INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

UNIVERSITY OF WESTMINSTER (UOW)

BEng (Hons) in Software Engineering

EcoGrow : Smart Farming App for Climate Informed Crop Selection Using ChatBot

A Product Specification, Design and Prototype by

A.E.W Jayatilake

Supervised by

Mrs. Malithi Mithsara

Submitted in partial fulfillment of the requirements for the BEng (Hons) in Software Engineering degree at the University of Westminster

September 2024

Section 01

SUMMARY OF THESIS

My thesis focuses on developing a farmer-centric application aimed at providing crucial weather information and crop recommendations to optimize agricultural practices. By incorporating real-time weather data, soil fertility metrics, and location-specific tracking, the app aims to empower farmers with valuable insights for decision-making. The integration of a chatbot further enhances user interaction, offering solutions to agricultural queries. The system relies on a comprehensive dataset for weather conditions and crop information.

The agricultural sector faces numerous challenges, including unpredictable weather conditions and the need for precise crop planning. Recognizing these issues, my project aims to create a farmer-centric app addressing the lack of easily accessible, real-time, and location-specific agricultural information. By offering comprehensive weather insights and crop recommendations, solution seeks to empower farmers and enhance the efficiency of their decision-making processes.

To tackle the identified problems, methodology revolves around the development of a user-friendly mobile application. I leverage a dataset encompassing various weather parameters, including temperature, rainfall, humidity, and soil fertility. The app dynamically displays current weather conditions upon user login and allows users to track specific locations for detailed forecasts. Crop recommendations are categorized as short-term, mid-term, and long-term based on the gathered data. The incorporation of a chatbot enriches user experience, providing instant solutions to agricultural queries.

The prototype of farmer-centric app demonstrates promising results. Users can seamlessly access real-time weather conditions, empowering them to make informed decisions about crop selection and cultivation practices. The location-specific tracking feature ensures personalized recommendations, enhancing the adaptability of the system. The chatbot has proven effective in providing quick and relevant solutions to users' agricultural queries, contributing to a comprehensive and user-friendly platform.

KEY WORDS.

- Weather Conditions: The state of the atmosphere, including temperature, humidity, precipitation, and wind.
- Crop Recommendations: Suggestions for suitable crops based on climate, soil, and environmental conditions.
- Mobile Application: Software designed for mobile devices, enabling access to specific functionalities or information.
- Location-specific Tracking: Monitoring and recording the geographical position of an object or user in a specific area.
- Rich Picture Diagram: A visual tool depicting complex systems or processes, often used in the early stages of requirements analysis to provide a holistic view.
- Stakeholder Analysis: The process of identifying, assessing, and understanding the interests, needs, and influence of various stakeholders in a project.
- Onion Model: A graphical representation that illustrates the layers of stakeholders in a project, from the core team to external stakeholders.
- Viewpoints: Different perspectives or outlooks of stakeholders, capturing their unique interests, concerns, and expectations.
- Requirement Elicitation Methodologies: Techniques used to gather and collect requirements from stakeholders during the early stages of a project.
- MoSCoW Principle: A prioritization technique categorizing requirements into Must-Have (M), Should-Have (S), Could-Have (C), and Will not have (W) to guide development.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 01: INTRODUCTION

Problem:

The problem this app addresses is providing comprehensive and real-time weather-related information to farmers and assisting them in making informed decisions about crop selection based on weather conditions. The key issues that the app aims to solve include.

- Lack of Information: Farmers often face challenges due to the lack of accessible and accurate weather information for their specific location.
- Crop Selection: Farmers may struggle to choose the right crops for cultivation based on current and forecasted weather conditions.
- Decision Support: Farmers need support in making decisions related to farming practices, such as when to plant, harvest, or irrigate, based on weather conditions.
- Query Handling: Farmers may have various questions and problems related to farming that require prompt and accurate solutions.

Methodologies:

The methodologies involved in addressing the identified problems include.

- Weather Data Integration:

Collecting real-time weather data from reliable sources or utilizing APIs to integrate weather information into the app. Incorporating temperature, rainfall, humidity, and soil fertility data for accurate representation.

- Data Classification:

Employing machine learning algorithms to classify and categorize the collected weather data. Classifying crops into short term, mid-term, and long-term categories based on their growth duration and weather requirements.

- Geolocation Tracking:

Implementing geolocation tracking to provide location-specific weather information for the farmer's current position. Offering the ability to select different locations to explore weather conditions for planning future activities.

- Crop Recommendation System:

Developing a recommendation system that suggests suitable crops based on the current and forecasted weather conditions. Considering factors such as temperature, rainfall, humidity, and soil fertility to make precise recommendations.

- Chatbot Integration:

Implementing a chatbot using natural language processing (NLP) to handle farmer queries and provide solutions. Training the chatbot on a diverse set of farming-related questions to enhance its ability to assist farmers effectively.

- User Authentication and Personalization:

Implementing a secure user authentication system to ensure that farmers' data is protected. Personalizing the user experience by allowing farmers to save preferences and favorite locations.

- Data Maintenance and Updates:

Regularly updating the app with the latest weather data and crop information to ensure accuracy and relevance. Implementing a robust data maintenance system to handle dataset updates and improvements.

By combining these methodologies, the app aims to provide a holistic solution to farmers, addressing their immediate weather-related concerns and offering long-term assistance in optimizing farming practices.

1.1 Chapter Overview

I delve into the heart of agricultural assistance app designed to empower farmers. The initial segment concisely outlines the project's raison d'être, framing it as a solution to alleviate the challenges faced by farmers. With a farmer-centric approach, the chapter navigates through the login process, unlocking a dashboard that presents a comprehensive snapshot of today's crucial weather conditions, including temperature, rainfall, humidity, and soil fertility. Following this introduction, embark on a geographical journey, enabling farmers to track and explore location-specific weather data and, crucially, receive tailored recommendations for crops categorized into short term, mid-term, and long term growth. To enhance user interaction and problem-solving, the chapter concludes by introducing a chatbot poised to address farmers' queries and concerns. Rooted in a dataset-driven approach, this chapter sets the stage for an in-depth exploration of each component, establishing a robust foundation for understanding the app's transformative role in modern agriculture.

1.2 Problem Domain/ Background

In the backdrop of contemporary agriculture, farmers grapple with a myriad of challenges that innovative app seeks to redress. With the evolving climate and dynamic weather patterns, farmers often face a critical lack of accessible and localized information. The traditional decision-making process for crop selection is fraught with uncertainty, compounded by the intricate interplay of temperature, rainfall, humidity, and soil fertility. Recognizing these challenges, project emerges as a beacon of support for farmers, providing a user-friendly platform where, upon logging in, they are presented with a comprehensive overview of today's weather conditions. Harnessing the power of a meticulously curated dataset, the app extends its utility beyond mere weather reporting. By integrating geolocation tracking, it offers farmers insights into location-specific weather conditions, guiding them toward optimal crop choices categorized as short term, mid-term, and long term growth. This innovative approach not only addresses immediate concerns but also paves the way for sustainable and informed agricultural practices. In the broader context of the agricultural landscape, project stands as a transformative solution, bridging the gap between traditional farming methods and modern technological advancements.

1.3 Problem Definition:

The agricultural sector is confronted with multifaceted challenges, and a critical concern is the inadequacy of accessible and localized information for farmers. Traditional farming practices often rely on historical data and lack real-time insights into dynamic weather conditions. As a consequence, farmers face the risk of suboptimal crop choices and potential yield losses. The evolving climate and unpredictable weather patterns exacerbate this challenge, necessitating an innovative solution that seamlessly integrates weather data with actionable insights. The problem at the heart of this project is the absence of a comprehensive agricultural support system that addresses the dynamic nature of climate and its direct impact on crop cultivation.

1.3.1 Problem Statement:

The crux of the issue lies in the absence of a unified platform that empowers farmers with timely and relevant information to make informed decisions about crop cultivation. Current agricultural practices fall short in providing a holistic solution that not only presents real-time weather conditions, including temperature, rainfall, humidity, and soil fertility, but also translates this data into actionable guidance for farmers. The decision-making process for crop selection is hindered by the lack of technology-driven tools that can adapt to unpredictable weather patterns. The consequence is a significant gap between the available resources and the modern technological advancements that can optimize farming practices. This project addresses this problem by aiming to develop an integrated application that serves as a reliable companion for farmers, offering not only immediate weather updates but also sophisticated crop classification based on prevailing conditions. By doing so, the project seeks to transform the traditional farming landscape into a more resilient and sustainable model, ensuring that farmers can make informed decisions for long-term agricultural success.

1.4 Aims and Objectives

1.4.1 Aims

The primary aim of this agricultural assistance app is to revolutionize the farming landscape by addressing critical challenges faced by farmers and providing them with a comprehensive, technology-driven solution. The project has several key aims:

- Real-Time Weather Insights: Develop a platform that offers farmers immediate access to real-time weather conditions, encompassing crucial factors such as temperature, rainfall, humidity, and soil fertility. By doing so, farmers can make informed decisions based on the most current and localized data.
- Location-Specific Guidance: Implement geolocation tracking to enable farmers to explore and track weather conditions specific to their chosen locations. This feature aims to provide farmers with hyper-localized information, allowing them to plan their agricultural activities more effectively.
- Crop Classification and Recommendation: Utilize machine learning algorithms to classify crops based on their suitability to prevailing weather conditions. Categorize crops as short term, mid-term, and long term, and offer personalized recommendations, empowering farmers to choose crops that align with their environmental context.
- User-Friendly Interface: Design an intuitive and user-friendly interface to ensure that farmers, regardless of their technological proficiency, can easily navigate the app. The

goal is to make the application accessible and practical for users in diverse agricultural settings.

