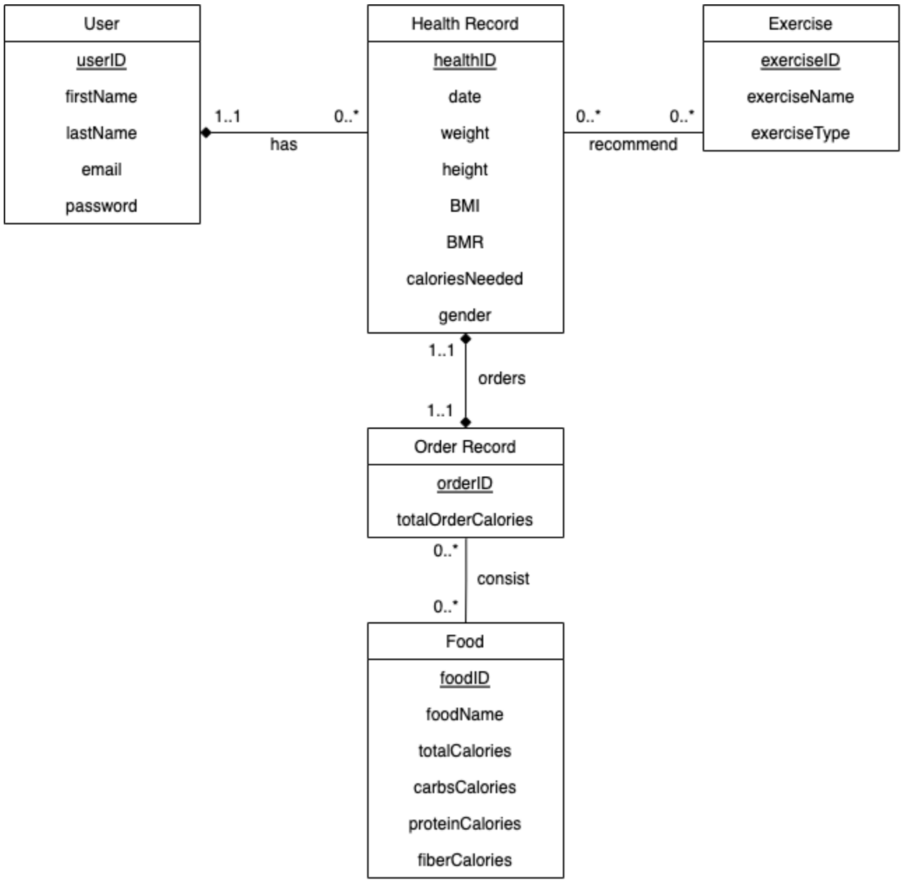
CS 411 Project Report

Team members- Gisella, Brenda, Heet, Ananya

Our original proposal was to create a calorie counter made easy. which we have committed to throughout the entire PT1. We made a few changes in functionalities for our app but the idea remained largely the same. The changes we implemented over time focused on providing the user with a more personalized calorie counter and help them achieve their desired goal by adding suggestive exercises and graphs to show their progress. In the calorie counter made easy the user is guided by steps on how to keep track of their calories and nutrients. In our final submission we have broken down calories into CarbsCalories, proteinCalories and FiberCalories.

The Data base for our project looks like this-



This application is useful in many ways as a lot of people want to be healthy, keep track of their calories and also know which food items and exercises can help them achieve that. Our program does exactly that in a way that a user can submit their BMI and know how many calories they need to take or reduce, which foods can help them achieve that and which exercises can aid them in reaching their desired weight. Some areas we could work more on are filtering calories, providing a better user experience and curating it according to each person and their health conditions. Right now our app depends on the average person and their body requirements but making it more personalized could be a feature we can implement in the future.

We discussed our UML and and highlighted the need for all the entities and their usefulness as follows-

1. USER

1. This will be a database for storing the user information regarding to log-in and sign-up

   Reason why we make it as an entity

2. To store all of the user information in one place, so users can keep track of their account.

2. HEALTH RECORD

1. This will be a database for storing the user information regarding its health condition at a certain time, characterized by healthID. Each user can have many health records.

