Hamdard University

Department of Computing

Final Year Project

Project Title

Dr. Crop Android Application

Project Code

FYP-002/SP25

Software Design Specifications

Submitted by

Name	Roll Number	Project Role
Muhammad Ibrahim	(1895-2021)	(Team Lead)
Zeeshan Ali	(1505-2021)	(Team Member 2)
Wali Muhammad	(1894-2021)	(Team Member 3)

Supervisor

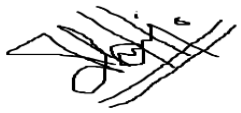
Teacher: Sir Saifullah Adnan

Fall 2024

Document Sign off Sheet

Document Information

Project Title	Dr.Crop Android Application
Project Code	FYP-002/SP25
Document Name	Software Detailed Design (SDS)
Document Version	<1.0>
Document Identifier	FYP-002/SP25-SDS
Document Status	Final
Author(s)	M.Ibrahim , Zeeshan Ali , Wali Muhammad
Approver(s)	Saifullah Adnan
Issue Date	

Name	Role	Signature	Date
Muhammad Ibrahim	Team Lead		02-07-2025
Zeeshan Ali	Team Member 2		02-07-2025
Wali Muhammad	Team Member 3		02-07-2025
Saifullah Adnan	Supervisor		
	Co-Supervisor		
	Project Coordinator		

Revision History

Date	Version	Description	Author
04-07-2025	1.0	Initial version of the document created, including system functions, non-functional requirements, use cases, and other essential sections.	Muhammad Ibrahim Zeesham ali Wali Muhammad

Definition of Terms, Acronyms, and Abbreviations

In the **Definition of Terms, Acronyms, and Abbreviations** section, you'll list and define all relevant terms, acronyms, and abbreviations used throughout the document. This ensures clarity and helps the reader understand the specialized language or abbreviations. Here's a general template for this section.

Term	Description
Actor	An entity that interacts with the system, e.g., user, external systems.
CNN	Convolutional Neural Network, a type of deep learning algorithm used for image recognition.
API	Application Programming Interface, a set of routines and tools for building software applications.
GUI	Graphical User Interface, a system that allows users to interact with software through graphical elements.
Camera Input	Input method where visual data is captured using a camera for processing.

Table of Contents

1. Introduction

1.1 Purpose of Document

The purpose of this Software Design Specification (SDS) document is to describe the detailed design of the Wheat Disease Detection prototype. It extends the SRS by defining architectural and technical design considerations.

1.2 Intended Audience

This document is intended for software developers, technical leads, UI designers, and future collaborators extending the prototype to other crops such as cotton.

1.4 Project Overview

The Wheat Disease Detection system uses a Convolutional Neural Network (CNN) model, trained and optimized using TensorFlow Lite, to classify wheat leaf images into specific disease categories.

1.5 Scope

This SDS covers the current prototype (wheat only) and lays the design foundation for future integration of additional crop models (e.g., cotton). It includes system architecture, interface design, and application layer descriptions.

2. Design Considerations

2.1 Assumptions & Dependencies

• User provides clear, single-leaf images
• Device has TensorFlow Lite compatibility
• Model accuracy depends on dataset quality

2.2 Risk and Volatile Areas

• Inconsistent image quality from users
• Limited scalability if more crops are added without refactoring
• Mobile device resource limitations

2.3 Scalability and Performance

The system is designed to be lightweight and support quick inference. Future scalability includes:
• Modular loading of crop models
• Efficient storage of datasets and results

3. System Architecture

3.1 System Level Architecture

3.1.1 System Decomposition into Elements

The system is composed of:

- User Interface (mobile app)
- Image Processing and Inference Module
- Pre-trained CNN Model (TFLite)
- Local result display module

3.1.2 The Relationship between the Elements

The mobile app captures input, sends it to the processing module, which communicates with the TFLite model and displays the prediction.

3.1.2.1 Interfaces to External Systems

No external interfaces are required. The system runs entirely offline on a mobile device.