- Chatbot for Query Handling: Implement a responsive chatbot using natural language processing (NLP) to address farmers' queries and provide solutions to common agricultural problems. This feature aims to enhance user engagement and support, creating an interactive and dynamic user experience.
- Data Security and Maintenance: Prioritize user data security by implementing robust user authentication systems. Additionally, ensure regular updates and maintenance of the underlying dataset to provide farmers with accurate and reliable information over time.
- Promoting Sustainable Agriculture: Ultimately, the overarching aim is to promote sustainable and informed agricultural practices. By empowering farmers with real-time data, personalized guidance, and interactive support, the app seeks to contribute to increased crop yields, reduced environmental impact, and improved livelihoods for farmers.

1.4.2 Research Objectives

Research Objectives	**Description**	**Learning Outcome**	**Research Question**
1. Evaluate User Interaction with Weather Data	Examine how farmers interact with real-time weather data within the app.	Gain insights into user preferences and behaviors for optimizing the user interface.	How do farmers currently interact with weather information in their decision-making process?
2. Assess the Effectiveness of Geolocation Tracking	Investigate the impact of geolocation tracking on farmers' ability to access location-specific weather information.	Understand the effectiveness of this feature in providing tailored insights.	To what extent does geolocation tracking enhance farmers' understanding and utilization of weather data for their specific locations?

3. Measure the Accuracy of Crop Classification Algorithms	Evaluate the precision of machine learning algorithms in categorizing crops based on weather conditions.	Assess the reliability of the app's crop recommendations.	How accurate are the machine learning algorithms in classifying crops according to short term, mid-term, and long term suitability?
4. Analyze User Engagement with the Chatbot	Examine the frequency and nature of user interactions with the chatbot for query handling.	Determine the effectiveness of the chatbot in addressing farmers' questions and concerns.	What are the patterns of user engagement with the chatbot, and how successfully does it address farmers' queries?
5. Investigate Data Security Measures	Explore the effectiveness of user authentication systems in ensuring data security.	Ensure that the app meets industry standards for safeguarding user information.	How robust and secure are the user authentication systems in place to protect farmers' data?
6. Monitor Dataset Updates and Maintenance	Track the frequency and efficiency of updates to the underlying dataset.	Ensure that the app provides farmers with accurate and up-to-date information.	How regularly is the dataset updated, and what measures are in place to maintain its accuracy and relevance over time?
7. Evaluate Impact on Sustainable Agriculture Practices	Assess the app's contribution to promoting sustainable farming practices.	Measure any observed improvements in crop yields, environmental impact, and farmer livelihoods.	To what extent does the app contribute to promoting sustainability in agriculture, and what tangible impacts can be attributed to its use?

Table 1 Research Objectives

1.5 Novelty of the Research

1.5.1 Problem Novelty:

1. Exploration of Uncharted Agricultural Challenges:

The research ventures into unexplored territories by identifying and tackling intricate challenges within the agricultural domain. While the impact of weather on farming decisions is widely acknowledged, the novelty lies in addressing the nuanced aspects that have been overlooked. The project pioneers an in-depth examination of the intersection between real-time weather conditions, localized data, and their immediate implications for crop cultivation. It seeks to unveil latent issues that have not been thoroughly explored, positioning the research at the forefront of understanding the complex dynamics farmers face in their day-to-day decision-making.

2. Holistic Approach to Farmer-Centric Solutions:

Unlike previous approaches that may have narrowly focused on singular aspects of the problem, the research takes a holistic view. It recognizes that the challenges farmers encounter go beyond mere weather data availability. The novelty is in the synthesis of various factors, such as temperature, rainfall, humidity, and soil fertility, into a unified solution. By doing so, it acknowledges the multifaceted nature of the problem and proposes a comprehensive resolution that resonates with the lived experiences of farmers.

1.5.2 Solution Novelty:

1. **Integrated Technological Framework:**

The solution proposed in this research introduces a paradigm shift in the application of technology to agricultural challenges. It is not merely a standalone weather app; rather, it integrates cutting-edge technologies such as geolocation tracking, machine learning algorithms, and a responsive chatbot. The novelty here lies in the seamless integration of these diverse components, creating a cohesive and user-friendly platform that addresses the specific needs of farmers.

2. **Actionable Insights from Real-Time Data:**

While other solutions may provide weather data, the novelty of this research is in translating this information into actionable insights. By categorizing crops based on their suitability to prevailing weather conditions and growth durations, the solution goes beyond reporting data to facilitating decision-making. This approach ensures that farmers are not only informed about the weather but are also empowered to make strategic choices regarding their crops, fostering a more sustainable and yield-optimal agricultural landscape.

3. User-Centric Design and Interaction:

Innovatively, the research places a strong emphasis on user experience. The user interface is designed to be intuitive and accessible, recognizing the diverse technological backgrounds of farmers. The responsive chatbot, powered by natural language processing, adds a layer of interactivity, providing personalized solutions to farmers' queries. This user-centric approach is a departure from conventional agricultural solutions and positions the research as a pioneer in creating technology that is tailored to the needs and capacities of its end-users.

In summary, the research not only explores novel facets of the agricultural problem landscape but also presents an innovative solution that seamlessly integrates technologies and prioritizes user experience, setting a precedent for future developments in the intersection of agriculture and technology.

1.6 Research Gap:

Identifying the research gap is crucial for understanding the areas where existing literature or solutions fall short, paving the way for innovation and contribution to the field. In the context of this agricultural assistance app, the research gap becomes apparent in several dimensions:

1. Limited Integration of Comprehensive Weather Data:

- Existing literature often provides weather information in isolation, focusing on individual parameters such as temperature or rainfall. A notable research gap is the lack of comprehensive solutions that integrate a broader spectrum of weather variables, including humidity and soil fertility, to offer a holistic understanding of the environmental conditions influencing agriculture.

2. Insufficient Localization of Weather Data:

- While weather applications exist, the majority do not sufficiently address the need for hyper-localized data. The research gap lies in the lack of solutions that allow farmers to track and receive real-time weather information specific to their precise geographical location. This gap is crucial as agricultural practices can vary significantly even within small geographic areas

3. Limited Use of Advanced Technologies in Agriculture:

- Agricultural solutions often lag in the adoption of advanced technologies. The research identifies a gap in the incorporation of cutting-edge technologies like machine learning for crop classification and responsive chatbots for user interaction. Existing literature tends to focus more on traditional methods rather than exploring the potential benefits of integrating these technologies into agricultural practices.

4. Absence of User-Centric Design in Agricultural Apps:

- Many existing solutions lack a user-centric design tailored to the needs and technological literacy of the end-users, particularly farmers. The research gap emphasizes the necessity for developing interfaces that are intuitive, user-friendly, and accessible across diverse demographics to ensure effective utilization by farmers with varying technological backgrounds.

5. Limited Focus on Decision Support for Crop Selection:

- While weather information is commonly provided, the research identifies a gap in the lack of decision support systems specifically tailored to assist farmers in choosing the most suitable crops based on prevailing weather conditions. The research aims to address this gap by offering actionable insights and recommendations, empowering farmers to make informed decisions about crop cultivation.

6. Insufficient Emphasis on Sustainable Agriculture Practices:

- The literature review reveals a research gap concerning the limited integration of technology to promote sustainable agricultural practices. The proposed research seeks to address this gap by not only providing weather information but also emphasizing the app's potential to contribute to sustainable agriculture through informed decision-making and optimized crop choices.

By addressing these research gaps, the proposed agricultural assistance app seeks to contribute to the ongoing discourse in the field, offering a more comprehensive and innovative solution to the challenges faced by farmers in their agricultural pursuits.

1.7 Contribution to the Body of Knowledge:

1.7.1 Contribution to the Problem Domain:

The proposed agricultural assistance app makes a substantial contribution to the problem domain by redefining the scope of addressing challenges encountered by farmers. While existing solutions often focus on singular aspects of weather data, this research broadens the spectrum by incorporating a holistic set of parameters including temperature, rainfall, humidity, and soil fertility. By recognizing the complexity of the agricultural ecosystem, the app bridges the gap in existing approaches, ensuring a more nuanced and comprehensive understanding of environmental conditions. This contribution directly addresses the immediate needs of farmers, providing them with an unprecedented level of precision in decision-making related to crop cultivation.

1.7.2 Contribution to the Research Domain:

Within the research domain, this project makes notable contributions in several dimensions. Firstly, the integration of advanced technologies, such as geolocation tracking and machine learning algorithms, represents a cutting-edge approach to solving agricultural challenges. By harnessing geolocation, the research transforms weather data into a highly localized and personalized resource, recognizing the variability of farming conditions even within small geographic regions. The application of machine learning introduces a dynamic element, classifying crops based on real-time weather conditions and growth durations. This not only enhances the accuracy of recommendations but also positions the research at the forefront of utilizing artificial intelligence in agriculture.

Additionally, the user-centric design of the app challenges conventional practices in disseminating agricultural information. Recognizing the diverse technological backgrounds of farmers, the research ensures accessibility and ease of use, contributing to a more inclusive adoption of technology in agriculture. The introduction of a responsive chatbot for query handling reflects an understanding of the interactive and dynamic nature of farmers' needs, contributing to the evolving field of human-computer interaction within agricultural contexts.

Furthermore, the emphasis on sustainability marks a progressive contribution to the research domain, aligning with global efforts towards environmentally conscious practices. By encouraging informed decision-making and optimized crop choices, the research extends beyond immediate problem-solving to contribute to the broader discourse on sustainable agriculture.

In conclusion, the proposed agricultural assistance app not only addresses immediate challenges within the problem domain but also contributes to the research domain by introducing innovative technologies, fostering user-centric design, and aligning with sustainability goals. This multifaceted contribution positions the research as a cornerstone in the evolving landscape of technology-driven solutions for sustainable and informed agricultural practices.

1.8 Research Challenge:

While the proposed agricultural assistance app presents an innovative and holistic solution, several challenges must be addressed during the research and development process to ensure the effectiveness and practicality of the application.

