

Software Requirements

Specification

for

NutrientNetwork

Version 1.0

Prepared by: Dylan Mejia, Jaydan Florian, Anthony Rock

October 2025

Table Of Contents

Cover Page.....	P. 1
Table of Contents.....	P. 2
1. Introduction	P. 3-4
1.1. Purpose.....	P. 3
1.2. Document Conventions.....	P. 3
1.3. Intended Audience and Suggested Reading Order.....	P. 3
1.4. Product Scope.....	P. 4
1.5. References.....	P. 4
2. Overall Description.....	P. 4-5
2.1. Product Perspective.....	P. 4
2.2. Product Functions.....	P. 4
2.3. User Classes and Characteristics.....	P. 5
2.4. Operating Environment.....	P. 5
2.5. Design and Implementation Constraints.....	P. 5
2.6. User Documentation.....	P. 5
2.7. Assumptions and Dependencies.....	P. 5
3. External Interface Requirements.....	P. 6-7
3.1. User Interfaces.....	P. 6
3.2. Hardware Interfaces.....	P. 6
3.3. Software Interfaces.....	P. 6
3.4. Communication Interfaces.....	P. 7
4. System Features.....	P. 7-8
4.1. Key Features.....	P. 7
4.2. Description and Goal.....	P. 7
4.3. Stimulus Response Sequences.....	P. 8
4.4. Functional Requirements.....	P. 8
5. Other Non-Functional Requirements.....	P. 8-9
5.1. Performance Requirements.....	P. 8
5.2. Safety Requirements.....	P. 8
5.3. Security Requirements.....	P. 9
5.4. Software Quality Attributes.....	P. 9
5.5. Business Rules.....	P. 9
5.6. Other Requirements.....	P. 9

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document outlines the requirements for the healthy recipe sharing app, NutrientNetwork. NutrientNetwork is a Java-based platform where users can share, browse, search, and save healthy recipes that meet their nutritional goals. This document will describe the full scope of the software project and the functions that the system is required to follow for version 1.0.

1.2 Document Conventions

This document was created based on the IEEE template for System Requirement Specification Documents. Functional Requirements are labeled as “FR-” and Non-functional Requirements are labeled as “NFR-”. Every requirement statement has its own priority.

1.3 Intended Audience and Suggested Reading Order

This document is intended for :

- Project Managers: For planning and allocation of resources
- Developers: For implementing the specified functional features for the application.
- QA/Testers: For testing different prototypes based on software requirements.
- Stakeholders: For validation and acceptance of the software’s functionality.

Readers of this document should begin with Section 1 for the project context, Section 2 for the system overview, and Sections 3–5 for the detailed requirements.

1.4 Product Scope

NutrientNetwork assists users to find and share healthy recipes, they can also track recipes to achieve certain dietary goals, and interact with like-minded people from anywhere in the world. The application aims to improve health through the management of diet and nutrition with the use of technology and a social platform.

1.5 References

- Google Firebase Documentation (2025)
- JavaFX Developer Guide (Oracle, 2024)
- Github Page

2. Overall Description

2.1 Product Perspective

NutrientNetwork is a standalone JavaFX desktop app with Firebase integration for backend services. It promotes active community engagement through recipe sharing and dietary goal tracking in a user-friendly environment. The application was developed to be able to run on operating systems like Windows, MacOS, and Linux.

2.2 Product Functions

- User registration and account authentication
- Recipe posting, editing, and saving
- Recipe tagging based on nutritional content
- Search, filter, and browse through recipes

- Social features (friends, likes, comments)
- Notifications and dietary goal tracking

2.3 User Classes and Characteristics

- Regular Users: Can register, upload, and browse recipes, and interact with other users
- Administrators: Can monitor posts and handle content moderation
- Developers: Can maintain, add, remove and test application features

2.4 Operating Environment

- Frontend: JavaFX desktop app (Windows/macOS/Linux)
- Backend: Firebase Firestore, Authentication, and Storage
- Development Tools: Java 21+, Firebase SDK
- Internet required for data operations

2.5 Design and Implementation Constraints

Constraints include:

- Must use Firebase as backend
- JavaFX as GUI framework
- Optional API integration must remain within free usage tier

2.6 User Documentation

User documentation includes a PDF guide, interactive help menus, video tutorials, and a README setup file for developers.

2.7 Assumptions and Dependencies

Assumptions:

- Users have stable internet access and compatible devices

- Firebase services need to remain operational

Dependencies:

- Firebase SDK, JavaFX libraries, Nutrition API (optional)

3. External Interface Requirements

3.1. User Interfaces

All screens should maintain a consistent design for easy use and transitions. Examples include the JavaFX GUI Home Feed, Recipe Details, Search, Creating A New Recipe, User Profile, and Account Settings. User Interfaces should be easy to navigate with different button labels. NutrientNetwork screens should have a consistent color palette, design and navigational flow for a better user experience.

3.2 Hardware Interfaces

Standard desktop hardware, and input devices are supported. The application uses device storage and camera for recipe image uploads.

3.3 Software Interfaces

NutrientNetwork requires Java to be installed on the system. The app also requires database integration with Firestore for the database along with Firebase Auth to control user access and Firebase Storage for media. There is also optional integration with a nutrition API for retrieving macro and calorie data.

3.4 Communications Interfaces

Firebase SDK handles all communications over HTTPS. Push notifications are powered by Firebase Cloud Messaging.

4. System Features

4.1 Key Features

- User account authentication
- Profile management and personalization
- Creating recipes to share publicly
- Adding other users as friends
- Search for specific recipes with filters
- Set and track dietary goals
- System notifications for recipe postings

4.2 Description and Goal

NutrientNetwork helps users find and share wholesome recipes, achieve dietary goals, and connect with a community of users focused on health. The wellbeing and health improvement of users is the primary focus of the project, and we incorporate technology to simplify the management of nutrition on a day-to-day basis.

4.3 Stimulus/Response Sequences

User opens app → selects ‘Sign Up or Login’ → user enters credentials → system verifies via Firebase → user redirected to dashboard.

4.4 Functional Requirements

- FR-1: Enable secure authentication for user registration and logins through Firebase.
- FR-2: Allow options to create and share a recipe by uploading pictures and text.
- FR-3: Allow users to set, edit, track, and save dietary goals.
- FR-4: Allow users to add friends on the app and view their recipes.
- FR-5: Create, store and retrieve user records in Firebase Firestore.
- FR-6: Perform secure login/logout of the application.

5. Other Nonfunctional Requirements

5.1. Performance Requirements

- NFR-1: The app screens should load in 2 seconds or less.
- NFR-2: Images can be uploaded in less than 10 seconds.
- NFR-3: The app should support 100 concurrent users during testing.

5.2. Safety Requirements

NFR-4: Safety moderation tools enable the removal of potentially inappropriate or unsafe materials. Administrators review community safety via reporting tools as well as curtains on posts.

5.3. Security Requirements

- NFR-5: All data secured using HTTPS
- NFR-6: All users authenticated via Firebase
- NFR-7: Passwords are encrypted and not stored
- NFR-8: Firestore rules are user access levels

5.4. Software Quality Attributes

- Usability: The interface is simple and intuitive
- Reliability: Persistence in the cloud
- Maintainability: Java structure is modular
- Scalability: Future iterations can be expanded to the web and/or mobile

5.5. Business Rules

- Users must post healthy recipes only, and must meet community guidelines.
- Administrators can remove inappropriate content
- Users can hide recipes and control who can see their private recipes

5.6 Other Requirements

- Minimum 70% of tests require coverage
- Data deletion is in line with user privacy policies