

# DYNAMIC GROCERY LIST APP

## SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

CS 4850 02

Spring 2024

Professor Perry

February 13, 2024

SP-5 Red

## Table of Contents

1.0 Introduction .....	3
1.1 Overview .....	3
1.2 Project Goals .....	3
1.3 Definitions and Acronyms .....	4
1.4 Assumptions .....	4
2.0 Design Constraints .....	4
2.1 Environment .....	4
2.2 User Characteristics .....	4
2.3 System .....	5
3.0 Functional Requirements .....	5
4.0 Non-Functional Requirements .....	7
4.1 Security .....	7
4.1.1 Encryption .....	8
4.1.2 Authentication .....	8
4.2 Capacity .....	8
4.3 Usability .....	8
4.3.1 UI .....	8
4.3.2 Sorted Lists .....	8
4.4 Other .....	8
5.0 External Interface Requirements .....	9
5.1 User Interface Requirements .....	9
5.2 Hardware Interface Requirements .....	9
5.3 Software Interface Requirements .....	9
5.4 Communication Interface Requirements .....	9
APPENDICES .....	10

## 1.0 Introduction

### 1.1 Overview

This software requirements document details all functional and non-functional requirements for the Dynamic Grocery List mobile application. The document begins by first discussing the project's goals then lists any acronyms used, along with their definitions, and the assumptions made while developing the application. Next, it details the design constraints for the application which includes the environment, user characteristics, and the system itself. Now, the functional requirements are listed one after another. Each begins with a short sentence stating what the requirement is and then they are followed by a longer description of the requirement. Once all functional requirements have been listed, the non-functional requirements are then described, sorted into four categories: security, capacity, usability, and other non-functional requirements that do not fit into those categories. Finally, the document lists all external interface requirements which include the user, hardware, software, and communication interfaces.

### 1.2 Project Goals

The modern world moves fast, and it can be hard to keep track of everything at once. Making trips to the store already takes more time than most people would like so it can be extremely frustrating to come back from a trip to find someone else needed an item, but it was not on the list. Now, an extra trip must be made to get the item that was forgotten. This project aims to save time by allowing users to collaborate on lists through their mobile devices.

One of the application's goals will be to allow users to create lists to add and remove items from. Users should then be able to share their lists with other users through the application so they can both add and remove items from the list. The users' views of the lists will also be updated in real-time to ensure they are both viewing the same list and items are not missed due to delays in updating list views. The

application should also be available to both iOS and Android users, which will be accomplished by using Flutter with Dart. We will use Firebase and its features as the backend for the application. In particular, the Firebase Authentication feature will allow users to create accounts using their emails and passwords. Users can then be identified through these accounts to collaborate on their grocery lists.

In the end, the project will allow users an efficient way to collaboratively manage their grocery lists through the simple application we will develop.

### 1.3 Definitions and Acronyms

UI: User Interface

### 1.4 Assumptions

The application is being produced in Flutter with Dart, so it should be able to run on most mainstream Android and iOS devices. With Firebase as the intended backend for this application, there should be no issues in terms of compatibility with Flutter and Firebase, due to both being developed by Google.

## 2.0 Design Constraints

### 2.1 Environment

This application is made purely for mobile devices and will require a reliable connection to the internet for full functionality.

### 2.2 User Characteristics

Users must have an iOS or Android device and have basic proficiency in navigation and use of said device.

## 2.3 System

This application will rely on Firebase as the backbone for basic functionality of the app and for services such as login and authentication.

## 3.0 Functional Requirements

### 3.1 Functional Requirement 1

The application allows users to create and modify lists.

#### 3.1.1 Description

The application allows users to create and name multiple lists within the app. Whenever a user creates a list, they are registered as the owner of that list.

### 3.2 Functional Requirement 2

The application allows users to delete lists.

#### 3.2.1 Description

Users should be able to manage their lists by deleting ones they no longer want. Furthermore, only users registered as owners should be able to delete those lists. Users invited to view and/or edit should not be able to delete the list.

### 3.3 Functional Requirement 3

The application allows users to share lists with other users.

#### 3.3.1 Description

Users who are registered as owners of the lists should be able to invite selected users to view and edit their lists.

### 3.4 Functional Requirement 4

The application updates both users' views of the lists in real time.

#### 3.4.1 Description

The application should be able to update the lists in real time. This ensures that any updates one user makes to a shared list are seen immediately by the other user.

### 3.5 Functional Requirement 5

The application is multiplatform.