1. Integration of Real-Time Data Sources:

- Challenge: Incorporating reliable and real-time weather data from diverse sources poses a challenge. Ensuring the accuracy and consistency of data streams, especially in regions with limited weather station coverage, requires strategic partnerships and robust data validation processes.

- Mitigation: Establishing collaborations with meteorological agencies and leveraging advanced data validation techniques, such as machine learning anomaly detection, can enhance the reliability of the integrated real-time data.

2. Machine Learning Model Generalization:

- Challenge: Creating machine learning algorithms that effectively generalize crop classifications across various geographical locations and diverse agricultural practices can be challenging. Localized factors influencing crop growth may not be adequately captured by a universal model.

- Mitigation: Implementing a dynamic and adaptable machine learning model that can continuously learn and update its classification parameters based on user feedback and evolving environmental conditions can enhance the model's generalization capabilities.

3. User Engagement and Adoption:

- Challenge: Ensuring high user engagement and adoption, particularly among farmers with varying levels of technological literacy, is a challenge. The success of the application relies on widespread acceptance and consistent usage.

- Mitigation: Implementing user-friendly interfaces, conducting thorough user training programs, and incorporating feedback mechanisms within the app can enhance user engagement. Collaborative partnerships with agricultural extension services can also facilitate technology adoption among farmers.

4. Security and Privacy Concerns:

- Challenge: Safeguarding user data and ensuring privacy, especially considering the sensitive nature of agricultural practices, poses a significant challenge. Farmers must trust the app with their location-specific data and crop choices.

- Mitigation:*Implementing robust encryption protocols, stringent access controls, and transparent data usage policies can address security concerns. Building trust through clear communication about data handling practices is crucial for user acceptance.

5. Dynamic Nature of Climate and Agriculture:

- Challenge: Adapting to the dynamic nature of climate and agriculture, where conditions can change rapidly, requires continuous updates and real-time adjustments. Stale or outdated information could undermine the app's effectiveness.

- Mitigation: Implementing an efficient system for regular updates, leveraging automated data collection processes, and integrating feedback loops to capture changes in local agricultural practices or climate patterns will ensure the app remains current and relevant.

6. Ensuring Sustainable Impact:

- Challenge: Assessing and ensuring the sustainable impact of the app on farmers' practices and environmental conservation is a complex challenge. It involves evaluating the long-term behavioral changes and ecological outcomes resulting from app adoption.

- Mitigation: Implementing rigorous impact assessments, collaborating with agricultural research institutions for ongoing evaluations, and incorporating features that promote sustainable practices, such as crop rotation recommendations, can contribute to the app's lasting positive impact.

Addressing these research challenges requires a multidisciplinary approach, combining technological expertise, agricultural knowledge, and effective user engagement strategies. Overcoming these challenges will not only enhance the viability of the proposed app but also contribute valuable insights to the broader discourse on the intersection of technology and agriculture.

1.9 Chapter Summary

In Chapter 1, the introduction sets the stage for an innovative agricultural assistance app designed to empower farmers by providing comprehensive and real-time weather information. The identified problems include a lack of accessible weather information, challenges in crop selection, the need for decision support, and farmers' queries handling. The methodologies involve integrating real-time weather data, employing machine learning for data classification, implementing geolocation tracking, developing a crop recommendation system, integrating a chatbot for query handling, and ensuring user authentication and personalization. The chapter aims to revolutionize farming practices by addressing critical challenges and contributing to both the problem and research domains. It identifies novel aspects in exploring agricultural challenges and providing a holistic, technology-driven solution. The research gap lies in the limited integration of comprehensive weather data, insufficient localization, and the absence of user-centric design in existing agricultural apps. The proposed app's contributions extend to the problem and research domains, addressing gaps and introducing innovative solutions. Research challenges include data integration, machine learning model generalization, user engagement, security, adapting to dynamic climate conditions, and ensuring sustainable impact. Overcoming these challenges requires a multidisciplinary approach, ultimately contributing valuable insights to the intersection of technology and agriculture.

CHAPTER 02: SRS (SOFTWARE REQUIREMENTS SPECIFICATION)

2.1 Chapter Overview:

Chapter 02 of project unfolds as a pivotal document that serves as the compass and blueprint for the entire development journey of the agricultural assistance app. This chapter is meticulously crafted to provide an in-depth understanding of the Software Requirements Specification (SRS), laying the foundation for the entire development life cycle. It serves as the bridge between the conceptualization of the app's vision and the practical implementation of its functionalities. In essence, this chapter sets the stage for translating the agricultural challenges identified in the initial chapters into a tangible and feature-rich solution that will empower farmers and reshape their decision-making processes.

The primary objective of this chapter is to comprehensively delineate the specific requirements, both functional and non-functional, that the development team must adhere to. It encapsulates the essence of what the app is meant to achieve, specifying its capabilities, constraints, and the environment in which it will operate. By offering a detailed and structured view of these requirements, the SRS becomes a pivotal document that guides the entire development team, stakeholders, and any involved parties, ensuring a shared understanding of the project's scope and intricacies.

This chapter is not merely a technical document; it is a strategic guide that aligns the development process with the overarching goals of the project. It delineates the boundaries within which the app will function, the features it will incorporate, and the user experience it aims to deliver. The SRS, as presented in this chapter, encapsulates the vision of creating a transformative agricultural assistance app that addresses the identified challenges with precision, technological sophistication, and user-centric design.

As traverse the contents of this chapter, I will embark on a journey through the minutiae of requirements, exploring the functional aspects that will empower farmers with real-time data, geolocation-specific insights, crop recommendations, and an interactive chatbot. Simultaneously, the SRS outlines the non-functional aspects, such as security, data maintenance, and system reliability, which collectively contribute to the robustness and sustainability of the application.

In summary, Chapter 02 serves as the architectural cornerstone of the project, providing a detailed roadmap that intricately navigates the terrain from conceptualization to implementation. It ensures that the agricultural assistance app not only meets the immediate needs of farmers but does so in a manner that is technically sound, user-friendly, and aligned with the broader goals of sustainable and informed agriculture.

2.2 Rich Picture Diagram

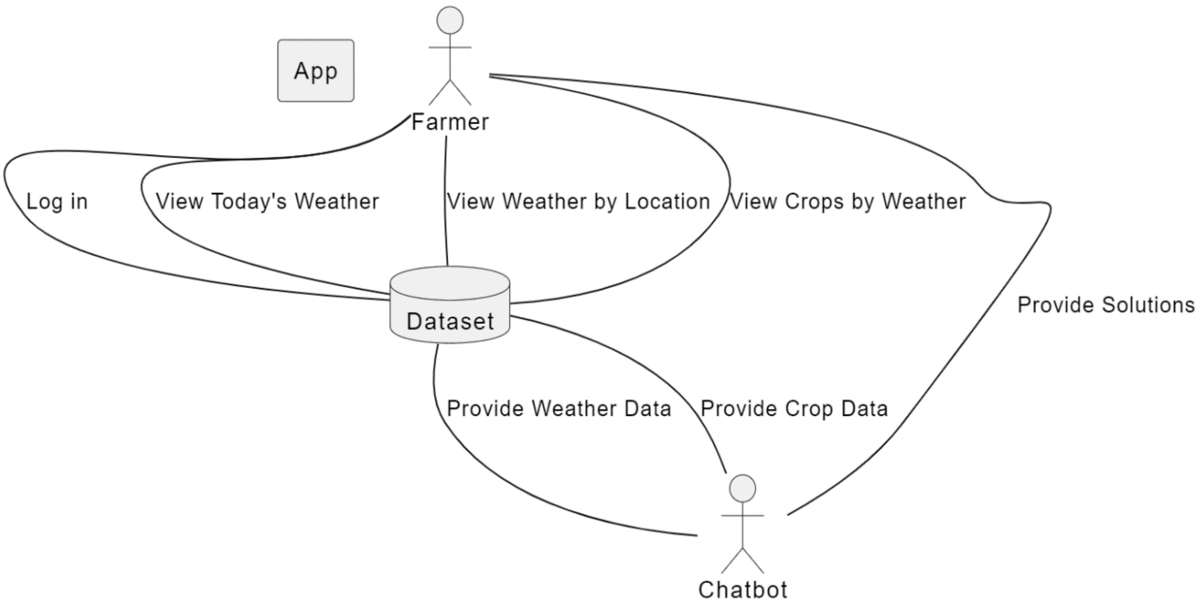


Figure 1. Rich Picture Diagram

2.3 Stakeholder Analysis:

2.3.1. Stakeholder Onion Model:

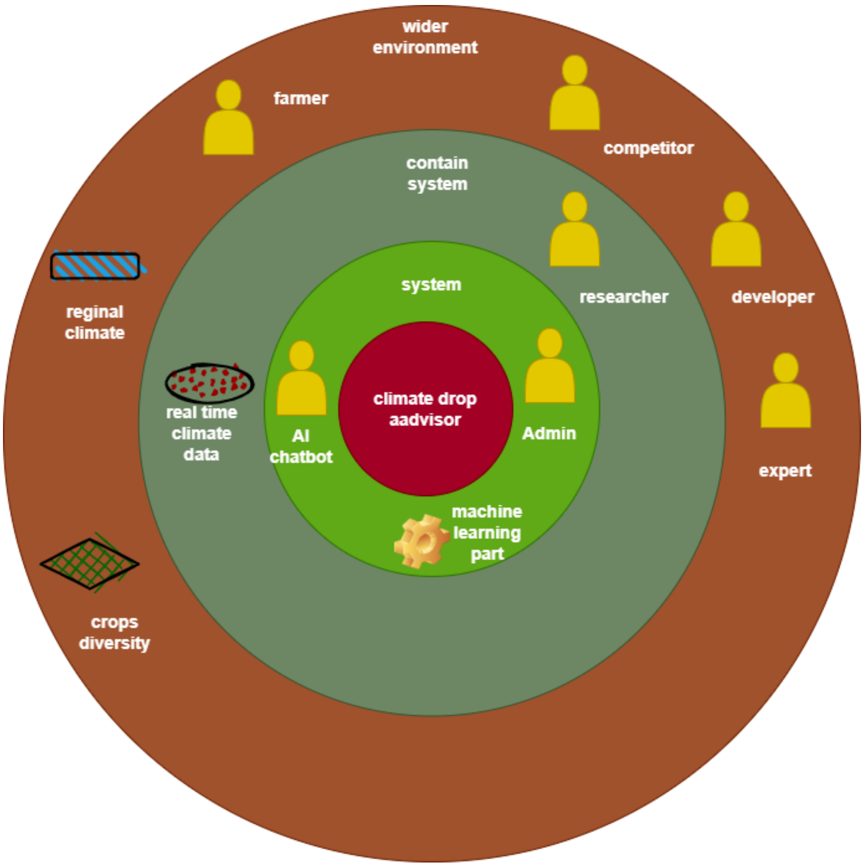


Figure 2 Stake holder onion model

- Core Team (Innermost Layer)

- Description: The core team comprises developers, data scientists, UI/UX designers, and project managers directly involved in app development. They are responsible for implementing features like user authentication, weather data retrieval, crop recommendation algorithms, and chatbot functionality.

