Software Requirements Specification

for

AI-based Crop Planning and Disease Prediction System

Prepared by

Anshika Agarwal	**23UCS538**	**23ucs538@lnmiit.ac.in**
Anuj Kamal Jain	**23UCS540**	**23ucs540@lnmiit.ac.in**
Maurya Jitendra Patel	**23UCS642**	**23ucs642@lnmiit.ac.in**
Sarthak Gupta	**23UCS697**	**23ucs697@lnmiit.ac.in**

Instructor: Dr. Ashish Kumar Dwivedi

Course: Software Development

Date: 26/09/2025

Contents

1 Introduction

1.1 Document Purpose

Many small and marginal farmers struggle to get reliable advice for growing their crops. They often don't have access to personalized farming plans, and it can be hard to identify or treat crop diseases on their own. On top of that, language barriers and limited literacy make it difficult to understand standard agricultural guidance.

Our software aims to help by providing reliable AI-driven personalized plans and crop disease diagnosis in local languages, with explanations in clear, natural-sounding audio. Farmers can also ask questions and adjust their plans, making it easier for them to take informed decisions and improve their crop yield and quality.

1.2 Product Scope

This platform is designed to make expert agricultural guidance accessible to small and marginal farmers. Its primary objective is to generate personalized farming plans and provide instant disease detection from farmer-uploaded images or videos. Crucially, all guidance, including practical solutions and detailed explanations, is delivered in the farmer's local language via both text and audio. The platform ensures relevance by allowing farmers to ask follow-up questions and adjust their plans. By combining AI-driven insights with easy-to-understand communication, the platform helps farmers make better decisions, improve crop health, and increase overall yield.

1.3 Intended Audience and Document Overview

This document is intended for all stakeholders involved in the development, evaluation, and usage of the AI-Powered Agricultural Assistance Platform. The key readers include:

- **Professor:** To review the requirements, ensure they align with project goals, and evaluate the proposed solution.
- **Development Team:** Software developers, AI specialists, and testers who will design, implement, and maintain the system based on these requirements.
- **Farmers:** Small and marginal farmers who will use the platform to receive personalized farming plans, crop analysis, and audio explanations.
- **Testers:** To understand the expected functionalities and constraints in order to verify and validate the system.

Document Overview:
This document specifies the functional and non-functional requirements of the platform, providing a comprehensive description of its intended capabilities. It includes the system's purpose, scope, objectives, detailed functional requirements, external interfaces, and design constraints. The SRS serves as a reference for developers, testers, and stakeholders to ensure a shared understanding of the system's requirements and expectations.

1.4 Definitions, Acronyms and Abbreviations

Acronyms	Full Form/Definition
AI	Artificial Intelligence
API	Application Programming Interface
DFD	Data Flow Diagram
TTS	Text-to-Speech
UML	Unified Modelling Language
SRS	Software Requirements Specification
JWT	JSON WebToken

1.5 Document Conventions

- Main Heading Titles:
  - Font: Arial
  - Face: Bold Black
  - Size: 18
- Subheading Titles:
  - Font: Arial
  - Face: Bold Black
  - Size: 14
- Main points:
  - Font: Arial
  - Face: Normal Black
  - Size: 12
- Other text explanations:
  - Font: Arial
  - Face: Italic Black
  - Size: 12

1.6 References and Acknowledgments

- ReactJS Documentation: Available at: *https://reactjs.org*
- NodeJS Documentation: Available at: https://nodejs.org/docs/latest/api/
- ExpressJS Documentation: Available at: https://expressjs.com/
- MongoDB Documentation: Available at: https://www.mongodb.com/docs/
- *FastAPI Documentation :* Available at : *https://fastapi.tiangolo.com/*
- Google Cloud Documentation : Available at: *https://ai.google.dev/gemini-api/docs*
- Lecture Slides

2 Overall Description

2.1 Product Overview

It is a completely new, self-contained AI-powered solution that replaces the current manual and fragmented approach farmers face in seeking agricultural advice with an integrated, intelligent platform. It will provide small and marginal farmers with instant, personalized crop planning and disease diagnosis using multimodal AI, eliminating the delays and accessibility issues of traditional expert consultation. The system will be delivered as a web-based application with a smooth, farmer-friendly interface, multi-language text and audio support, and dedicated storage for user profiles and crop history. Unlike conventional methods, which often involve multiple disconnected tools or physical visits to agricultural offices, CropGenesis unifies these essential functions into a single, accessible system designed to reduce errors, increase efficiency, and empower farmers to make timely, informed decisions.

