جـامعة بيروت العربية
BEIRUT ARAB UNIVERSITY

Faculty of Science

Department of Mathematics and Computer Science

Senior Project - I (CMPS 443)

ePantry: A Digital Pantry to Organize Your Life

Student Name:	**Student ID:**
Salwa Hammoud	202101663
Mohammad Chatila	202102096
Shireen Al Sabea	202101672
Khaled Abdul Samad	202100287

Supervisor:

Dr. Lama Affara — Principal Supervisor

May 30, 2023

Acknowledgements

As we reach the culmination of our academic journey, we are filled with a sense of accomplishment and profound gratitude. Today, we dedicate this letter to express our deepest appreciation to you, Dr. Lama Affara, for your unwavering support, guidance, and belief in our abilities throughout the entire process of completing our senior project.

From the very beginning, your unwavering mentorship and encouragement have been pivotal in shaping our project and transforming it into a remarkable endeavor. Your passion for the subject matter, your expertise, and your commitment to nurturing aspiring minds have been truly inspiring. Your invaluable guidance has not only enhanced our knowledge but also instilled in us the confidence to explore new horizons, think critically, and overcome challenges.

Furthermore, we would like to express our heartfelt gratitude to our family and friends for their unyielding support, encouragement, and patience throughout this demanding endeavor. Their unwavering belief in us, their love, and their constant motivation have been the driving force behind our success. We are blessed to have such a strong support system, and we dedicate this project to them as a testament to their unwavering support.

We are also deeply grateful to the esteemed faculty members and mentors who have played a significant role in shaping our academic journey. Their passion for teaching, their dedication to academic excellence, and their commitment to nurturing students' potential have had a profound impact on our intellectual growth. We are grateful for the opportunities they have provided us to expand our knowledge, challenge our assumptions, and foster a love for lifelong learning.

With utmost gratitude and warm regards,
ePantry’s Team.

Abstract

Did you know that one-third of all the food produced for us to eat is wasted or lost globally? It's a staggering amount—about 1.3 billion tons per year, worth a whopping US$1 trillion! This wastefulness not only harms the environment but also puts a strain on our wallets. That's where ePantry comes in. It's an incredible app designed to tackle the problem of food waste caused by spoiled items in our pantries.

ePantry helps us keep track of what we have in our pantry and makes sure we use up the perishable items before they go bad. By managing our pantry inventory, we can reduce the amount of food that ends up in the trash and be more mindful of what we consume. And speaking of saving, ePantry helps us cut down on our grocery expenses too. By keeping track of our pantry inventory and preventing duplicate purchases, we can avoid wasting money on unnecessary items. The app allows us to quickly check what we already have at home, so we don't end up buying more than we need.

But here's the icing on the cake, ePantry offers us the option to browse recipes based on the ingredients we already have in our pantry. No more staring blankly to our supplies wondering what to cook. The app previews creative and delicious recipes that we can whip up using what we already have. It's like having a personal chef right at our fingertips!

In a nutshell, ePantry is a game-changer. It helps us reduce food waste, saves us time and money on grocery shopping, and even sparks culinary inspiration with its recipe recommendations. Let's embrace this innovative solution and make our lives easier, while also taking a step towards a more sustainable and efficient approach to food management.

Table of Contents

Chapter 1. Introduction

A. Main Project Description

How often do you find yourself buying groceries that you already have at home or forgetting to buy certain ingredients? Well, you are not alone! ePantry is a mobile application that aims to solve these problems among others. Our application will make your life easier by providing you with a digital version of your pantry. Every item that is currently in your pantry or your fridge will be shown clearly on your device. This way you can check what groceries are missing from your house even if you were not home. Moreover, ePantry will allow you to browse multiple recipes with ingredients you already have on hand. This project can be implemented on a very large scale with multiple advanced features. We could personalize recommendations of recipes based on preferences or health issues or allow users to buy their groceries through our application in future updates.

B. Problem Statement

In the realm of home automation, little progress has been done in terms of improving organizing systems for consumer-purchased items. Lack of effective organization can cause a lot of problems for normal consumers, especially when it comes to perishable food products. The amount of time a consumer spends food shopping is the first problem. This takes into account both the actual time spent shopping and any prospective return journeys for things that were left behind. Where if we suppose that any customer spends at least 1-2 hours weekly for grocery shopping, this will result in wasting up to hundreds of hours shopping in a year. Standard grocery lists help with order and efficiency, although it's still possible that you'll forget something or buy something you've already bought twice. Such repeated purchases may result in unused foods going bad and being thrown out, which contributes to needless food waste. So the main goal of this project was to develop an active inventory tracking pantry system that cuts down on food waste and allows for better organization of food.

C. Project Goal

The main purpose of our project is to create a platform to track purchases and usages of household groceries for families and shared households. ePantry also highlights solving the issue of food waste since we will be tracking everything that exists in the user's fridge. We will be notifying the user of products' expiration dates which results in reducing food waste and thereby helps in saving the world's resources. Furthermore, ePantry will increase awareness through information on shelf life and nutrition and by filtering and previewing a few recipes where the user can easily decide what to eat considering sometimes it is a confusing choice. Last but not least, the user will be saving time and money where he will be able to know what's available in his/her fridge at all times. As a result, the user won't go shopping every time he/she finds out something is missing or buys something twice resulting in food and money waste.

D. System and Domain Review

The basic idea on which ePantry was built isn't something new. However, existing applications lack certain features that we hope our system can provide. Among these existing systems, the most commonly used are Cooklist, KitchenPal, and PantryCheck. Summarized below is the availability of a few features present in our application. As shown, none of these three applications have all the features found in ePantry. Our users will be able to scan their receipts to insert items, browse recipes, view reports on savings, and create a favorites list of items they do not want to run out of.

Criteria	Cooklist	KitchenPal	PantryCheck	e-Pantry
Scan Receipts	✔	✘	✘	✔
Browse Recipes	✔	✔	✘	✔
View Reports	✘	✘	✘	✔
Favorites List	✘	✘	✘	✔

Chapter 2. Project Plan

A. SDLC Model

To ensure organization throughout planning and implementing, we used a software development lifecycle model that properly fits ePantry. Following through with an SDLC model allowed us to manage our project, divide and properly assign tasks, and assess and manage risks. Scrum is an agile development methodology; it emphasizes teamwork, accountability, and iterative progress towards a well-defined goal. Moreover, scrum is an adaptable, fast, flexible, and effective framework that is designed to deliver value to the customer throughout the development of the project. All these characteristics make scrum the perfect model for us. We had weekly sprints that are preceded by a meeting where we will specify each sprint's tasks. Our development team consists of us four – Salwa, Mohammad, Khaled, and Shireen. The team met the scrum master – Dr. Lama Affara, our supervisor – weekly to ensure that the process is on the right track.

B. Project Organization

One of the most important aspects of implementing a project is specifying every member's role. To do so, we teamed up and discussed what task should every member be responsible of. Khaled was responsible for the design; that includes UX/UI design. He made sure the designs of both the application and the website are user friendly. Next, Shireen created the database structure and models. This made it easier to implement the backend of the application. Finally, Mohammad and Salwa were responsible for the frontend of the application.

C. Ethical standards and guidelines

Ethics are the foundation of any business. Therefore, it is our top priority to ensure that ePantry meets the highest professional standards possible. We will always act consistently with our community's interest, be fair to and supportive of every team member, participate in lifelong learning, and promote an ethical approach for ourselves. Moreover, ePantry will:

- guarantee its user's privacy and confidentiality is well maintained by protecting their data.
- ask the user for authorization before accessing their cameras to scan receipts.
- preview user-appropriate advertisements.
- not access the user's locations before future updates where location may be needed.

D. Schedule/Timeline

Time (in Weeks)	Task	Team Member(s)	Progress
Week 2	Generate an idea	-	100%
Week 2	Competitive market research	Salwa/Khaled	100%
Week 3	Project Proposal	-	100%
Week 4	Discuss idea with Advisor	-	100%
Week 4	Project Description	Salwa/Mohammad	100%
Week 4	Problem Statement	Shireen	100%
Week 4	Project Goal	Shireen	100%
Week 5	System and Domain Review	Salwa	100%
Week 6	SDLC Model	Mohammad	100%
Week 6	Project Organization	Mohammad	100%
Week 6	Ethical Standards and Guidelines	Salwa	100%
Week 6	Timeline	Khaled	100%
Week 6	Feasibility study	Khaled	100%
Week 6	Project Plan Report	All	100%
Week 7	Present to the Committee	Khaled	100%
Week 8	Software Requirements	Salwa/Mohammad	100%
Week 9	Use case Diagram	Shireen	100%
Week 10	Tech stack	Shireen	100%
Week 11	Data Flow Diagram	Shireen/Salwa	100%
Week 11	Database Diagram	Shireen/Salwa/Mohammad	100%
Week 12	UML Diagram	Shireen	100%

Week 12	User Interface Prototype	Khaled	100%
Week 13	Sequence Diagram	Shireen	100%
Week 14	Final Report Draft	All	100%
Week 15	Present to the Committee	All	100%
Week 15	Final Senior I Report	All	100%

E. Feasibility Study

The main purpose of this feasibility study is to check if the proposed system is viable especially in this economy. The result of this feasibility study will determine whether to proceed with this project, terminate this project, or change a few things to fit the economy. To study the feasibility of our project, we will analyze multiple aspects including operational, technical, and economic feasibility.

a. Risk Management

Risk management is the process of minimizing any potential problems that may negatively impact a project's timetable. 'Risk' is any unexpected event that might affect the people, processes, technology, and resources involved in a project.

There are three main classifications of risks which can affect a software project:

- Project risks: Project risks are concerned with different forms of budgetary, schedule, personnel, resources, and customer-related problems. The project risks we faced are:
  - Scope creep which occured when the initial project objectives weren't well-defined. We had too many features and too many ideas at the beginning. To control this risk, we created clear project parameters from to strengthen the project scope, agreed upon the project scope, and communicated that vision from the beginning which left less room for scope creep. We also scheduled regular progress check-ins that can also ensure the project stays in line with the original project scope.
  - Low performance which occurred when the project didn't perform as well as initially expected. While you can't always identify the root cause of low performance, we were able to identify the reason being our app was taking too long to load. To control this risk, we used a package that puts images in cache so they won't have to be downloaded every time our users open the app. This enhanced ePantry's performance.
  - Time risk, also known as project schedule risk, is the risk that tasks in your project will take longer than expected. Delayed timelines might impact other things like your budget, delivery date, or overall performance. Therefore, to mitigate time risk, we overestimated the time needed to complete various tasks for various features in the planning phase and build in time contingency. That way, we were able to be one step ahead of our deadlines.

- Technical risks: Technical risks are concerned with potential method, implementation, interfacing, testing, and maintenance issue. The technical risks we might have faced is:
  - Loss of data which cannot be restored is a major technical risk. This risk is too dangerous to face; therefore, we followed various strategies to control this risk before it even occurred. Regular data backup is one of the critical strategies that we implemented. When backing up data, ensure that we have several

copies of it, keep the copies separate, and store them on an external hard drive, a remote server, or in the cloud.

b. Operational Feasibility

Operational feasibility is the measure of how well a proposed system solves the problem and will be used once it is installed. An online survey researched the need for us humans to have a more organized kitchen. The results showed a dire need of a tool to organize our lives (see Appendix A). Moreover, as Lebanon is passing through this economic crisis, numerous families started storing food items in their homes as a prevention method for what may come. This resulted in

- unorganized pantries and fridges filled with items that are going bad,
- people not knowing what they have stored,
- people not knowing what to cook, or even what to dispose!

Hence, ePantry, a fridge, freezer, and pantry organizing app with many other features, will help those families know what food items they have stored, as well as know which foods are expired or close to their expiration date.

And as the awareness of having a healthy diet is increasing, the importance of cooking at home and with fresh and healthy ingredients is rising as well. So, with EPantry featuring a recipe generator and recommender that gives users healthy recipes, it will be the perfect solution for those wanting a healthy diet.

c. Technical Feasibility

Technical feasibility is the study is concerned with specifying equipment and what software are required to properly implement the idea and satisfy user requirements. ePantry, our application, can easily be considered technically feasible since all that is required is software and programs that already exist in the market. This includes languages and frameworks to create a mobile application and a website, which makes our project technically feasible.

- Hardware needed: Computers (our own laptops)
- Software needed:

- Editors, Libraries, and Languages (Flutter, Dart, Drift, Html…),
- Design Software (Canva),
- Prototype Software (Adobe Xd),
- Presentation Software (Microsoft PowerPoint),
- Report Writing Software (Microsoft Word).

- Human minds needed: at least four intermediate level programmers.

d. Economic Feasibility

Economic feasibility is a study that determines the cost and benefit of our project. Our application won't cost a lot to build, we will be able to develop the app with no extra costs. All the editors and frameworks that we will use are free of charge, but if we want to take our app and website public then we will have to pay the hosting fees.

Now to discuss profit, the estimated total number of households in Lebanon is around 1.266 million (according to the UNHCR), and each household has at least one fridge and a pantry. And there are 4.29 million mobile cellular in Lebanon (according to the World Data), if only 50% of households in Lebanon had a mobile device, then this gives us 612,000 potential users.

However, ePantry is not limited to a geographic location and can be used worldwide. So this means each person with a fridge or pantry is a potential user. Having this large number of potential users makes our app profitable and allows us to use different methods to increase our profits, such as Ads, in-app purchases, or monthly subscription for a premium account.

- **Budget Needed:**
  - Server / Cloud Server: $400 - $800
  - App Store and Play Store fees: $125
  - Dart Courses: $100-$200
  - Website Domain Name: $20 / year
  - Microsoft Office: $49
  - Canva Subscription: $112
  - Meeting Fees (printing data and surveys, transportation fees…): $150 - $300
  - Average Total Cost: $1,200 - $1,500

So the budget needed to fulfill our project objectives is about $1,350.

- **Revenue:**
  - In-App Ads: Apps can make anywhere from $10 up to $200 a day in average revenue from ads, and that's a conservative estimate for an app with only 1000 active users. We'll assume 100$ / day.
  - In-App Purchases: for example, we can sell users diet plans, extra recipes, add another fridge, etc. If only 10% of users bought one of these extra features, assuming we have 1000 active users, we'll get 100 purchases. We'll assume $400 / 1000 users.
  - Therefore, for an average of 1000 active users per month with no user growth: $4000
  - And for an average of 5000 active users per month with no user growth: $20000

- **Break Even Analysis:**

  The break even point where we'll start making profit is the 8th month with about 400 active users (see Appendix B). Hence, with this estimated cost and with the potential revenue of our project, we can say that our project is economically feasible.

e. Delivery

We will be supplying our users with a User (or end-user) Documentation. Which refers to explanatory and informational materials that describe the application we created to end-users. Essentially, user documentation is written to provide information and assist consumers in using the application.

After reading the application's user manual, they can learn:

- the features of our application,
- how to use our application,
- what not to do to avoid problems,
- how to address issues that arise despite all precautions,
- useful tricks, shortcuts, and tips

Depending on our needs, we choose to use a quick start guide as our User Documentation. A quick start guide is a short version of a user manual. It's more popular than its full-blown relative because, because people just don't want to read that much. A quick start guide typically consists of a few pages of the most important information on a product and short-listed instructions on how to use it. It's created to help users quickly set up and start using the product/application straight away.