Reason why we make it as an entity:

2. We want to use the information stored in health records to suggest exercises to the user based on their statistics.
3. We want to save the user's health record so we can display its progress (which can't be done if we only use the local storage). This progress is tracked per healthID and the time is saved in the date attribute.

3. ORDER

1. In this database, we will be storing the food orders made by the user at a certain time. An order is characterized uniquely by orderID. Reason we make it as entity:
   1. We want to keep track of what the user orders at a certain time (which is why each healthID is linked to exactly one order).

4. FOOD

1. This will be the database for all the foods taken from https://www.ars.usda.gov/northeast-area/beltsville-md-bhnrc/beltsville-human-nutrition-research-center/food-surveys-research-group/docs/fndds-download-databases/ and its nutritional value

5. EXERCISE

1. This will be the database for exercise taken from https://docs.google.com/spreadsheets/d/120azDT8KnASyr9HySl142pNp3dznGBbgbGC-KjLlpEE/edit?usp=sharing. Each exercise is characterized by an exerciseID, and we have additional attributes exerciseName and exerciseType . Reason why we make it as an entity:

1. So based on the health record, we can also suggest exercises to help them maintain/increase/decrease their BMI. For example, if the person's health record determines that he/she is underweight, we would suggest low intensity workouts from this database.

Changes to schema or source of data for application-

Our source of data and schema remained largely the same throughout as we tried to make sure we were filtering out the most useful data and entities in our first stage of the group project.The functionality we added to our schema later was to add BMI history so that the user can keep track of their BMI over time and we could present them with that information through our line chart as a part of our creative component.

Some changes we made and functionalities we added over the course of this project are -

1. Instead of having our app as a public food calculator, we decided to make it more personalized. This is a big improvement from the proposal, since now the user has a personal account where their BMIs and the date where they use the app is saved. With this change, they can view their histories, clear history, see their average, maximum, minimum BMI, and being able to change their password.
2. On our original proposal, we were going to do Table scan, however after learning about indexes, we implemented some indexing so that we don't have to process the entire data and directly jump to the index, which makes our queries faster.
3. We also wanted to take this health website to the next level by suggesting exercises based on their current BMI, so the user gets a more curated recommendation for them and gets their calories, food, and exercises in one place in a more streamlined manner. For example if they are underweight we suggest them low intensity exercises and if they are overweight we recommend high intensity exercises, and etc.

4. We removed the filtering component based on food calories range as we thought it might be redundant and doesn't really add significant value since we already have other 2 filter functions. We ended up having a search bar, and filter based on the nutritional value of the foods.
5. We removed the protein needed proposed on our original proposal so we can focus more into calories needed and to make our website more "easy-to-use"

How does our advanced programs complement our application

We added 2 Advanced queries and the reasons on how they complement our application are as follows:

Our first advanced query is the filter when displaying the list of foods. We have high carbohydrates, high protein, and high fiber. Since our website emphasizes a healthy diet and conscious eating, having this advanced query allows the user to easily select foods with high nutritional value based on their needs.

Our second advanced query allows the user to see their maximum BMI, minimum BMI, and average BMI at any given moment. This is helpful for people that want to keep track of their progress and have a visualization on how their eating and exercise habits impact their health. We also used a line chart library to display their BMI histories to create a more personalized feel.

What our application achieved or failed to achieve regarding its usefulness.

Achievements:

- Our app enables the user to maintain a healthy lifestyle by allowing them to keep a track of their BMI.
- It provides the user with recommendations for food items and exercises according to their BMI and the required calories by their body.
- Overall, the app has accomplished the goal to provide a healthy lifestyle to the people and aid in their diet.

Failures:

- Not being able to view past health records. Currently only a summary is being given (highest, lowest, and average BMI).