2.2 Product Functionality

- **User Authentication**: Allow farmers to register, log in, and manage profiles securely, authentication is done using the JWT token and customized backend for login and registration.
- **Personalized Crop Planning**: Generate tailored crop plans based on farmer-provided inputs (e.g., soil type, irrigation, crop preferences).
- **Plan Explanation in Local Languages**: Provide crop plan details in both text and audio, supporting multiple languages for accessibility.
- **Plan Modification & Follow-up**: Enable farmers to ask queries or request updates to generated crop plans.
- **Crop Diagnosis**: Analyze images or videos of crops to detect diseases or pests and provide expert recommendations.
- **Audio Explanations for Diagnosis**: Deliver diagnosis results and remedies in human-like audio for ease of understanding.
- **Crop History Management**: Store and retrieve past crop plans and diagnosis results for future reference.

2.3 Design and Implementation Constraints

Hardware Constraints:

- The system must run efficiently on standard smartphones with limited processing power and memory, as farmers may not have access to high-end devices.
- Servers hosting the AI engine must have sufficient GPU/CPU capacity to process image and video inputs in real-time for disease diagnosis.

Software Constraints:

- The AI functionalities for generating crop plans, diagnosing diseases, and text-to-speech conversion rely on the external Gemini API. Any downtime, rate-limiting, or API changes may affect system performance.
- The system must be compatible with Android and iOS mobile platforms, with responsive web support as a supplementary access channel.

Data Constraints:

- Farmer-provided data, such as images, videos, and crop history, must adhere to defined formats (e.g., JPEG/PNG for images, MP4 for videos).
- The system must ensure secure storage of sensitive farmer information, complying with local data protection regulations.

Network Constraints:

- As the system relies on cloud-based AI services, a stable internet connection is required for accessing the Gemini API.
- The system should handle intermittent connectivity gracefully, with offline caching for previously accessed crop plans and audio content.

Language and Accessibility Constraints:

- The system must support multiple local languages for all text and audio outputs.
- Text-to-speech output must be clear, natural-sounding, and understandable by users with low literacy levels.

Usability Constraints:

- The system should have a simple, intuitive interface suitable for farmers who may have limited exposure to technology.
- Interactive guidance must allow users to easily ask follow-up questions or adjust crop plans without complex navigation.

Integration Constraints:

- Integration with the User Database and Crop History Database must ensure data consistency and reliability.
- The system must be designed to accommodate future expansion, such as adding advanced AI models, or additional third-party APIs.

2.4 Assumptions and Dependencies

Assumptions:

- **Farmer Device Availability**: It is assumed that farmers will have access to a smartphone or tablet capable of running the application and capturing images/videos of crops.
- **Internet Access:** A stable internet connection is assumed for the system to interact with cloud-based AI services, especially for generating crop plans, diagnosing diseases, and converting text to speech.
- **Farmer Literacy:** While the system supports multiple local languages and audio guidance, it is assumed that farmers can interact with basic mobile interfaces (tapping buttons, selecting options).
- **Image/Video Quality:** It is assumed that farmers will provide images or videos with sufficient quality for accurate AI-based disease diagnosis.
- **API Availability:** The external Gemini API, used for AI functionalities and text-to-speech services, is assumed to be operational and accessible during system use.

Dependencies:

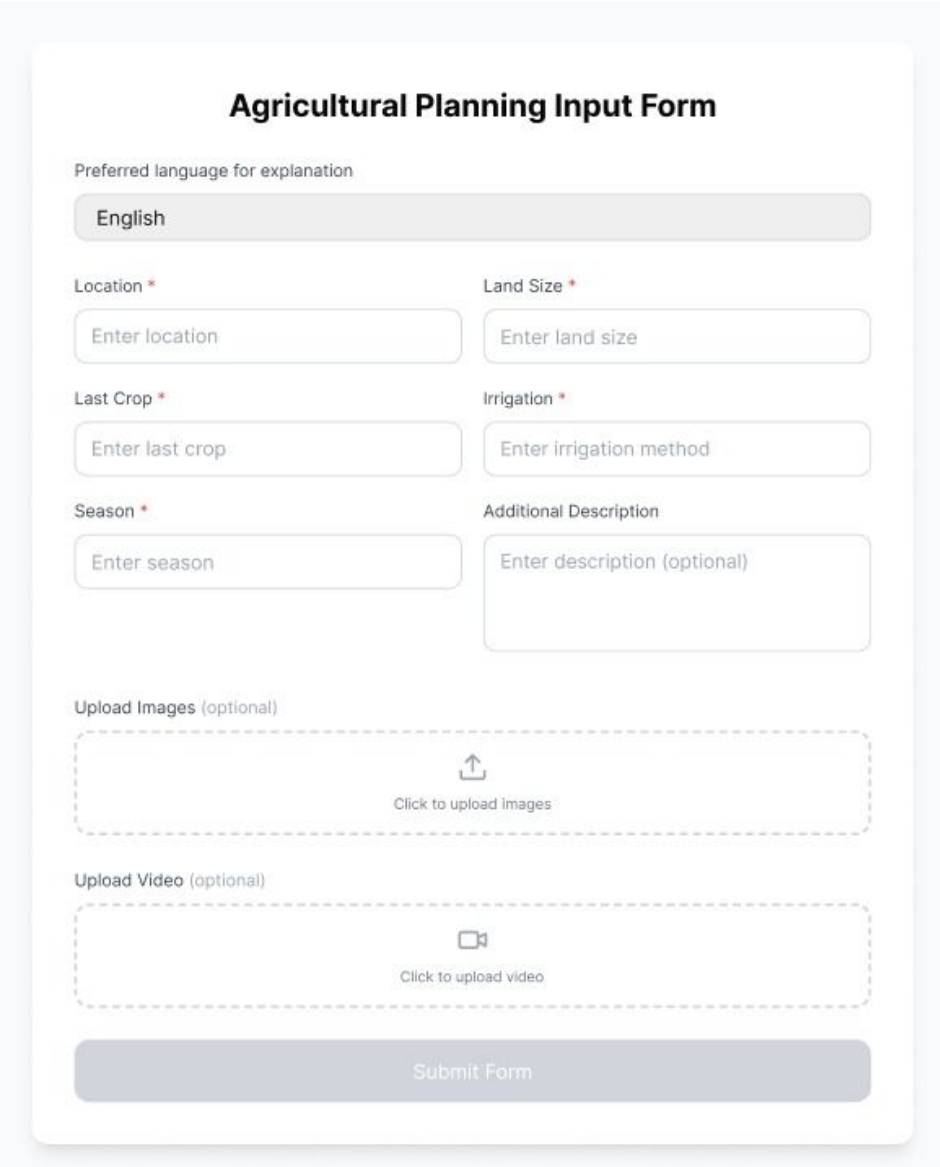
- **External AI API (Gemini API):** The system depends on Gemini API for:
    1. Generating personalized crop plans
    2. Diagnosing crop diseases from images or videos
    3. Converting text guidance into natural-sounding audio
- **Database Systems:** The system relies on a User Database for farmer registration/login and a Crop History Database for storing past plans and diagnoses.
- **Third-party Libraries:** The system may use external libraries or SDKs for image processing, audio output, or multi-language support, which must remain compatible throughout the project lifecycle.
- **Regulatory Compliance:** The system must comply with local data privacy and protection regulations. Any changes in legal requirements could affect how farmer data is collected, stored, or processed.

3 Specific Requirements

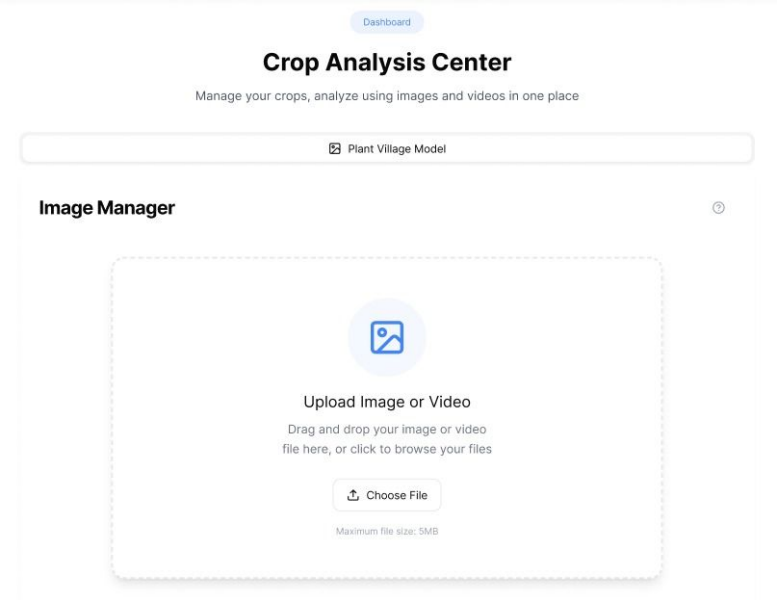
3.1 External Interface Requirements

3.1.1 User Interfaces

Crop Planning Input Form:



Crop Diseases Diagnosis:



3.1.2 Hardware Interfaces

1. **User Devices (Smartphones, Tablets, Laptops, Desktops)**

   - Purpose: To make users access the system utilizing web browsers.
   - Interaction: Users access the system using GUIs that can display dashboards and system features using input devices such as keyboards, mice, and touchscreens to perform operations, like logging in, getting personalized crop plans and diagnosing plants related diseases by uploading images.
   - Data: Processing input data like text, images, and uploaded files and output is personalized crop plan.
   - Control: Interaction through browser-based or mobile application interfaces, using HTTP/HTTPS for communication with the backend.