Chapter 3. Software Requirement Specification

A. Product Functions

Certain features distinguish ePantry from similar apps on the market and give it its identity. Among these features are:

- Scanning receipts to insert items into the pantry. This feature allows our users to effortlessly enter items they have bought into their digital pantry, fridge, or freezer.
- Browsing recipes with ingredients on hand. One problem every mother faces is deciding what to cook for lunch or dinner. We want to provide ease of mind for our users by ensuring they always have various options on ePantry.
- Viewing reports to save money. We will add a feature that will show our users how much food was wasted in their household – expired before usage – this current month and a comparison to last month's wastage (whether their wastage is up or down from last month and by what percentage).
- Adding items to favorites list. This is a list of items our users always want on hand. When a user adds an item into his/her favorites list, it will automatically be added into their shopping list.

As previously mentioned, ePantry can be improved and updated on a larger scale. We can add features in future update that will make ePantry a better application. Among these "stretch" features are:

- Ordering items through our application from nearby markets. In the future, we hope to implement a feature that will allow us to scan nearby stores from our user's location and either be the middleman between the market's delivery and our users or provide our own fast delivery system.
- Checking availability of items in nearby markets. ePantry aims to improve the quality of life for every user, whether it be by saving them money or time, by reducing their wastes, or by providing them with a platform that will organize their lifestyle. In later versions, we plan add a feature that will allow our users to

check the items found in every market nearby. This will save so much time. Instead of visiting multiple supermarkets or stores to buy a few items, our users can check what store has all items in their shopping list.

- Customize the app based on your dietary preferences. We know that different people live different lifestyles when it comes to what they eat. We want our app to be completely customizable to suit each user's preferences. Therefore, when a dietary preference is set, the app will only show recipes that fall under that specific category. For example, if one user is vegan, the app will only show vegan recipes.
- Customize the app based on your health condition. We also want different users with different health conditions to feel comfortable using ePantry. Hence, we also plan to add a feature in the future that allows our users to provide the health issues they're comfortable with sharing. Based on these issues, we will recommend certain recipes more and other recipes less. For example, a user with diabetes will see less high sugar recipes and a user with hypertension will be recommended recipes with less salt levels.

We realize that when we plan to implement the last two features, a household account will not suffice. This is because every household member might have different preferences or health problems. To solve this issue, in the future, we will also implement multiple private user accounts in every household account. This way every user can enter to their account in a household.

B. User Characteristics

Our application – ePantry– is meant to help anyone who is struggling to maintain and manage their pantry, fridge, or freezer. The team behind ePantry are trying their best to minimize users' involvement to make it much easier to use. Therefore, our users don't need to have prior skills to use the application. ePantry's user-friendly interface will easily guide the user through the application. Thus, only basic knowledge on using your phone and your phone's camera is required.

C. Non-functional Requirements

Non-functional requirements are requirements that put constraints on the services or functions offered by the system or on timing, process, standards, etc. There are three main non-functional requirements: product, organizational, and external requirements.

1) Product Requirements:

   a. Usability:

      - User Friendly, simple, and interactive design

   b. Portability:
      - Same account can be used over multiple devices.
      - Every account can have different users for different preferences.
   c. Efficiency:
      - Easy navigations through different sections of the app.
      - Launching the application should not take more than 5 seconds.
      - The application will not have sudden crashes.
   d. Reliability:
      - Contacts For customer support on feedbacks and any issues regarding the usage of the app.

2) Organizational Requirements:

   a. Standards:
      - The app will follow all rules and regulations set by international committees for the safety and privacy of users.
   b. Delivery:
      - The application will be launched first in Lebanon.
      - It will be followed by an immediate launch globally.
      - The application must be delivered totally, a semi-finished product won’t be accepted.

c. Implementation:

- The application will be available on both iOS and Android devices.

3) External Requirements:

a. Privacy:

- The application will not ask to access device location until future updates.
- Password cannot be viewed once set.
- The application will not access nor save photos from the gallery without permission.
- The application won't access the microphone and camera without permission.

b. License/Copyright Issues:

- The application shall not display recipe or ingredient information that go against copyright laws.

c. Social Issues:

- The application shall be designed with a user-friendly interface and intuitive navigation, ensuring accessibility for users with limited technological literacy or disabilities.

d. Ethical Issues:

- The application shall prioritize user data security and confidentiality, implementing encryption algorithm for sensitive data.

e. Legal Issues:

- The application shall comply with applicable data retention and deletion laws by facilitating data deletion upon user request.

D. Domain Requirements

Domain requirements are related to the domain of our application. It is important to specify these requirements and make sure they are satisfied because if these requirements were not satisfied, our system might not work. Among these requirements are:

- The application will scan receipt in English only.
- The application clearly divides products into distinct storage locations (pantry, fridge, or freezer).
- The application will assume each item is stored in specifically one place (either pantry, fridge, or freezer).
- The application will not differentiate among different brands of products.
- The receipt will not include non-food items.
- The application will assume each item has a certain shelf-life.

E. Functional Requirements

Functional requirements are divided into two kinds: user requirements and system requirements. These requirements specify our application's functionalities in detail. To create flawless application similar to the one we have in mind, we will implement every requirement mentioned. Below, these requirements are summarized:

a. User Requirements

- Users will be able to create a household account through credentials.
- Users will be able insert items to pantry, fridge, or freezer by scanning receipts.
- Users will be able to insert items to pantry, fridge, or freezer manually.
- Users will be able to delete items.
- Users will be able to view all items in their pantry, fridge, or freezer.
- Users will be able to browse recipes.
- Users will be able to add their own recipes.
- Users will be able to add items to their favorites list.

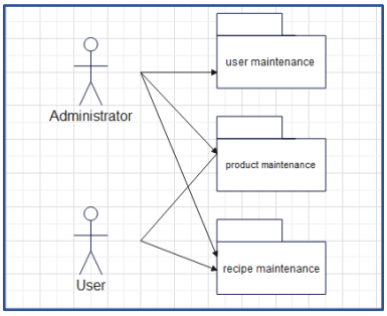
- Users will be able to view reports.
- Users will be able to check nutritional information about each item.
- Users can search for a specific item in their pantry, fridge, or freezer.
- Users will be able to update account information.

b. System Requirements

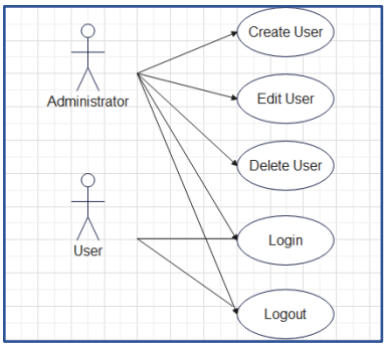
- The system shall provide multiple ways to insert items.
- The system shall provide multiple ways create account.
- The system shall send notifications to users about nearly expiring items.
- The system shall calculate money paid and money saved monthly.
- The system shall calculate monthly food wastage.

Use Case Diagrams

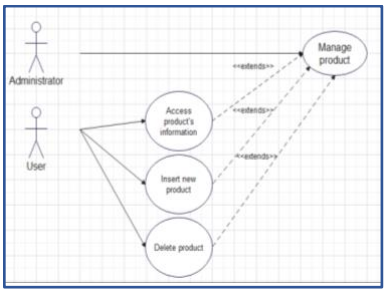
A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. Our proposed system consists of few packages of services, each of these packages is going to be explained in the upcoming figures.



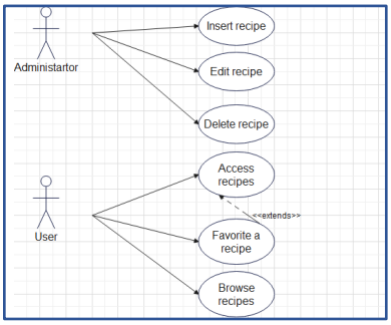
1- Package user maintenance allows to identify the different users of the system and to manage its privileges. This way, the system will be able to interact of an adjusted form with the users.



2- The product maintenance package explains how the user and the system are going to deal with the product in addition to the powers of each of them.



3- In the recipe maintenance package we can observe that the only the administrator has the access to manipulate or deal with the recipes whereas user may request to access the generated recipes or search for any product.

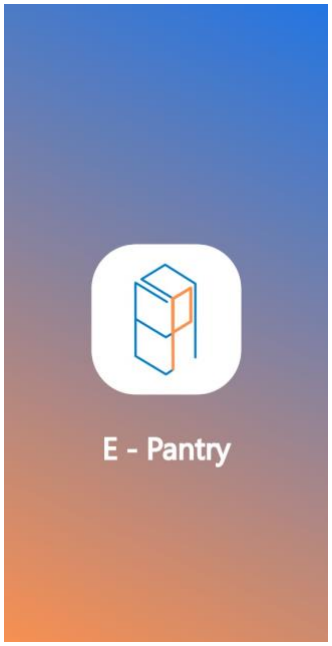


Chapter 4. Project Design

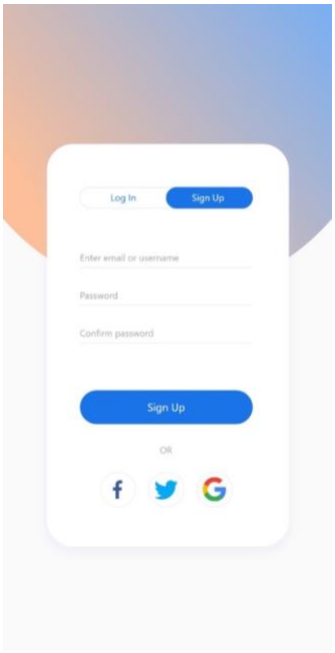
A. User Interface Prototype

Designing a prototype is a critical step in the creation of our application mainly because it will allow us as developers to determine which aspects are functional and which ones still need refining. Moreover, the user interface prototype will enable us to test the workflow of ePantry with multiple scenarios. This is mainly to check that the most common scenarios are optimized for minimal effort during use. As previously mentioned, and as clearly displayed in our prototype, ePantry has a highly user-friendly interface; any user can browse our application with ease. Below, our prototype depicts all the use cases of ePantry.

