Market Tracker

Application For Tracking Variation of Prices Across Different Supermarkets

André Graça, n.º 47224, e-mail: a47224@alunos.isel.pt, tel.: 929130440
Daniel Caseiro, n.º 46052, e-mail: a46052@alunos.isel.pt, tel.: 911124858
Diogo Santos, n.º 48459, e-mail: a48459@alunos.isel.pt, tel.: 939046442

Supervisor: Filipe Freitas, e-mail: filipe.freitas@isel.pt

28 of February of 2024

1 Introduction

This project aims to provide users with a comprehensive web service for comparing supermarket prices, promotions, and product insights. Our platform includes a rating system, advanced filters, a price and promotion tracking feature, empowering users to make informed purchasing decisions. The system will be designed with 2 types of users in mind: **operators** and **clients**.

Both will be to compare prices across various supermarkets and monitor price fluctuations over time, but each has a unique set of functionalities. An **operator** will be able to perform insertions and updates of products in their store(s), while **clients** can create a personalized shopping cart based on their preferences, such as company, price, city, etc.…

For example, an **operator** who represents a certain supermarket can update a price of a chocolate bar in his supermarket, making it the cheapest in our application. Market Tracker will support a graphic representation of the product's price variation. Considering the previous update, users will now be able to search by chocolate bars applying a lowest price filter, obtaining that one chocolate bar. Finally, a **client** can add this chocolate bar to their cart.

2 System Requirements

2.1 Functional Requirements

- Users can authenticate in our application Google´s OAuth 2.0 or by creating an account.
- Users can check the graph of variation of price of a product during a period.
- Operators:
  - Insert and update products, only the ones that belong to them.
  - Insert their stores in our DB.
  - Insert promotions across various products in their stores.

- Clients:
    - Fill a Cart with products.
    - Rate and comment products.
    - Compare and search by filtering products by various keywords (e.g. brand, store, overall rating, etc…).
    - After completing the cart then can procced to generate a digital grocery list which will tell how much they will spend in each store and how much stores they need to go to fulfill the cart. They can also generate the lower total price of a basket.

2.2 Non-Functional Requirements

- Software testing to increase our confidence in the system's security and features. It should also allow us to move faster without fear of breaking the previous working code.
- Responsive design in our web application.
- Overall good software practices that should allow the system to grow in complexity more easily.
- Deploying the application components and making them available to the public.

2.3 Optional Features

- Map application to tell the smallest route between each store.
- Android application.

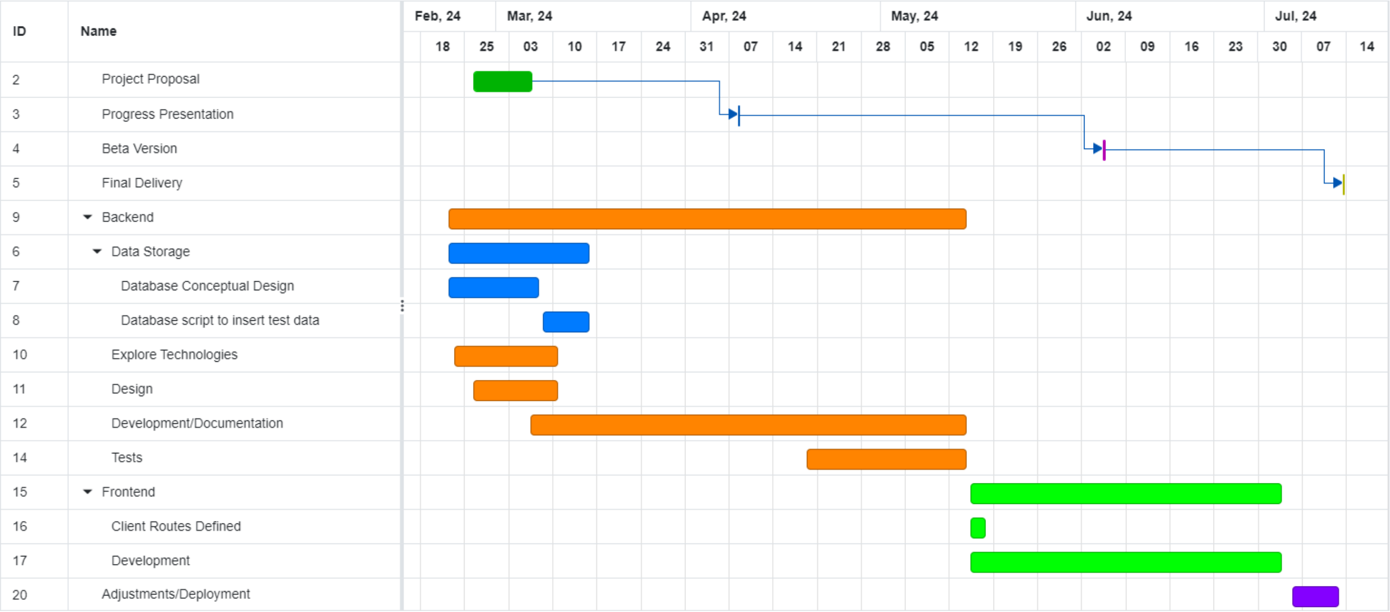
3 Technologies

We plan to use C# [1] alongside it's Entity [2] and MVC framework [3] to serve application backend service. The database we will use a persistently store application like PostgresSQL [4]. The web application we will use Html [5] and TypeScript [6] for web pages, as well NextJS [7] (typescript framework for frontend development). If we can develop a mobile application, it will be developed in Jetpack Compose [8].

4 Risks

Developing the backend service can take longer than usual thanks to our inexperience with C# programing language as well its frameworks (Entity framework and MVC framework) and the same can happen with in the frontend when experimenting with the NextJS framework.

5 Planning



ID	Name
2	Project Proposal
3	Progress Presentation
4	Beta Version
5	Final Delivery
9	Backend
6	Data Storage
7	Database Conceptual Design
8	Database script to insert test data
10	Explore Technologies
11	Design
12	Development/Documentation
14	Tests
15	Frontend
16	Client Routes Defined
17	Development
20	Adjustments/Deployment

Figure 1: Gantt Chart Project Plan

References

[1] https://learn.microsoft.com/en-us/dotnet/csharp/ - C#, a programming language on the .NET platform, last accessed on 29 February 2024.

[2] https://learn.microsoft.com/en-us/ef/ - C#'s Entity Framework, the framework for a high-level data access, last accessed on 29 February 2024.

[3] https://learn.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/overview/asp-net-mvc-overview/ - C#'s MVC Framework, the framework for architectural pattern base application, last accessed on 28 February 2024.

[4] https://www.postgresql.org/ - PostgreSQL, a persistent store application, last accessed on 28 February 2024.

[5] https://developer.mozilla.org/en-US/docs/Web/HTML/ - HTML, HyperText Markup Language, last accessed on 29 February 2024.

[6] https://www.typescriptlang.org/ - TypeScript, javascript with syntax for types, last accessed on 29 February 2024.

[7] https://nextjs.org/ - NextJS, the react framework for the web, last accessed on 29 February 2024.

[8] https://developer.android.com/jetpack/compose/ - Jetpack Compose, a toolkit for building native UI, last accessed on 29 February 2023.