2. **Server Hardware**

   - **Purpose:** To process requests from users, authenticate users, and deliver customized crop plan and diagnose plant diseases.

- **Interaction:** the software interacts with server hardware to handle tasks such as user authentication, dashboard rendering, data validation, and error handling.
- **Data:** Data includes login credentials, crop input details, crop images, and user profile information.
- **Control:** The server controls actions such as file uploads, validation, dashboard updates, and plan fetching.

3.1.3 Software Interfaces

- **Web Application**: CropGenesis will be accessed via a web interface, allowing farmers to interact with the system through standard web browsers on desktops, laptops, or tablets.
- **Gemini API**: The system communicates with the external Gemini API for AI functionalities, including crop plan generation, disease diagnosis, and text-to-speech conversion.
- **Database Systems**: CropGenesis interfaces with the User Database for authentication and the Crop History Database to store crop plans, diagnoses, and user activity.
- **Data Exchange**: User inputs (crop details, images, or videos) are sent to the backend via HTTP/HTTPS requests, and responses from the AI engine are displayed in text or audio format on the web interface.

3.2 Functional Requirements

The functional requirements of CropGenesis specify the system's intended behavior, capturing the services, tasks, and functions it must perform to support farmers effectively.

3.2.1 User Authentication and Management

- FR-1: Each user (farmer) must be able to register with CropGenesis using basic information such as name, phone number, location, and password.
- FR-2: Each user must be able to log in using their registered credentials.
- FR-3: Each user must be able to log out from the system securely.
- FR-4: Each user must be able to change their password.
- FR-5: Each user must have a personalized dashboard showing saved crop plans, disease diagnoses, and follow-up recommendations.

3.2.2 Crop Plan Generation

- FR-1: Users must be able to enter farm-specific details such as land size, irrigation, soil type, language preferred and season.
- FR-2: The system shall generate a personalized crop plan using AI based on the input parameters.

- FR-3: Users must be able to view and download the crop plan in text format.
- FR-4: Users must be able to listen to the crop plan through text-to-speech audio output in their local language.
- FR-5: Users must be able to ask follow-up questions or request adjustments to the crop plan.

3.2.3 Disease Diagnosis

- FR-1: Users must be able to upload images or videos of crops for disease diagnosis.
- FR-2: The system shall analyze the input using AI and identify potential crop diseases.
- FR-3: The system shall provide suggested remedies or treatment plans for identified diseases.
- FR-4: Users must be able to receive diagnostic results in both text and audio formats.
- FR-5: Users must be able to save the diagnosis in their crop history for future reference.

3.2.4 Crop Plan History Management

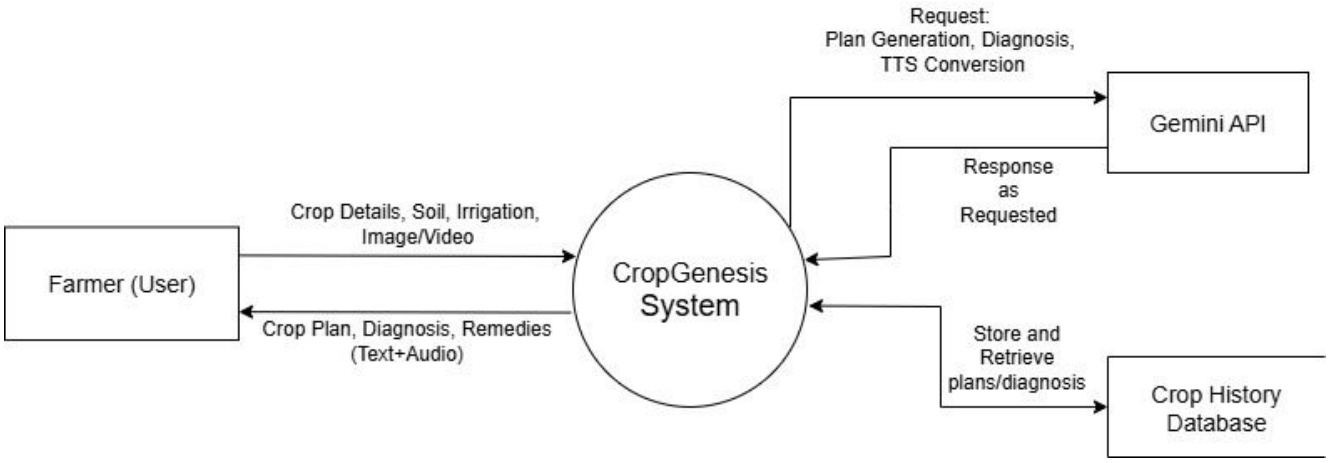
- FR-1: Users must be able to view a chronological list of past crop plans and disease diagnoses.
- FR-2: Users must be able to search, retrieve, and review specific historical records.
- FR-3: Users must be able to delete records from their crop history if desired.

3.2.5 Multi-language Support

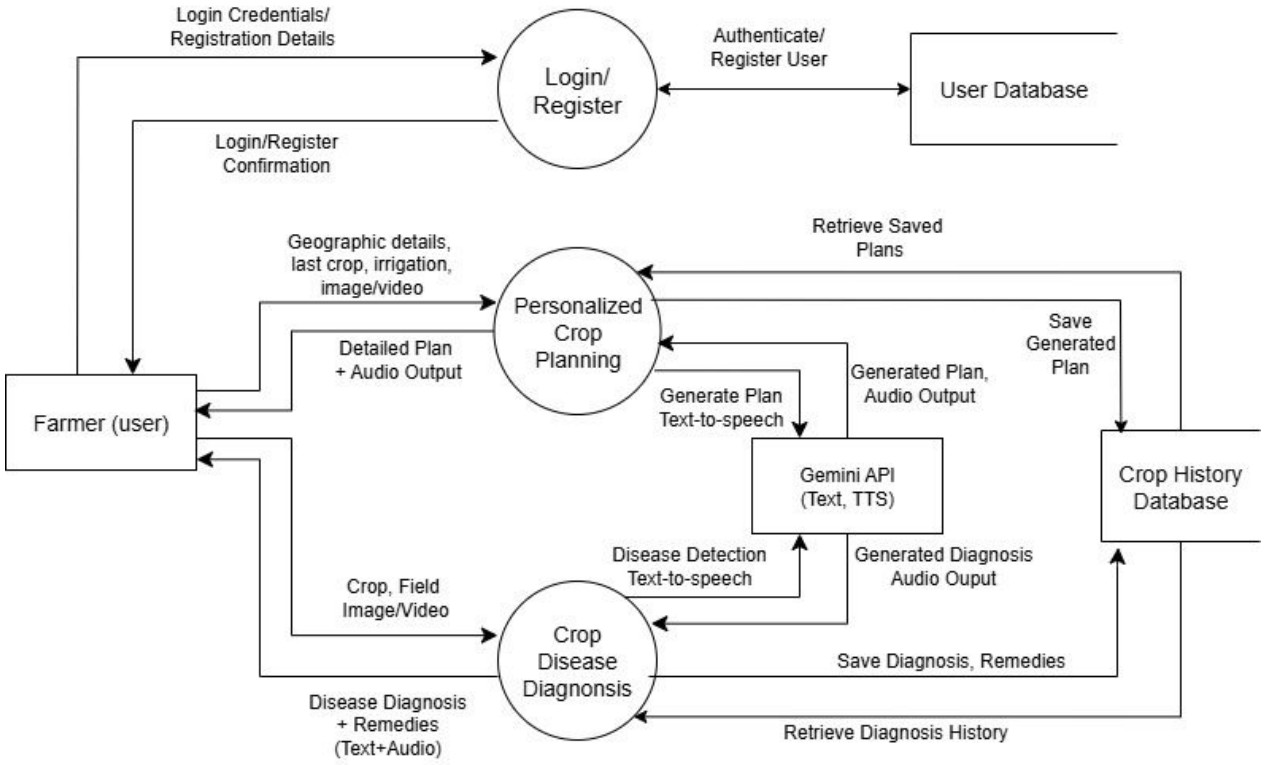
- FR-1: Users must be able to select their preferred local language for all text and audio outputs.
- FR-2: The system shall provide accurate translations for crop plans, diagnostic results, and guidance in the selected language.

3.3 Use Case Model

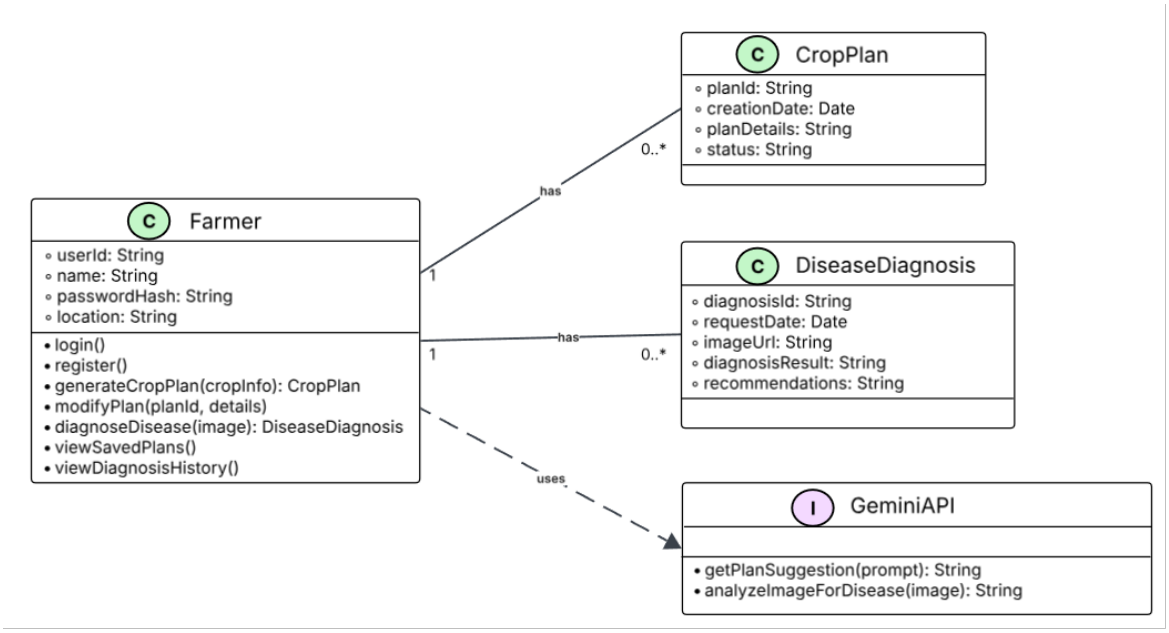
3.3.1 Data Flow Diagram (Level 0): (Full Diagram Link)



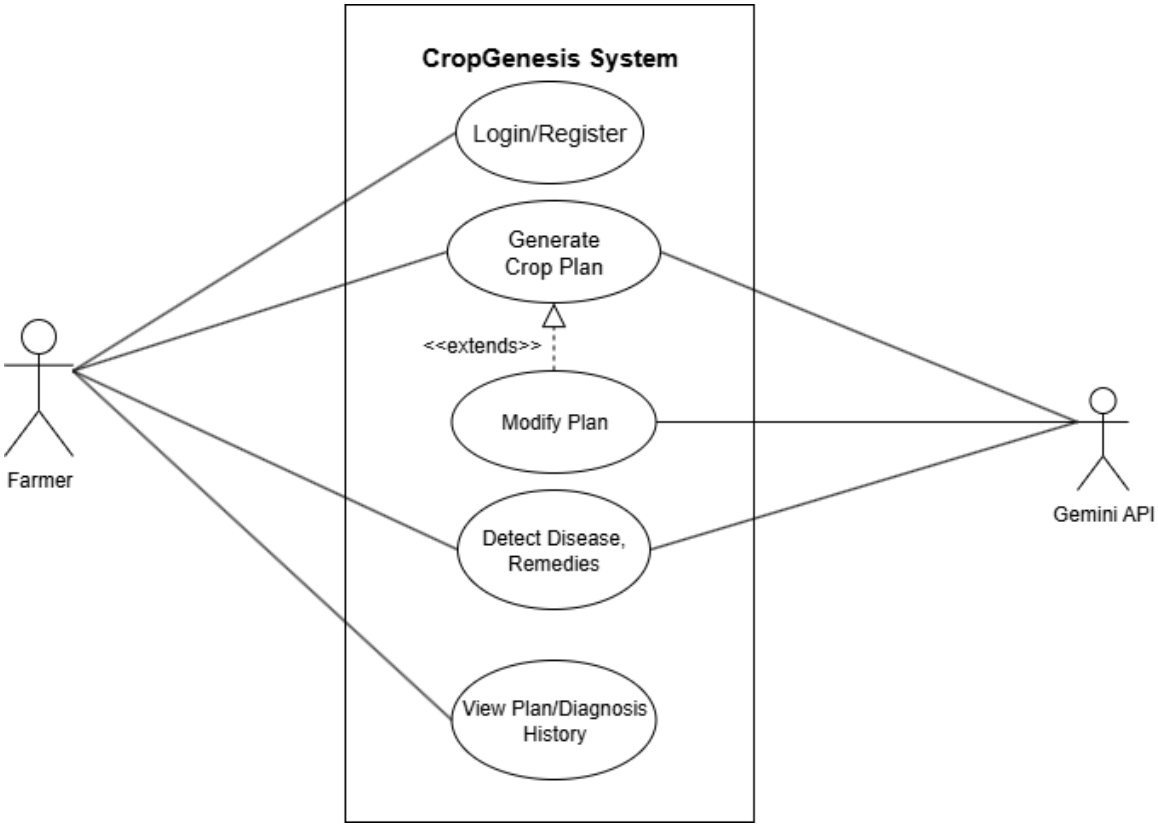
3.3.2 Data Flow Diagram (Level 1): (Full Diagram Link)



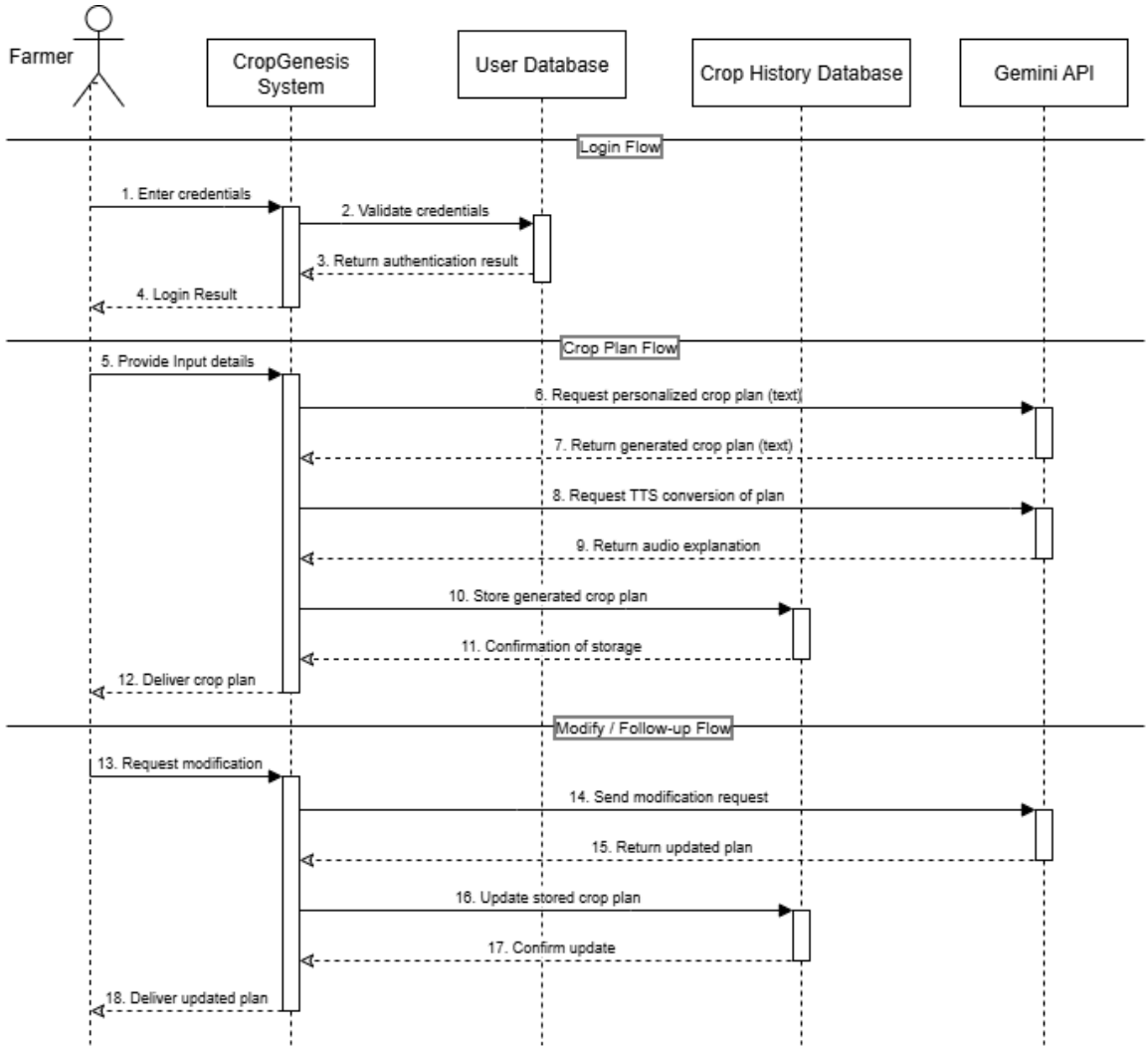
3.3.3 Class Diagram: (Full Diagram Link)



3.3.4 Use Case Diagram: (Full Diagram Link)



3.3.5 Sequence Diagram: (Full Diagram Link)



3.3.6 Decision Table: (Decision Table 1) , (Decision Table 2)

Decision Table 1

Conditions	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
User Action	Login	Login	Login	Register	Register
User Exists in DB?	Yes	Yes	No	No	Yes
Credentials Match?	Yes	No	N/A	N/A	N/A
Actions					
Grant System Access	X				
Show Login Error		X	X		
Create New User in DB				X	
Show Registration Success				X	
Show User Already Exists					X

Decision Table 2

Conditions	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
User Request	Generate/Modify Plan	Diagnose Disease	View Saved Plans	View Diagnosis History	Generate/Diagnose
Valid Inputs Provided?	Yes	Yes	N/A	N/A	No
Actions					
Call Gemini API (Plan/Diagnosis)	X	X			
Call Gemini API for TTS	X	X			
Save/Update Data in Crop History DB	X	X			
Retrieve Data from Crop History DB			X	X	
Display Result to User (Text+Audio)	X	X			
Display History List to User			X	X	
Show Input Error Message					X

4 Other Non-functional Requirements

4.1 Performance Requirements

Performance is critical for user retention, especially in environments with variable network connectivity. The system's responsiveness directly impacts its usability and trustworthiness for farmers.