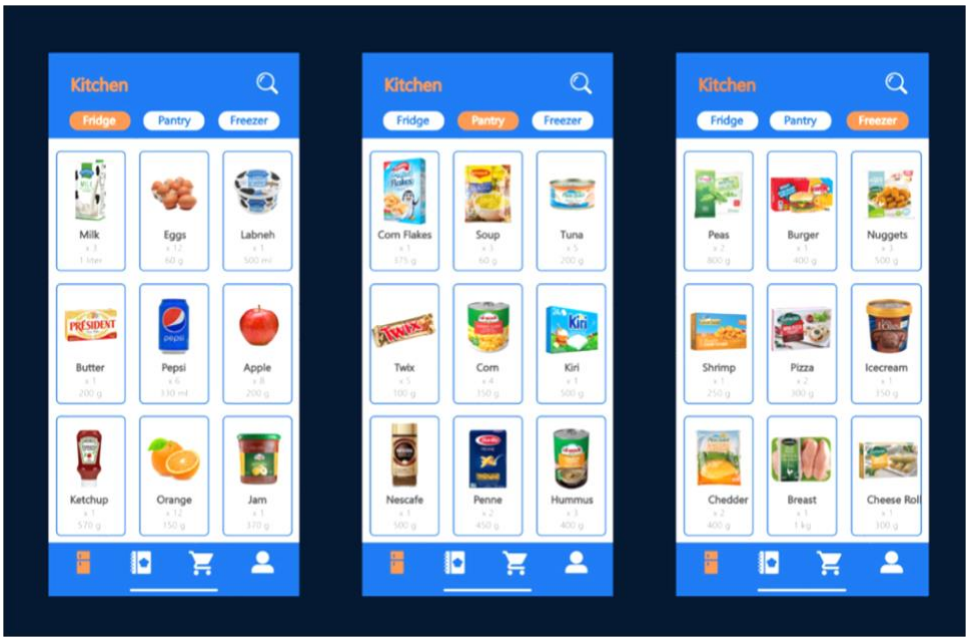
1) Launch Page



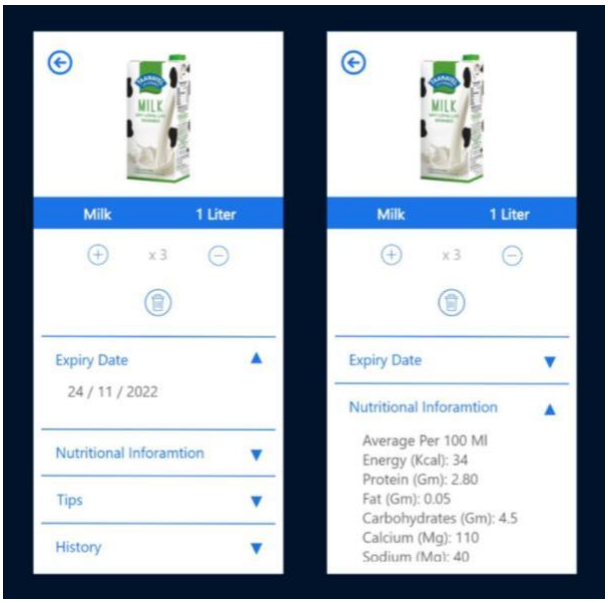
2) Sign In/ Sign Up Page



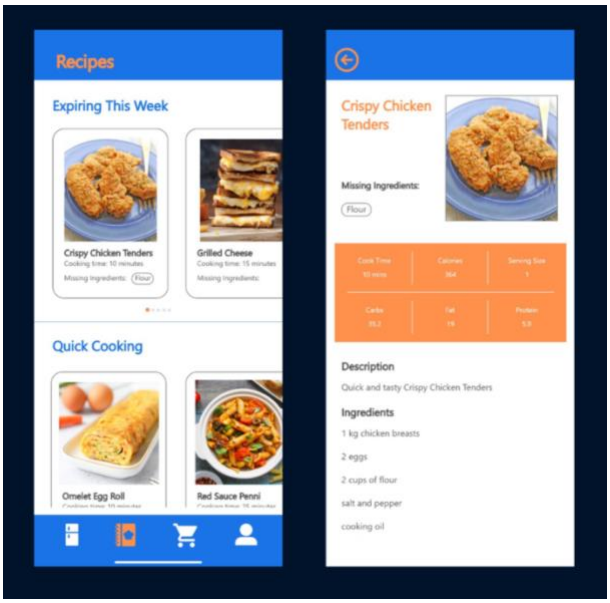
3) Fridge, Pantry, Freezer Storage Page



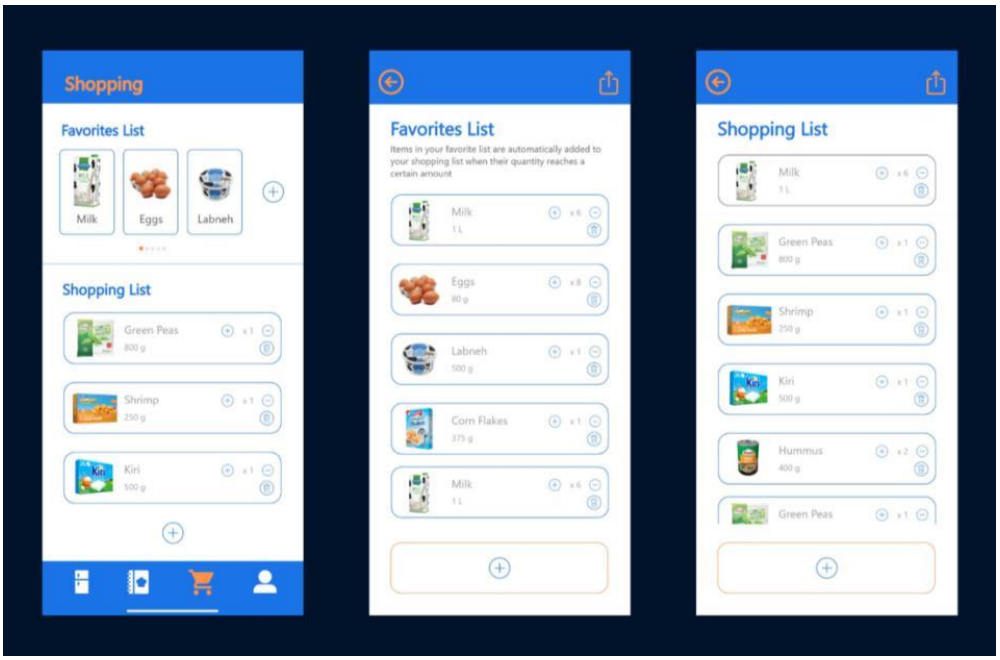
4) Items Info Page



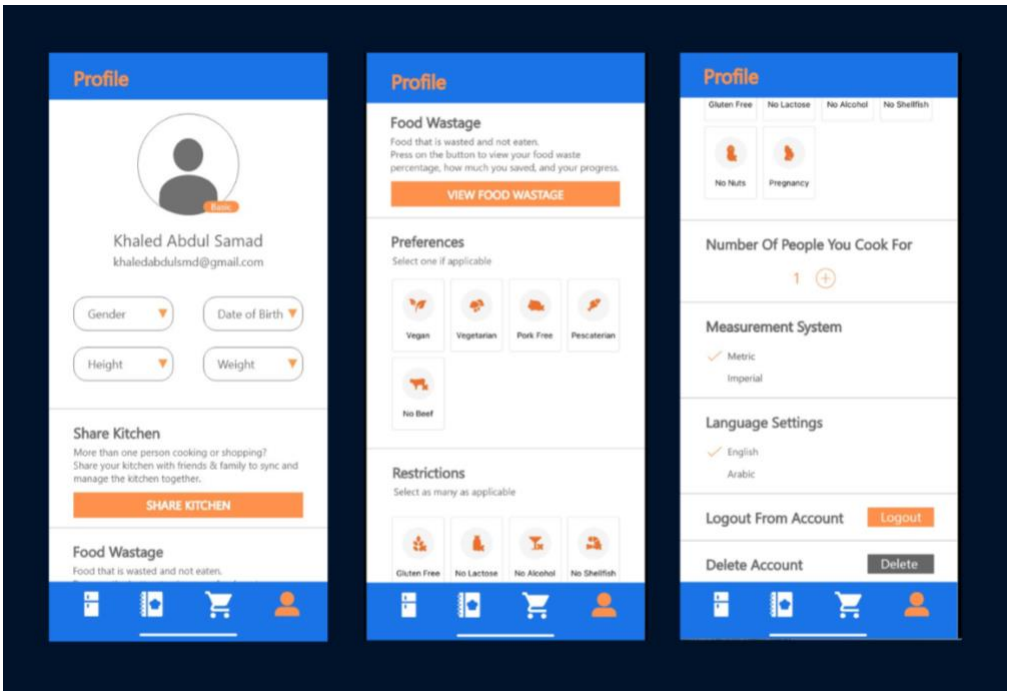
5) Recipe Page



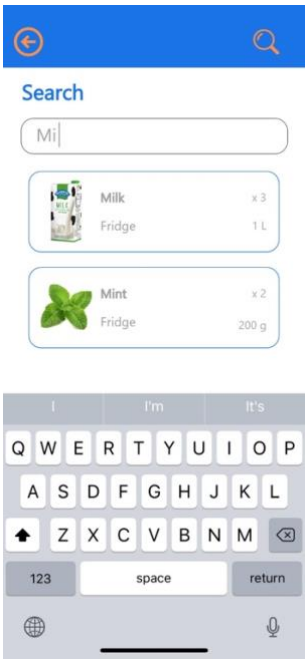
6) Favorites List/ Shopping List Page



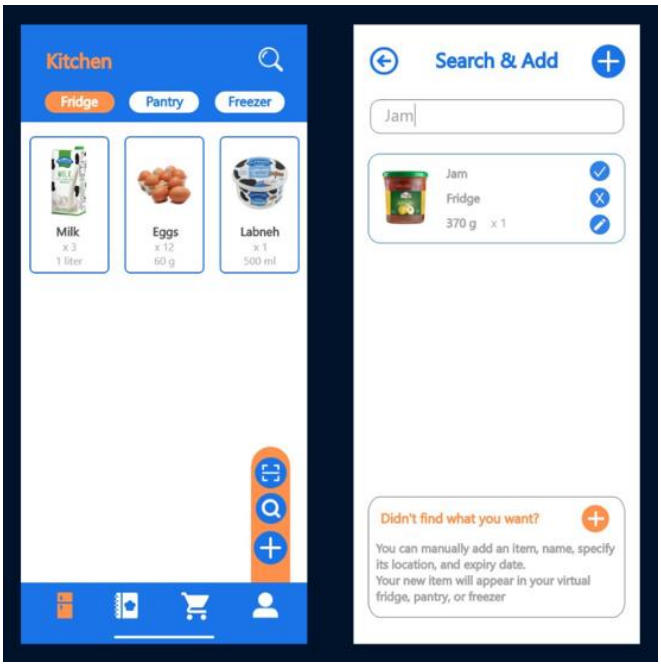
7) Account Settings Page



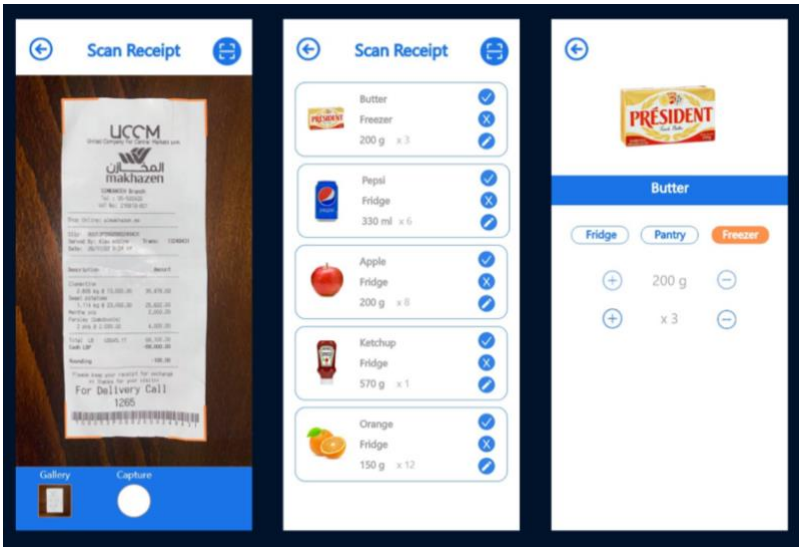
8) Search Page



9) Insert Item by Searching



11) Insert Item by Scanning



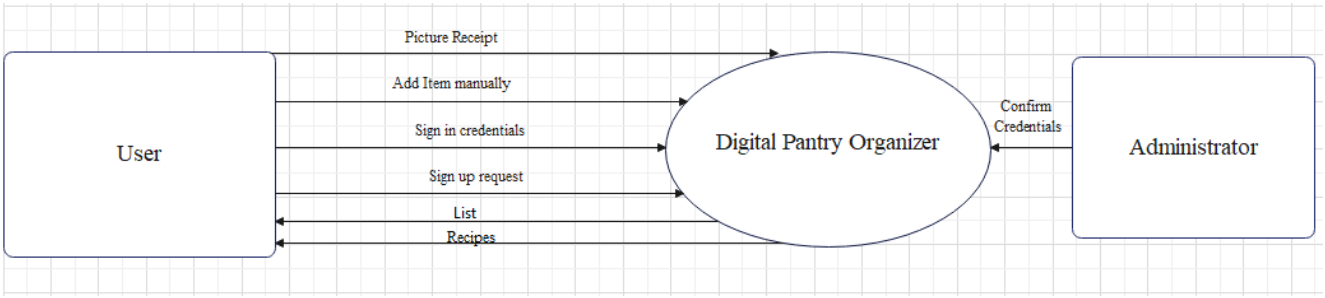
12) Food Wastage Report

B. Data Flow Diagram

DFD describes the processes that are involved in a system to transfer data from the input to the file storage. In other words, a data flow diagram shows how data flows throughout a system. ePantry is an app that depends on data from multiple aspects. First and foremost, a user should create a household account. To do so, the user can use credentials or sign in through a third-party application. When the user signs in, all the data stored in our system of all the items in his/her pantry, fridge, or freezer will be in display. We also need to store data regarding the user's favorite items and shopping list. Therefore, when a user uses our app, multiple types of data flow in and out of the app's database.

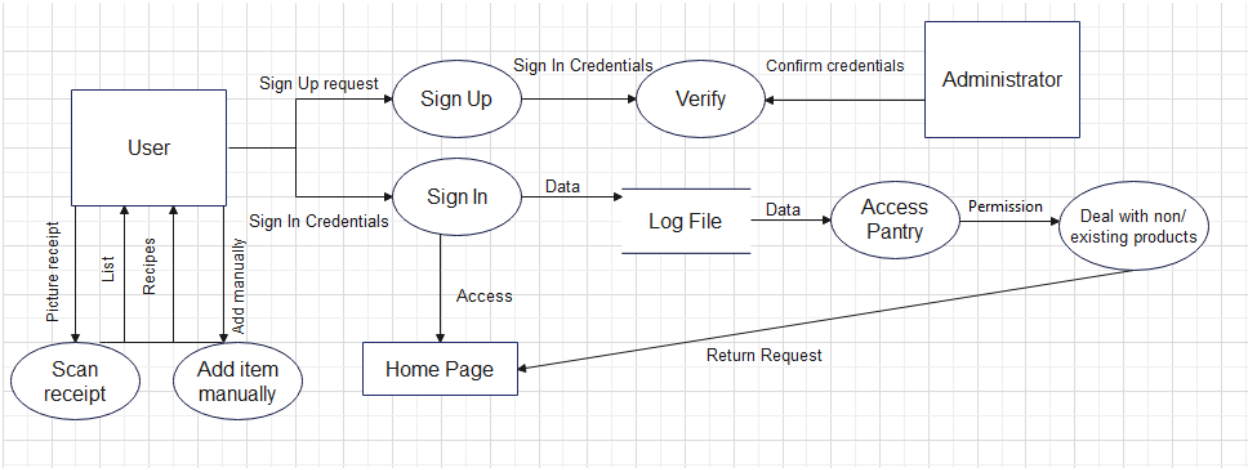
A. Level 0

Also known as a "context diagram," this is the highest level and represents a very simple, top-level view of the system being represented. So, this level will just contain an overview of the production and receiving of data in our system.



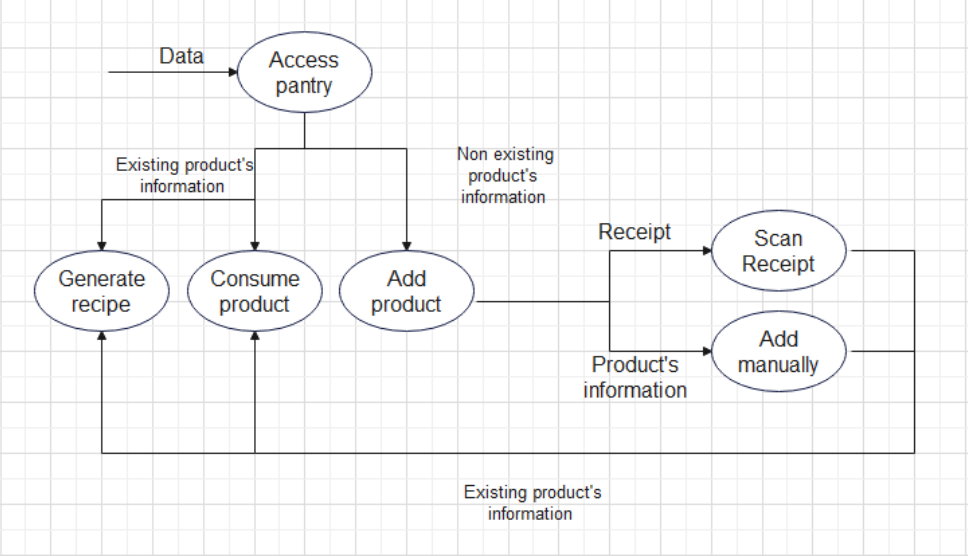
B. Level 1

This level is still a relatively broad view of the system but incorporates subprocesses and more detail. In this level, we will show the most basic process the user will go through which is signing in or signing up to a household account.



C. Level 2

We will be heading to more detailed description of the “access pantry” process in addition to the “deal with non/existing products” one.



C. Database Diagram

The nested NoSQL schema is a complex and versatile database structure that allows for flexible and scalable data storage. At its core, the schema consists of collections that contain documents, which in turn have fields that hold the data. This structure enables the database to accommodate various types of data, from user profiles to recipes and food items.

One of the key collections in the schema is the "users" collection. This collection is our root collection where it contains a document for each ePantry user with five fields: country, email, name, profile picture, and cuisines. The first four fields are of type string and the “cuisines" field is a list of strings that contains the user's preferred cuisines to ensure that the generated recipes are to our user’s liking. This field is an example of how the schema can handle complex data types like lists. From here, each user document has nested collections inside, where each collection in turn has its own documents and fields as well.

Moving on to the nested group of collections, we start with the "favList" collection. This collection has documents with two fields each: name and image both of type string. In this collection, we can find our user’s favorite food items. If an item is found in this list and it is about to expire, ePantry will directly add said item to the user’s shopping list.

The "shopping" collection has documents with the name, quantity, and image fields which are also of type string where they represent the name of the product with the quantity needed and the image of that product. This collection will store our user’s shopping list to make it easier for our users to organize their life with a visual shopping list.

The "freezerItems," "pantryItems," and "fridgeItems" collections have documents with name, quantity, expiryDate, nutrition, and image fields which are related to various items that are found in the kitchen of our users. The documents in these collections are clearly divided in the application on different pages so the user can check the availability of an item in a certain storage location. We also created an "allItems" collection which contains

all the items found in the user's household. This collection contains documents with image, ingredients, and keywords fields. We created this collection because it allowed us to implement real-time search in ePantry where the user can search if an item exists in their kitchen.

