|  |
| --- |
| **Software Design Specification** |
|  |

AgroVision

Farmer’s Companion Application

Authors :-

Balendhu S Ajay, Kevin Tom Varghese, Philip John, Pranav Krishna P

Reviewers :-

Dr. Aneesh G Nath, Prof. Shyna A

**Contents**

1. Introduction
   1. Document Purpose
   2. Product Scope
   3. Intended Audience and Document Overview
   4. Document Conventions
   5. References and Acknowledgements
2. System Architecture Design
3. Technology Stack
4. Software Development Life Cycle
5. Mock User Interface

**1.Introduction**

The crop disease detection mobile app is designed to help farmers identify and manage crop diseases quickly and accurately. The app uses machine learning algorithms to analyze crop images and diagnose diseases, providing farmers with treatment recommendations to prevent further crop damage. The app is intended for use by farmers and agricultural experts who want to manage crop diseases efficiently.

1.1 Document Purpose

The purpose of the Software Design Document is to provide a clear, documented model of the design of the crop disease detection system, that will provide following Designs of the Project.

1.System Architecture

2.Application Architecture

3.Mock UI

1.2 Product Scope

The crop disease detection mobile app is a machine learning-based solution designed to help farmers and agricultural experts diagnose and manage crop diseases quickly and accurately. The app uses algorithms and machine learning models to analyze crop images and identify signs of disease, and provides treatment recommendations based on sound agricultural principles and best practices.

The crop disease detection mobile app will have the following user roles:

* Farmers: Farmers will be the primary users of the app, using it to diagnose and manage crop diseases.
* Agricultural experts: Agricultural experts, such as agronomists and extension agents, may also use the app to assist farmers in diagnosing and managing crop diseases.
* App administrators: App administrators will be responsible for managing user accounts, app updates, and other administrative tasks.

The user interface of the crop disease detection mobile app will be designed to be intuitive and easy to use. It will include features such as:

* Navigation menu: A menu that allows users to access different features of the app.
* Image upload: A feature that allows users to upload images of their crops.
* Disease diagnosis: A feature that displays the diagnosis of the crop disease and recommended treatments.
* Chatbot: A feature that provide assistance to users regarding crop-related queries.
* Management Tools: A functionality that provides several features such as support for farmers when it comes to irrigation related info, fertilizers & pesticides supply, and displaying weather patterns.
* Locate: A feature that displays and guides users to nearby offering essential farming supplies.

1.3 Intended Audience and Document Overview

Readers include the panel of professors, and the team members consist of developers and documentation writers. The document consists of description of all the various designs used in the projects. It has all the features that are to be implemented in the product.

1.4 Document Conventions

Use Segoe UI font for the entire report: –

Chapter/Sections Title – Segoe UI. 18, Bold.

Heading 2 – Segoe UI, 16, Normal, Underlined.

Heading 3 – Segoe UI, 14, Normal, Underlined.

Body – Segoe UI, 11, Normal.

* Line Spacing – Between Heading, 1 to 2 lines; between lines in paragraph, 1.25 lines.
* Alignments – Chapter/Sections Title, Heading 2 & 3 – Left Aligned.

Ensure that all body text is paragraph justified.

* Figures and Tables – Ensures that all Figures and Tables are suitably numbered and given

proper names/headings. Write figure title under the figure and title above the table.

1.5 References and Acknowledgements

1. <https://www.researchgate.net/publication/314436486_An_Overview_of_the_Research_on_Plant_Leaves_Disease_detection_using_Image_Processing_Techniques>

**An Overview of the Research on Plant Leaves Disease detection using Image Processing Techniques by Ms. Kiran R. Gavhale, Prof. Ujwalla Gawande**

        Methods and techniques for disease Detection

1. <https://www.annualreviews.org/doi/10.1146/annurev.phyto.43.113004.133839>

**Plant Disease: A Threat to Global Food Security by** **Richard N. Strange** **and Peter R. Scott**

1. <https://www.jandonline.org/article/S0002-8223(09)01821-5/fulltext>

**Pandemics Aren't Just for People: How Disease Can Affect Crops and the Food Supply by Brian Boyce**

**2.** **System Architecture Design**

A clear understanding and a wise choice of software architecture is very important for developing this application and is mandatory when a handful of people are involved in the mobile app development process. There are several suitable architectures for this project, from which we chose Three-Tier Architecture.



* Presentation Layer

The presentation layer consists of the user interface that the user interacts with. The user interface would be designed to be intuitive and user-friendly, allowing the user to easily navigate the app and access its features.

* Business Layer

The business layer for a leaf disease detection app would handle the business logic and processing of data related to image analysis, chatbot queries and responses, and crop management tools.

* Data Layer

The data layer would store the chatbot queries and responses, which could include information about the user's location, the type of crop they are growing, and the specific questions they are asking. The data layer would work in conjunction with the application, client, and business tiers to ensure that the data is processed and analyzed efficiently and accurately. The data layer would also be responsible for ensuring that the data is secure and that it meets the needs of the users.

The Three-tier architecture consists of;

1. Client

The client tier architecture would consist of a mobile application developed using a framework such as Flutter. The app would incorporate a chatbot that can answer crop-related queries from the user.

The app would also need to incorporate maps to locate nearby fertilizer shops and crop management tools to help farmers manage their crops more efficiently.

The client tier would be responsible for implementing the user interface for the chatbot, maps, and crop management tools.

The app would also need to incorporate a camera feature to allow users to take pictures of plant leaves for disease detection. The client tier would send the images to the server for processing and analysis. The server then uses artificial intelligence and computer vision techniques to classify the images and detect the presence of leaf diseases.

Several studies have proposed the use of deep learning models and convolutional neural networks (CNN) for the detection of leaf diseases in plants. These models are trained on large datasets of images of diseased and healthy leaves. The proposed models have achieved high accuracy rates, ranging from 99.35% to 99.9%.

The server tier would handle the business logic and processing of data related to image analysis, chatbot queries and responses, maps, and crop management tools.

The app has the potential to improve the efficiency of leaf disease detection and improve crop yields for farmers and agriculturalists.

1. Application

The application tier architecture would handle the business logic and processing of data related to image analysis, chatbot queries and responses, maps, and crop management tools.

The app would use artificial intelligence and computer vision techniques to classify images of plant leaves and identify the presence of diseases.

The app would also incorporate a chatbot that can answer crop-related queries from the user. The chatbot could use natural language processing techniques to understand the user's queries and provide relevant responses.

The app would also need to incorporate maps to locate nearby fertilizer shops and crop management tools, which could include features such as crop monitoring, irrigation management, and pest control; in order to help farmers manage their crops more efficiently.

1. Database

The database tier architecture would store the knowledge base or database of information used to feed the chatbot with the information required to give a suitable response to the user.

The database would need to be able to handle large amounts of data related to crop-related queries and their corresponding responses. The database could be designed to store information related to crop types, weather patterns, soil types, and other relevant information that the chatbot could use to provide relevant responses to the user.

The database tier would work in conjunction with the application and client tiers to ensure that the data is processed and analyzed efficiently and accurately.

The chatbot would use natural language processing techniques to understand the user's queries and provide relevant responses. The application tier would be responsible for implementing the natural language processing techniques used for the chatbot.

**3. Technology Stack**

* Flutter

Flutter is a useful technology enabling the development of this application, that can be used on both Android and iOS devices. The mobile application can incorporate a camera feature to allow users to take pictures of plant leaves for disease detection.

Flutter provides a fast and efficient way to develop mobile applications, which is important for a project that requires real-time image analysis and chatbot responses. Additionally, Flutter provides a rich set of widgets and tools that can be used to create a user-friendly and intuitive interface for the application.