#### 3.5.1 Description

The application should be available and function equally on both the iOS and Android operating systems via Flutter.

### 3.6 Functional Requirement 6

The application has multiple views in the UI to access.

#### 3.6.1 Description

The application should have multiple views, one for the user's available lists and another to view or edit a single list.

### 3.7 Functional Requirement 7

The application allows users to create and store accounts in Firebase.

### 3.7.1 Description

Users can create an account using an email and their own password. They can then use this account to share lists with other users. Once the account is created users will be able to set their own display name.

## 3.8 Functional Requirement 8

The application has a login system that utilizes created accounts.

### 3.8.1 Description

The application has a login interface to allow users to access and share lists. It should also allow users to request a password change if they forget their original one.

## 3.9 Functional Requirement 9

The application has a search function to search for specific items in grocery lists.

### 3.9.1 Description

The application allows users to search for keywords in their grocery lists.

## 4.0 Non-Functional Requirements

### 4.1 Security

Application has security features implemented to both secure the application and user data.

#### 4.1.1 Encryption

The application has a form of encryption to ensure the confidentiality and integrity of data as it is sent between the application and the database.

#### 4.1.2 Authentication

The application ensures that user accounts and their lists cannot be accessed without the appropriate login details. This guarantees that malicious users cannot tamper with or delete the lists of other users.

#### 4.2 Capacity

Application is able to function seamlessly for multiple users using the databases concurrently.

#### 4.3 Usability

##### 4.3.1 UI

The UI has a pleasing look and simplistic feel that ensures the app is easy to use. The design should be able to work for several types of mobile devices.

##### 4.3.2 Sorted Lists

The application allows users to sort lists in order of last updated. This ensures that lists more recently created or updated by the user will be at the top for easy access.

#### 4.4 Other

The application has a feature to notify users when changes are made to lists that they created or have permission to edit. The notifications should be set to occur at reasonable intervals and not at every minor change to ensure the other user's device is not sent too many notifications.



## 5.0 External Interface Requirements

### 5.1 User Interface Requirements

Intuitive user interface allows the options to create, access, and delete grocery lists. Gives users options to share grocery lists with other users and can view a list of shared lists.

### 5.2 Hardware Interface Requirements

Compatible with most Android and IOS smartphones and tablets.

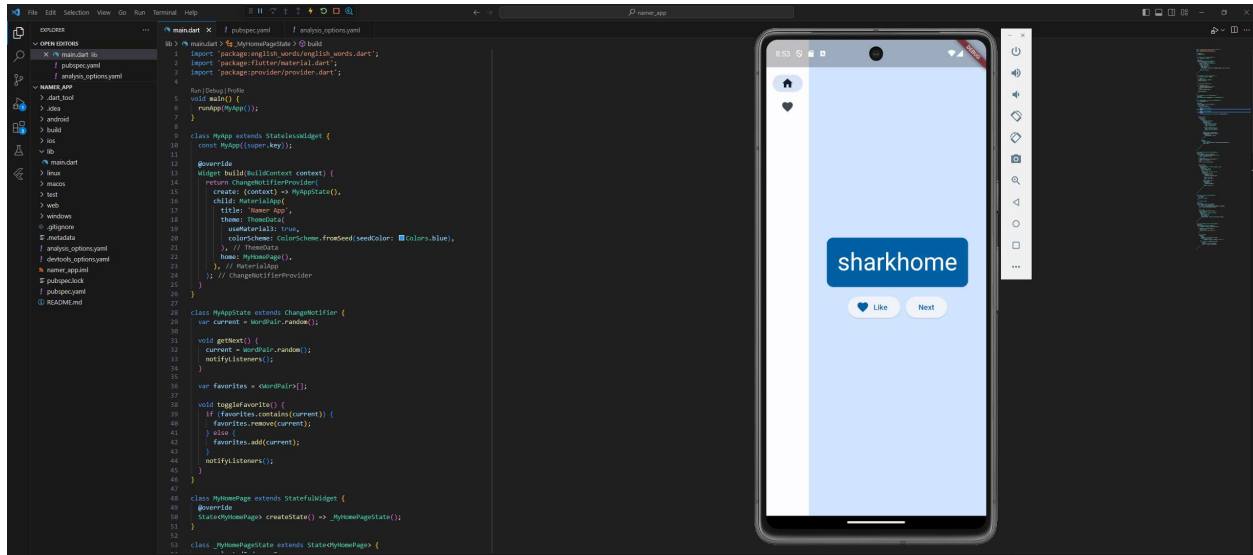
### 5.3 Software Interface Requirements

Application connects to some form of database such as Firebase Firestore to store and access the grocery lists.