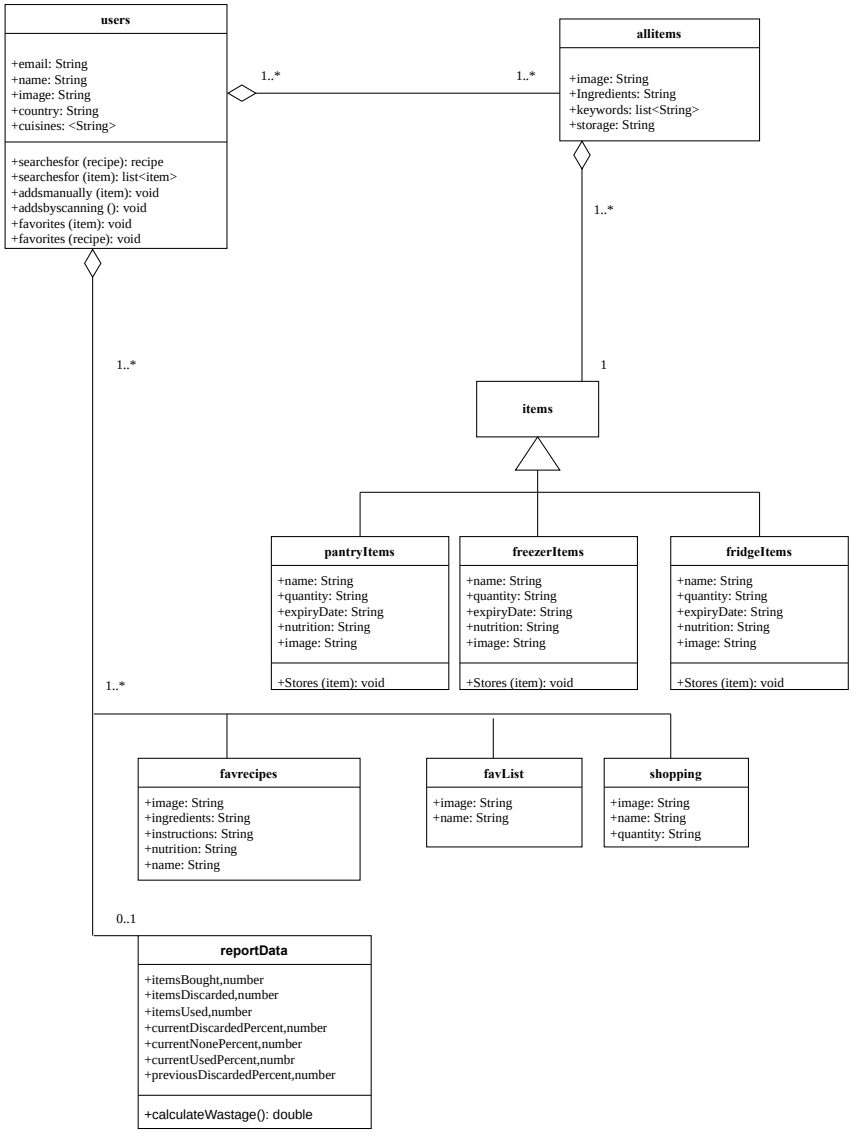
The "favRecipes" collection is another important collection in the schema. This collection has documents with image, ingredients, instructions, nutrition, and name fields. Since ePantry generates random recipes all the time, we wanted our users to be able to save or "favorite" a certain recipe so they can access it even if they reloaded the recipes page. Therefore, we implemented a favorite button that will save recipes into the database and display them all the time. To enable the viewing of reports about used and saved items, we have created a collection called "reportData." This collection includes the following fields, itemsBought where this field represents the total number of items bought by the user. Whenever a user purchases an item, the value of itemsBought is incremented to reflect the new total. Another field is itemsDiscarded where this field represents the total number of items that have been discarded by the user. If a user decides to get rid of an item, the value of itemsDiscarded is incremented accordingly. Last field is itemsUsed where it represents the total number of items that have been used by the user. Whenever a user utilizes an item, the value of itemsUsed is incremented to keep track of the overall usage. We also kept a count of the current and previous percentage of discarded items as well as current used and remaining items in pantry.

In summary, the nested NoSQL schema is a powerful tool for storing and retrieving complex data. It allows for flexible and scalable data storage, making it ideal for a wide range of applications. From user profiles to recipes to food items, the schema is designed to handle a variety of data types. By providing a hierarchical structure with collections, documents, and fields, the schema makes it easy to organize and retrieve data. Whether you're building an e-commerce site, a recipe database, or a food-tracking app, the nested NoSQL schema can help you manage your data effectively.


Collection
Document
Field
users
country, String
email, String
name, String
cuisines, List <String>
favList
shopping
freezerItems
fridgeItems
pantryItems
name, String
image, String
name, String
quantity, String
image, String
name, String
quantity, String
expiryDate,String
nutrition, String
image, String
name, String
quantity, String
expiryDate,String
nutrition, String
image, String
name, String
quantity, String
expiryDate,String
nutrition, String
image, String
favRecipes
allItems
reportData
image, String
ingredients, String
instructions, String
name, String
nutrition, String
image, String
name, String
keywords, list<String>
storage, String
itemsBought, number
itemsDiscarded, number
itemsUsed, number
currentDiscardedPercent, number
currentUsedPercent, number
currentUsedPercent, number
previousDiscardedPercent,number

D. Class Diagram

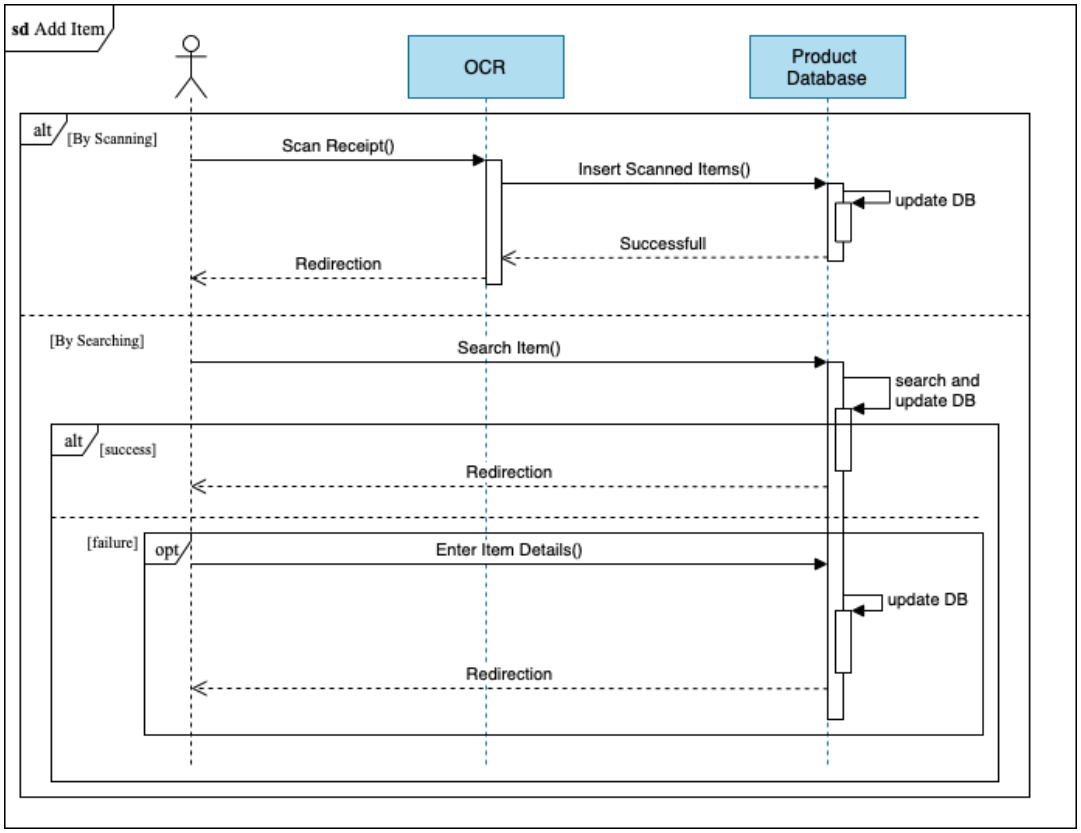
The classes required for the class diagram are exactly the same as the collections assigned with their fields as it is in the NOSQL scheme; these are User, items, favrecipe, Storage Type and much more classes. These said classes have the same attributes and methods as above. For example, the class item will have attributes image, ingredients, and keywords. In addition to the relationships that exist between these classes same as in the above scheme.



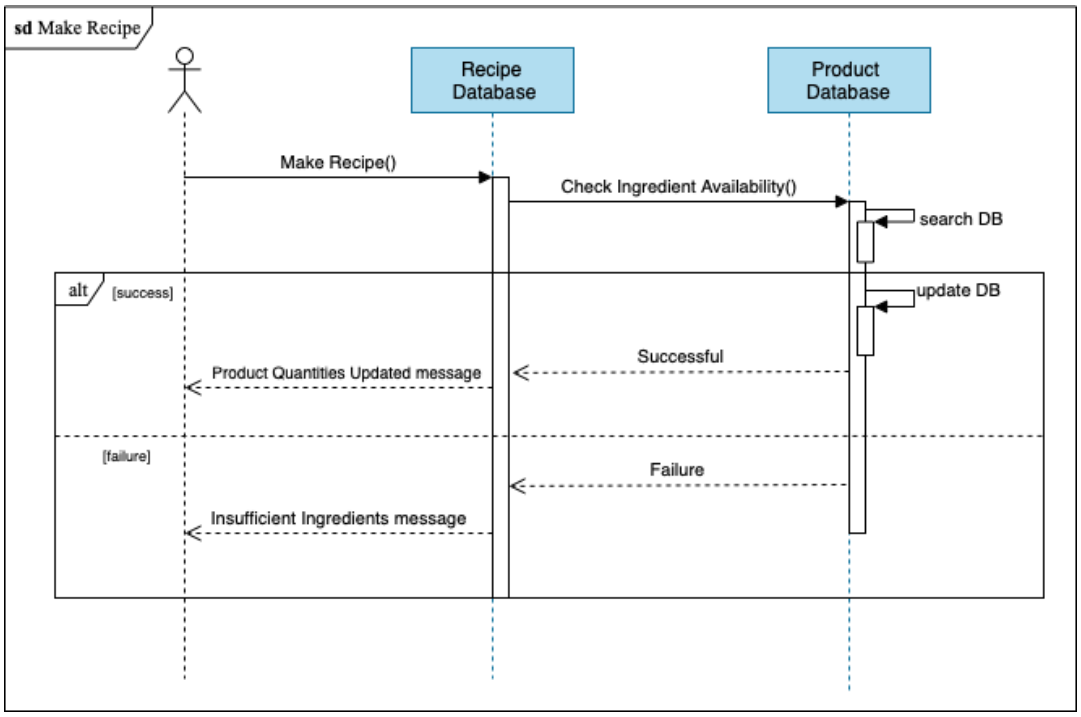
E. Sequence Diagram

ePantry has multiple use cases given its features. To show the "life" of a user's interaction with the system, we created two sequence diagrams for the most prominent use cases of our application.

- The first sequence diagram is for adding an item to the user's pantry, fridge, or freezer. The user can do so either by scanning a receipt or by searching for a product. If the product the user is searching for is not found in our database, the user has the option to add an item manually by entering the details required to add a product. This is all summarized in the sequence diagram below.



- The second sequence diagram describes the process by which a user can make a recipe from the recipes in our recipe database. When the user decides to make a recipe, we will fetch the required ingredients from the recipe, then verify that the user has them in his/her pantry. If not, a “Insufficient Ingredients” message will pop up.



Chapter 5. Methodology

A. Implementation (ePantry App)

Frontend

Flutter and Dart were the primary technologies used to implement the frontend of the application. Flutter's rich widget library, coupled with Dart's expressive and flexible programming language, provided an efficient and streamlined approach to building the frontend of the application.

One of the significant advantages of using Flutter and Dart is their ability to code once and deploy for both iOS and Android platforms simultaneously. This helped in reducing development time and effort, and ensured that the app's user experience and functionality remain consistent across both platforms.

The Flutter framework's hot-reload feature, coupled with Dart's Ahead-of-Time (AOT) compilation, made the development process even more efficient. These features allowed me to make changes to the app's frontend code and instantly see the updates without needing to rebuild the entire application.

Backend

Firebase was the primary technology used for the application's backend, providing robust authentication and database services. Leveraging Firebase's authentication service, users were able to sign up, log in, and reset their passwords securely.

Firebase's real-time database service was utilized to store and retrieve data dynamically, providing a seamless experience for the app's users. Additionally, Firebase's cloud storage was used to store images and other data, ensuring fast and reliable access to these files for the app's users. Firebase's scalable and flexible infrastructure will allow for the app to handle a large number of users and data without any downtime or performance issues.

As mentioned previously in the Database Diagram section, we used a document database, which is a NoSQL database model. We opted for a nested collections design, where each user document has multiple nested collections inside. For example, a user document X itself has 8 collections inside, including "fridgeItems." Inside this collection, we can find documents of all items found in user X's fridge. If we press on a document Y, we will see the corresponding fields of item Y. (see Appendix C for visual)

- ### Packages

As part of the development process for ePantry, we utilized over 30 packages (see Appendix D) to optimize implementation and enhance the overall user experience. The package manager used in our app is Pub because it is the default and widely adopted package manager for Dart and Flutter development. Pub provides a seamless and efficient way to manage dependencies, allowing us to easily incorporate external libraries and packages into our codebase. These packages were carefully selected based on their capabilities and compatibility with ePantry's specific needs. Using these packages not only saved time and effort during development, but also resulted in a polished final product that meets the highest standards of quality and usability as mentioned in our non-functional requirements. Some of these packages were:

- permission_handler allowed us to ensure our users' privacies by asking for permission before accessing their cameras and their photo galleries. This ensured that we satisfy the non-functional requirements related to privacy.
- flutter_staggered_animations and google_nav_bar made ePantry visually appealing and provided it with interactive navigations. These packages, along with a few more, ensured that our application is user-friendly and interactive as previously mentioned in the non-functional requirements. They also made sure our application is efficient when it comes to navigating different pages of ePantry.

- firebase_auth and cloud_firestore facilitated a seamless backend integration and reliable data storage. These packages also ensured our users' privacy is well maintained by authenticating everything and storing their data safely. Moreover, it allowed us to make our application portable since the user can sign in with the same account over multiple devices.
- scanbot_sdk saved us development time and reduced our costs by utilizing its pre-built powerful OCR. This package ensured as an efficient and accurate scanning of receipts through the powerful scanning algorithm that it uses.

○ APIs

In order to provide enhanced functionalities and accurate information to the users, the app utilizes the power of multiple APIs. These APIs not only made the app more robust but also helped in improving its overall performance. These APIs were:

- Spoonacular API provides access to a vast collection of recipes, nutrition, and food-related data. We used to for it allowed us to bring diverse recipes from different cuisines to our users.
- Edamam API provides nutrition analysis and recipe search functionalities. It also offers a comprehensive nutritional database. We used to for it provided ePantry with generic images for groceries and food items.
- USDA FoodData Central API offers access to a wide range of food and nutrition-related data, including nutrient profiles for different foods, branded food products, and more. We used it for it supplied ePantry with accurate nutritional information for all possible food items.

○ Performance Optimization

To ensure an optimal user experience, our team made sure to implement performance optimization techniques. In particular, we focused on improving the

loading time of images displayed in our app, as we found that they were causing significant delays and affecting the app's overall performance.

To address this issue, we leveraged the cached_network_image package, which allowed us to cache images locally and load them quickly, without requiring the app to download the same image multiple times. This technique reduced load times and made the app more responsive, resulting in a smoother and more visually appealing user experience.

- **Deployment and Delivery**

For deployment and delivery, we will submit documentation that contains all the pages of ePantry, along with a detailed description of each page and its feature. The documentation will look something like this:

- **Login/Signup Pages**

From the moment users open the app, they are greeted by a welcoming login/signup page that offers several options for account creation. This page has been thoughtfully designed to be user-friendly and intuitive.

- Signup: Users can sign up for a new account by entering their email and creating a new password, or they can sign up using their existing Google, Facebook, or Twitter accounts.
- Login: For users who already have an account, the login process is straightforward, and they can easily reset their password if needed.

- **Kitchen Page**

The Kitchen page is the heart of the ePantry application. It is divided into three different sections that allow users to view their pantry, fridge, or freezer. This page has been designed with an intuitive and user-friendly interface, which makes navigation through the different screens a breeze. There is a navigation bar located at the top of the screen which provides

easy access to the different storage types (pantry, fridge, or freezer). Another navigation bar is located at the bottom of the screen and provides easy access to the different pages of the application (Recipes, Lists, and Account Pages).

- Search Feature: Users can use the search button located in the top of the screen to quickly find specific items in their kitchen. The search function is particularly useful when users need to locate an item quickly. When they press the search button, they are directed to another page where they can enter the name of the item they are looking for, and it will appear in a list with its location whether it was in fridge, pantry or freezer.
- Add Items Feature: Adding items to the fridge, pantry, or freezer is simple and straightforward with the intuitive Add button, which is always visible on the bottom right of the screen. Clicking on the Add button opens a popup menu that offers two options: scanning or manual adding items.

    - Adding Manually: If users choose manual search, they are directed to a separate page where they can enter the name of the item, quantity, expiry date, and storage type. They can also choose whether to auto-fill the nutritional information or not.
    - Adding by Scanning: If users choose to scan, they are directed to another page where they can either take a photo of their receipt or upload a photo from their gallery, and ePantry's innovative technology will extract the items and their quantities from the receipt and add them to the relevant storage type.

- View Items Feature: Once users add items to their kitchen, they are displayed in a grid on the screen of their relevant storage type.

Each item is placed in a box containing a photo, name, and the quantity of that item. Clicking on an item in the grid opens the item page, where users can view specific details about each item, adjust the quantity, or remove the item by using or discarding it. Users can view nutritional facts about the item by pressing on the arrow beside the nutritional information. A pop down list will appear having nutritional facts such as fat, carbohydrates, energy, calcium… There is also a favorite button that allows users to save their favorite items to a separate list called the favorites list. The favorite button is located at the top of the screen. The user can also adjust the expiry date and the quantity of any item.

- **Recipes Page**

The Recipe page is a fantastic addition to ePantry. It is a great resource for users who want to make the most of the items in their kitchen. Users navigate to this page by pressing on the recipe icon in nav bar. This page features three main sections for breakfast, main course, and desserts, as well as a saved recipes section. In each section several recipes are displayed to the user. At the top of the screen, there is a menu button with two features: cuisine preference and refresh.

- Cuisine Preferences Feature: The cuisine preference feature enables the user to select their preferred cuisine, and ePantry will generate and display recipes that align with their preferences.
- Refresh Feature: The refresh feature updates the recipe list, providing the user with a fresh selection of recipes to explore and prepare.
- Recipes Feature: When users click on a recipe, they are directed to a page where they can view the ingredients, directions, and nutritional information. Whereas we mentioned these recipes are tailored based on their cuisine preferences. Users could save

recipes to a separate list which is a great feature that will undoubtedly be appreciated by users, they can do so by pressing the like button on the top of the screen.

- **Lists Page**

When selecting the third icon from the navbar, users are directed to the Lists page. The Lists page is another excellent feature of ePantry. It has two sections: the favorites list and the shopping list. Users can view the items in both lists from the main Lists page, but they can click on "View more" to access the specific pages for each list.

- Favorites Page: When selecting the third icon from the navbar, users are directed to the Lists page. The Lists page is another excellent feature of ePantry. It has two sections: the favorites list and the shopping list. Users can view the items in both lists from the main Lists page, but they can click on "View more" to access the specific pages for each list.

    - Adding Items: Users can add new items to this list by pressing on the add button at the top of the page. A popup window appears, and the user is prompted to enter the name of the item to be added. Or the user can select the like button from each item page, by doing so the item will automatically be added to the favorites list.
    - Deleting Items: Users can remove an item from their favorites list by pressing on the delete button beside each item.

- Shopping List Page: The shopping list is for items the users need to buy. Users can add, delete, and adjust quantities of items in this lists.

- Adding Items: Users can add new items to this list by pressing on the add button at the top of the page. A popup window appears, and the user is prompted to enter the name of the item to be added. Or the user can select the like button from each item page, by doing so the item will automatically be added to the shopping list.
- Deleting Items: Users can remove an item from their shopping list by pressing on the delete button beside each item.
- Adjust Quantities: Users can increase or decrease the quantity of an item by pressing on the add or minus buttons respectively.

- **Account Page**

The Account page provides users access to their account information, including their name, email, and profile picture. Additionally, it offers various functionalities, such as Edit Profile, Change Password, Change Country, View Food Wastage Report, Know More About ePantry, and Logout options.

- Edit profile and change password: By selecting Edit Profile or Change Password, users are redirected to another page where they can modify their account photo, username, email, and password.
- Change Country: The Change Country option displays a list of countries from which users can select their location.
- Food Wastage Report: This page shows our users their food wastage report. It shows them the percentage of food that was used vs discarded this month, along with a comparison between this month's wastage and last month's wastage.

Chapter 5. Methodology

A. Implementation (ePantry App)

Frontend

Flutter and Dart were the primary technologies used to implement the frontend of the application. Flutter's rich widget library, coupled with Dart's expressive and flexible programming language, provided an efficient and streamlined approach to building the frontend of the application.

One of the significant advantages of using Flutter and Dart is their ability to code once and deploy for both iOS and Android platforms simultaneously. This helped in reducing development time and effort, and ensured that the app's user experience and functionality remain consistent across both platforms.

The Flutter framework's hot-reload feature, coupled with Dart's Ahead-of-Time (AOT) compilation, made the development process even more efficient. These features allowed me to make changes to the app's frontend code and instantly see the updates without needing to rebuild the entire application.

Backend

Firebase was the primary technology used for the application's backend, providing robust authentication and database services. Leveraging Firebase's authentication service, users were able to sign up, log in, and reset their passwords securely.

Firebase's real-time database service was utilized to store and retrieve data dynamically, providing a seamless experience for the app's users. Additionally, Firebase's cloud storage was used to store images and other data, ensuring fast and reliable access to these files for the app's users. Firebase's scalable and flexible infrastructure will allow for the app to handle a large number of users and data without any downtime or performance issues.

As mentioned previously in the Database Diagram section, we used a document database, which is a NoSQL database model. We opted for a nested collections design, where each user document has multiple nested collections inside. For example, a user document X itself has 8 collections inside, including "fridgeItems." Inside this collection, we can find documents of all items found in user X's fridge. If we press on a document Y, we will see the corresponding fields of item Y. (see Appendix C for visual)

- ### Packages

As part of the development process for ePantry, we utilized over 30 packages (see Appendix D) to optimize implementation and enhance the overall user experience. The package manager used in our app is Pub because it is the default and widely adopted package manager for Dart and Flutter development. Pub provides a seamless and efficient way to manage dependencies, allowing us to easily incorporate external libraries and packages into our codebase. These packages were carefully selected based on their capabilities and compatibility with ePantry's specific needs. Using these packages not only saved time and effort during development, but also resulted in a polished final product that meets the highest standards of quality and usability as mentioned in our non-functional requirements. Some of these packages were:

- permission_handler allowed us to ensure our users' privacies by asking for permission before accessing their cameras and their photo galleries. This ensured that we satisfy the non-functional requirements related to privacy.
- flutter_staggered_animations and google_nav_bar made ePantry visually appealing and provided it with interactive navigations. These packages, along with a few more, ensured that our application is user-friendly and interactive as previously mentioned in the non-functional requirements. They also made sure our application is efficient when it comes to navigating different pages of ePantry.

- firebase_auth and cloud_firestore facilitated a seamless backend integration and reliable data storage. These packages also ensured our users' privacy is well maintained by authenticating everything and storing their data safely. Moreover, it allowed us to make our application portable since the user can sign in with the same account over multiple devices.
- scanbot_sdk saved us development time and reduced our costs by utilizing its pre-built powerful OCR. This package ensured as an efficient and accurate scanning of receipts through the powerful scanning algorithm that it uses.

○ APIs

In order to provide enhanced functionalities and accurate information to the users, the app utilizes the power of multiple APIs. These APIs not only made the app more robust but also helped in improving its overall performance. These APIs were:

- Spoonacular API provides access to a vast collection of recipes, nutrition, and food-related data. We used to for it allowed us to bring diverse recipes from different cuisines to our users.
- Edamam API provides nutrition analysis and recipe search functionalities. It also offers a comprehensive nutritional database. We used to for it provided ePantry with generic images for groceries and food items.
- USDA FoodData Central API offers access to a wide range of food and nutrition-related data, including nutrient profiles for different foods, branded food products, and more. We used it for it supplied ePantry with accurate nutritional information for all possible food items.

○ Performance Optimization

To ensure an optimal user experience, our team made sure to implement performance optimization techniques. In particular, we focused on improving the

loading time of images displayed in our app, as we found that they were causing significant delays and affecting the app's overall performance.

To address this issue, we leveraged the cached_network_image package, which allowed us to cache images locally and load them quickly, without requiring the app to download the same image multiple times. This technique reduced load times and made the app more responsive, resulting in a smoother and more visually appealing user experience.

- **Deployment and Delivery**

For deployment and delivery, we will submit documentation that contains all the pages of ePantry, along with a detailed description of each page and its feature. The documentation will look something like this:

- **Login/Signup Pages**

From the moment users open the app, they are greeted by a welcoming login/signup page that offers several options for account creation. This page has been thoughtfully designed to be user-friendly and intuitive.

- Signup: Users can sign up for a new account by entering their email and creating a new password, or they can sign up using their existing Google, Facebook, or Twitter accounts.
- Login: For users who already have an account, the login process is straightforward, and they can easily reset their password if needed.

- **Kitchen Page**

The Kitchen page is the heart of the ePantry application. It is divided into three different sections that allow users to view their pantry, fridge, or freezer. This page has been designed with an intuitive and user-friendly interface, which makes navigation through the different screens a breeze. There is a navigation bar located at the top of the screen which provides

easy access to the different storage types (pantry, fridge, or freezer). Another navigation bar is located at the bottom of the screen and provides easy access to the different pages of the application (Recipes, Lists, and Account Pages).

- Search Feature: Users can use the search button located in the top of the screen to quickly find specific items in their kitchen. The search function is particularly useful when users need to locate an item quickly. When they press the search button, they are directed to another page where they can enter the name of the item they are looking for, and it will appear in a list with its location whether it was in fridge, pantry or freezer.
- Add Items Feature: Adding items to the fridge, pantry, or freezer is simple and straightforward with the intuitive Add button, which is always visible on the bottom right of the screen. Clicking on the Add button opens a popup menu that offers two options: scanning or manual adding items.

    - Adding Manually: If users choose manual search, they are directed to a separate page where they can enter the name of the item, quantity, expiry date, and storage type. They can also choose whether to auto-fill the nutritional information or not.
    - Adding by Scanning: If users choose to scan, they are directed to another page where they can either take a photo of their receipt or upload a photo from their gallery, and ePantry's innovative technology will extract the items and their quantities from the receipt and add them to the relevant storage type.

- View Items Feature: Once users add items to their kitchen, they are displayed in a grid on the screen of their relevant storage type.

Each item is placed in a box containing a photo, name, and the quantity of that item. Clicking on an item in the grid opens the item page, where users can view specific details about each item, adjust the quantity, or remove the item by using or discarding it. Users can view nutritional facts about the item by pressing on the arrow beside the nutritional information. A pop down list will appear having nutritional facts such as fat, carbohydrates, energy, calcium… There is also a favorite button that allows users to save their favorite items to a separate list called the favorites list. The favorite button is located at the top of the screen. The user can also adjust the expiry date and the quantity of any item.

- **Recipes Page**

The Recipe page is a fantastic addition to ePantry. It is a great resource for users who want to make the most of the items in their kitchen. Users navigate to this page by pressing on the recipe icon in nav bar. This page features three main sections for breakfast, main course, and desserts, as well as a saved recipes section. In each section several recipes are displayed to the user. At the top of the screen, there is a menu button with two features: cuisine preference and refresh.

- Cuisine Preferences Feature: The cuisine preference feature enables the user to select their preferred cuisine, and ePantry will generate and display recipes that align with their preferences.
- Refresh Feature: The refresh feature updates the recipe list, providing the user with a fresh selection of recipes to explore and prepare.
- Recipes Feature: When users click on a recipe, they are directed to a page where they can view the ingredients, directions, and nutritional information. Whereas we mentioned these recipes are tailored based on their cuisine preferences. Users could save

recipes to a separate list which is a great feature that will undoubtedly be appreciated by users, they can do so by pressing the like button on the top of the screen.

- **Lists Page**

When selecting the third icon from the navbar, users are directed to the Lists page. The Lists page is another excellent feature of ePantry. It has two sections: the favorites list and the shopping list. Users can view the items in both lists from the main Lists page, but they can click on "View more" to access the specific pages for each list.

- Favorites Page: When selecting the third icon from the navbar, users are directed to the Lists page. The Lists page is another excellent feature of ePantry. It has two sections: the favorites list and the shopping list. Users can view the items in both lists from the main Lists page, but they can click on "View more" to access the specific pages for each list.

    - Adding Items: Users can add new items to this list by pressing on the add button at the top of the page. A popup window appears, and the user is prompted to enter the name of the item to be added. Or the user can select the like button from each item page, by doing so the item will automatically be added to the favorites list.
    - Deleting Items: Users can remove an item from their favorites list by pressing on the delete button beside each item.

- Shopping List Page: The shopping list is for items the users need to buy. Users can add, delete, and adjust quantities of items in this lists.

- Adding Items: Users can add new items to this list by pressing on the add button at the top of the page. A popup window appears, and the user is prompted to enter the name of the item to be added. Or the user can select the like button from each item page, by doing so the item will automatically be added to the shopping list.
- Deleting Items: Users can remove an item from their shopping list by pressing on the delete button beside each item.
- Adjust Quantities: Users can increase or decrease the quantity of an item by pressing on the add or minus buttons respectively.

- **Account Page**

The Account page provides users access to their account information, including their name, email, and profile picture. Additionally, it offers various functionalities, such as Edit Profile, Change Password, Change Country, View Food Wastage Report, Know More About ePantry, and Logout options.

- Edit profile and change password: By selecting Edit Profile or Change Password, users are redirected to another page where they can modify their account photo, username, email, and password.
- Change Country: The Change Country option displays a list of countries from which users can select their location.
- Food Wastage Report: This page shows our users their food wastage report. It shows them the percentage of food that was used vs discarded this month, along with a comparison between this month's wastage and last month's wastage.

- About ePantry: The About ePantry option directs users to a page where they can learn more about the ePantry service, along with a link to ePantry's website.
- Logout: Lastly, the Logout option allows users to securely log out of their account.

CS Knowledge Application and Learning

Data Mining and Image Processing

To make the process of tracking items in the fridge easier, we integrated an object detection model using Roboflow. This feature utilizes computer vision to detect and identify items in real-time, allowing users to easily keep track of what they have in their fridge. We used concepts learned during the Introduction to Data Mining (CMPS452) and Image Processing (CMPS327) courses to implement the system. Topics covered throughout these courses facilitated the development of such a system.

- During the Data Mining course, we learned that object detection is a supervised learning task, where the model learns to classify and localize objects in an image based on labeled training data. Therefore, we had to make sure the dataset we used was labeled. Moreover, augmenting the dataset by applying transformations like noise, blur, and cutout is a common technique to artificially increase the size and diversity of the training data. This helps the model generalize better by exposing it to different variations of the objects and improving its robustness. Throughout the training process, we also monitored the performance of our model using evaluation metrics mentioned in class such as mean Average Precision (mAP) and recall.
- During the Image Processing course, we learned about many image processing techniques and how to filter out certain noise in images. We also learned that images taken in real-life scenarios tend to be

noisy, blurry, and/or partially occluded. We wanted to simulate real-world scenarios and make the model more robust to variations. Therefore, we used these concepts and introduced noise, blur, rotation, exposure, and cutout to our initial dataset.

First, we found an initial dataset of 516 images and gave it to Roboflow; we then performed data augmentation on the dataset using Roboflow by applying blur, noise, and cutout on the existing images to obtain a dataset of 3050 images. The resulting model had a 99.4% precision and a 99.0% recall.

Although the developed model takes images as input, detects objects from the images, and adds/deletes items from the fridge items in the database, we will assume that this system is running in real-time in our users' kitchens. Whenever the fridge opens and closes, we will assume a change has been made in the fridge; therefore, we will run the code and update the database accordingly. This machine learning process have been handled through the following steps that are explained below and shown in the appendices.

- Appendix E

  This appendix presents a sample image taken from a fridge, which serves as a part of the training dataset. The dataset was augmented by adding noise and outliers to increase its size and account for various possible captures.

- Appendix F

  This appendix displays the results obtained when object detection is applied to the training dataset. The results include a confidence score and a label for each detected object. For example, the

analysis may determine that a particular object in the image is a tomato with a confidence score of 91%.

- Appendix G

  In this appendix, a histogram is shown to represent the frequency of class balances in the training set. The items represented by the green histogram are well-captured and noticeable in the dataset. For instance, the histogram may indicate that tomatoes have been captured in 1959 pictures, indicating a good representation. The items represented by the red histogram are classified as "underrepresented," suggesting a higher chance of errors in their detection if they were found in the fridge.

- Appendix H

  This appendix contains a graph illustrating the training process of the object detection model. The graph displays the variation of mean average precision (mAP) as a function of the number of training iterations (epochs). As the number of epochs increases, the mean average precision also increases. This relationship is logical since more training allows the model to learn and improve its object detection performance.

By examining these appendices in conjunction, we gain insights into the machine learning process involved in training the object detection model. It starts with the augmented training dataset (Appendix E), which is then used to perform object detection, yielding results with confidence scores and labels (Appendix F). The class balance of the training set is analyzed through a histogram (Appendix G), highlighting both well-represented and underrepresented classes. Finally, the training progress of the object detection model is visualized through a graph showcasing the

improvement in mean average precision with increasing epochs (Appendix H).

- **Database Systems and NoSQL Systems**

  - In the Database Systems (CMPS342) course, we learned about designing data models to represent real-world entities and their relationships. This knowledge helped us in structuring our Firebase database by defining collections, documents, and fields based on the entities and their attributes in ePantry. Moreover, our understanding of queries from the course allowed us to write queries to perform various tasks in the application. For example, this is a query to perform a search on the items in the user's kitchen:

```
- final result = await
  FirebaseFirestore.instance.collection('users')

  .doc(FirebaseAuth.instance.currentUser!.uid).collecti
  on('allItems').where(
    'keywords', arrayContains: query).get();
```

  - The NoSQL course on Coursera covered various NoSQL database models, including key-value stores, document databases, columnar databases, and graph databases. Understanding these concepts helped us grasp the advantages and characteristics of NoSQL databases, which Firebase follows. NoSQL databases like Firebase often use a document-oriented data model, where data is stored in flexible, schema-less documents. The knowledge gained from the NoSQL course helped us design and manipulate these documents effectively.

- **Computer Security**

Our users' security and privacy are of highest importance to us; therefore, we made sure that their passwords are encrypted safely using Firebase's password hashing algorithm. Concepts explained throughout the Computer

Security (CMPS455) course covered authentication and authorization mechanisms. We learned about hashing functions and encryption; these concepts made us more aware of the technique used to protect our user's passwords. The algorithm used for password hashing is SCRYPT; it is a password-based key derivation function designed to be computationally expensive, making it more resistant to brute-force attacks, especially after specifying the number of rounds to 8.

B. Implementation (ePantry Website)

We designed and implemented an engaging and visually appealing website to effectively showcase our senior project, the ePantry application. The website serves as an informative and advertisement platform, aimed at providing users with a comprehensive overview of the features and benefits of ePantry, while also allowing them to easily navigate through different sections and interact with our team.

The website features a sleek and intuitive user interface, with a menu located on the top right corner, providing easy access to different pages including the Home, About, Features, and Contact Us sections. The menu is designed to be responsive and user-friendly, allowing users to seamlessly explore the website and access relevant information. At the top of the website, a captivating slider showcases different screens of the ePantry app, giving users a sneak peek into the functionalities and user interface of the application. Accompanying the slider are two prominent download buttons that direct users to the app store and play store, providing a convenient and straightforward way for users to download the ePantry app and experience its features firsthand.

As users scroll down the home page, they are greeted with an eye-catching Feature section that highlights the three most important features of ePantry in an interesting and visually appealing manner. This section is carefully designed to capture users' attention and provide them with a clear understanding of the unique functionalities that ePantry offers. To further enhance users' engagement, a "Read More" button is prominently

placed, allowing them to delve deeper into the features of ePantry on a dedicated Features page.

Continuing their journey through the website, users are introduced to the Download section, which provides them with essential information about how they can download the ePantry app from the app store and play store. The section also emphasizes the user-friendly interface and security features of the app, instilling trust and confidence in potential users.

The About Us section is strategically placed to provide users with insights into the team behind ePantry, their dedication and hard work in creating the application, and their vision of transforming pantry management for users. This section not only showcases the expertise and passion of our team but also builds a connection with potential users by conveying our commitment to making their lives more efficient and organized. To further pique users' interest, a "Read More" button is provided, leading them to a dedicated About Us page where they can learn more about our team, our journey, and our values.

To keep users engaged and informed, the website includes a Subscribe section that allows them to enter their email address and subscribe to our newsletter for updates on ePantry. This section is strategically placed to capture users' attention and encourage them to stay connected with us for the latest information and developments.

The Testimonial section on the website showcases reviews from satisfied users of the ePantry app, providing social proof and testimonials about the effectiveness and usability of the application. Users can read through the testimonials and gain insights from real-life experiences, building trust and confidence in the ePantry app. A "Read More" button is also provided for users to access more reviews and further validate the credibility of ePantry.

The Contact Us section serves as a means for users to connect with our team by sending messages. Users can provide their name, email address, subject, and message content, allowing them to reach out to us with any inquiries, feedback, or suggestions. This

section is designed to be user-friendly and convenient, encouraging users to engage with us and establish a direct communication channel.

Finally, the footer of the website summarizes important information, including links to different pages, contact details, newsletter subscription, and links to ePantry's social media accounts.

This section is designed to provide users with easy access to relevant information and further opportunities to connect with us through different channels. The footer is strategically placed at the bottom of the website, ensuring that users can easily access important information without having to scroll back to the top.

In addition to the various sections and pages, the overall design of the website is carefully crafted to create a visually appealing and engaging experience for users. The use of captivating visuals, including images and graphics, complemented by concise and impactful text, effectively conveys the key messages and benefits of ePantry. The color scheme and typography are chosen to align with the brand identity of ePantry, creating a cohesive and professional look and feel throughout the website.

Furthermore, the website is designed to be responsive, ensuring that it is accessible and functional on different devices, including desktop computers, laptops, tablets, and smartphones. This ensures that users can access the website and navigate through its pages seamlessly, regardless of the device they are using, providing a consistent and user-friendly experience.

Overall, our website serves as a comprehensive and engaging platform for showcasing the features and benefits of ePantry. It not only provides users with information about the app, but also offers them an interactive experience, allowing them to easily navigate through different sections, learn more about the app's features, read testimonials, subscribe to the newsletter, and connect with our team. The visually appealing design, intuitive user interface, and compelling content work together to create an informative and engaging website that effectively promotes the ePantry app and encourages users to download and try it out for themselves.

As our team continues to refine and enhance the ePantry app, we are committed to keeping the website up to date with the latest information and developments. We will continuously listen to user feedback, analyze website analytics, and make necessary improvements to ensure that our website remains a valuable resource for users interested in learning about ePantry and its benefits. We are excited to share our project with the world through our website and look forward to welcoming new users to experience the convenience and efficiency of ePantry for managing their pantry and simplifying their lives.

- **Header Section**

  - Includes the basic meta tags for encoding, compatibility, and mobile views.
  - Sets the title of the webpage to "ePantry - Your Digital Pantry".
  - Links external stylesheets for slider, bootstrap, fonts, custom styles, and responsive styles.
  - Defines a header section with a navbar, containing a logo and a menu button.
  - Utilizes JavaScript to handle the opening and closing of a navigation overlay with links to different pages (Home, About, Feature, Contact Us). The JavaScript code for the navbar performs the following functions:

    1. Defines a function called "openNav()" that is called when the menu button is clicked.
    2. Uses "getElementById()" to select the HTML element with the id "myNav", which represents the navigation overlay.
    3. Utilizes "classList.toggle()" to add or remove the "menu_width" class to the selected element, which controls the width of the overlay, effectively toggling its visibility
    4. Uses "querySelector()" to select the HTML element with the class "custom_menu-btn", which represents the menu button.
    5. Utilizes "classList.toggle()" again to add or remove the "menu_btn-

style" class to the selected element, which controls the style of the menu button, effectively toggling its appearance.

6. Overall, the JavaScript code provides functionality to open and close the navigation overlay, as well as change the style of the menu button when it is clicked, enhancing the user experience and allowing for easy navigation through the website.

- **Slider Section**

  - The section has a container with two columns, where the left column (col-md-7) contains a detail-box with headings "your digital Pantry" and "DOWNLOAD NOW". It also has two links for downloading the app from App Store and Play Store.
  - The right column (col-md-5) contains a carousel (image slider) with four slides (carouselitems) displaying images of the app screens.
  - The carousel has indicators (carousel-indicators) at the bottom to show the current slide, and previous and next controls (carousel-control-prev and carousel-control-next) for navigating through the slides.
  - Each slide (carousel-item) has an image (img-box) with a source (src) pointing to the respective image file (images/Screens/1.png, 2.png, etc.) to be displayed in the slider.
  - The carousel is set up with "data-ride" attribute to enable auto sliding of the images.
  - The section is positioned relatively, allowing for positioning of elements within it using CSS.

- **Feature Section**

  - The section is divided into three columns of class (col-md-4), each showcasing a different feature of the application.
  - Each feature is presented in a box format with an image, a heading, and a description.

- The section also includes a "Read More" link that redirects to another page for more information about the features.
- The section is designed with responsive layout and margin for proper alignment and visual appeal.

Download Section

- Heading container with an H2 tag for "Download Anytime, Anywhere" title.
- Row div containing three columns with class "col-md-4".
- First column with two boxes, each with a "box" class.
- Box 1 with an image, "Download on App Store" title, and a paragraph for app description.
- Box 2 with an image, "User Friendly Interface" title, and a paragraph for app description.
- "Read More" link at the bottom.
- Second column with a main image in a "main-img-box" div.
- Third column with two boxes, each with a "box" class.
    1- Box 1 with an image, "Download on Play Store" title, and a paragraph for app description.
    2- Box 2 with an image, "Secure" title, and a paragraph for app description.
    3- "Download Now" link at the bottom.

About Us Section

- The "about" section is contained within a section element with the class "about_section" and has layout padding.
- The container class is used to wrap the content within the section.
- The heading_container class is used to align the heading to the left.
- The about section consists of two columns, each with a width of 5 columns for mediumsized scree

- Each column contains a detail-box element with class "b-1" and "b-2" respectively, which have a border.
- The detail-box elements contain paragraphs with the class "custom_About" for custom styling.
- The paragraphs contain information about the ePantry app, including its features, team, and purpose.
- The paragraphs are styled with line breaks and formatted text for easy reading.
- The "Read More" link within the first detail-box element has a link to "about.html" for further information.

Contact Section

This section helps in satisfying one of the most important points in the non-functional requirements which is the user has the ability to give a feedback or a review about ePantry highlighting about the importance of the user's opinions.

- The section contains a heading with the text "Contact Us" styled as an h2 element.
- The section contains a row class to create a row for a two-column layout.
- The left column is defined using the col-md-6 class and contains a contact form.
- The contact form is defined using a form element with an action attribute pointing to "send-email.php" for form submission.
- The contact form contains a contact_form-container class to wrap the form elements.
- The form includes input fields for name, email, subject, and message, each with their respective labels.
- The name, email, and subject fields are defined as input elements with type "text" and required attribute.
- The message field is defined as a textarea element with required attribute, and rows and cols attributes for size.

- The form ends with a submit button with the text "SEND" and a button element.
  - The PHP code is used to send an email from the contact section of the website.
  - The code starts with checking if the email parameter is set in the POST data.
  - It defines the email recipient address and subject.
  - It defines a function 'died' to handle errors in case of form submission issues.
  - It validates that the required form fields (name, email, subject, message) are set in the POST data, and displays error messages if they are not.
  - It sanitizes the form input data using regular expressions to prevent potential security issues.
  - It creates an email message body with the form data.
  - It sets the email headers, including 'From', 'Reply-To', and 'X-Mailer'.
  - It uses the 'mail' function to send the email with the defined recipient address, subject, message, and headers.
  - It redirects the user to a thank you page after successful submission.
  - The code uses PHP's built-in mail() function to send the email. Note that this function requires appropriate configuration on the server to work properly, such as setting up a mail server and SMTP settings.
- The right column is defined using the col-md-6 class and contains a contact image displayed using an img element with a source file "images/contactUs.png".

Information Section

- The section contains a logo box with the heading "ePantry".
- The section is divided into four columns using the Bootstrap grid system (col-md-3).
- The first column contains information "About Us" with a description of

ePantry.

- The second column contains a list of useful links (e.g., "About", "Features", "Contact Us") using an unordered list (ul) with anchor (a) elements.
- The third column contains "Contact Us" information with a description of how to contact ePantry.
- The fourth column contains a "Newsletter" subscription form with an input field for email and a submit button.
- The section ends with a social media container containing links to Facebook, Twitter, LinkedIn, and Instagram.

Footer Section

- The "footer" section is contained within a section element with the class "containerfluid" and is used for the website's copyright information.
- The copyright information includes the current year and "All Rights Reserved" text, followed by a link to the "About" page of ePantry.

C. Testing

After implementing our application, we will go into the testing phase. Mobile application testing enables us as developers to build applications that are scalable and accessible across multiple platforms. It's a process to build an application software by testing it for its functionality, usability, and consistency. This can be done by automation as well as with manual testing. There are multiple key factors to consider when testing a mobile app, the ones we will be working on are:

- Choosing the proper mobile devices to test on; it is important to select the most suitable devices for testing according to the preferences of the target group and the usage patterns.
- Choosing whether to test on simulators or real devices; while simulators help in providing real environment for testing, real devices help to hard-test the applications. Therefore, we will be testing on both.

As part of our testing process for ePantry, we conducted various tests to ensure its functionality, performance, and usability. The following is a summary of the aspects we tested and the reasons behind their sufficiency:

- We performed unit tests to validate individual components, functions, and methods of the application. Test cases were designed to cover different scenarios, including input validation, error handling, and edge cases. By conducting unit tests, we aimed to identify and fix any issues at the code level, ensuring the reliability and correctness of the app's core functionalities. First, we tested features related to user authentication. We made sure that if the user enters an invalid email or if the passwords do not match, a message will be displayed indicating the error as shown below. Next, we also tested every feature that takes in user input. We made sure that the input is valid and not empty. If the text fields were empty and the user was to press “Insert Item” button when adding items manually, the application will not react to the pressed button. Moreover, one of the features of ePantry is the ability to search for items in your household. If the user searches for an item that does not exist in their kitchen, at first nothing showed. However, we added a feature that showed “No such item exists...” to make it visually clear for our users. (see Appendix I)
- We carried out system tests to verify the integration of various modules and components within the ePantry app. Test cases were designed to assess the interactions between different features, APIs, and databases. By performing system testing, we aimed to identify any potential issues arising from the integration of multiple parts of the application and ensure the smooth functioning of the app as a whole. We first made sure that the APIs used obtained the correct information every time. For example, nutritional information provided by USDA FoodData Central API was accurate and not null. Next, we also tested the navigation between pages of the application, especially when certain data had to be sent from one page to another. To illustrate, when the user presses on a certain item, details of that item will be displayed. Therefore, we had to make sure that

the details related to the pressed item specifically were sent from the Kitchen page to the Item's Details page. (see Appendix J)

- We conducted usability tests to evaluate the user-friendliness and intuitiveness of ePantry. Test cases were designed to simulate real-world user interactions and scenarios. To do so, we gave the app to some of our friends with no prior knowledge to anything related to ePantry and allowed them to navigate it freely. We wanted to see if we have missed any bugs or errors that might arise. We noticed that a few features needed some modifications and modified accordingly.

D. Maintenance

Building and launching a mobile app doesn't mean that the work is over. In fact, maintaining and continuously updating the app is a significant task for the app development teams. Therefore, we will always strive to ensure a bug-free and seamless working mobile app. First, and throughout development, we will be using Git for version control. Since our team is made of more than one member and since we will all be working on each phase of the project, we will need a version control system to help us manage and to help us work simultaneously. We chose Git because it helps us cope with the confusion that tends to happen when multiple people from our team are editing the same files. It clearly displays the changes that were made by any team member. This way, every person of the team will always be up to date. Moreover, we acknowledge the fact that technology is forever evolving, and for that reason, once the ePantry is launched, we will continue working on adaptive maintenance by updating the app regularly and ensure that it is compatible with the latest devices and operating systems. Our team also aims to continuously work on perfective and preventive maintenance. All this does not require a special tool or software; we will simply work on our code to make sure it is up to standards at all times.

Chapter 6. Conclusion and Future Work

Conclusively, developing ePantry was not an easy journey. There was multiple steps and roads to follow to develop an application that we can be proud of, an application that will make people's lives easier and smoother. ePantry will fulfill its duties by organizing our users' pantries, fridges, and freezers, saving their time on shopping for groceries, and browsing the internet for recipes. Instead of throwing away items that have gone bad because you forgot about them, ePantry will remind you to use them. Although this is a tiny detail in our lives, yet we should always strive to organize our lives one tiny detail at a time. For it has been proved that an organized life leads to improved mental health and reduced stress.

It should be noted, however, that to bring our vision of ePantry to life, we had to follow a certain pathway. We studied the risks and ways to mitigate them prior to beginning the development, we had to be organized in working, we had frequent meetings with each other and with our advisor, and finally, we had to test the application before putting it on the market. When our application is a hundred percent ready, we must work continuously on maintaining it up to the highest of standards.

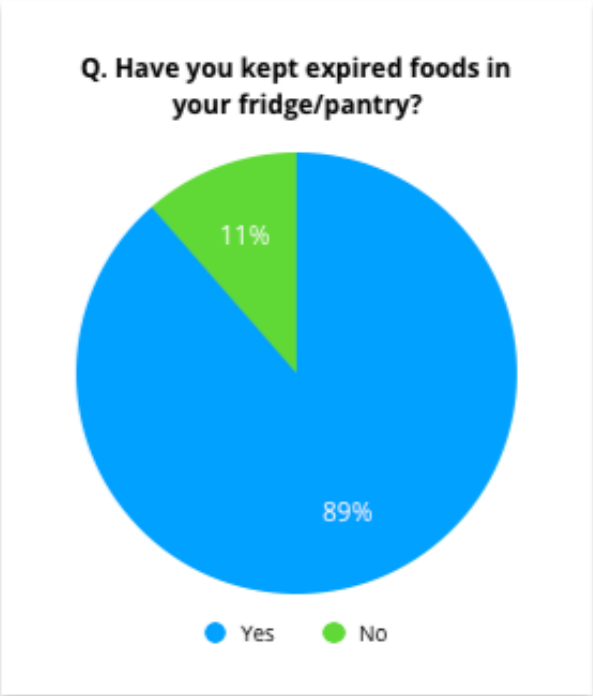
The first version of ePantry is nothing but a drop in the ocean of our idea's potential. A multitude of stretch features will be added in the future after the release of ePantry 1.0. Our team will always work to enhance the quality of life for everyone. To do so, we plan on implementing features such as multiple account households where each account can have his/her own dietary preferences and health conditions, a delivery system from nearby supermarkets, and a system that checks the availability of items from your shopping list in any nearby supermarket of your choice. The team behind ePantry will work continuously to ensure that ePantry always follows market trends and the basic needs of our lives.

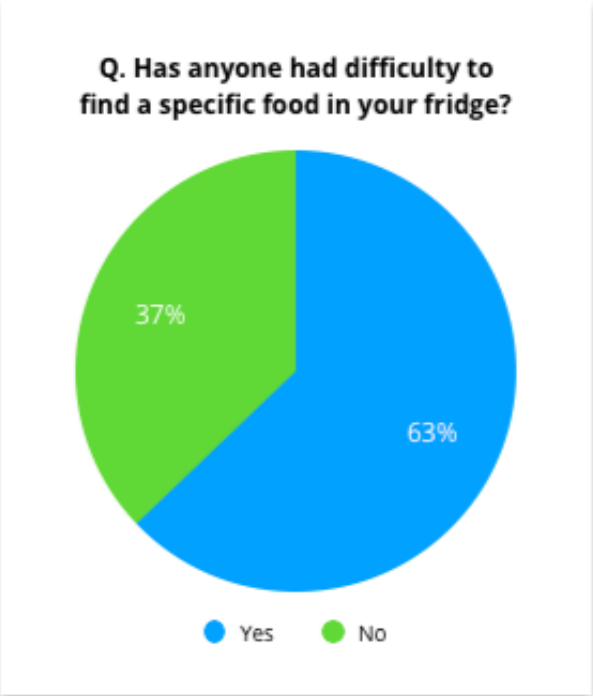
Chapter 7. References

- https://techdocs.akamai.com/cloudtest/docs
- https://kaleiwhite33.medium.com/top-10-mobile-performance-testing-tools-in-2020-affea817d820
- https://www.openxcell.com/blog/best-practices-of-mobile-app-maintenance/
- https://www.ranorex.com/blog/how-to-successfully-utilize-a-unit-test-vs-an-integration-test/
- https://pub.dev/packages/scanbot_sdk
- https://docs.scanbot.io/document-scanner-sdk/flutter/introduction/
- https://github.com/doo/scanbot-sdk-example-flutter
- https://medium.com/@mustafatahirhussein/drift-moor-database-for-flutter-an-overview-f17ff8548d85
- https://medium.com/dhiwise/how-to-choose-right-flutter-database-a1f35237a7f9

Appendix A

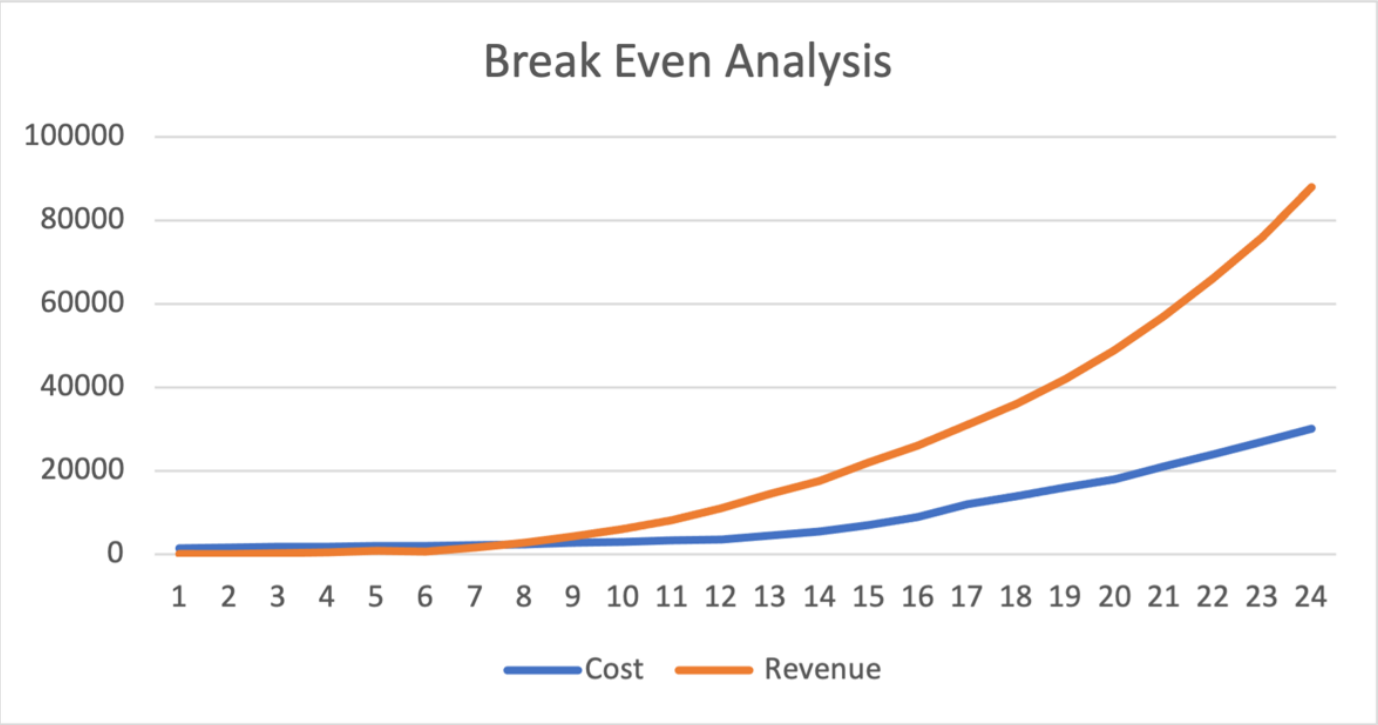
An Online Survey for Operational Feasibility





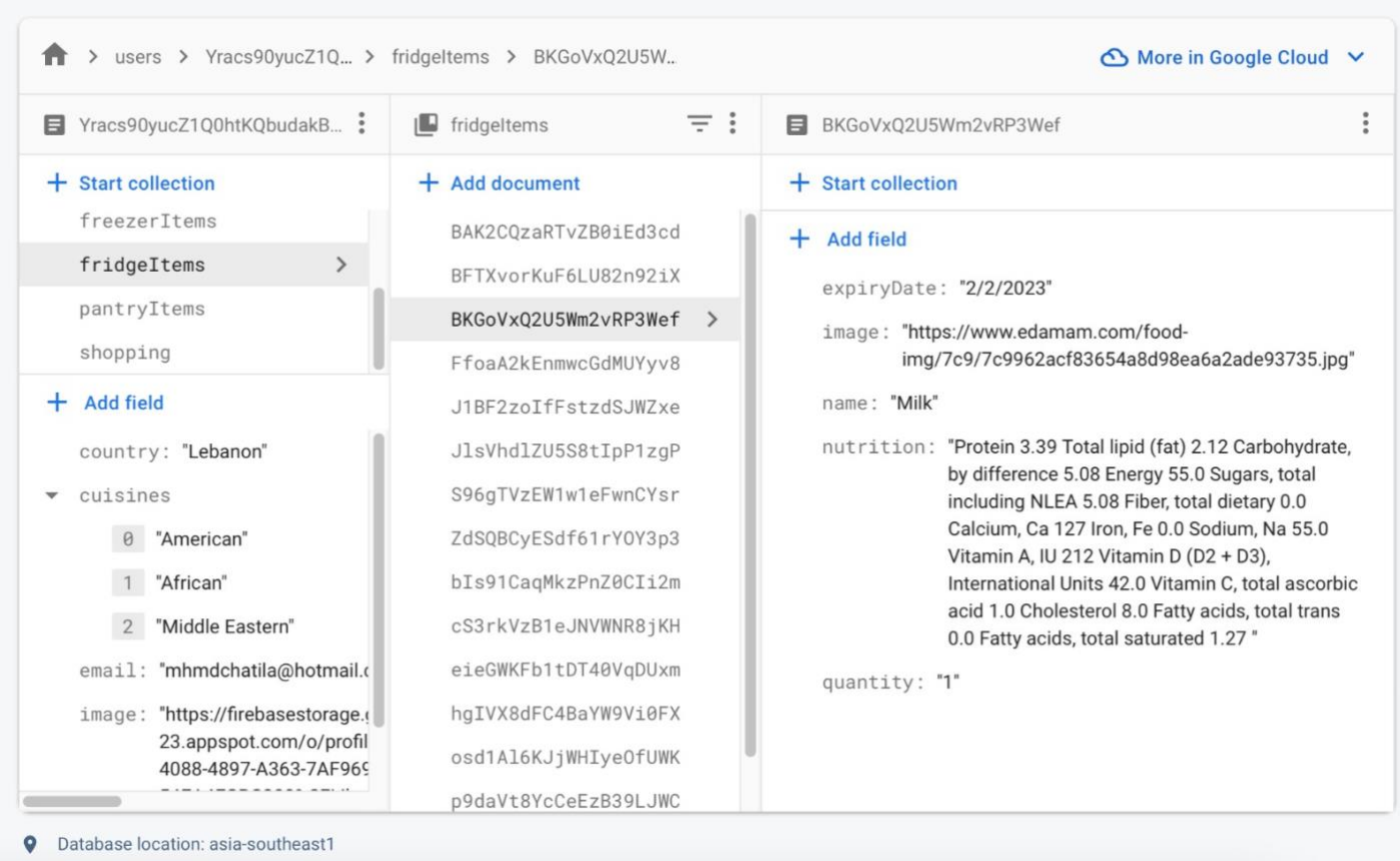
Appendix B

Break Even Analysis



Appendix C

Cloud Firestore Nested Collections



Appendix D

Packages Used In ePantry

```
cupertino_icons: ^1.0.2
get: ^4.6.5
google_nav_bar: ^5.0.6
line_icons: ^2.0.1
ionicons: ^0.1.2
icons_plus: ^3.0.0
flutter_staggered_animations: ^1.1.1
firebase_core: ^2.5.0
firebase_auth: ^4.2.6
firebase_database: ^10.0.10
google_sign_in: ^5.4.4
flutter_facebook_auth: ^5.0.7
awesome_snackbar_content: ^0.1.0
flutter_svg: ^1.1.6
animations: ^2.0.7
flutter_sticky_header: ^0.6.5
csc_picker: ^0.2.7
country_picker: ^2.0.19
image_picker: ^0.8.6+3
get_it: ^7.2.0
flutter_image_compress: ^1.1.3
permission_handler: ^10.2.0
```

