JAVA: Vedant Kothari, Aidan Wong, Alex Luo, Jonathan Metzler
SoftDev
P04: Makers Makin' It, Act II -- The Seequel
2025-03-25
Time Spent: 2
TARGET SHIP DATE: 4/25/25

DESIGN DOCUMENT (VERSION 0)

I. Description

Over time, obesity has become a serious issue, especially in the United States. In fact, since 1975, global obesity rates have nearly tripled. To spread awareness, we created a website that extrapolates information from various datasets to graph and visualize obesity by location, physical composition, and cases of death from obesity. In addition, our site analyzes previous cases of obesity to warn people and improve their quality of life by providing lifestyle change advice to avoid obesity.

A. Program Components

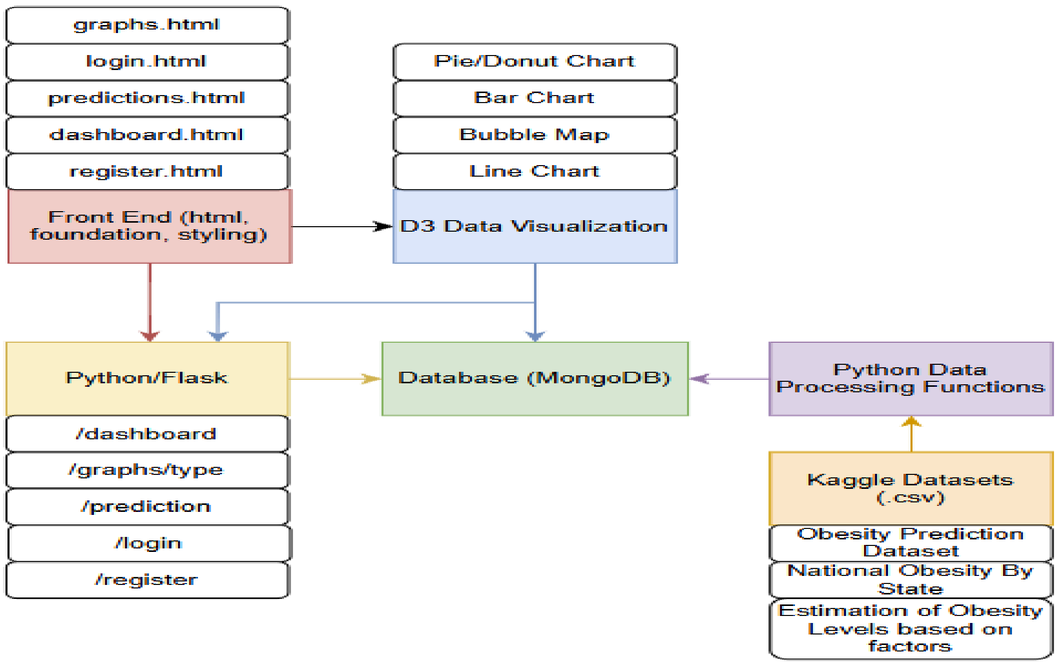
1. Flask/Middleware
    a. Handles URL routes and serves web app while managing session flow.
    b. Session Management: Keeps track of logged-in states and user data inputs for obesity predictions
2. MongoDB Database
    a. User Information: stores username, password, and lifestyle data, prediction records
    b. Obesity Dataset: stores aggregated data from the Kaggle CSVs (e.g., obesity levels by country/state, risk classifications, etc.)
3. Data Processing
    a. Python functions to homogenize data from the Kaggle datasets such that they have a consistent format to load into MongoDB
4. Obesity Prediction/Analysis
    a. Prediction module: user-entered data references the obesity dataset to generate a risk assessment
    b. Risk factor explanation: provides insights on the factors that are most significant for that user
5. D3
    a. Bar charts, line charts, pie charts, bubble maps, and spike maps to reflect obesity trends over time and across regions

    b. User-driven filtering to update charts based on specific parameters
6. Front End
    a. HTML templates to display data from Flask routes
    b. Foundation provides grids and styling for multi-screen UI
    c. Integrates with D3 such that data visualization is well structured within the UI

