TomoMacro (Nutrient Tracker)

Team Members: Luke Pan, Kyle Shi

Status Summary

Work Done:

In the first week of the project, we set up all of our dependencies. We installed tailwind, ReactJS, NodeJS, Typescript, and Mongoose into our project and hooked up the frontend to the backend. We set up the schema of our database and linked our app with MongoWe implemented a basic login system such that user data (email, name, password, etc) is stored in our MongoDB database, and user passwords are hashed. We created the majority of our frontend UI (yet to be styled) with React. The main components of our application include the Recipes and Meals, which we've completed. We're still in the process of styling the UI and creating the final Date component.

Changes or Issues Encountered:

We had some issues getting everything to work together initially and there was quite a bit of a learning curve because we hadn't used React or NodeJS/Mongoose before. So the initial setup and tinkering took longer than expected. We're thinking that we might need to re-approach the design of some of our functionalities to include OO patterns because the technologies that we're using handle a lot of the features that we want.

Patterns:

So far the only pattern that we're using is the MVC pattern and the Observer pattern. The model in this case is our NoSQL database which we're using through MongoDB, and our View is the UI which is created with React, and our Controller is express/NodeJS. React updates its components whenever the state of a component changes using the Observer pattern.

Class Diagram

As the main parts of our application built out right now are the Frontend/UI, we do NOT have a UML Class diagram. Although we are using the Observer Pattern, this is a built in feature of the React library. React components are built upon the Observer pattern and it's not typical (from our understanding) to create UML Class diagrams of React code which is again, why we have not provided a UML Class diagram for this report.

Plan for Next Iteration

For the next iteration we want to complete the entire project. We will think about our features and see where we can utilize design patterns. We are thinking of implementing the Flyweight pattern for adding foods to meals as there could be similar foods being added as well as using the Factory pattern to create recipes/foods. We need complete user authentication so that the changes the user makes are able to be associated with the correct user in the database. Also so that a user will only be able to access their own data and not the data of other users. We also need to link the frontend components that we created to the backend endpoints. Next, we need to assemble the final user interface using our react components and style it using tailwind. Finally, we will test and then publicly host the final product.

Our plan in order below:

1. Complete frontend react components
2. Plan out which OO patterns to use for backend
3. Code out backend
4. Connect frontend to backend
5. Test
6. Host