Fourth report of Open Banking Data Analysis project

Team 45 7/2/2020

Update from the meeting with our client:

- Sent use cases that we discussed to Jannen
- Extend self-generated data from October 2019 to Jan 2020 to create a time series
- Instead of Sort Code, could use IBAN as well
- Even though total credit card payments > balance, they are not paid until a certain date, meaning, even if the balance is hypothetically negative, it does not necessarily mean that you are in overdraft. Use some algorithm or intelligence to prevent users from going into overdraft.
- Azure virtual machine shut down but we got it fixed, IP Address changed to 51.132.14.212

To do list:

- Complete dataset from October 2019 to Jan 2020 for 2 users.
- We need to check for when payments are made, make sure dates correspond correctly and look out for mistakes
- We need to add the feature that shows billing date and Payment date for credit cards
- Add interest rates from Credit Cards
- Maybe we can find inspiration from a similar app (etc. Money Dashboard)
- Add suggestion feature: Tell user and suggest as to when to pay off debts and recommend what expenses could be reduced
- Gives predictions on how much will be spent in the upcoming days
- Every company in the UK has a company associated to it, if we go to resources.companieshouse.gov.uk/sic/, if wikidata provides sic's for companies, we could try and get the category from the website to make sure it's correct

- Make sure to implement intelligence into our app to differentiate our app

- If open banking already provides the credit card interest rates, then assume we already know the credit card interest, therefore we don’t need a page to input the interest rates manually.

- Savings pot won’t work. Remove savings pot

Individual progress and plan:

Raghib:

Progress:

- Wrote new code that supports one user being able to connect multiple different bank accounts to their single “Icy bank” account.
- Learned about NLP, used AzureML to write categorisation algorithm
- Wrote new improved categorisation algorithm using MCC on Open Banking API data format

Plan:

- Front end and back end for functionality to set caps and budgets- need to write a function that gets the total spending for each category for further calculation
- Need to write the front end for list of accounts for users to easily switch between different accounts
- Need to work on the date filter that allows users to choose the time range of the transactions to be shown

Yuheng:

Progress:

- Discussed the use case we want to achieve with our app and the corresponding backend algorithm
- Studied the structure of the generated data and assume that data we get from open banking api is of the same format
- Developed the algorithm to create a timeline with all transactions, and also add all active direct debit as future payments on the timeline

- Wrote the pseudocode for the prediction algorithm and average daily spending calculation

Plan:

- Translate pseudocode into working python code
- Implement the algorithm as working python code
- Test the code separately to make sure it does what it supposes to do
- Test the code in the webapp to make sure it works as an app
- Return the result as a dictionary for further graph generation (handled by Lib Kai)

Lib Kai

Progress:

- Started to populate data in excel for a data time series from October 2019 to January 2020
- Studied the data format provided by Yuheng to better generate dummy data
- Start looking into alternatives for our visualisations

Plan:

- Write an excel program that converts the Excel datasheet to a JSON object withOpen Banking data format
- Decide which charting tool to use and learn it
- Continuously improve the visuals of the web app