

OKAYFinances

Group 18: OKAYGroup

Ang Yi Min, 2000575@sit.singaporetech.edu.sg

Bryan Ong Ming En, 2001011@sit.singaporetech.edu.sg

Chua Sheng Yu, 2001206@sit.singaporetech.edu.sg

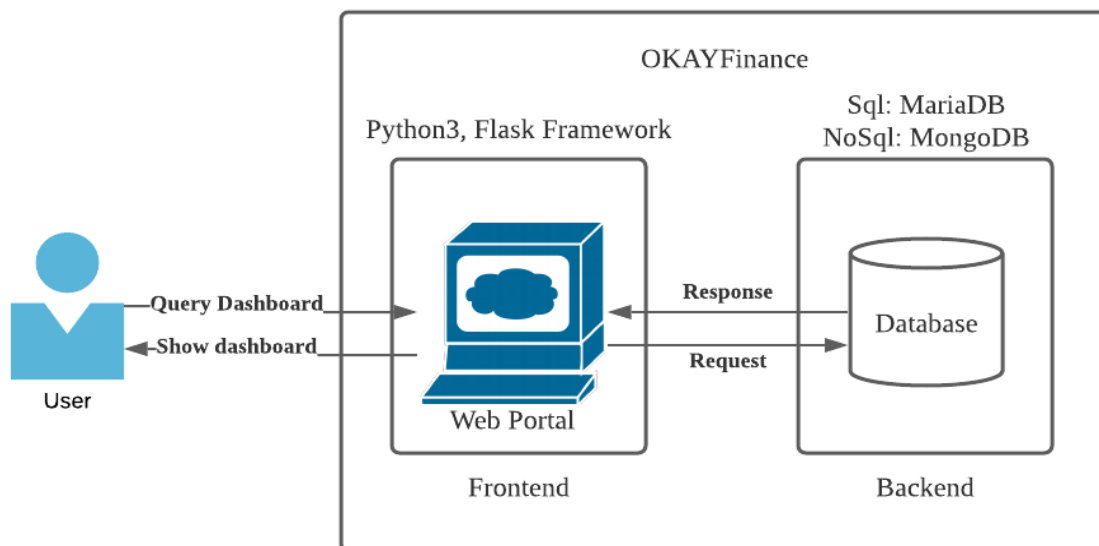
Kwang Guan Cong, 2001411@sit.singaporetech.edu.sg

Application Description

Our application would target solving our own personal finances. This application would help us track our spending habits and set our financial goals back on track by putting budget caps and sending notifications on irregular spending such as when spending too much on food when compared to our average spending or if our personal budget cap has been exceeded. Additional features include comparing statistics within the application users.

System Architecture and Requirements

The application would be using Python3 and SQL as the main backend development language alongside the Flask framework for the front-end web-based application. Database development will mostly be on MariaDB as the main relational database management system and NoSQL for non-relational database.



Initial data set to be used:

https://www.kaggle.com/bukolafatunde/personal-finance?select=personal_transactions.csv

The dataset contains transaction data of a person, such as shopping on Amazon.

We will also be using our own bank transaction data that is masked to exclude sensitive information to add on to the dataset.

Relational Database

User table

In our User Detail table, it would contain details such as their email address and password pertaining to their login information. The keys would then be used to relate their transactional data that they have uploaded onto their transactional table.

Transaction table

The users then would have their own database containing their own transactional information. They then can insert, update, and delete their transaction data such as categories, date, etcetera.

Analyzer

The data from RSDB will then be pulled to generate analytical charts and statistics to formulate useful actions such as total amount spent in a month, customer segmentation based on gender and age, etcetera. The data can also be used to compare other users of the application.

Budget

User will be able to create and view budget set by themselves. They will be able to view past budget set to see if they are on the right track with their spending.

Non-relational Database

We plan to convert our current database into NoSQL format or to implement the dataset into the NoSQL database instead, along with integrating the new database for use in our application.

One difficulty that we will be expecting is to be able to think and plan on how to change a relational format SQL into a NoSQL format which has no relationships, and foreign keys. Since NoSQL databases are document, key-value, graph, or wide-column stores, it would be harder to use more complicated select statements to search or sort data in a NoSQL database.

A difference that we expect is that NoSQL will allow the database to scale better than SQL. NoSQL can improve the performance of the database as join queries are eliminated. SQL database uses structured query language and has a predefined schema while NoSQL database has dynamic schemas for unstructured data [1].

We hope to reuse the same functionalities as the relational database. We also plan to investigate NoSQL solutions like MongoDB or Apache Cassandra.

References

- [1] L. Schaefer, "What is nosql? Nosql databases explained," *MongoDB*. [Online]. Available: <https://www.mongodb.com/nosql-explained>. [Accessed: 17-Sep-2021].