Design Document

Team 3

Aman Goyal (2019101097)

Eshan Gupta (2019102044)

Keshav Bajaj (2019115010)

Key Requirements

1. Make a web interface to interact with the user and a backend server to handle queries
2. Take JSON dataset as input
3. Read column names from metadata and segregate them into primary, secondary, or tertiary filters
4. Provide a checkbox on every column name to provide filtering functionality
5. Get the targeted dataset from the backend server through API

Requirements Specifications

Design Document

Introduction

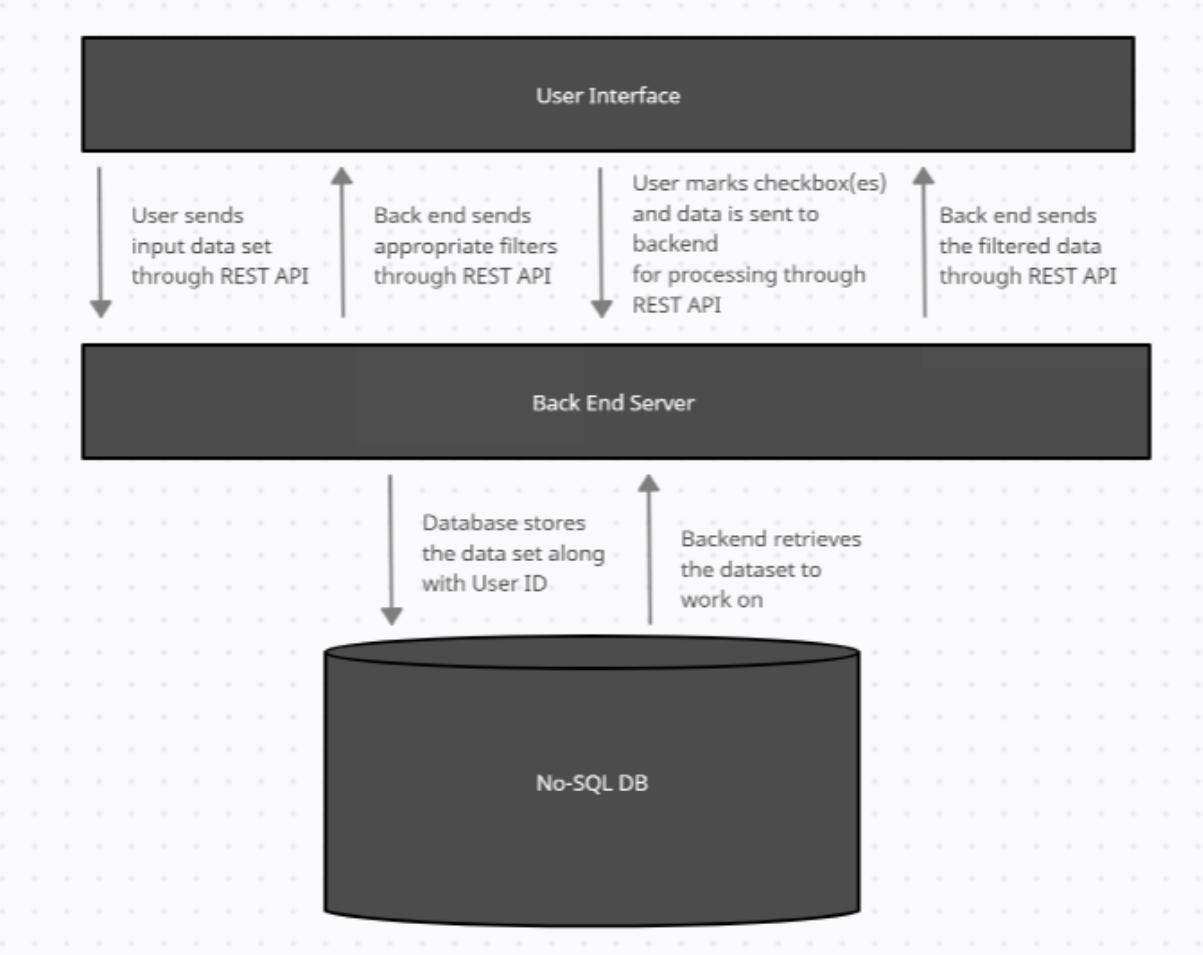
The project aims to return the user appropriate filters based on the input data set and use them in order to filter out the data as desired and download the targeted data set.

System Overview

The software system would include a basic front end that enables the user to input the data set, display the data set, select filters in a checkbox fashion and enable the user to download the targeted data set. The back end would support reading the metadata and segregating them

into primary, secondary, or tertiary filters. It would also be involved in filtering the data based upon user inputs on the filters.

High-Level Approach



Project Workflow

1. The use case begins when the user inputs the JSON dataset onto the web interface
2. Use Case: Verifying dataset format and retrieving metadata
3. The dataset and metadata are sent to the backend server
4. Use Case: The server reads the data and the metadata and segregates them into primary, secondary, or tertiary filters

5. The server sends the list of filters with their category to the frontend
6. The web interface displays the filters in 3 sections according to their level with a checkbox alongside filters
7. Use Case: The user selects the filters using the checkboxes and selects the desired value for the filter
8. After the user has selected their filters, the filter information is sent to the server
9. The server applies the filters on the dataset and sends the targeted dataset back to the interface
10. The targeted dataset is displayed on the web interface

Use Case: The user can download the targeted dataset in JSON and CSV formats

The above workflow presumes correct input and a running server, in case either of the aforementioned conditions is violated suitable error messages will be displayed.

UX

A basic front end that enables the user to

1. Input the data set
2. Display the data set
3. Select filters in a checkbox fashion
4. Enable the user to download the targeted data set.

High-Level API

REST API will be used for all the data exchanges between the backend and frontend.

Database models

A No-SQL database would be used to store the input dataset along with a user ID. User ID would be a string containing alphanumeric characters, while the dataset would be stored in accordance with the provided metadata.

Deployment Models

The project is expected to be deployed using free hosting services such as Heroku to host the application. Testing will be done on the local computers.

Tools/ Libraries used

The tools and libraries we are planning to use are Flask, MongoDB, and ReactJS. Additionally, for REST APIs we will be using the request library from python.