B. Program Component Connections

1. Flask <-> Database Helper Functions
    a. Calls prediction modules and integrates with MongoDB data processing(i.e. data retrieval)
    b. Serves data to the front-end
2. Flask <-> Templates/Front-End
    a. Flask serves pages with obesity data and prediction
    b. User interaction on templates triggers requests back to Flask routes
3. Templates <-> D3
    a. Templates embed D3 scripts to visualize obesity data
4. Data Processing Python <-> Database
    a. Python functions load and clean the Kaggle datasets into MongoDB
5. Database <-> D3
    a. D3 visualizes data directly from the database

C. Component Map

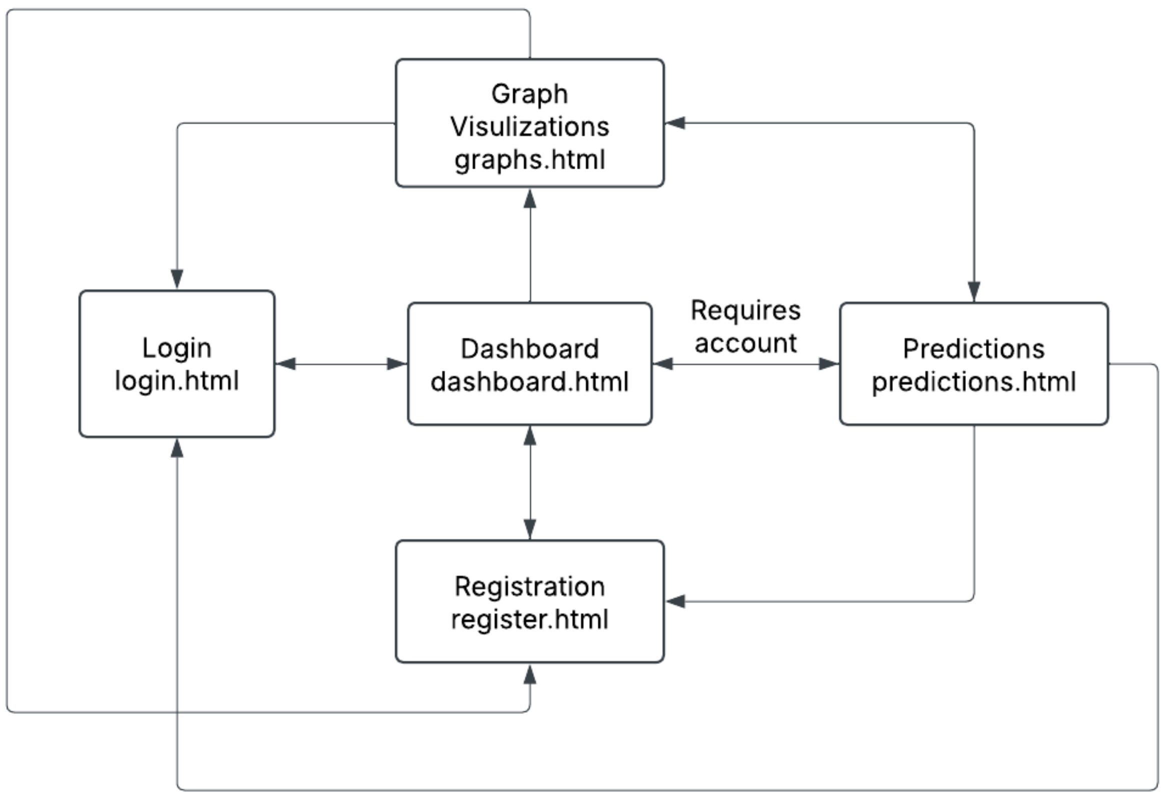


D. Database Organization

Mongo Document Database

```
{
  "_id": ObjectId("..."), unique ID
 "personal_information": {
        "gender": Male or Female
        "age": Integer
        "height": Float
        "weight": Float
        "family_history_with_overweight": yes or no
 },
 "lifestyle_factors": {
        "FAVC": Frequently consumes high-calorie foods (yes/no)
        "FCVC": Frequency of vegetable consumption (scale 1-3)
        "NCP": Number of main meals per day (integer)
        "CAEC": Food consumption between meals
        "SMOKE": Whether the person smokes (yes/no)
        "CH2O": Daily water intake (scale 1-3)
        "SCC": Monitors calorie intake (yes/no)
        "FAF": Physical activity frequency (scale 0-3)
        "TUE": Time spent using technology (scale 0-3)
        "CALC": Alcohol consumption frequency
        "MTRANS": Main mode of transportation
 },
 "obesity_level": The target label: Obesity level
 "location": {
        "country": Country of origin
        "state": State of origin
        "city": Optional field for city
 }
}
```

E. Site Map + Descriptions



1. Dashboard (/dashboard) - Displays user details (if logged in) and provides links to the graph visualization tool and prediction content
    a. User details include name, the user's biological information (if provided), and the prediction analysis that the user has run
2. Graph Visualization (/graphs/type) - Displays various graphs on different aspects of the obesity datasets
    a. Includes filtering and selection menus to show different kinds of graphs
3. Predictions (/prediction) - Uses datasets to predict a user's chances of obesity, risk factors, and prevention methods
    a. Requires the user to be logged in to use
