CSCI 5253 Project Proposal

Project Title

Shopping history summarization application

Team Members

Sitong Lu, John Salame

Project Goals

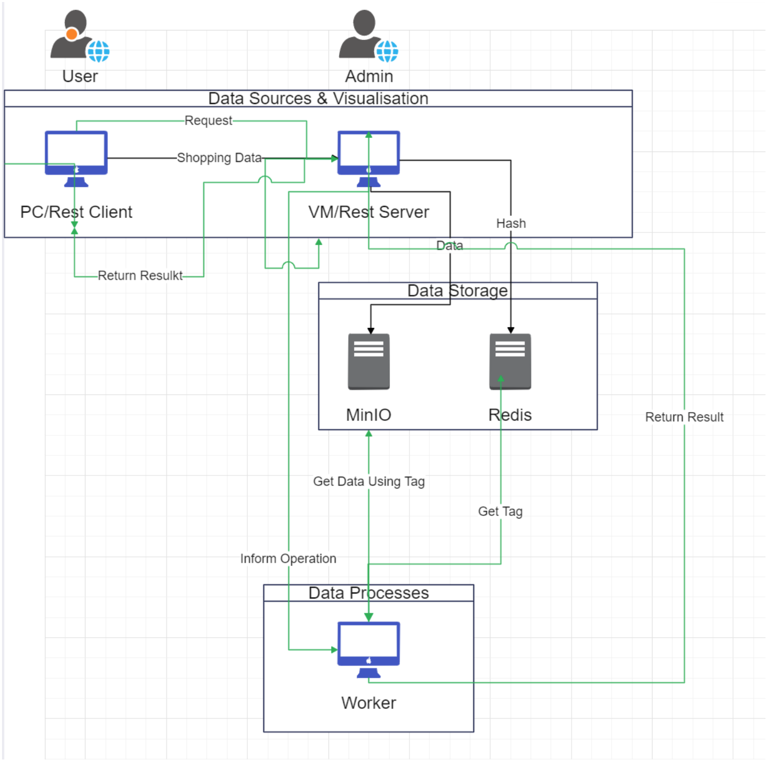
The goal of our project is to provide a server that can record and summarize the shopping history for the user. This program will allow the user to submit JSON file that records his/her raw shopping history and display the whole history in the way the user ordered. The server will contain other useful methods to allow the program to show the user the most useful information he/she needed on the user interface.

Software/Hardware Components

- **Software Components**: VS Code (IDE for programming), Redis (for hosting quick access dictionary), MinIO (for storing all the user data and their reference that can be checked in Redis), Flask (for providing Flask environment), REST (for building client and server side with visible interface displaying all the necessary data), Google Cloud VM (for deploying the server online and let it be accessible from the rest of the world).
- **Hardware Components**: Surface Pro 8 (i7-1185G7, 16GB RAM + 256GB ROM)

Architectural Diagram

Below is the diagram showing the interactions between system components:



Interaction Description

The project will use the following 4 software/hardware components:

- API Interfaces (for passing/accessing the JSON file to/from the client server)
- Message queues (acting as buffer for making sure that I am always passing JSON data in order, instead of facing the issue that my server is overloaded with a lot of data and pass all of them at once at the end due to the no-buffer issue)
- Databases (for storing my JSON data on the server and access them efficiently using SQL queries)
- Virtual Machines (for hosting the server on Google Cloud)

The **API interface** provides both the user and the administrator a visible interface for checking data and results gained from commands. It also passes the data requested from both the user and the administrator to **databases** using technique that acts like **message queues** (for avoiding issues like data overload) for data storage. We will host these servers on **virtual machines** provided by Google Cloud.

Possible Debugging

We will record the logs of all the programs running for our project down, similar to the way introduced in Lab 7, to debug all the issues during development.

We will be mostly using integration tests or functional tests since it best reflects how good a server/node is running with all their features. Afterall, it is pretty hard to set up unit tests in this situation as a server contains multiple setup and methods before really make one feature (e.g. send/get response from another place) work.

Project Fit

We believe this project idea will meet the eventual project requirements because all the technologies we are going to apply into are all from the previous lab assignments and lecture notes. As a result, this project on the first look is going to be similar to what we have done in Lab 7, but with all different kinds of methods for data processing/transitioning and some use of SQL queries for sorting and finding necessary data the user need. Overall, this project contains everything we have learned in the last 7 lab assignments and we think it is the best example of showing how much we have learned from this course at the end of the semester.