HoMePIT

Erik Anderson, Jerome Chestnut, Maxwell Fugette

Dept. Of Computer Science and Information Technology

Abstract

Menu planning is an inescapable and oft time-consuming part of human existence. Some individuals attempt to avoid it by subsisting on fast and highly processed food, while others try their best to manage it using whatever tools are available to organize and categorize this part of their lives. With healthy eating being the preferred option, it is imperative for it to be made as simple and convenient as possible. We believe that many of the tasks associated with meal planning, such as the tracking of household ingredients and foodstuffs, planning what meals to eat on what days, and the composition of shopping lists are often too tedious and time-consuming for the average person, which will inevitably lead to unhealthy dietary choices. All of this can be simplified with the appropriate application of technology. HoMePIT is our attempt at making this simplification.
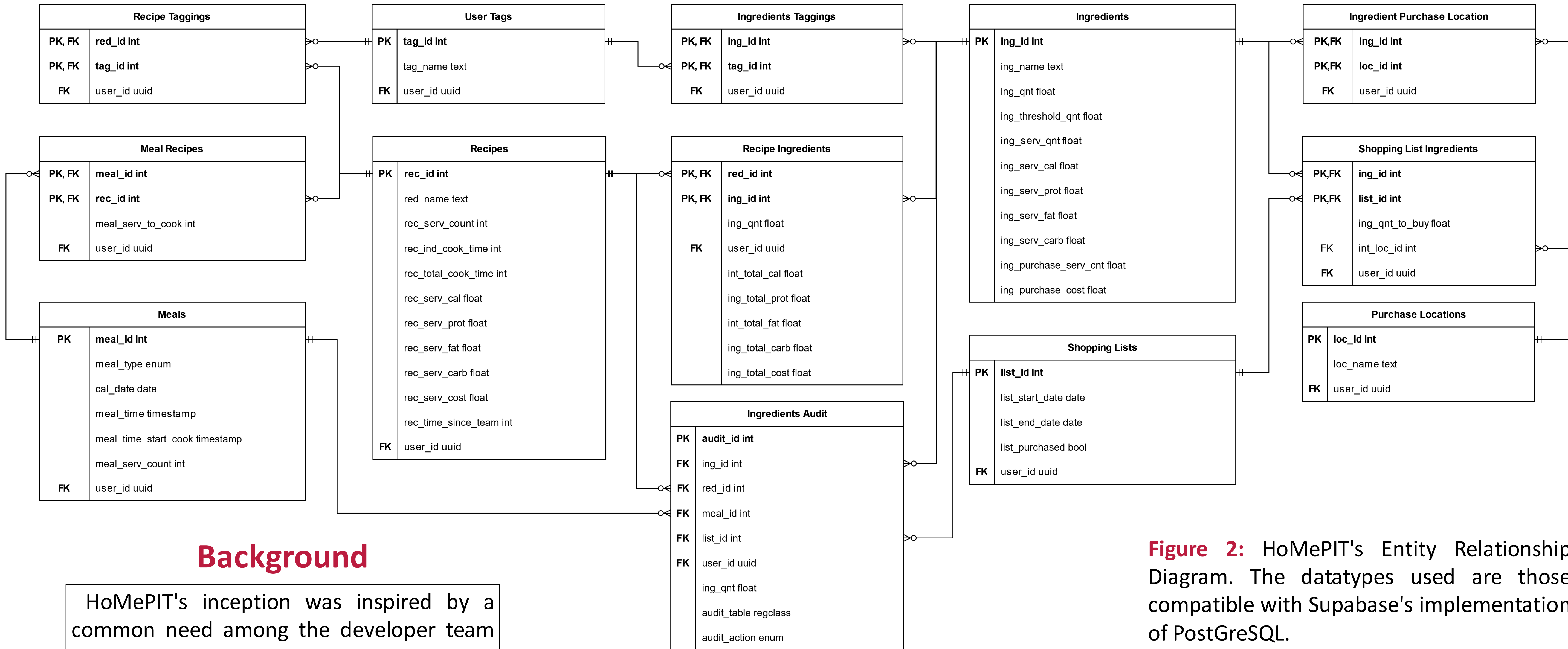
Figure 1: HoMePIT's user interface utilizes MaterialUI and Bootstrap for its look and feel, as seen here in the Pantry page as viewed on a mobile device.

Background

HoMePIT's inception was inspired by a common need among the developer team for a simple application to assist in meal planning and ingredient tracking. Most applications of the same nature as HoMePIT go well beyond the scope of what is necessary for our purposes, often resembling fitness applications more so than meal applications. One such example of this is MyFitnessPal, which while accomplishing mostly the same task as HoMePIT resembles a fitness app first and foremost. The developer team wanted a simple and sleek application that does only what it says it will, avoiding fluff and filler features that will take away from its core purpose.

Technology

HoMePIT utilizes numerous technologies to enable its functionality. First, HoMePIT's design was implemented using the open-source development framework Next.js, which in turn enables the creation of React-based applications. The functionality of Next.js and React were enhanced through the usage of Material UI, which better enabled the creation of useful and appealing user interfaces. HoMePIT's CSS was developed using the Bootstrap framework. To enable the implementation of an interactive calendar within the Meal Planner, the FullCalendar library was used.

HoMePIT's website is hosted using Firebase Hosting. The database backing it uses the PostgreSQL technology and is hosted using the cloud database service SupaBase. Supabase provides functionality such as social-based authentication, row level security, and security policies.

Figure 2: HoMePIT's Entity Relationship Diagram. The datatypes used are those compatible with Supabase's implementation of PostGreSQL.

Design

HoMePIT was designed with the goal of being a web application compatible with both mobile and desktop browsers, with focus being placed on ensuring compatibility with the Firefox browser on Android and Windows 10. Its visual design was derived with the intent of having a modern, reactive appearance. To organize HoMePIT's functions, several pages are available for navigation and use, as listed below.

Pantry
The Pantry page contains a list of ingredients specified by the user, with each ingredient containing information relating to its name, quantity, macronutrient amounts, etc. There are options to edit and remove current ingredients.

Recipe Book
Like the Pantry, the Recipe Book page consists of a list of recipes which the user has created, with relevant information such as cook time, serving amount, macronutrients, and so on.

Meal Planner
The Meal Planner page contains an interactive calendar in which the user can specify meals for any given day using the recipes a user has added. Data such as total monthly cost and nutrition is displayed here as well.

Shopping List
The Shopping List creates shopping lists based off of what ingredients need purchasing to satisfy the meals within a the time period that the list is set to satisfy.

Future Work

Early in development, many features were identified that would improve the functionality of HoMePIT yet were unlikely to be implemented due to scope and time constraints. Some examples include being able to input ingredients with barcodes, the ability to import recipes from websites, generating meal planes based on user-specified constraints, and having a shopping list minimize the number of stores that need to be visited. Likewise, during development, some features were found to require excessive development time and were cut for the initial release of HoMePIT. These features included leftover tracking, expiration date tracking and automatically subtracting expired ingredients from the Pantry. All of these features would be reasonable to implement in the future and would improve HoMePIT accordingly.

Acknowledgements

References

1. Next.js - https://nextjs.org/
2. Supabase - https://supabase.com/
3. Firebase - https://firebase.google.com/
4. FullCalendar - https://fullcalendar.io/
5. Material UI - https://mui.com/
6. Bootstrap - https://getbootstrap.com/