

Grocery Comparer

Alex Min

@alexsnowm on GitHub

Description

In a nutshell, this is a web app that tells you where the cheapest groceries are.

I noticed that when grocery shopping, there was no easy way for me to calculate which store offered the lowest price per unit for an item without doing it by hand, especially if an item is offered at different units of measure (lb vs. oz).

This app does that for you.



Features

- Can browse all items and stores inputted by user, displaying current lowest prices for items and addresses for stores
- Can view individual items and stores, displaying detailed info, such as aisle, item price history, and user notes
- Can see the current lowest price for an item from its price history, converted to or matching the user's chosen unit of measure if possible



Planning - User Stories

- Users can create, update, and delete items and stores
- Users can create and delete prices for an item
- Users can see the current lowest price for an item
- Users can select a unit of measure to convert the current lowest price to
- Users can view an individual item and its info, like stores offering it and user notes
- Users can view an individual store and its info, like its website link and user notes



Planning - Database

The database includes:

- Items table - holds name, notes, the number ID of the current lowest Price, and a one-to-many relationship to Prices
- Prices table - holds price in decimal and formatted String form, date added, item's aisle, and the price's unit of measure
- Stores table - holds name, phone, website link, address in String form as well as broken into street, city, state, and zipcode, and a one-to-many relationship to Prices



Technology Stack

- Java 8
- Spring Boot
- Thymeleaf
- MySQL
- Bootstrap
- Hibernate



Demo



What I Learned

- Better understanding of Java persistence
- Spring Boot, Hibernate, MySQL - learned more validation functionality, how data interfaces from the View to the database and back again, and how tables interact with each other in relationships
- Thymeleaf - learned more conditional and literals functionality
- Learned more Model-View-Controller (MVC) best practices and how to implement it



What's Next

- Shopping lists - allow users to export selected items and prices to a printer-friendly PDF or as a formatted email for offline reference at the store
- User login - allow different users to maintain and view their own list of items and stores