3.1.3 Major Physical Design Issues

- Device storage limitations
- Processing speed of older mobile devices
- Model size and compatibility

3.2 Software Architecture

3.2.1 User Interface Layer

Displays image input option, prediction results, and help content. Designed using native Android components or Flutter

3.2.2 Middle Tier

Handles image preprocessing, model inference, and result formatting.

3.2.3 Data Access Layer

No permanent database is used. Images and results are processed in-memory only.

4. Design Strategy

4.1 Future System Extension or Enhancement

The system is divided into loosely coupled modules for image input, model processing, and output display. This modular approach supports easier debugging, future updates, and adding new crops (like cotton) in later versions.

4.2 System Reuse

The design ensures scalability by:

- Allowing model replacement for different crops.
- Maintaining lightweight inference using TensorFlow Lite.
- Ensuring fast predictions even on low-spec devices.

4.3 User Interface Paradigms

Since the application operates offline, the security risks are minimal. However, app permissions are strictly limited to camera access. No data is transmitted or stored externally.

4.4 Data Management (Storage, Distribution, Persistence)

UX is designed for non-technical users (farmers):

- Clean, minimal UI with icons and native language labels.
- Single-tap prediction.

4.5 Concurrency and Synchronization

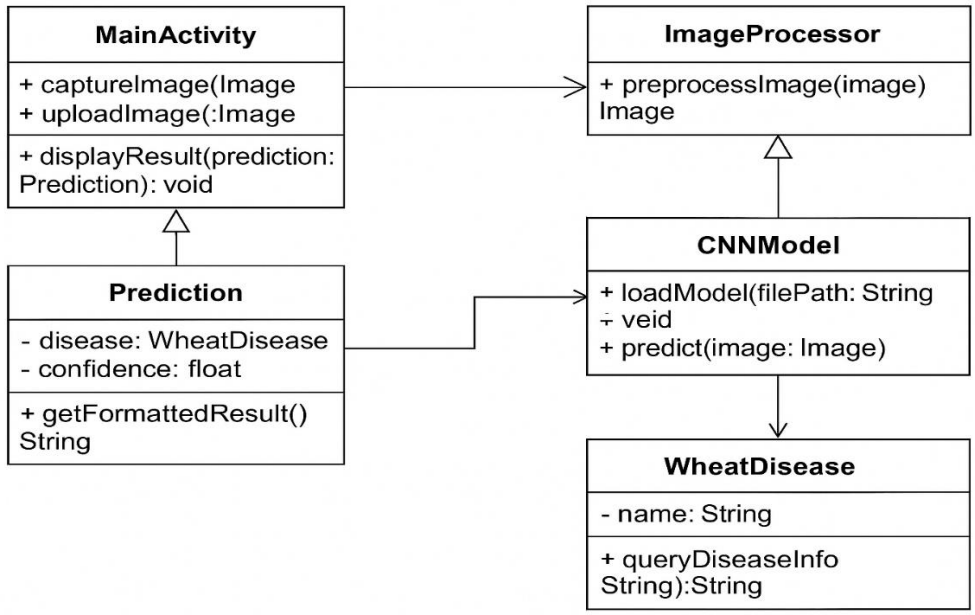
All major components such as model, image processing pipeline, and UI are separated into their own modules, making maintenance and future enhancements straightforward.

4.6 Trade-offs

The app is designed to function reliably offline. Model loads on app start and remains in memory. In the case of failure, fallback options such as image re-selection and guidance are provided.

