Software Requirements Specification (SRS)

Project Title : KrushiDava – Village Pesticide Availability App
Platform : Flutter (Cross-platform mobile app)
Date : July 2025

Table of Contents

1. INTRODUCTION

1.1 Purpose of this Document

This SRS outlines the requirements for developing a mobile application that displays information about agricultural pesticides. Admin users can manage pesticide entries, and registered users can view details and maintain a wishlist. The system is intended for educational use and field reference but not for direct purchasing.

1.2 Scope of the System

The app will :

- Allow Admin to add, update, delete pesticide details.

- Allow registered users to browse pesticide information.
- Allow users to add pesticides to a wishlist.
- Not handle online purchases; users must visit agro stores for actual transactions.

1.3 Overview

The system includes two main roles :

- **Admin** : Full CRUD control over pesticide items.
- **User** : View items, read descriptions, and manage a wishlist.

1.4 Definitions, Acronyms, and Abbreviations

Acronym	Definition
CRUD	Create, Read, Update, Delete
UI	User Interface
SRS	Software Requirements Specification
Flutter	UI toolkit for mobile development

1.5 Business Context

Farmers and students often need fast access to pesticide information. This app acts as a lightweight mobile catalog. It helps educate users about available pesticides and guides them to buy offline from agro shops.

1.6 References

- chatgpt.com
- https://agritech.tnau.ac.in/govt_schemes_services/pdf/2013/SRS%20Service%201_Part1_Pesticide1_1.pdf
- https://agritech.tnau.ac.in/govt_schemes_services/pdf/2013/SRS%20Service%203%20V%201.1_02072012.pdf
- https://www.wikipedia.org/
- https://www.coewm.com/documents/trg-ref-materials/ref-Software-requirement-sepcification-document-Odisha-training-on-establishment-of-DL-portal.pdf

2. GENERAL DESCRIPTION

2.1 Product Functions

- Admin login and dashboard.
- CRUD operations on pesticide items.
- User registration and login.
- Pesticide browsing and filtering.
- Wishlist management.
- Offline buying reminder.

2.2 Similar Systems

Currently, most agro-based applications are either :

- **E-commerce platforms** (e.g., AgroStar, BigHaat) : These apps offer complete online shopping experiences for seeds, fertilizers, and pesticides, including payment gateways

and delivery logistics.

- **Government or research portals** (e.g., Krishi Vigyan Kendra portals, agropedia) : These provide scattered and text-heavy information, often not optimized for mobile usage or general public accessibility.

2.3 User Characteristics

- **Admin** : Agro experts, shop owners, or college staff. Familiar with pesticide names and uses.

- **Users** : Students, farmers, or general public with basic mobile skills.

2.4 User Problem Statement

Users often lack reliable info about pesticides. They rely on verbal recommendations or outdated catalogs. There's no central place to compare products.

2.5 User Objectives

- To check pesticide details before buying.

- To manage a personal wishlist for reference.

- To avoid confusion in choosing products offline.

2.6 General Constraints

- Internet is required to sync data.

- No purchase through the app.

- Only admin can update product info.
- Simple UI for semi-literate users.

3. FUNCTIONAL REQUIREMENTS

3.1 Admin Module

- **FR1** : Admin can log in securely.
- **FR2** : Admin can add new pesticide entries (name, brand, use, dosage, image).
- **FR3** : Admin can update pesticide details.
- **FR4** : Admin can delete unwanted or outdated pesticide entries.
- **FR5** : Admin can view a list of all entries.

3.2 Registered User Module

- **FR6** : User registration and secure login.
- **FR7** : Users can view all pesticide entries with images and details.
- **FR8** : Users can filter by brand, category, or crop type.
- **FR9** : Users can add pesticides to their wishlist.
- **FR10** : Users can view and remove items from the wishlist.

- **FR11** : Reminder shown to "Visit agro store to buy".

4. INTERFACE REQUIREMENTS

4.1 User Interfaces

- Mobile-friendly UI.
- Separate dashboards for Admin and User.
- Product cards with image, name, and short info.
- Detail screen on tap.
- Wishlist section for users.

4.2 Hardware Interfaces

- Android phone (min 2GB RAM, Android 9+)
- Optional: Camera for barcode (future scope)
- Internet access required

4.3 Communications Interfaces

- Firebase or cloud-based backend.
- REST API or Firebase sync.

- Realtime sync not critical.

4.4 Software Interfaces

- **Frontend** : Flutter SDK
- **Backend** : Firebase Firestore or Cloud MySQL
- **IDE** : Android Studio or VS Code
- **Authentication** : Firebase Auth

5. PERFORMANCE REQUIREMENTS

- Login/CRUD actions must respond within 2–3 seconds.
- App must support 50 concurrent users.
- Wishlist and browsing should be smooth under moderate data load.
- Offline caching (optional future feature).

6. OTHER NON-FUNCTIONAL ATTRIBUTES

6.1 Security

- Admin actions password protected.

- User data stored securely.
- No payment, so lower financial risk.

6.2 Reliability

- App should avoid crashes or data loss.
- Firebase provides backup and sync.

6.3 Maintainability

- Modular code structure for easy updates.
- Admin can update entries without developer help.

6.4 Portability

- Runs on Android.
- Flutter-based iOS compatibility.

6.5 Extensibility

- Future features: barcode scanning, crop-wise recommendations, chat with agro experts.

6.6 Reusability

- Code modules like login, wishlist, and CRUD reusable for any catalog-type app.