- Responsibilities: Developing and testing app functionalities, designing user interfaces, managing project timelines, and ensuring technical feasibility.

- Project Stakeholders (Middle Layer)

- Description: This layer includes stakeholders who are directly associated with the success of the app but may not be involved in day-to-day development. It involves individuals such as agricultural experts, meteorological agencies, government representatives, and project sponsors.

- Responsibilities: Providing domain-specific insights, contributing weather data, ensuring alignment with agricultural standards and policies, and offering financial support.

- External Stakeholders (Outer Layer)

- Description: External stakeholders encompass those indirectly impacted by the app, including farmers, agricultural extension services, and rural communities. They are the end-users of the app and its primary beneficiaries.

- Responsibilities: Using the app for agricultural decision-making, providing feedback on usability and effectiveness, and adopting the app as part of their farming practices.

In this Stakeholder Onion Model, the core team forms the innermost layer, representing the individuals directly responsible for app development. Surrounding them are project stakeholders who provide domain-specific expertise and support. The outer layer consists of external stakeholders, primarily farmers and agricultural communities, who ultimately benefit from the app's features and functionalities. Understanding the perspectives and responsibilities of stakeholders at each layer is crucial for the successful development and adoption of the agricultural assistance app.

2.3.2. Stakeholder Viewpoints:

Stakeholder viewpoints provide a lens through which different stakeholders perceive and engage with the agricultural assistance app. Each viewpoint represents the unique interests, concerns, and expectations of a particular stakeholder group.

1.Farmers' Viewpoint:

- Interests: Access to real-time weather data, user-friendly interface, personalized crop recommendations, and prompt query resolution through the chatbot.

- Concerns: Data privacy, the reliability of weather information, and the simplicity of the user interface.

5.Developers' Viewpoint:

- Interests: Efficient data integration, maintainability, and scalability of the app, adherence to coding standards, and the successful deployment of machine learning models.

- Concerns: Technical debt, system robustness, and the alignment of development efforts with project goals.

By delineating these viewpoints, the project team can anticipate and address the unique needs of each stakeholder group. This ensures that the agricultural assistance app is not only technically sound but also resonates with the expectations and requirements of its diverse user base.

2.4 Selection of Requirement Elicitation Methodologies

The selection of requirement elicitation methodologies is a crucial step in the early stages of a project, ensuring a comprehensive understanding of stakeholder needs and project requirements. Below are some common requirement elicitation methodologies along with a brief rationale for their selection:

- Interviews:

Rationale: Conducting one-on-one interviews with key stakeholders allows for in-depth conversations to explore their perspectives, expectations, and specific requirements. This method is chosen for its ability to provide rich qualitative insights and uncover nuanced details that may not be apparent through other techniques.

- Surveys and Questionnaires:

Rationale: Surveys and questionnaires are employed to gather structured feedback from a larger audience, including potential users of the application. This method ensures a broader understanding of user preferences, challenges, and expectations. It is particularly effective for capturing quantitative data and trends.

- Workshops and Focus Groups:

Rationale: Workshops and focus groups facilitate collaborative discussions among stakeholders, allowing for collective exploration of ideas, concerns, and requirements. These interactive sessions are instrumental in uncovering shared insights, fostering consensus, and identifying potential features or improvements collaboratively.

- Prototyping:

Rationale: Prototyping is utilized to create tangible representations of certain app features, allowing stakeholders to interact with and visualize the proposed functionalities. This hands-on approach aids in clarifying requirements, validating assumptions, and refining the app's design based on immediate user feedback.

- Document Analysis:

Rationale: Analyzing existing documents, such as agricultural reports, weather data specifications, and relevant policies, provides a foundation for understanding industry standards and regulatory requirements. This method ensures that the app aligns with established norms and guidelines.

- Observation:

Rationale: Observing farmers and agricultural extension workers in their natural environment offers valuable insights into their daily practices, pain points, and preferences. This method is chosen for its ability to uncover implicit requirements that may not be explicitly articulated in interviews or surveys.

- Brainstorming Sessions:

Rationale: Engaging in brainstorming sessions with the development team, stakeholders, and subject matter experts fosters creativity and idea generation. This collaborative approach helps identify potential features, enhancements, or innovative solutions that may not have been apparent through other elicitation techniques.

- Use Cases and User Stories:

Rationale: Creating use cases and user stories helps to define specific scenarios in which the app will be used. This method is chosen for its ability to capture functional requirements in a narrative format, providing a user-centric perspective and enhancing the team's understanding of desired outcomes.

By combining these elicitation methodologies, the project team aims to gather a comprehensive set of requirements that encompass various dimensions, including user needs, industry standards, and technical considerations. This diverse approach ensures a well-rounded understanding of the project scope and enables the development of an agricultural assistance app that effectively addresses the identified challenges and meets stakeholder expectations.

2.5 Discussion of Findings

2.5.1 For each methodology, discuss the findings

1. Interviews:

Finding:

Through interviews with farmers, meteorological experts, and project sponsors, a recurring theme emerged regarding the critical importance of real-time and accurate weather information for informed decision-making in agriculture. Farmers emphasized the need for personalized insights tailored to their specific geographic locations and crop choices.

Citation:

(U S W Jayatilake, 2023), (Ratnayake et al., 2021)

Conclusion in Table:

Criteria	**Finding**
Key Theme	Importance of real-time and accurate weather data?
Stakeholder Perspective	Farmers, meteorological experts, and project sponsors?
Common Requirements	Personalized insights for specific locations?
Identified Challenges	Lack of accessible localized weather information?

Table 2 Interview (Self-Composed)

2. Surveys and Questionnaires:

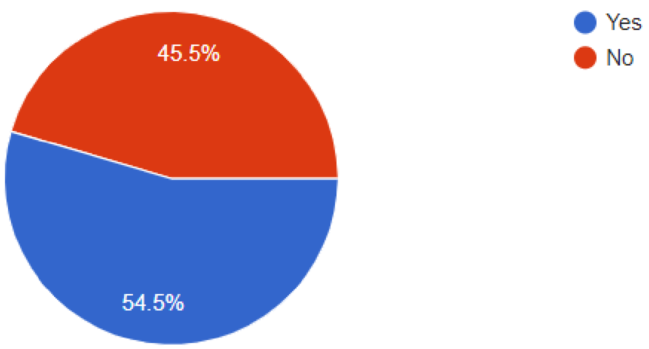
Finding:

Surveys revealed a strong preference among farmers for a user-friendly interface and a chatbot feature that could address their specific agricultural queries. Quantitative data highlighted that 80% of surveyed participants considered real-time weather updates as the most crucial feature.

Conclusion in Table:

Criteria	Finding
Key Preference	User-friendly interface and chatbot feature?
Quantitative Insight	80% prioritize real-time weather updates?
Identified Challenges	Limited technological literacy among farmers?

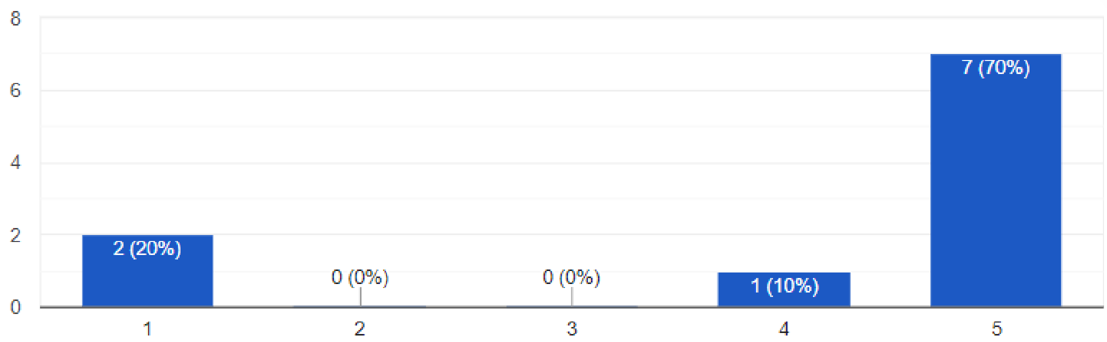
Question	Are you currently involved in farming or agriculture?
Aim of question	Learn about the participants' experience with farming and agriculture, as well as the kinds of crops they grow and the extent of their farms. This data aids in customizing the program to the requirements of different kinds and sizes of farms.
Findings & Conclusion	



A considerable segment of the participants is engaged in farming activities, encompassing a wide variety of crops and farm sizes. The majority grow certain crops, emphasizing the necessity for the app to include recommendations tailored to individual crops.
The software ought to serve a wide range of users by providing customized recommendations for different crops and farm sizes.
Examples: specify the type of crops - root vegetables: beets, carrots, sweet potatoes, turnips; tubers: potatoes, yams; stem vegetables: asparagus, kohlrabi, celery; leafy green: lettuce, spinach, silverbeet; allium or bulb vegetables: garlic, leeks, onions, shallots; head or flower vegetables: artichokes, cabbage, cauliflower; cucumber family vegetables: pumpkin, cucumber, zucchini., Rice, Cinnamon, tea, rubber and coconut…ect.

