REPORT

This application contains different paths to such an application that we have chosen to produce. This is the app briefly summarized.

We have 7 different activities. The first is a splash screen on the mainactivity.kt which starts the Crypto List Activity with a 3 seconds delay. Here you can see the different crypto currencies the API provides with symbol, image, price, name and percentage which are updated during the last 24 hours. You have two options on this screen, either or you can press **points** which redirect you to Portfolio activity (screen 3) or the crypto currency you would like to get more info about (Screen 4). In the Portfolio screen you can see your portfolio and press the button transactions which redirect you to screen 7 to se your transactions. In screen 4 where we have called it Buy Sell Activity, here you can see the data of the selected currency. And you can choose to buy (screen 5) or sell (screen 6).

We have used Retrofit fetching for fetching the data from the API. The live source technique is for fetching the live data posting it to the API. The application use SQLite with Room to store buy, sell activities. The view model along with Fragments to show data. We use Adapters to fetch the data from database and API to the activity, where we can also pass data/information from e.g., one screen to another. In this case we pass information from screen 2 to screen 4, 5 and 6. Where the application will pass the information on buy and sell activity to the database.

In listing the different data's, we use recycler view method and a fragment view method. These are some of the different UI-components. Some other components we have used are as follows: TextView, ConstraintLayout, ImageView, ConstraintLayout, Button and LineayarLayout. Which we believe are a very good approach for our application. Some alternatives we could have focused more on is the fragment-part. We could have use more of this to make the application smoother.

Some pros for this application are that is a simple but unique in fetching data from retrofit API. Is easy to add different components in the code after the exam is delivered. Some cons are that the application is rather unstable, it can sometimes give some rather strange bugs but still work. We have tried to fix this but did simply not have the time.

Known bugs:

- Every time when starting application, you get 10.000 USD