4. Detail System Design

5.1 Design Class Diagram

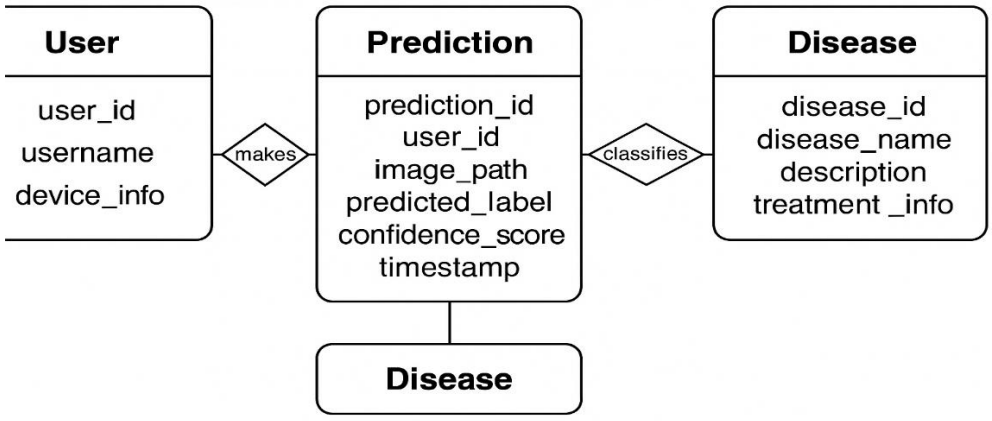


5.2 Database Design

[A detailed Database design should include the following:

- *Logical data model (E/R model)*
- *Data dictionary]*

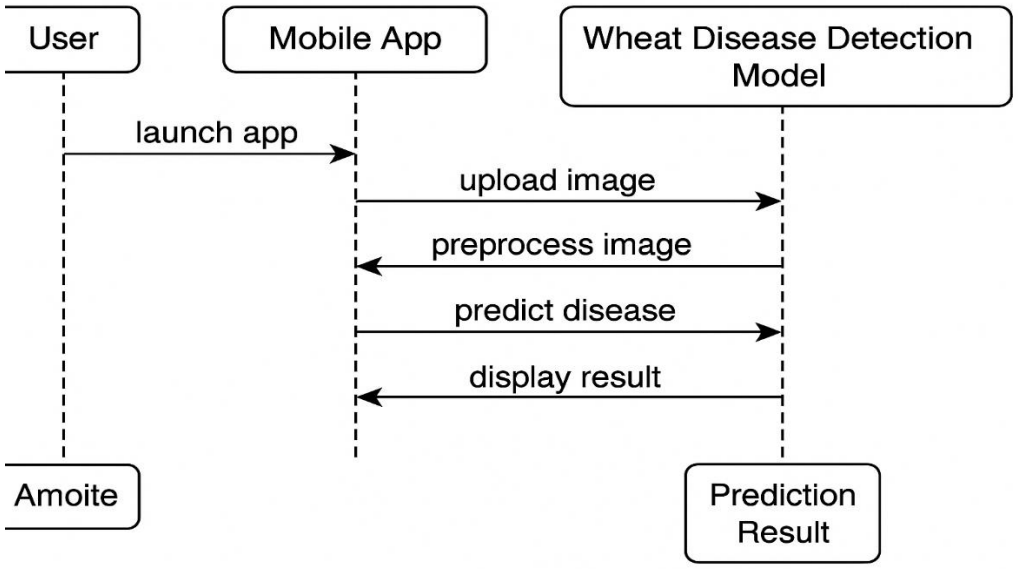
5.3 *ER Diagram*



5.4 Application Design

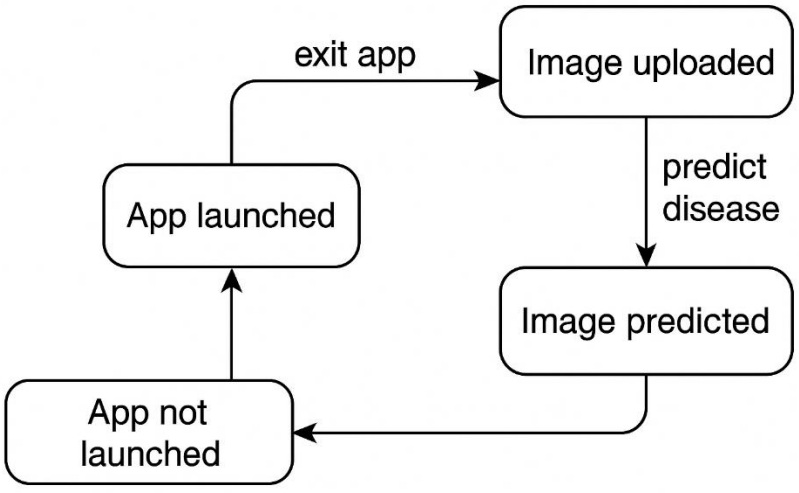
5.4.1 Sequence Diagram.