The technical difficulties all of us faced

Gisella - I initially had difficulties retrieving data from NodeJS and MySQL backend to be displayed in our React frontend since I have never worked with these frameworks before. Through this project I become familiar with CORS, JSON body parser, and API calls.

Brenda - I mainly had difficulties learning about react and how the pages render (using useState and useEffect). This is important because we want to make sure the API calls (executing the query) are called in the correct order since most of them are dependent on one another and if not done properly, a null value could be passed to the query and it would output incorrect results.

Heet - Initially, I faced many difficulties in connecting to the database using NodeJS and then retrieving the data using queries in the APIs I had created. Also, I didn't have any experience with NodeJS before so I had to learn the syntax for the backend code and setting up a server. Moreover, I also faced major difficulties connecting the server to frontend and displaying the data on our web page.

Ananya - I faced difficulties while creating advanced query blocks as I had no previous knowledge of it as much and solely understand how to create queries that function properly with our code. Also I faced difficulties while creating code block for frontend line chart as we had to create user BMI history to JSON and my teammates helped me with that.

Changes comparing the final application with the original proposal:

In addition to all the functionalities we decided on the original proposal, we added the creative component after our stage 5 which is a line chart for the History of the User's BMI along with their dates. This adds a more personalized touch to the application so the user can keep track of their progress. We also created a stored procedure with exercise types that enables the user to get the type of exercise they want; be it high intensity, cardio or low intensity, then automatically call the trigger to display if the calories are deficit or surplus.

Describe future work that you think, other than the interface, that the application can improve on:

- As for right now we only display exercise recommendations to the user, what we can improve on is to also attach the calories burnt for specific exercises and deduct that from the total calories of the user.
- Add history of exercise so they can revisit them.
- Add more food items or maybe even add recipes so that they can make low calorie dishes.
- Add quantity of the food in the food table, so that the user's food calorie intake is more accurate based on how much they are eating

Final Division of Labor

We divide the team into a front-end team and back-end team. Gisella and Brenda are on the front-end team and we are responsible for creating all the user interface and connecting the backend to the front-end. Towards the later stages, we also helped the back-end implementing the query to node js, making stored procedures and triggers.

Heet and Ananya were on the back-end team and were responsible for setting up the server, writing SQL queries, setting up the database, and linking the backend and frontend. In stage 5, they also devised the backend for the creative component.

Overall our team did a good job on working together in the team with few mishaps along the way due to personal schedules. We usually meet once or twice a week to divide the task, and update each other along the way when something is working/ not working.

What we did in further details:

Gisella:

- Did most of the components in the Landing page:
    - Taking a user's input for their health record and storing it into the database
    - Rendering data from Food database into the food table
    - Creating a summary of the user's statistics (whether they are on deficit or surplus based on the foods they chose)

    - Generate a user's health summary (history of their BMI)
    - Incorporate stored procedure and trigger to show recommended exercises for the user
- I also helped Heet write the backend APIs for each of these get and post requests in the Landing page

Brenda:

- Did SignUp and Login and all functionalities within such as: authentication, forgot password, incorrect email and password, routing.
- Implementing the Search feature for displaying the foods on the table.
- Implementing React's useState and useEffect to get user input before passing it to various api calls on the backend (such as handling checkbox, textfield, etc).
- Display exercise front-end.
- Working with Gisella to call the stored procedure from the front-end.
- Connecting the creative component line chart to the backend by doing some JSON parsing.

Heet:

- Worked on most of the backend
- Set up the connection to the database from the server file
- Implemented and developed SQL queries to retrieve the data from the database
- Set up the backend APIs for the login, signup, and landing page.
- Implemented the backend for the filters, advanced components, creative components, stored procedures and trigger
- Implemented the get and post calls in the backend and connect them with the frontend
- Helped Brenda to develop the creative component

Ananya:

- Created formulas for calorie calculation
- Implemented front end line chart for creative component
- Created advanced query such as history and search
- Worked with Heet on adding code block for order and delete

- Worked with Heet for registration and login backend