Flutter also allows for easy integration with other technologies, such as Python and AWS, which can be used for the server and data layers of the project. Overall, Flutter is a useful technology for this project because it allows for the development of a mobile application that is fast, efficient, and user-friendly, and can be easily integrated with other technologies.

* Python

Python is needed for this project because it is a powerful programming language that can be used for machine learning and image processing tasks. Python has a wide range of libraries and frameworks that can be used for image analysis, such as OpenCV and TensorFlow.

These libraries can be used to train machine learning models to classify images of plant leaves and identify the presence of diseases. Python can also be used to develop the server-side of the application, which can be hosted on AWS. The server can use Flask, a Python web framework, to provide an API for the mobile application to send images for processing and analysis. The server can also use Python to implement the chatbot and crop management tools.

Overall, Python is a useful technology for this project because it allows for the development of machine learning models for image analysis and provides a powerful and flexible platform for developing the server-side of the application.

* TensorFlow

TensorFlow is a useful technology for this project because it is a powerful open-source library for machine learning and deep learning tasks. TensorFlow provides a wide range of tools and frameworks that can be used for image analysis, such as convolutional neural networks (CNNs) and transfer learning.

These tools can be used to train machine learning models to classify images of plant leaves and identify the presence of diseases. TensorFlow also provides a simple and intuitive interface for defining and training machine learning models, which can help to reduce the time and effort required to develop the models. Additionally, TensorFlow provides built-in support for distributed computing, which can improve the performance of the machine learning models by allowing them to be trained on multiple machines simultaneously.

Overall, TensorFlow is a useful technology for this project because it allows for the development of machine learning models for image analysis, and provides a powerful and flexible platform for training and deploying these models.

* Figma

Figma was used to create the design for this project. Figma is a powerful design tool that allows for the creation of high-fidelity designs for web and mobile applications. Figma provides a wide range of tools and features that can be used to create wireframes, prototypes, and visual designs for the application.

The design process began with the creation of wireframes, which were used to define the layout and structure of the application. Once the wireframes were finalized, the visual design was created using Figma's design tools and components. The design was then refined and iterated upon based on feedback from stakeholders and user testing. Figma's collaboration features were also used to share the design with the development team and ensure that the design was implemented accurately.

Overall, Figma was a useful tool for this project because it allowed for the creation of a high-quality design that was user-friendly and visually appealing.

1. **Software Development Life Cycle**

For this project, we follow the Agile model, as it can be used to develop a prototype application that can be tested and refined through user feedback. The development process can involve the following steps:

* Planning: The team identifies the requirements and goals of the application, and creates a backlog of features and tasks to be completed.
* Design: The team designs the user interface and the architecture of the application, and creates wireframes and mockups to visualize the application.
* Development: The team develops the application in iterations, where each iteration involves the implementation of a subset of features and tasks from the backlog. The team tests the application after each iteration to ensure that it meets the requirements and goals.
* Testing: The team tests the application to identify and fix bugs and issues, and to ensure that it is user-friendly and meets the requirements and goals.
* Deployment: The team deploys the application to a production environment, where it can be accessed by users.
* Maintenance: The team maintains the application by fixing bugs and issues, and by adding new features and functionalities based on user feedback.

In the case of a leaf disease detection application, the Agile model can be used to develop a prototype application that uses image analysis algorithms to detect leaf diseases in plants. The application can be tested and refined through user feedback, and can be deployed to farmers and agricultural institutions to improve the detection and prevention of plant diseases.

1. **Mock User Interface**

Graphical user interface, application

Description automatically generated  A picture containing plant

Description automatically generated

1.Splash Screen 2.Home Page 3.Camera

A screenshot of a phone

Description automatically generated with low confidence A picture containing text

Description automatically generated Graphical user interface

Description automatically generated

4.Management Tools 5.Gallery 6.Location Graphical user interface, text, application, chat or text message

Description automatically generated 7. Chatbot