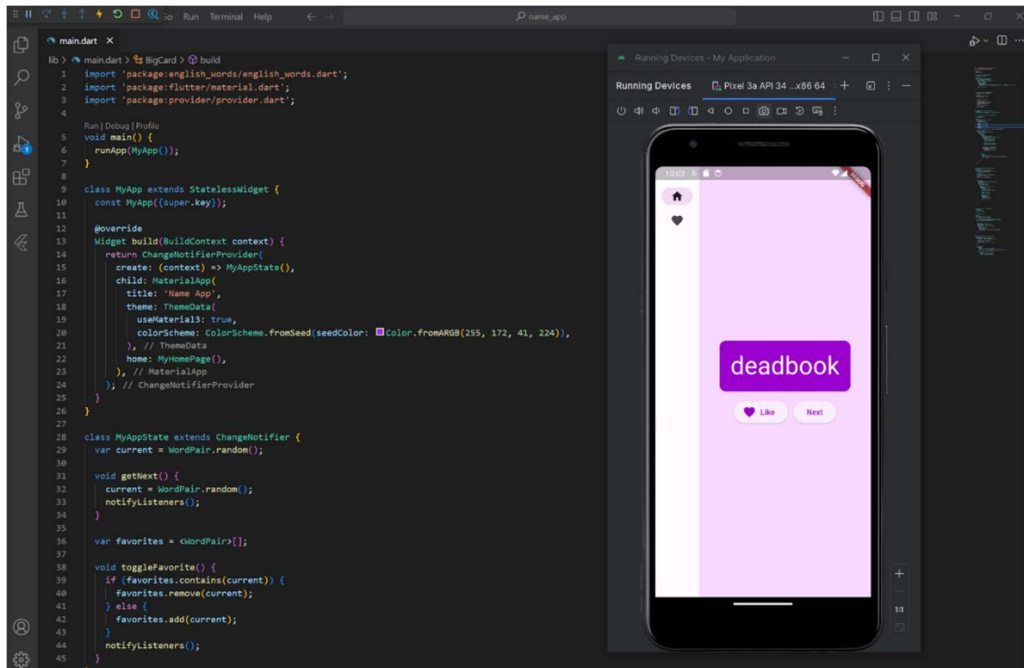
### 5.4 Communication Interface Requirements

Accessed lists are stored locally as a cache for viewing when mobile devices are offline.

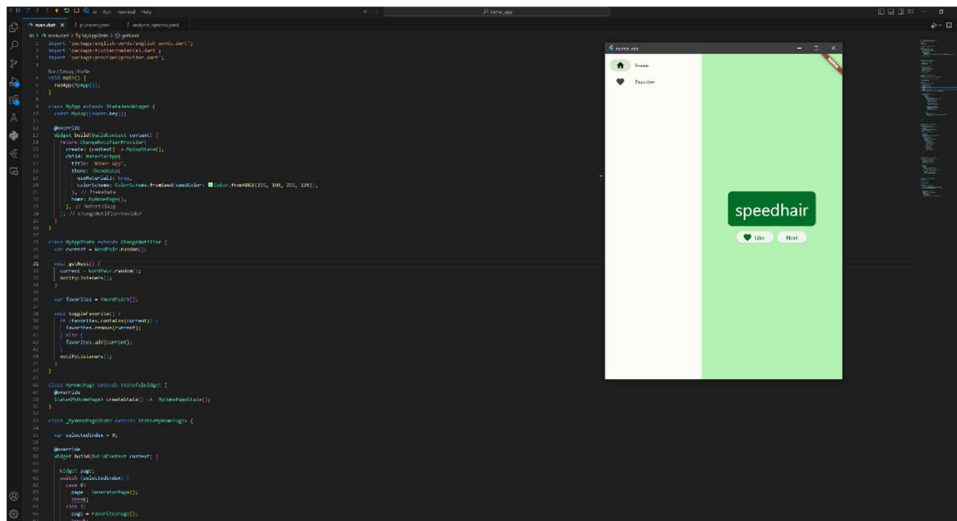
## APPENDICES



Cole Young completed Flutter tutorial screenshot.



Elizabeth Krummert completed the Flutter tutorial screenshot.



Aaron Figueroa completed Flutter tutorial screenshot.