ISEL
INSTITUTO SUPERIOR DE
ENGENHARIA DE LISBOA

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 Android application for comparing supermarket prices, promotions, and product insights. By offering a mobile platform, it aims to provide users with a more convenient and accessible alternative to traditional website-based comparisons, enhancing the shopping experience while on the go. Our platform will include a rating system, advanced filters, and a price and promotion tracking feature so anyone can access organized, filtered and valuable information through a user-friendly interface empowering more informed purchasing decisions.

While leading platforms like **KuantoKusta** already provide valuable price comparison services, we identified key areas for improvement and differentiation. Our project not only offers price comparison but also integrates features such as a dynamic user rating system and more advanced filters to provide users with a more holistic shopping experience, e.g. maximum number of supermarkets to visit and save lists to later compare prices of products users buy regularly.

By addressing these gaps and building upon existing solutions, we aim to set ourselves apart in the market while also contributing to the evolution of the supermarket shopping landscape. Our motivation lies in creating a platform that truly meets the needs of modern consumers, facilitating smarter shopping decisions and acknowledging the challenges of its development.

2. System Requirements

The system will be designed with 2 types of users in mind: **operators** and **clients**. Both will 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 available in their physical store(s), while **clients** can create a personalized shopping cart based on their preferences, e.g. such as company, price and city.

2.1 Functional Requirements

Both clients and operators have the option to authenticate using their own credentials or via Google's OAuth 2.0 for clients. Additionally, they can access price variation graphs for specific products over a defined period.

- Operators:
  - Insert, update and list products.
  - Insert their stores in our DB.
  - Insert promotions across various products in their stores.
- Clients:
  - Compare and search by filtering products by various keywords (e.g. brand, store, price, rating, etc.).
  - Share a Cart with friends/family.
  - Fill a Cart with products.
  - Rate and review products.
  - 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 cart and reduce the number of stores they must visit.

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.
- 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

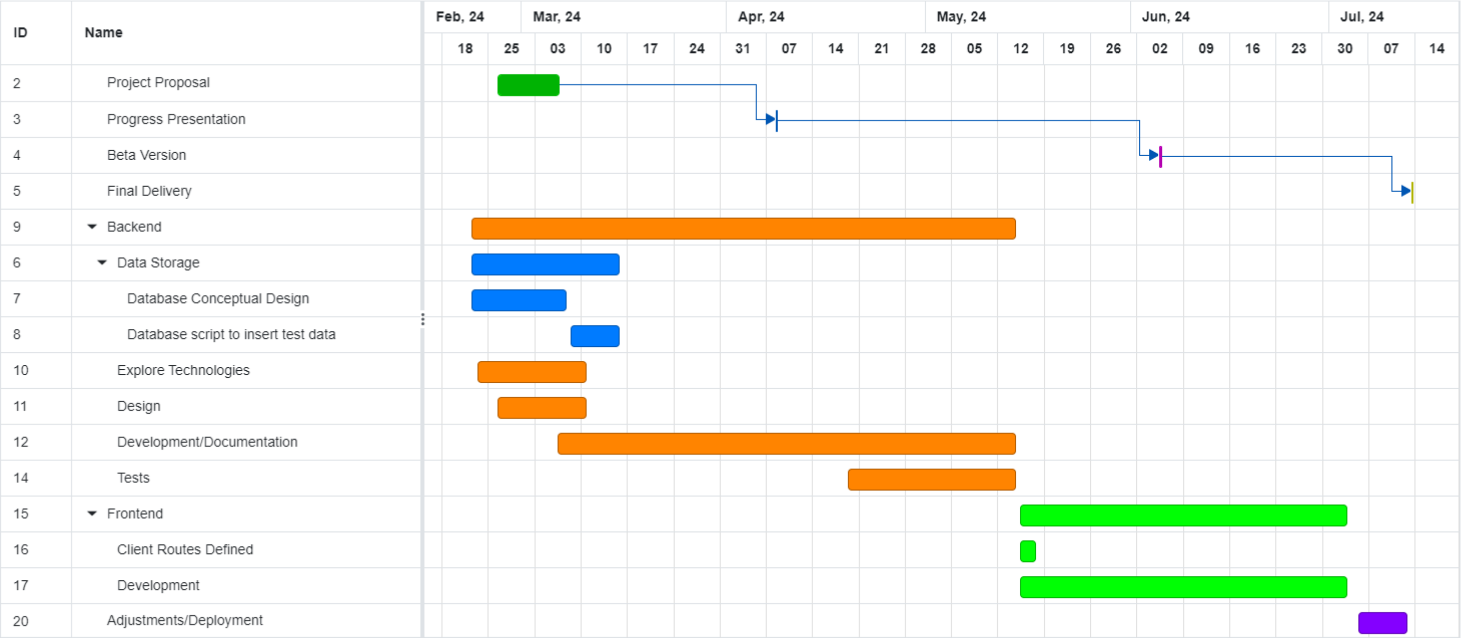
- Clients will be able to insert products.
- Clients can share their low-cost recipes and other clients can add all products in that said recipe to their carts.
- Operator's products stats.
- Admin role to accept admission of an Operator.
- Users can have a community where they share lists that they choose to.

3. Technologies

We want to take this opportunity not only to learn new languages and technologies but also apply what we learned along this course over the years.

To accomplish this, we plan to build an organized .NET backend using C# [1] along with its Entity [2] and MVC framework [3], enabling us to take some time to learn and explore these new technologies. To manage data, we'll rely on PostgreSQL [4], a familiar relational database. For the web interface we will use the React [5] library writing TypeScript [6] to take advantage of its compiler type check. The android application will be developed using Jetpack Compose [8]. To test all the backend services, we'll rely on unit tests using the Moq [9] library. The final step will be to build the Android application (apk file) and deploy our back and frontend in the cloud, we're still considering choices like Render [10] or Heroku [11] to leave information available 24/7.

4. 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

Backend: 22/02 – 14/05 (**82 days**)

Frontend: 15/05 – 03/07 (**49 days**)

Deploy: 05/07 – 12/07 (**7 days**)

References

[1] https://learn.microsoft.com/en-us/dotnet/csharp/ - C#, a programming language on the .NET platform. – last accessed on 15 of March

[2] https://learn.microsoft.com/en-us/ef/ - C#'s Entity Framework, the framework for a high-level data access. – last accessed on 15 of March

[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 15 of March

[4] https://www.postgresql.org/ - PostgreSQL, a persistent store application. – last accessed on 15 of March

[5] https://www.react.dev/ - React, a library for web and native user interfaces. – last accessed on 15 of March

[6] https://www.typescriptlang.org/ - TypeScript, JavaScript with syntax for types. – last accessed on 15 March

[7] https://vitejs.dev/ - Vite, a build tool that enhances the front-end development experience. – last accessed on 15 March

[8] https://developer.android.com/jetpack/compose/ - Jetpack Compose, a toolkit for building native UI. – last accessed on 15 March

[9] https://github.com/devlooped/moq - Moq, the most popular and friendly mocking framework for .NET. – last accessed on 15 March

[10] https://render.com/ - Render, a unified cloud to build and run all your apps and websites with free TLS certificates, global CDN, private networks and auto deploys from Git. – last accessed on 15 March

[11] https://www.heroku.com/ - Heroku, a platform as a service (PaaS) that enables developers to build, run, and operate applications entirely in the cloud. – last accessed on 15 March