- **P1 (Disease Diagnosis – Image/Video):** The system shall process a user-uploaded image/video (up to 5MB) and return the AI-generated disease diagnosis and remedy within 20 seconds, assuming a stable 3G network connection.
- **P2 (Crop Plan Generation):** A request for a new personalized crop plan shall be processed and the complete plan (text and audio) delivered to the user's device within 30 seconds.
- **P3 (Audio Generation):** For any text response generated by the system (e.g., diagnosis, plan step, follow-up answer), the corresponding audio file shall be generated and available for streaming/playback within 5 seconds of the text appearing.
- **P4 (Database Access):** User authentication (login) shall be completed within 5-10 seconds. Saved plans and past diagnoses from the Crop History Database shall be retrieved and displayed within 5-10 seconds.
- **P5 (Concurrency):** The system must be able to support multiple concurrent users (e.g., users requesting diagnoses or plans simultaneously) with no more than a 25% degradation in the response times specified in P1-P4.

4.2 Safety and Security Requirements

Given the system handles personal farmer data and provides critical agricultural advice, safety and security are paramount.

- **S1 (Advice Disclaimer):** All AI-generated advice (diagnoses, remedy suggestions, crop plans) must be presented with a clear, non-technical disclaimer in the user's local language. The disclaimer shall state that the guidance is AI-generated, may not be 100% accurate, and should be used as a suggestion, not as a replacement for professional agricultural extension services where available.
- **SEC1 (User Authentication):** Access to personal data (saved plans, crop history) shall be protected. The system must provide secure authentication via a simple mechanism.
- **SEC2 (API Key Security):** The external Gemini API key must **not** be stored within the web application, It shall be stored securely on the backend server. and all API calls shall be proxied through this server to protect the key from being compromised.

- **SEC3 (Privacy):** Individual farmer data (location, crop history, diagnoses) will not be shared with any third party without explicit, opt-in consent from the user.

4.3 Software Quality Attributes

- **SQA1 Reliability:** The system shall be highly available, especially during critical planting and harvesting seasons. It must provide accurate AI-driven advice for crop planning and disease diagnosis. The application should handle poor network connectivity and API errors gracefully by notifying the user, ensuring it remains a dependable tool for farmers.
- **SQA2 Scalability:** The system's architecture must support horizontal scaling. It must be able to handle a growing number of farmers, simultaneous image uploads for diagnosis, and crop plan requests without a drop in performance, ensuring a consistent experience as the user base expands across different regions.
- **SQA3 Maintainability:** The software must be designed in a modular fashion. This will allow for easy updates, such as adding new languages, incorporating information about new crops or diseases, or upgrading to a newer version of the Gemini API with minimal impact on the overall application.
- **SQA4 Usability:** The user interface must be extremely intuitive and accessible to its target audience of small and marginal farmers, who may have varying levels of literacy. The interface will have a clear text-to-audio feature to ensure that all guidance is easily understood and actionable.

Appendix A – Data Dictionary

Item Name	Type	Description	Possible Values / Format	Related Operations/ Requirements
FarmerID	UID	Unique identifier for each registered farmer	Auto-generated alphanumeric ID	Used in registration, login, crop plan retrieval, and history tracking
FarmerName	String	Full name of the farmer	Text	Captured during registration; displayed in UI and reports
FarmerLanguage	String	Preferred language for communication	English, Hindi, Local Language	Used for text and audio guidance output
CropPlanDetails	Text / JSON	Detailed AI-generated plan for the crop	JSON with tasks, timelines, and instructions	Displayed to the farmer and used for follow-up queries
DiagnosisResult	String / JSON	Result of disease detection	Disease name, severity, and recommended remedy	Displayed to farmer with text and audio output

Appendix B - Group Log

1. **25/08/2025**: Discussed the overall vision of the software and explored several websites to search which database service and API key would be best for this purpose.

2. **1/09/2025**: Started writing in the SRS Document according to the guidelines given in the document and the instructions given by the professor. Completed writing the SRS till section 1.6

3. **5/09/2025**: Completed the SRS Document till section 2 (Product Overview, Product Functionality, Design and Implementation Constraints and Assumptions & Dependencies).

4. **13/09/2025**: Started specifying the details in section 3 of this document. Also, completed building the use case model of the application. Other diagrams were pending to be completed.

5. **17/09/2025**: Finished making the pending diagrams: DFD, Class Diagram, Sequence diagram and decision tables.

6. **26/09/202**5: Completed the remaining parts of SRS.