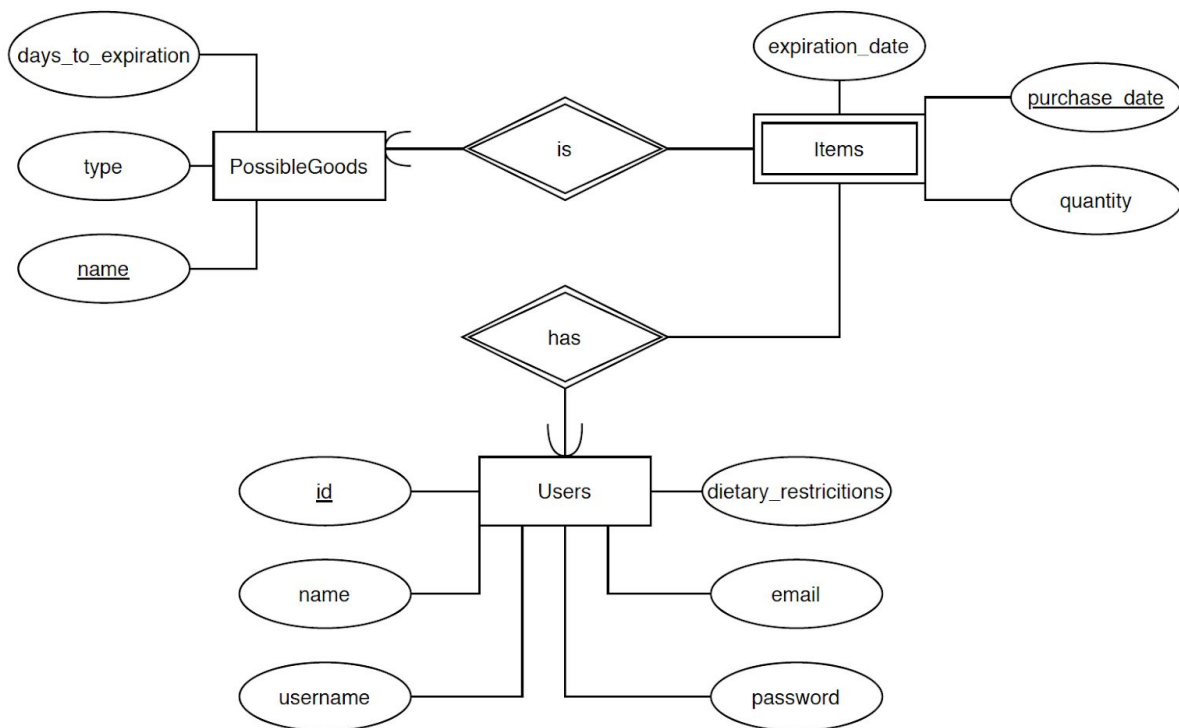


Progress Report:

New assumptions, E/R diagram, and list of tables (if they have changed since Milestone 1):

Since the last milestone, a few changes have been made to the E/R diagram and schema of the database. To begin with, the 6 dietary restrictions (pescetarian, vegetarian, vegan, dairy_free, gluten_free, and peanut_allergy) in the Users table were replaced with a single array that contains a binary variable indicating whether or not that user has a dietary restriction. Furthermore, the Recipes relation was completely abandoned. This was because recipes will be found using the Edamam API, opposed to being stored in the database and found via a query. An updated E/R diagram can be seen below:



Since we are using Firebase for users (and potentially the rest of our DBMS), there will need to be some changes in the SQL files to accommodate this cross platform functionality. We will continue to explore what DBMS is optimal for our purposes.

A brief description of the platform you chose in Task 6. Not much details are needed if you are using Flask-based platforms.

We chose to use React, specifically, the create-react-app system for easy functionality purposes. This is because some group members had previous experience working in React, so we found that this method would be the quickest and most efficient way to work on the frontend. Additionally, create-react-app comes preinstalled with bootstrap template functions, as well as other needed files, so that we could quickly get started on this project without having to waste time downloading every essential file individually. While this is not as well suited for heavy, high-traffic websites as React.js Boilerplate, we felt that for this small-scale, starter project, it would be appropriate to use a platform that was easy to set up and work with, as well as perform the functions we needed.

For the backend, we chose to use Google's Firebase. This framework allows for straightforward user authentication and provides Firestore as a cloud database that implements NoSQL. This database will eventually store all of our data, such as the foods in each user's fridge. We chose Firebase as it was quite simple to learn and use, with API calls easily implemented into our React.js code.

Changes you made to the database during performance tuning in Task 8, e.g., additional indexes created.

There were a couple changes of changes made to the database during performance tuning in Task 8. In order to improve the performance of the database, two indexes were added: one for the food type in the possible goods database, and one for the expiration dates in the fridge database. In general, our database has a lot of unique fields that act as indexes, but the two added help with improving performance.