4. Login (/login) - Allows the user to log into their existing account
5. Register (/register) - Allows the user to create an account

F. Frontend Framework: Foundation

1. Why Foundation?
    a. Easy-to-use and easy-to-follow tutorials

    b. A vast amount of CSS and JS components that will make things look and feel good
2. How Foundation?
    a. Grid System: Structure layout of pages like the dashboard and graph visualizers (Website will be able to adapt to size changes)
    b. Pre-designed Components: Buttons, Forms, etc
    c. JS Plugins: Modal Windows for confirmation messages

G. APIS

a. None at the moment

H. Data Visualization Library: D3

1. Why D3?
    a. Flexibility - Precise control over various parts of the graph, enabling highly customizable visualizations
    b. Data Binding - Binds data to DOM elements, allowing real-time updates
    c. Appearance - Looks extremely clean and has many built-in animations/interactable elements
    d. Declarative Syntax - Simplifies mapping data to graphical elements
2. How D3?
    a. Bar charts - Compare obesity rates among different demographics
    b. Line charts - Track how obesity has changed over time
    c. Pie charts - Show the distribution of the different types of obesity (overweight, extremely overweight, obese, etc)
    d. Bubble Maps - Show where obesity is more prevalent around the world/country

I. Datasets

1. Obesity level estimation: https://www.kaggle.com/datasets/adeniranstephen/obesity-prediction-dataset
2. National obesity by state: https://catalog.data.gov/dataset/national-obesity-by-state-d765a
3. Estimation of obesity levels: https://www.kaggle.com/datasets/aravindpcoder/obesity-or-cvd-risk-classifyregressorcluster -

J. Task Breakdown

1. PM + Frontend Lead - Vedant Kothari

    a. Frontend - HTML templates, CSS, and FEF implementation
    b. Facilitating clear communications
2. Graph Lead - Aidan Wong
    a. Graph Visualization → Work with Database and Middleware leaders for implementation
    b. User authentication functionality
3. Middleware Lead - Alex Luo
    a. Setting up Flask environment and creating functionality for the website
    b. Working with Database and Graph leaders to develop specialized functions
4. Database Lead - Jonathan Metzler
    a. Create a database using Mongo that combines CSV datasets from external sources
    b. Working with Middleware and Graph leaders to create necessary general functions to interact with the database