5.4.1.1 <Sequence Diagram 1>

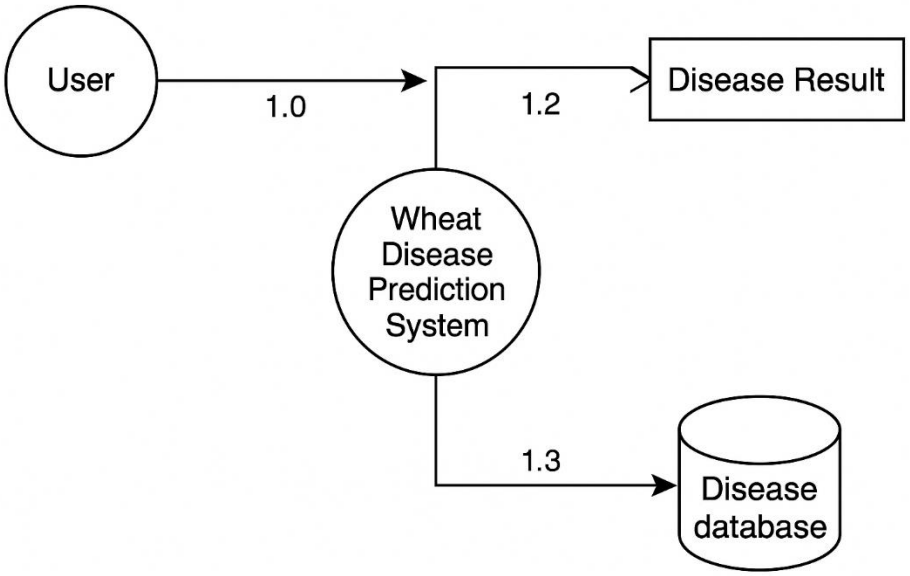


5.4.2 State Diagram

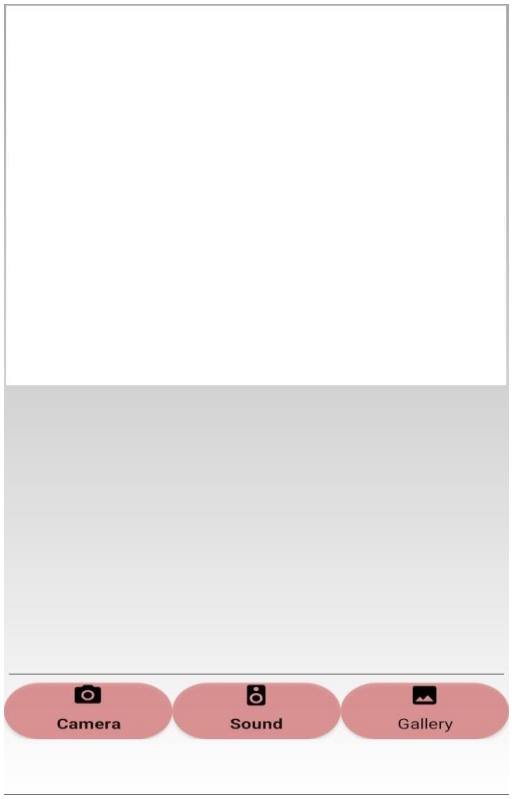
5.4.2.1 <State Diagram 1>



5.4.3 DFD Level 1



5.5 GUI Design



6. References

• TensorFlow Lite documentation: https://www.tensorflow.org/lite
• Wheat Disease datasets (custom/collected from Kaggle)
• Research papers on CNN for plant disease detection
• Android Developer Documentation: https://developer.android.com
• Agricultural datasets from open repositories

7. Appendices

Appendix A: Terminologies
• Actor: An entity that interacts with the system, e.g., user, external systems.
• CNN: Convolutional Neural Network, a type of deep learning algorithm used for image recognition.
• API: Application Programming Interface, a set of routines and tools for building software applications.
• GUI: Graphical User Interface, a system that allows users to interact with software through graphical elements.
• Camera Input: Input method where visual data is captured using a camera for processing.