving the lists with the currencies, amounts and colors of the bars, passing them to **BarData** and initializing the Individual Bars. Then it makes use of **BarChart** Widget from the fl_chart package to show the data in the form of a bar graph.

Pie Chart

The pie chart has a slice corresponding to each currency in the user's wallet, each with a different color and with their size corresponding to the percentage of the total value after exchange that the currency represents.



A key was also added to show the exact value of each currency as well as which color corresponds to each currency.

To create this chart we created a **Sector** class which stores the value, title and color of each slice and a **SectorData** class which receives the lists with the currencies, amounts and colors of the bars and turns them into **Sector** objects.

Finally, the Widget **PieChartWidget** was created, receiving the lists with the currencies, amounts and colors of the slices, passing them to **SectorData** and initializing the Individual sectors. Then it makes use of **PieChart** Widget from the fl_chart package to show the data in the form of a pie graph.

Testing Performance:

The application is capable of working without internet connection as long as the rates for the currencies are stored in local memory from previous uses of the application, warning users that the rates used might not be up to date.

However, if there is no internet connection and the exchange rates are not stored in the local memory then the user is shown an error page asking them to connect to the internet.

Use Cases:

Our application has the following use cases diagram: