Project Title:
Demeter

Functional Specification

Supervisor: Hyowon Lee
Submission/Completion Date: 09/11/2023

Name	Student ID
Kline Borromeo	19372763
Ronghui Lin	19354553

Table of Contents

Introduction

1.1 Overview

Demeter, named after the Greek Goddess of Harvest, is an innovative mobile application that seeks to transform how people engage with food. Demeter aims to offer a wide range of features, including meal and recipe recommendations, grocery/pantry assistance, dietary restriction and allergen detection, meal difficulty sorting, and seasonal dish suggestions. The app will be user-centric, streamlining the user experience, and promoting healthier food choices while saving time and effort. It will also consider the needs of individuals with eating disorders and have the potential for integration with fitness trackers and wearable devices for comprehensive health management. The user-centric design will also ensure a streamlined experience, fostering healthier food choices while saving time and effort.

1.2 Business Context

There are a few ways that this application can be implemented for profit. There can be sponsorships and partnerships with participating retail stores, offering real-time price comparisons and special offers available. A subscription based offering could be implemented where premium features are locked behind, examples include access to a wider recipe database, meal planning and enhanced user support. Collaboration with celebrity chefs, nutritionists or cooking experts to create premium recipes which can be available for subscribers or one time microtransactions. Collaborating with educational institutions to offer student discounts and provide meal planning tools tailored to the student lifestyle.

1.3 Glossary

Dart	● A programming language developed by Google, used for building web, server, and mobile applications. In this project, Dart is used for mobile app development.

3

Python	● A high-level programming language used for various purposes, including machine learning and backend development.
Android Studio	● An integrated development environment (IDE) for Android app development.
iOS Application	● Refers to the mobile application developed for Apple's iOS operating system.
PyTorch	● An open-source machine learning framework used for tasks like implementing recommendation algorithms.
FireBase/FireStore	● Firebase is a cloud-based platform for building mobile and web applications, and Firestore is a NoSQL database offered by Firebase.
Flutter Framework	● An open-source UI software development kit created by Google, used for developing natively compiled applications for mobile, web, and desktop from a single codebase.
Speech/Voice Processing	● Technologies that enable the app to understand and process spoken language, such as apiai or SpeechRecognition.
Cloud Hosting	● Storing and managing application data and logic on remote servers, such as Amazon EC2 (Elastic Compute Cloud) or Google Cloud.
FrontEnd Development	● The process of designing and implementing the user interface (UI) of the mobile app.
BackEnd Development	● Involves setting up the server infrastructure, handling server-side logic, and managing the database.
User Experience UX	● Focused on ensuring that the app provides a positive and user-friendly experience for its users.

Machine Learning	● A branch of artificial intelligence that involves the development of algorithms and models to enable the app to make data-driven recommendations and perform allergen detection.
Server Management	● The ongoing maintenance and management of the server infrastructure.
Database Management	● The process of designing and maintaining the database structure, ensuring data integrity, and data security.
Allergen Detection System	● A system that identifies potential allergens in food products, often using image analysis or other methods.
Recommendation Algorithm	● Algorithms that suggest personalised recipes and meals based on user preferences and available ingredients.
User Testing	● The process of gathering feedback and testing the app with real users to refine its design and functionality.
Wearable Devices	● Devices like fitness trackers (e.g., Fitbit) and smartwatches (e.g., Apple Watch) that can be integrated with the app for comprehensive health management.
Project Management Software	● Tools like TeamGantt used for planning, tracking, and managing the project's progress.

2. General Description

2.1 Product / System Functions

The preliminary list for necessary functions for Demeter are, but not limited to:
- Register

- Login
- Pantry Tracking
- Grocery Tracking
- Recipe Recommendation
- Various Inputs, Text, Voice, Visual
- Allergen/Diet detection
- Seasonal Dishes

2.2 User Characteristics & Objectives

We expect our users to be accustomed to using mobile apps. The demographic of our users would be the college students and/or homecooks who enjoy cooking food. One of the **main learning challenges** of this project is to find a way to reduce the burden users face when faced with a lot of input actions (e.g. tapping to add items) used in other applications. The user objectives will inform us on how we can reduce these burdens.

Objective 1: Efficient input methods
- Users will have multiple options when it comes to adding ingredients to their pantry
- This could be by speech-to-text, by camera/barcode and by manual input

Objective 2: A Personalised Recommendation system
- When users are registering, they will have options on what food/cuisine they like, their diet restrictions and this will filter the recipes that Demeter will recommend to them based on these preferences.

Objective 3: Encourage the user to use the app
- If we cannot reduce user input, then we can entice users that there is worth to doing these repetitive input tasks.

2.3 Operational Scenarios

Scenario 1:
User wakes up and wants to plan a healthy breakfast for themselves. They proceed to open the Demeter app to see what ingredients they have in their kitchen and receive personalised breakfast recipes

Scenario 2: User wants to add items from their shopping trip to the pantry. They can do so while they shop or at the end when they get home. User can use various inputs such as machine vision, barcode scanning, voice recognition or by simple manual input. Ingredients inputted will be added to their user specific pantry and will be used to create recipe recommendations.

Scenario 3:
Users want to add in their dietary restriction, they can input their restrictions and it will be saved to their preferences. Then when they scan items to add to their pantry, the system will warn the user if an ingredient in the item matches the restricted items from the user. Recipe recommendations will also take into consideration the restrictions that the user has set.

6

2.4 Constraints

Experience Constraints:
In this project, we will be using technology which we haven't had experience with, such as Dart, Flutter, Mobile App Development, Server hosting on Cloud services and its associated network components for security and client connection. This will likely be the largest challenge and constraint we have as it requires us to learn and become proficient in multiple new technologies in order to develop this project.

Financial Constraints:
Due to the nature of this application using Firebase and a Cloud service for the database and server, testing large scalability will incur costs as Firebase Spark Tier (Free) is limited as well as the free cloud tier hosting from Amazon EC2 and Google Cloud Free tier.

Input Constraints:
As mentioned above, one of the machine challenges will be reducing the amount of user input which is why this is also one of our constraints, as manually scanning every item, dictating every item or even manually entering text for each item is time consuming and puts a burden on the user.

Internet Constraint:
Some of the features will require the users to have an internet connection, we must determine which features will need an internet connection and which do not. We must also consider if we will store information locally or have it all server side.

3. Functional Requirements

3.1 User Registration

Name	User Registration
Description	Allows the user to register to use the application
Criticality	Critical - Users to need to register in order to access the application
Technical Issues	User information must be secure, so the database needs to be protected.
Dependencies with other Requirements	

3.2 User Login

Name	User Login
Description	Allows users with existing username and password to log in.
Criticality	Highly important- (users will not be able to create notifications without logging in.)
Technical Issues	User information must be secure, so the database needs to be protected.
Dependencies with other Requirements	User Registration

3.3 Meal Recommendations

Name	Meal Recommendations
Description	Users will be given recommendations based on their registration details.
Criticality	It is a core feature of Demeter so we will need to ensure that this feature is working as intended
Technical Issues	Dependent on machine learning and algorithms.
Dependencies with other Requirements	User Registration and User Login

3.4 Grocery/Pantry Assistant

Name	Pantry Assistant
Description	Users will input ingredients into the Demeter app to see what items they have in their kitchen.
Criticality	
Technical Issues	Output needs to be the same for all inputs
Dependencies with other Requirements	

3.5 Dietary Restrictions/ Allergy Detection

Name	Dietary Restrictions / Allergy Detection
Description	Two Functions: 1. Demeter will check the contents of the ingredients inputted and see if it includes any allergens for restricted foods in it, 2. Restriction Exclusion: Will only show recipes that exclude restricted foods or allergens.
Criticality	Users' information must be stored securely as this is very sensitive/private.
Technical Issues	Function 2 will have certain technical challenges.
Dependencies with other Requirements	Login

3.6 Computer Vision Input

Name	Computer Vision for Ingredient Detection
Description	Computer vision will be used for detecting individual loose ingredients such as loose fruits.
Criticality	High - Would be one of the main ways of easing user input.
Technical Issues	Integrating it with the other 3 ways of user input and making them output the same data.
Dependencies with other Requirements	Login & Grocery Pantry Assistant

3.7 Voice Recognition Input

| Name	Voice Recognition
Description	Voice Processing will be used to allow users to dictate their purchases and items in their pantry.
Criticality	High - Another one of the main ways of easing manual user input.

Technical Issues	Ensuring quality of data being outputted
Dependencies with other Requirements	Login & Grocery/Pantry Assistant

3.8 Barcode Scanning

| Name	Barcode Scanning
Description	This will allow users to detect allergens and add ingredients to their pantry via their mobile phone cameras to scan barcodes.
Criticality	Critical - The ideal main way of inputting ingredients and is also used to detect allergens/dietary restrictions
Technical Issues	Ensuring quality output and ensuring correct data is pulled from the barcode.
Dependencies with other Requirements	Login, Grocery/Pantry Assistant, Allergen/Dietary Restriction Detection

4. System Architecture

Figure 4.1
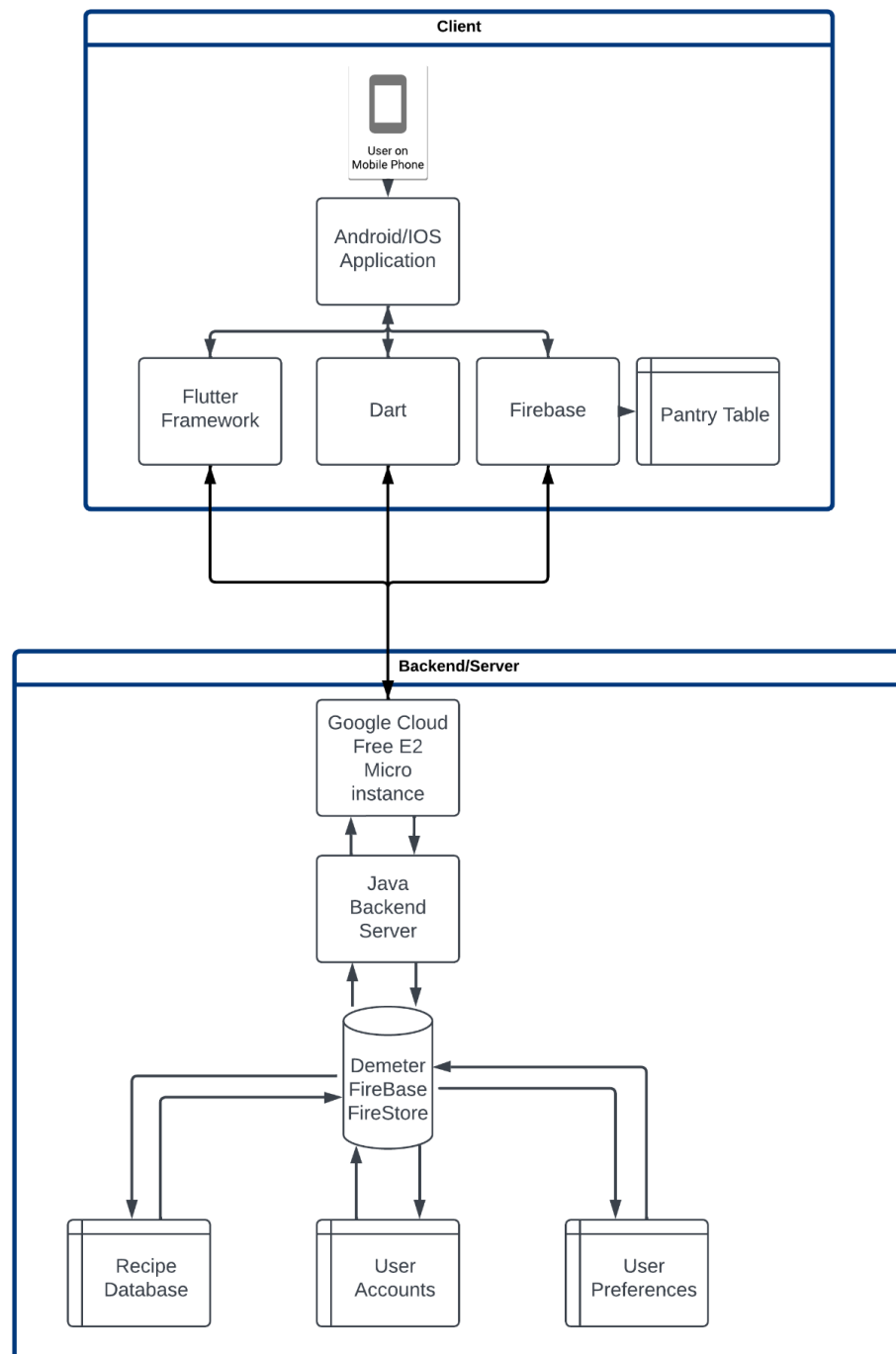
The figure 4.1 above illustrates how our system architecture would ideally look. User connects via mobile application which is either an IOS application or Android application. The client will be built on Flutter Framework and Dart in order to allow cross platform development. It will then send requests to the server which will be hosted on Google Cloud's free E2 Micro instance, which will be programmed in Java. This is to allow the server to process requests for recommendations, login requests and perform any additional processing required. In this, we have 2 main databases, one of which is client side which stores what is in each user's pantry, allowing them to add items into the "pantry" without having an internet connection, this will be synced with the Cloud server once they establish connection again. Then a Firebase database on the cloud, which will store all our recipes, user accounts and any user preferences which would need to be stored securely.

5. High Level Design

Figure 5.1
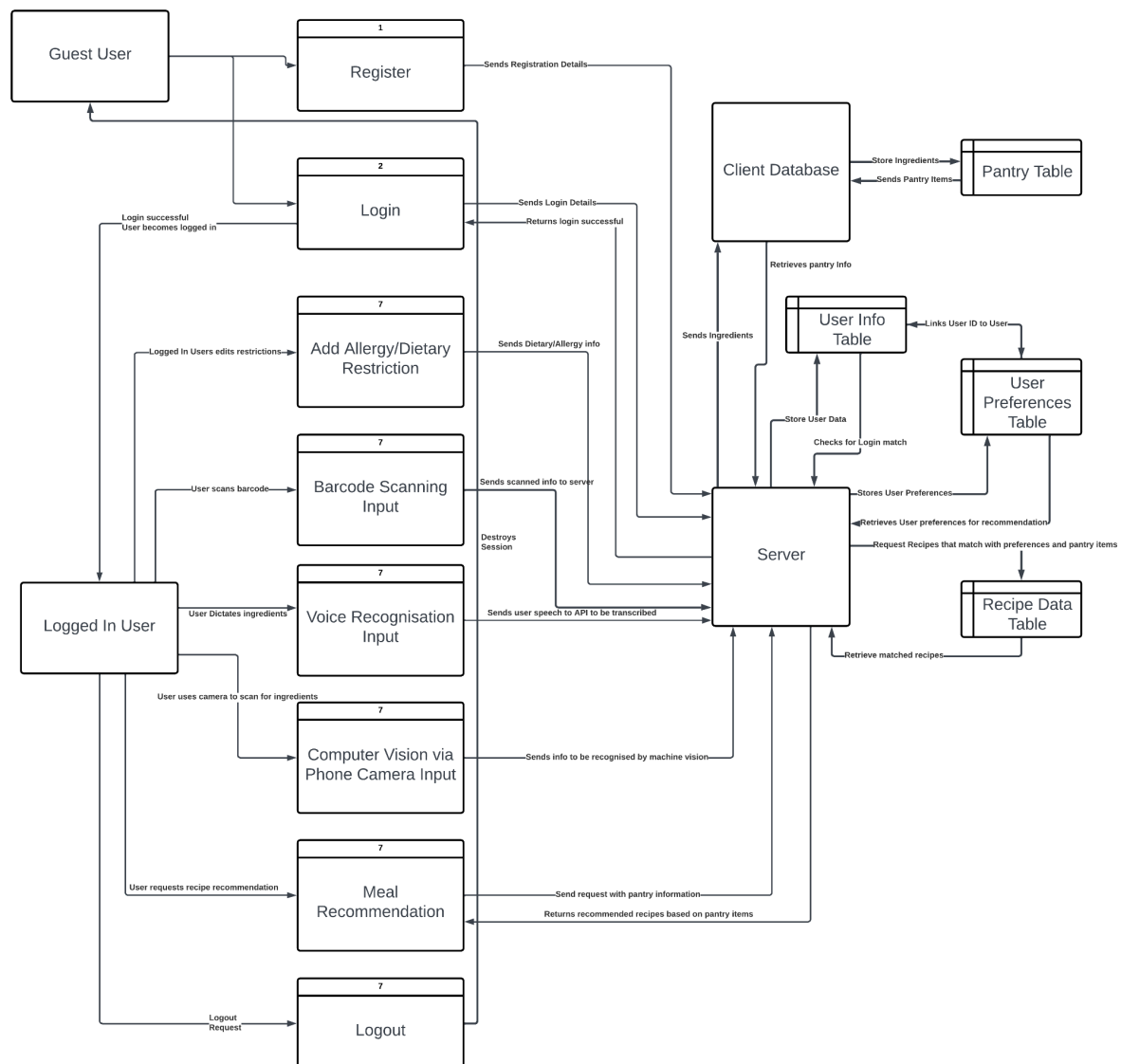
In the figure 5.1 above, it displays how the program would flow through a Data Flow Diagram (DFD), it shows how each user will interact with the system and then how the systems interact with each other in response to the user. It also shows how data flows and is processed through the program and outputted to different parts of the system to another.
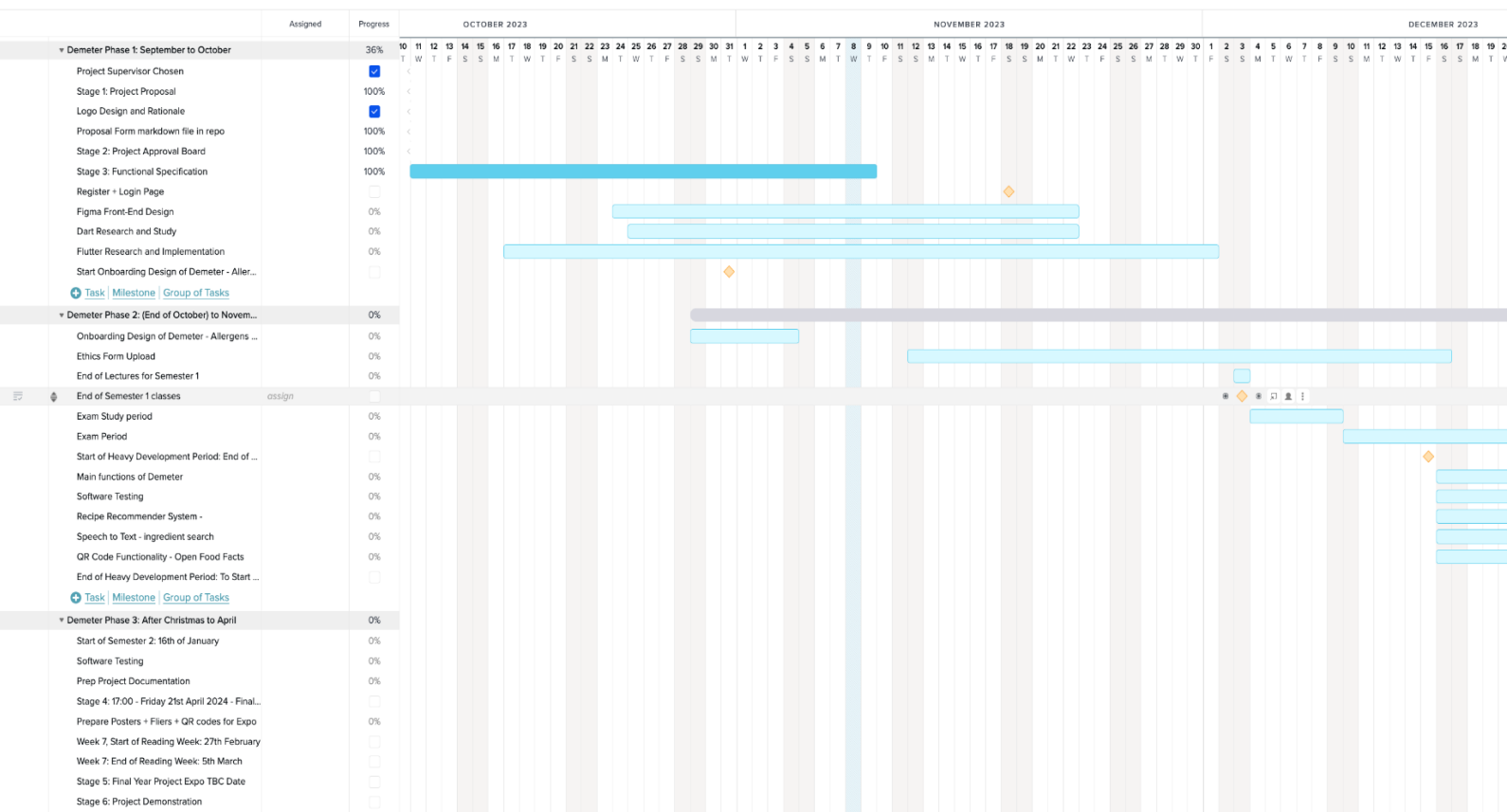
6. Preliminary Schedule

This was our first schedule that we made, but as we got our timeline given to us we adapted this into our new Timeline as shown in **Figure 6.1**.
Project Management software used: TeamGantt
- Month 1: Project planning, requirements gathering, and initial design.
    - Logo design
    - App design and feel
- Month 2: Frontend and mobile app development.
- Month 3-6: Backend development, database setup, and initial machine learning model integration.
- Month 6-7: Allergen detection system refinement and testing.
- Month 7: Meal recommendation algorithm implementation and testing.
- Month 8: Final testing, user feedback, and app launch.

Figure 6.1

[Gantt Chart Demeter]