

# Green List

## SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

CS-4850 Section I

Spring 2026

Professor Perry

February 1st 2026

SP-5 Green: Tate Swidorski, Kevin Gomes, Chris Heyward

## Table of Contents

1.0 Introduction	3
1.1 Overview	3
1.2 Project Goals	3
1.3 Assumptions	3
2.0 Design Constraints	3
2.1 Environment	3
2.2 User Characteristics	3
2.3 System	3
3.0 Functional Requirements	3
3.1 Login / Authentication	3
3.2 Your Lists Page	4
3.3 Inside a List	4
4.0 Non-Functional Requirements (use if applicable)	4
4.1 Security	4
4.2 Capacity	4
4.3 Usability	4
4.4 Other	4
5.0 External Interface Requirements	4
5.1 User Interface Requirements	4
5.2 Software Interface Requirements	4
5.3 Communication Interface Requirements	4

## 1.0 Introduction

### 1.1 Overview

Green List is a mobile grocery list application built using the flutter framework. The app will allow users to create and manage multiple grocery lists, collaborate on said lists with friends, and the lists will update automatically in real time.

### 1.2 Project Goals

- Provide a simple grocery list application for iOS (Android is TBD).
- Allow users to create an account and log in securely.
- Allow users to create, edit, and delete multiple grocery lists.
- Allow users to add, edit, and delete items within lists.
- Provide real-time synchronization of grocery lists across multiple users.
- Allow users to invite other users to join their grocery list (allow removal as well).
- Provide logout functionality.
- Provide a clean, user-friendly interface.

### 1.3 Assumptions

- The user must have internet access to use the application.
- The user must have an email in order to sign up.

## 2.0 Design Constraints

### 2.1 Environment

- The application will be developed as a mobile application using Flutter.
- The application shall support iOS devices (Android is TBD).
- Firebase shall be used for authentication and database services.
- Spring Boot will be used for backend API services.
- Figma may be used in order to help design UI.

### 2.2 User Characteristics

- Intended for general users
- Roommates, family, friends are collaborators

### 2.3 System

- Flutter mobile application
- Firebase authentication
- Firebase database
- Spring Boot for backend

## 3.0 Functional Requirements

### 3.1 Login / Authentication

- Create account
- Login with username and password
- Should be redirected to this page if the user is not logged in

### 3.2 Your Lists Page

- Display all of your lists / lists you are a collaborator on
- Add or remove lists from this page

### 3.3 Inside a List

- Display all of the items on the list
- Allowed to add, remove, and edit items on the list
- Can invite collaborators on this page
- List updates in real time

## 4.0 Non-Functional Requirements (use if applicable)

### 4.1 Security

- User authentication will be handled using Firebase Authentication.
- Users will only be able to manage and view lists they have access to.

### 4.2 Capacity

- The system shall support multiple grocery lists per user.
- The system shall support multiple items per grocery list.
- The system shall support multiple collaborators per grocery list.

### 4.3 Usability

- The system shall provide a simple and user-friendly interface.
- The system shall allow users to add, edit, or delete items with minimal steps.
- The system shall be usable without requiring training or technical knowledge.

### 4.4 Other

- Grocery list updates shall appear in real time across all users in a list.

## 5.0 External Interface Requirements

### 5.1 User Interface Requirements

- The system shall provide a login/account creation screen.
- The system shall provide a "Your Lists" page to display and manage grocery lists.
- The system shall provide a grocery list page showing all list items.
- The system shall provide buttons/options for adding, editing, and deleting grocery list items.
- The system shall provide an interface for inviting and removing collaborators.
- The system shall have a button to logout.

### 5.2 Software Interface Requirements

- The system shall interface with Firebase Authentication.
- The system shall interface with Firebase Database services.
- The system may interface with Spring Boot backend API services if needed.

### 5.3 Communication Interface Requirements

- The system shall require internet access for real-time updates.