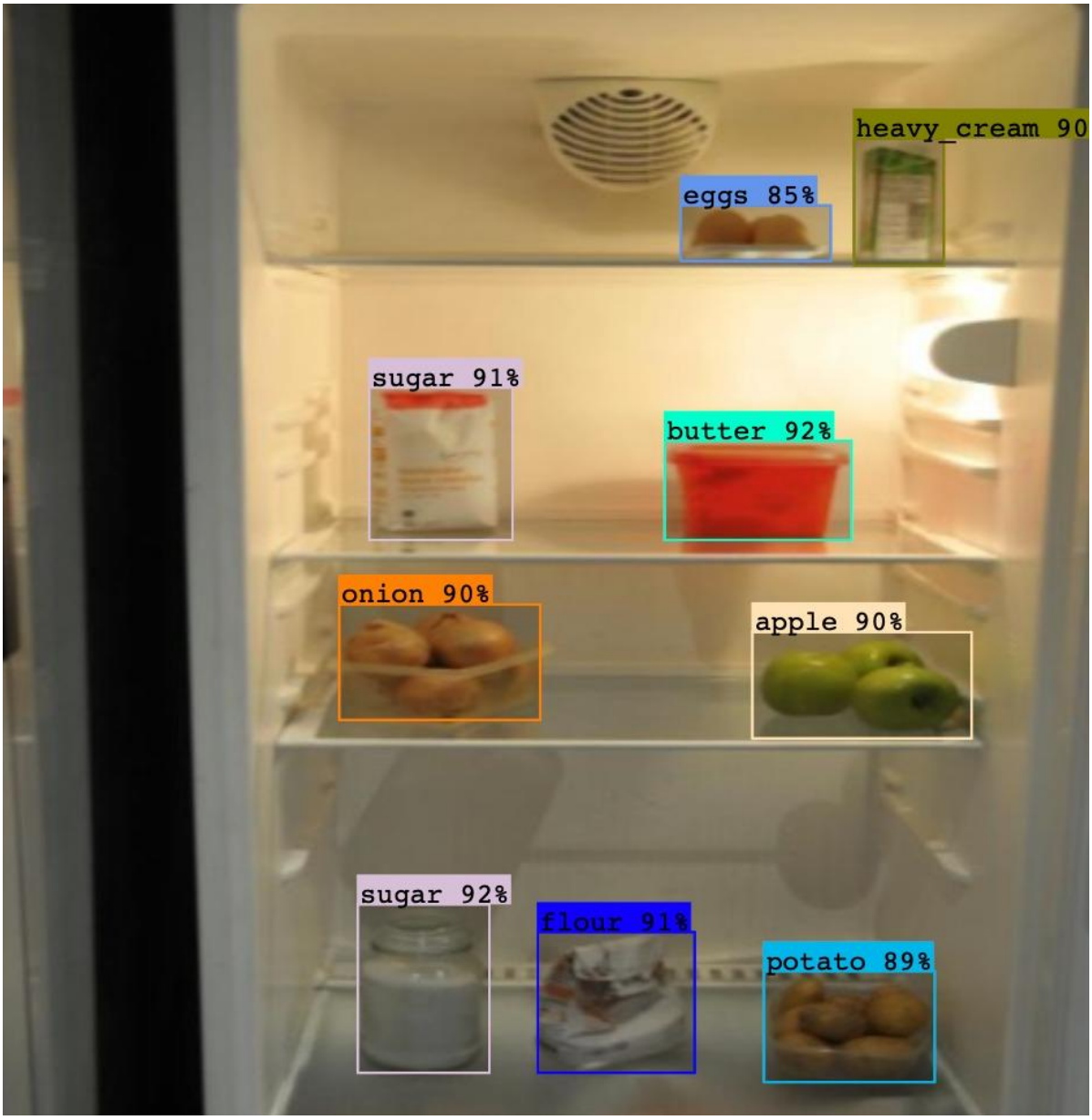
```
path_provider: ^2.0.13
cloud_firestore: ^4.4.3
intl: ^0.17.0
flutter_spinkit: ^5.1.0
http: ^0.13.5
open_filex: ^4.3.2
scanbot_sdk: ^2.11.0
shared_preferences: ^2.1.0
scanbot_image_picker: 1.0.0
logging: ^1.1.1
json_annotation: ^4.8.0
flutter_json: ^0.0.1
openfoodfacts: ^1.30.1
flutter_localization: ^0.1.10
firebase_storage: ^11.1.0
path: ^1.8.2
flutter_native_splash: ^2.2.16
cached_network_image: ^3.2.3
pie_chart: ^5.3.2
percent_indicator: ^4.2.3
```

Appendix E

A Sample of Training Dataset

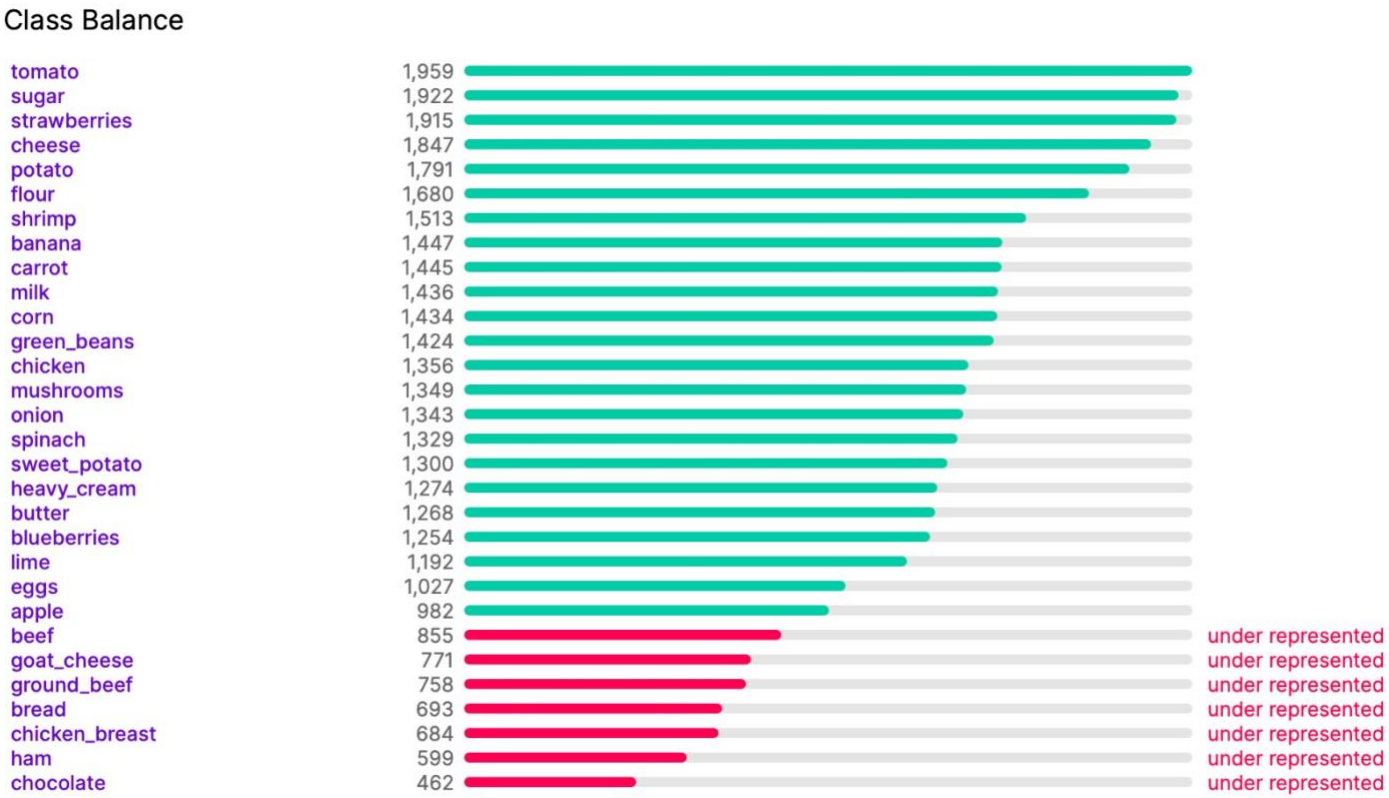
Appendix F

A Sample of Testing Dataset Result



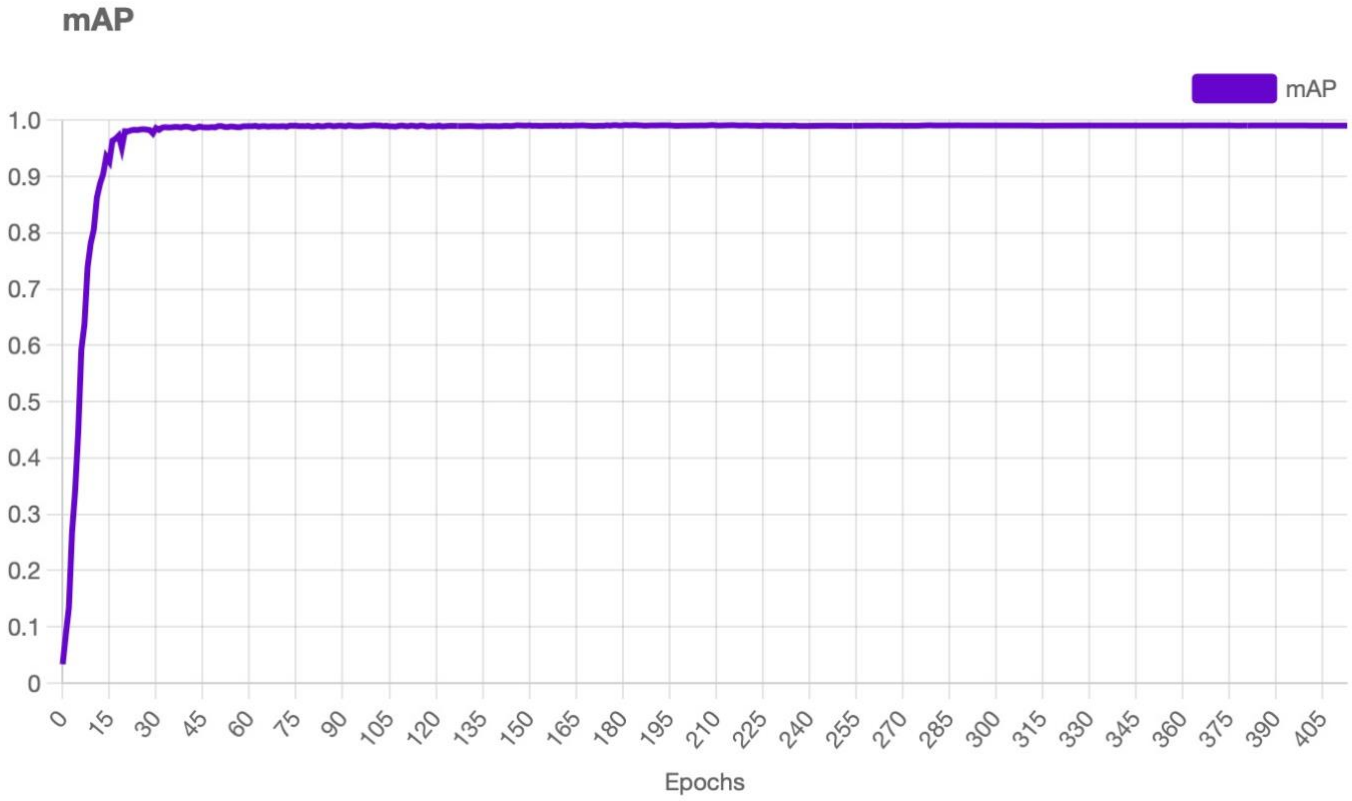
Appendix G

Object Representation Histogram



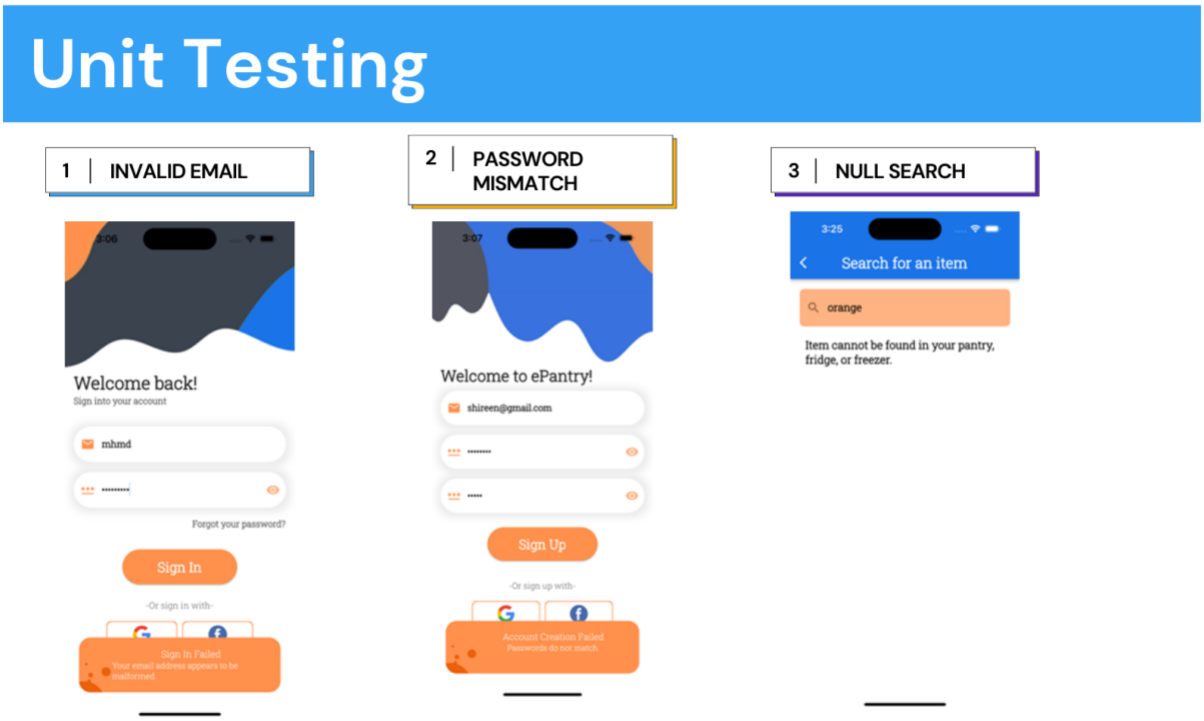
Appendix H

Mean Average Precision Graph



Appendix I

Unit Testing



Appendix J

System Testing