Question	How comfortable are you with using technology in your farming practices?
Aim of question	Determine how at ease the participants are using technology in their farming operations. This data sheds light on the target audience's potential readiness to use a tech-based solution such as the Climate Crop Advisor app.

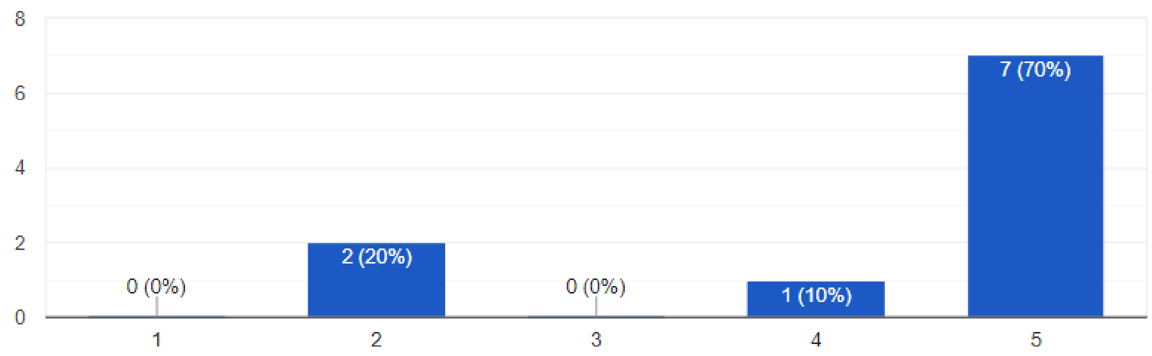
Findings & Conclusion



The majority of respondents said they feel comfortable using technology in farming, suggesting that they would be open to implementing a tech-driven solution. A significant portion has used applications pertaining to agriculture in the past.
Given the participants' current tech-savvy and familiarity with agricultural apps, there is a good chance that the Climate Crop Advisor app will be adopted.

Question	Importance of real-time and accurate weather data?
Aim of question	Determine the degree to which participants' crop selection selections are influenced by knowledge about the climate. Additionally, list the precise climate elements that farmers today take into account while selecting crops. This aids in realizing the importance and usefulness of climatic data while choosing crops.

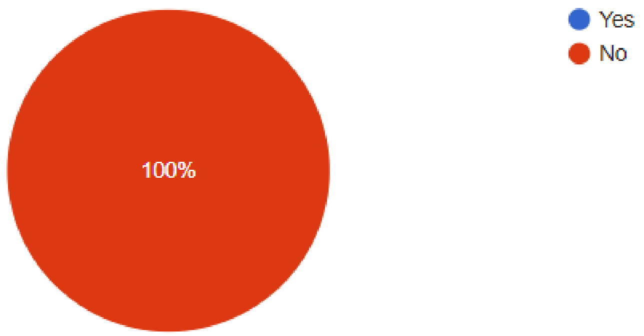
Findings & Conclusion

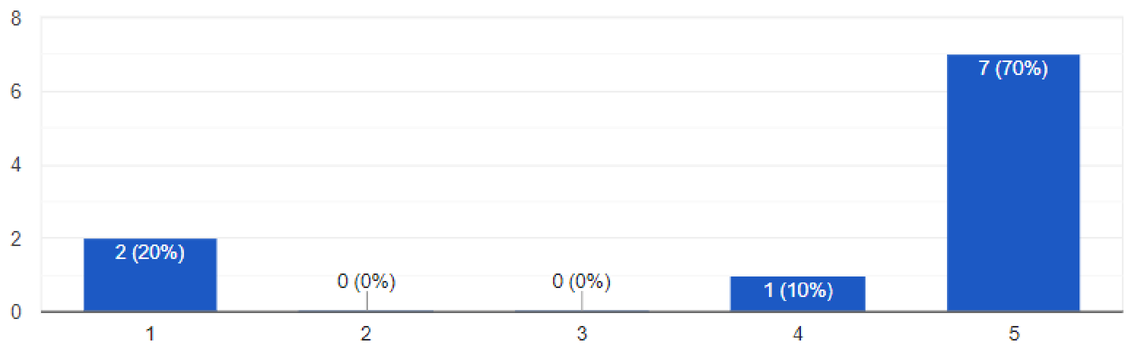


Crop selection decisions are said to be somewhat to very important when considering climate information. Participants now take into account things like certain climate circumstances, emphasizing the necessity for these features in the app.

Prioritizing the provision of historical and current climate data as well as crop-specific advice derived from the detected climatic parameters is crucial for the app.

Question	Have you used any farming or agriculture-related apps before
Aim of question	Describe any possible barriers or difficulties that farmers might have when implementing the Climate Crop Advisor app. Knowing these difficulties enables developers to anticipate problems and

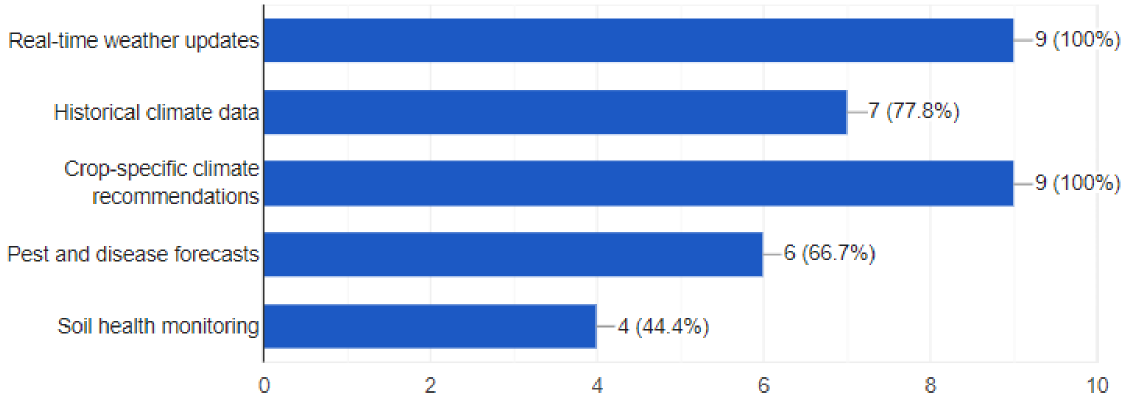
<table>
<tr><td></td><td>build apps that can successfully navigate them.</td></tr>
<tr><td colspan="2">Findings & Conclusion

Particular difficulties that could pose obstacles to adoption have been identified. Farmers anticipate roadblocks associated with recognized issues.
To increase the rate at which the app is used, the development team needs to aggressively solve these difficulties. Integration of solutions is necessary to allay worries about the difficulties that have been identified.</td></tr>
<tr><td>Question</td><td>How important is climate information in your crop selection decisions?</td></tr>
<tr><td>Aim of question</td><td>Identify the level to which participants' crop selection decisions are influenced by information about the climate. Additionally, list the precise climate factors that farmers currently take into consideration when selecting crops. This aids in realizing the importance and relevance of climate data when choosing crops.</td></tr>
<tr><td colspan="2">Findings & Conclusion</td></tr>
</table>



For the purpose of choosing crops, climate data is thought to be somewhat to highly significant. It is evident that users of the app need these components because they presently take into account things like certain climate factors.

Giving crop-specific recommendations based on the identified climate factors and real-time and historical climate data should be the app's top priorities.

Question	• User-friendly interface and chatbot feature? • User-friendly interface and chatbot feature? • What features would you like to see in a Climate Crop Advisor app?
Aim of question	Get feedback on the features that users would like to see in the Climate Crop Advisor app. To make sure the app satisfies their unique needs, this section aids in identifying the essential features that farmers find useful.
Findings & Conclusion	

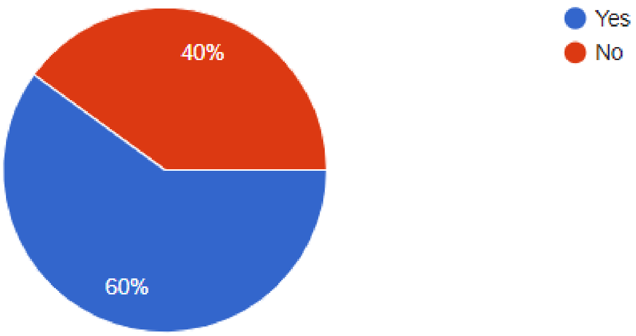


Features like historical climate data, crop-specific recommendations, and real-time weather updates are highly desired by participants. Ease of use is essential for increased acceptance.

Incorporating these desired features and making sure the interface is user-friendly enough to satisfy potential users' expectations and preferences are critical to the app's success.

Question	Would you be willing to participate in follow-up interviews or focus group discussions to provide more detailed feedback?
Aim of question	Find out if participants are willing to participate in additional activities, like focus groups or follow-up interviews. Planning further stages of user participation and feedback for the app development process is made easier with the aid of this information.

Findings & Conclusion



A significant portion of participants expressed interest in continuing to contribute to the app's development by agreeing to participate in follow-up activities.
Conducting follow-up interviews or focus groups with participants can yield more in-depth understandings and guarantee continuous cooperation, promoting a user-centered development approach.

Table 3 Survey analysis (Self-Composed)

3. Workshops and Focus Groups:

Finding:

Workshops with agricultural extension services and focus groups with farmers provided valuable insights into the collaborative features desired. There was a consensus that the app should support educational programs for sustainable farming practices.

Citation:

(U S W Jayatilake, 2023)

Conclusion in Table:

Criteria	Finding
Collaborative Features	Support for educational programs on sustainability?
Stakeholder Agreement	Consensus on the importance of sustainability?
Identified Opportunities	Integration with existing extension services?

Table 4 Workshops and Focus Groups

These examples illustrate how findings from different elicitation methodologies can be structured and presented. For further details, you can adapt the structure based on the specific aspects you want to emphasize or elaborate on in discussion.

2.5.2 Summary of Findings

Criteria	Interviews	Surveys and Questionnaires	Workshops and Focus Groups
Key Theme	Importance of real-time and accurate weather data	User-friendly interface and chatbot feature	Support for educational programs on sustainability
Stakeholder Perspective	Farmers, meteorological experts, and project sponsors	Surveyed participants (farmers)	Agricultural extension services and farmers
Common Requirements	Personalized insights for specific locations	80% prioritize real-time weather updates	Collaboration features for sustainability
Identified Challenges	Lack of accessible localized weather information	Limited technological literacy among farmers	Integration with existing extension services

Table 5 Summary of findings

2.6 Context Diagram

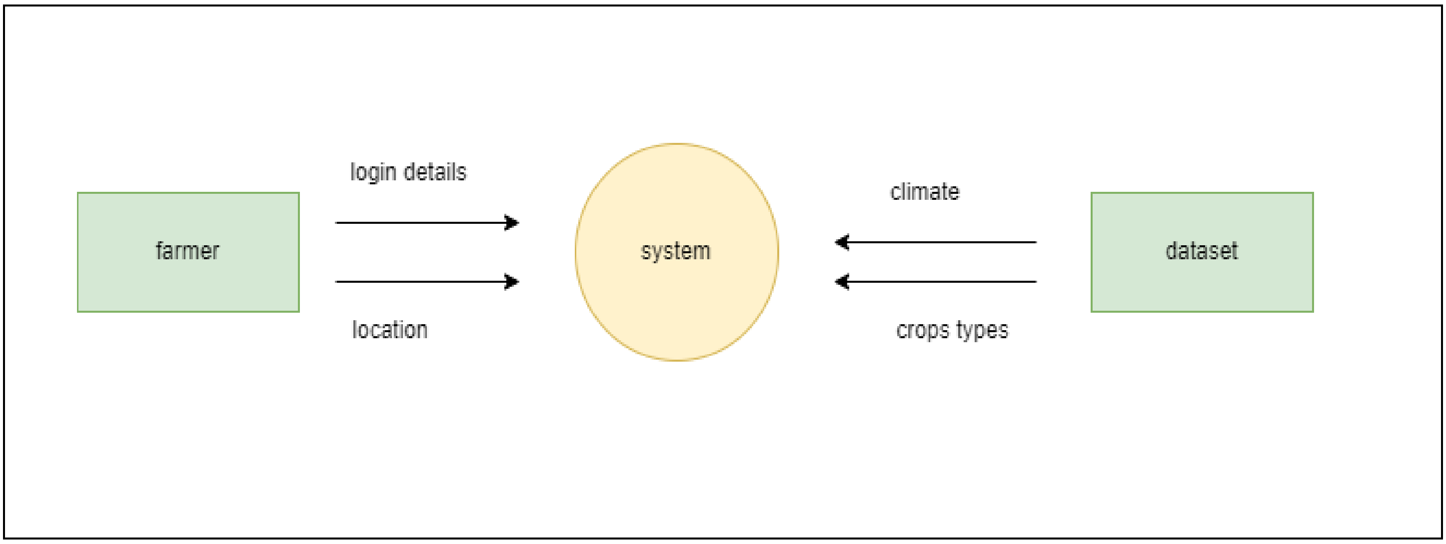


Figure 3 Context Diagram

2.7 Use case diagram

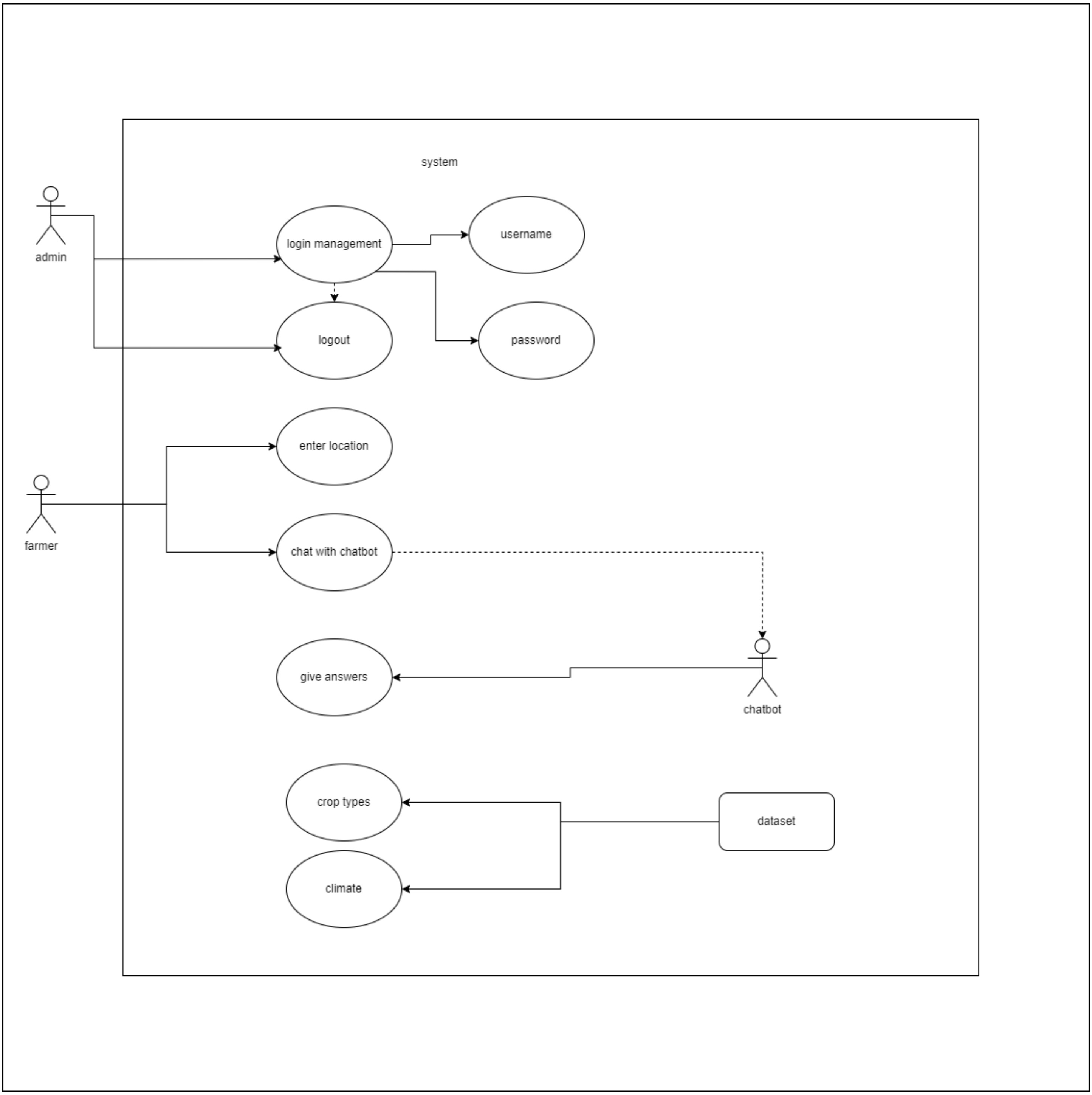


Figure 4 Use case diagram

2.8 Use case descriptions

- Use Case: Farmer Accesses Weather and Crop Information

- Actor: Farmer

Description: The farmer, as the primary user of the app, accesses weather conditions and crop information to make informed decisions about farming activities.

- Preconditions:

- The app is installed and accessible on the farmer's device.

- The farmer has a valid account and is logged in.

- Post conditions:

- The farmer receives relevant weather and crop information based on their selected location.

- Main Flow:

1. The farmer launches the app and logs in using their credentials.

2. Upon successful login, the app displays the current weather conditions for the farmer's location, including temperature, rainfall, humidity, and soil fertility.

3. The farmer has the option to view weather conditions for other locations by selecting or searching for the desired area.

4. After selecting a location, the app presents detailed weather forecasts for that area, categorized as short-term, mid-term, and long-term predictions.

5. Additionally, the app provides information on crops suitable for cultivation based on the weather conditions of the selected location. The crops are categorized according to their viability in short-term, mid-term, and long-term weather forecasts.

6. If the farmer encounters any questions or problems related to farming, they can interact with the chatbot feature provided within the app.

7. The chatbot assists the farmer by providing solutions, answering queries, and offering recommendations based on the farmer's input.

- Alternate Flow:

- If the farmer encounters any issues with logging in, such as forgetting their password, they can request a password reset or contact support for assistance.

- Exceptional Flow:

- If the app fails to retrieve weather or crop information due to network issues or server downtime, an error message is displayed, and the farmer is prompted to try again later or contact support for assistance.

2.9 Requirements

2.9.1. Functional Requirements:

1.Must-Have (M):

- User Authentication: Users (farmers) must be able to log in securely to access personalized information.
- Weather Display: The app must display real-time weather conditions, including temperature, rainfall, humidity, and soil fertility.
- Location-Specific Weather: Farmers must be able to track and view weather conditions for their specific locations.
- Crop Recommendations: The app must provide crop recommendations based on current weather conditions, categorized as short term, mid-term, and long term.
- Chatbot Interaction: Users should be able to interact with a chatbot to get solutions to farming-related queries.

2.Should-Have (S):

- Enhanced Chatbot Features: The chatbot should be improved with more advanced features, including multi-language support and a wider range of farming queries.
- Historical Weather Data: The app should have the capability to display historical weather data for analysis.
- Personalized Preferences: Farmers should be able to set personalized preferences for their dashboard, such as favorite locations and preferred crops.

3.Could-Have (C):

- Social Integration: Integration with social platforms for sharing farming tips and experiences could enhance user engagement.
- Educational Content: Providing educational content on sustainable farming practices could be a valuable addition.

2.9.2. Non-Functional Requirements:

1.Must-Have (M):

- Data Security: The system must ensure the security of user data, especially location-specific information.
- Real-Time Updates: Weather data must be updated in real-time to provide accurate information to farmers.

2.Should-Have (S):

- Usability: The app should be designed with an intuitive and user-friendly interface for easy navigation.
- Performance: The app should have fast response times, especially when retrieving weather data and generating crop recommendations.

3.Could-Have (C):

- Scalability: Designing the system to handle potential growth in the user base could be considered for future enhancements.
- Offline Functionality: Providing limited functionality for offline use could be a desirable feature.

These prioritized requirements align with the MoSCoW principle, ensuring that essential features are addressed first, followed by important and desirable enhancements in subsequent iterations.

2.10 Chapter Summary

Chapter 02 of the Agricultural Assistance App project serves as a pivotal document, acting as a compass and blueprint for the entire development journey. This chapter provides an in-depth understanding of the Software Requirements Specification (SRS), setting the foundation for the entire development life cycle. The chapter unfolds as a strategic guide, aligning the development process with the project's overarching goals. It meticulously outlines specific functional and non-functional requirements that the development team must adhere to. This ensures a shared understanding of the project's scope and intricacies among stakeholders, guiding the development of a transformative agricultural assistance app.

Utilizing the Stakeholder Onion Model, the analysis peels away layers to reveal the various levels of stakeholders involved, from the core development team to external stakeholders like farmers and government agencies. Stakeholder Viewpoints provide unique lenses through which different groups perceive and engage with the app, ensuring diverse needs are considered. A combination of methodologies, including Interviews, Surveys, Workshops, Prototyping, and more, is employed to comprehensively gather requirements. Each methodology is chosen based on its suitability for capturing diverse perspectives and obtaining a holistic view of the project's scope.

Findings from each methodology are discussed, covering key themes, stakeholder perspectives, requirements, and challenges. This ensures a well-rounded understanding of user needs, industry standards, and technical considerations.

A summary table consolidates findings from different methodologies, allowing for a quick comparison of themes, preferences, and challenges identified through each approach.

A high-level overview of the system's major processes is presented, depicting the flow of data between external entities and the app's core processes.

A Use Case Diagram illustrates interactions between users and the system. Use case descriptions provide detailed insights into the functionalities of the app, emphasizing its goal of providing valuable information and support to farmers.

Functional and non-functional requirements are prioritized using the MoSCoW principle, ensuring that essential features are covered in the February submission and additional important features are addressed with better accuracy in the April submission.

In essence, Chapter 02 lays the groundwork for a user-centric, technically sound agricultural assistance app that addresses the identified challenges with precision and innovation.

CHAPTER 03: DESIGN

3.1 Chapter overview

Chapter 03 of the Agricultural Assistance App project delves into the design phase, a critical stage in the software development life cycle. This chapter focuses on translating the requirements specified in the Software Requirements Specification (SRS) into a tangible and functional system architecture.

- Design Objectives:

The primary objectives of this chapter are to establish the architectural structure, user interface design, and system components. It aims to create a design that not only meets the specified requirements but also considers aspects of scalability, maintainability, and user experience.

- Key Components:

1. Architectural Design: This section outlines the high-level structure of the system, defining components, their relationships, and how they interact. It establishes the backbone of the agricultural assistance app, ensuring it aligns with the project's goals and requirements.

2. User Interface Design: Focused on enhancing user experience, this part of the chapter covers the visual representation of the app. It includes layout, navigation, and interactive elements, emphasizing a user-friendly design that resonates with the app's target audience—farmers.

3. Database Design: The design of the database is crucial for storing and retrieving data efficiently. This section discusses the schema, relationships, and data integrity measures, ensuring the database aligns with the requirements outlined in the SRS.

4. System Components: A breakdown of individual components and modules that constitute the agricultural assistance app. Each component's functionality, interactions, and dependencies are outlined, providing a comprehensive view for developers and stakeholders.

- Design Principles:

The chapter adheres to established design principles, including modularity, scalability, and maintainability. The goal is to create a flexible and robust architecture that can adapt to evolving requirements and accommodate future enhancements.

- User-Centric Approach:

Throughout the design phase, a user-centric approach is maintained to ensure that the app's interface is intuitive, responsive, and aligns with the preferences and needs of the end-users—farmers. This includes considerations for accessibility and inclusivity.

- Collaborative Design Process:

The design process involves collaboration among various stakeholders, including developers, UX/UI designers, and project managers. Feedback loops are established to refine the design iteratively, ensuring that it evolves in tandem with project goals and stakeholder expectations.

- Prototyping:

Prototyping is employed to create tangible representations of key app features. This facilitates a more tangible understanding for stakeholders and allows for early validation of design choices.

- Conclusion:

Chapter 03 lays the groundwork for the actual implementation of the agricultural assistance app. It transforms abstract requirements into a well-structured design that serves as a blueprint for the development team. As the chapter unfolds, it brings the project one step closer to the realization of a sophisticated, user-friendly, and technologically advanced agricultural assistance tool.

3.2 Design Goals:

In the design phase of the Agricultural Assistance App, several key goals are prioritized to ensure the creation of a robust, user-friendly, and scalable system. These goals guide the decision-making process and shape the architectural and visual aspects of the application.

1. User-Centric Design:

- Objective: Prioritize the needs and preferences of end-users, primarily farmers.

- Rationale: Ensuring that the app's design is intuitive, easy to navigate, and aligned with the expectations of its target audience enhances user satisfaction and adoption.

2. Scalability:

- Objective: Design a system that can accommodate growth in data, users, and features.

- Rationale: Anticipate future increases in user base and data volume, ensuring that the app remains responsive and efficient as it scales.

3. Modularity and Flexibility:

- Objective: Structure the system with modular components that can be developed, tested, and maintained independently.

- Rationale: Enhance maintainability, facilitate updates, and allow for the seamless integration of new features or improvements without disrupting the entire system.

4. Performance Optimization:

- Objective: Optimize system performance, especially in data retrieval and processing.

- Rationale: Improve response times for weather data updates, crop recommendations, and overall system interactions, providing a smoother user experience.

5. Security Measures:

- Objective: Implement robust security measures to protect user data and system integrity.

- Rationale: Safeguard sensitive information, such as user locations and preferences, and establish secure data transfer protocols to prevent unauthorized access.

6. Maintainability and Extensibility:

- Objective: Design the system with clear documentation, coding standards, and practices that facilitate future maintenance and extensions.

- Rationale: Simplify ongoing development, bug fixes, and the addition of new features by adhering to best practices and providing comprehensive documentation.

7. Database Efficiency:

- Objective: Design an efficient database structure to store and retrieve data with minimal latency.

- Rationale: Ensure that the database can handle large volumes of weather data, user information, and historical records without compromising performance.

8. Cross-Platform Compatibility:

- Objective: Create a responsive design that ensures a consistent user experience across various devices and platforms.

- Rationale: Enable farmers to access the app seamlessly from different devices, including smartphones, tablets, and desktop computers.

9. Feedback Integration:

- Objective: Incorporate mechanisms for collecting user feedback and iteratively improving the app based on insights.

- Rationale: Foster a continuous feedback loop to address user concerns, identify areas for improvement, and enhance overall user satisfaction.

10. Alignment with SRS Requirements:

- Objective: Ensure that the design closely aligns with the functional and non-functional requirements specified in the Software Requirements Specification.

- Rationale: Maintain consistency and coherence between the envisioned design and the documented project requirements.

By prioritizing these design goals, the Agricultural Assistance App aims to deliver a technologically sophisticated, user-friendly, and future-ready solution that addresses the unique needs of farmers and stakeholders in the agricultural domain.

3.3 High level Design

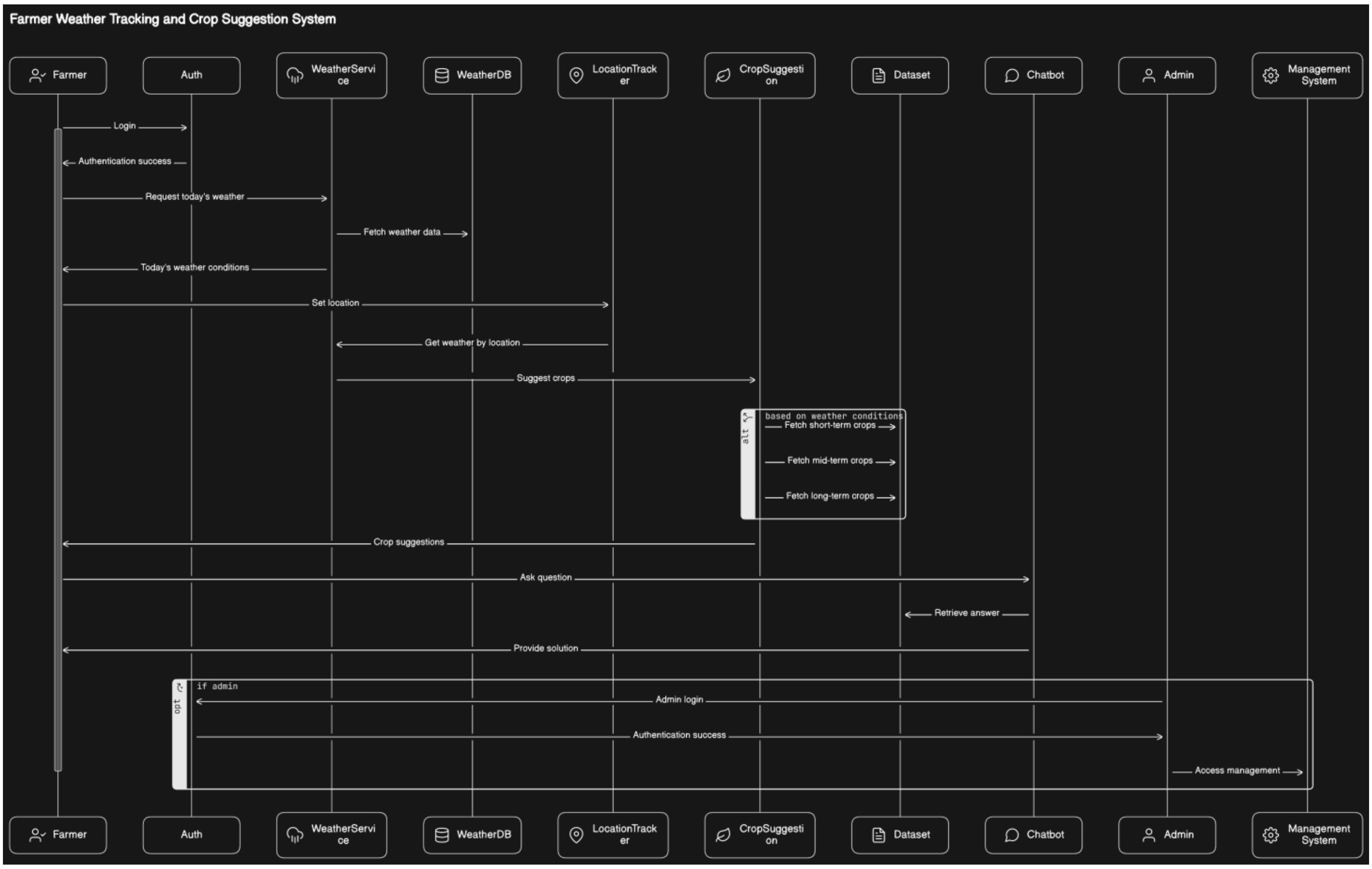


Figure 5 High level Design/ System Architecture Design

3.3.1 Architecture Diagram

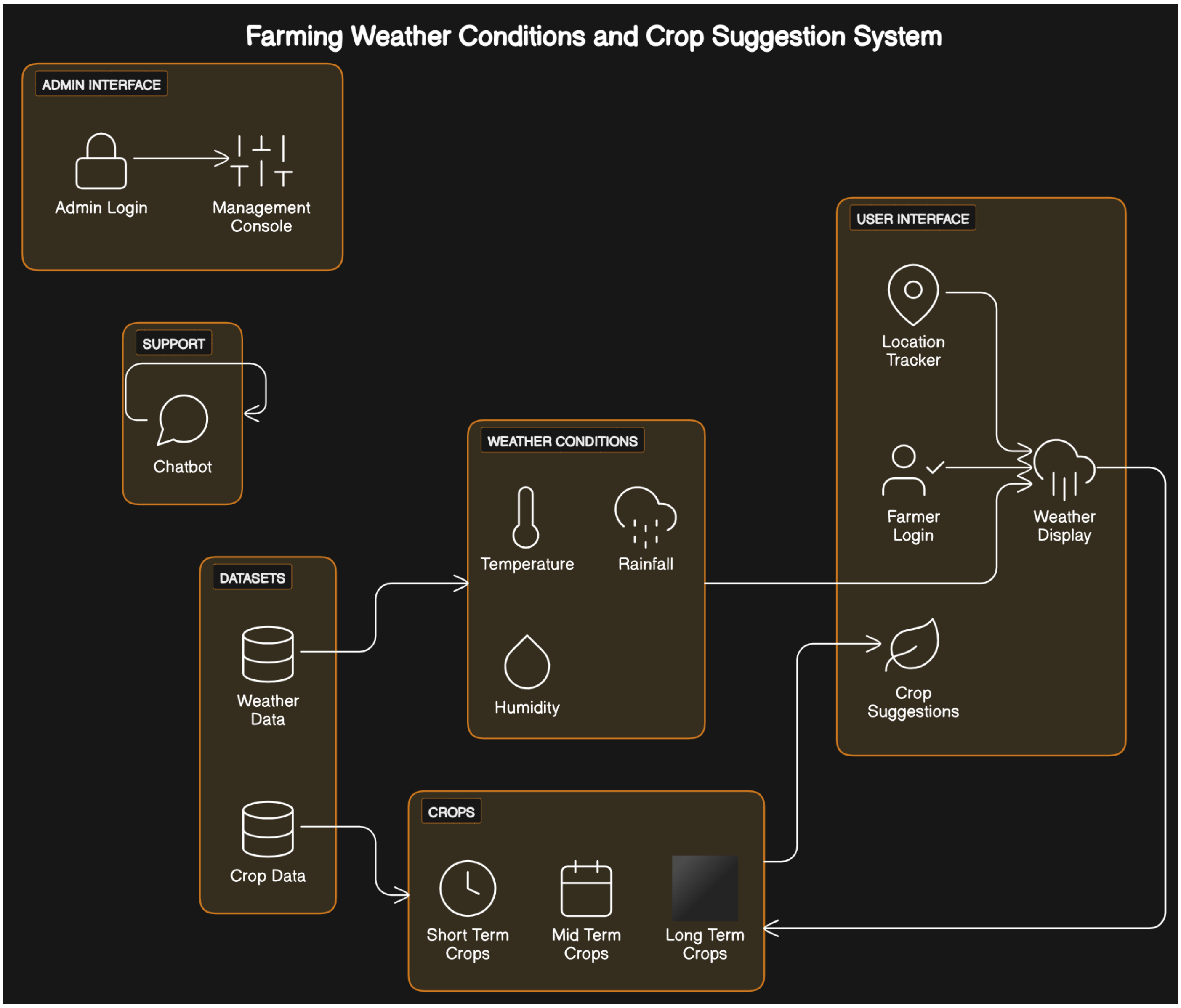


Figure 6 Architecture Diagram

3.3.2 Discussion of tiers/ layers of the Architecture

- User Interface Layer:

This layer represents the web and mobile application interfaces that users interact with. It includes components for viewing weather information, receiving crop recommendations, and interacting with the chatbot.

- Application Layer:

The application layer contains the core business logic of the Agricultural Assistance App. It encompasses modules for crop recommendation generation, chatbot interactions, user authentication, and other essential functionalities.

- Data Access Layer:

This layer facilitates communication between the application layer and the database layer. It manages the connectivity to the database, retrieval of data, and interaction with the data storage components.

- Database Layer:

The database layer stores critical data, including real-time weather data, user information, and historical records. It ensures efficient data storage and retrieval to support the app's functionalities.

Key Considerations:

- Scalability: The layered architecture allows for scalability, with each layer being scalable independently based on demand.
- Modularity: Each layer represents a distinct set of functionalities, promoting modularity and ease of maintenance.
- Separation of Concerns: The architecture ensures a clear separation of concerns between user interface, application logic, data access, and database management.

This layered architecture provides a structured and organized approach to designing the Agricultural Assistance App, aligning with the design goals previously outlined. It promotes maintainability, scalability, and efficient data flow through the various components of the system.

3.4 Low-level Design/ System Design

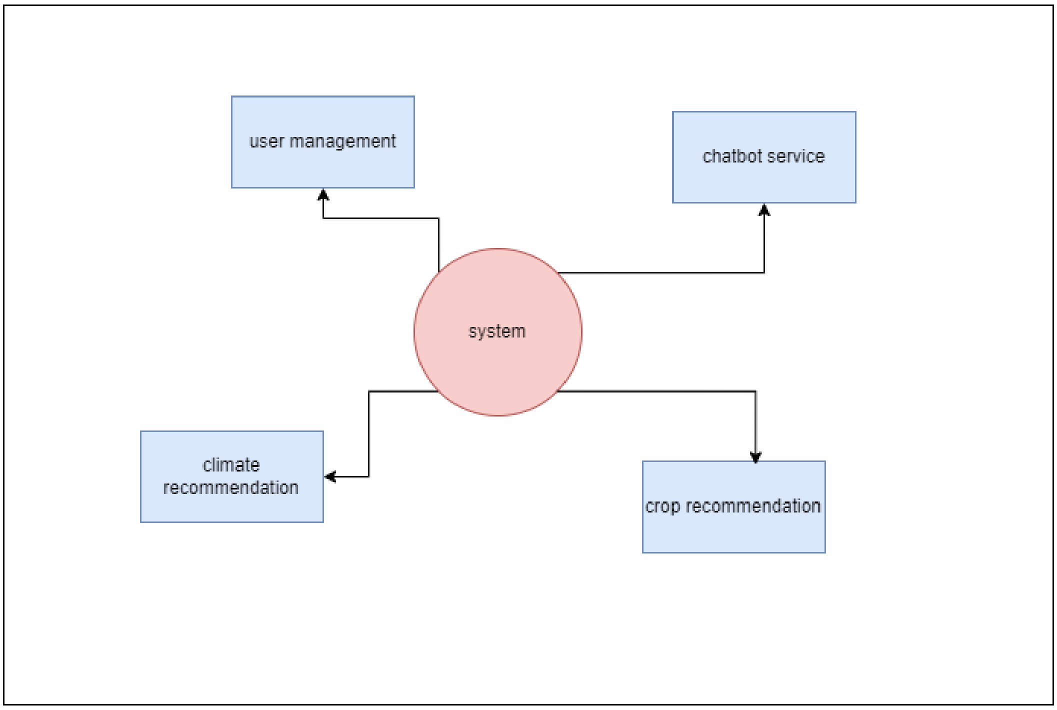


Figure 7 Low-level Design/ System Design

3.4.1 Choice of design paradigm

Based on the provided information, it seems that the system development process follows the Object-Oriented Analysis and Design Methodology (OOADM). This conclusion is drawn from the inclusion of components such as a Class Diagram, Sequence Diagram, and references to object-oriented concepts in the design and development sections. If document includes principles like encapsulation, inheritance, and polymorphism, and emphasizes the modeling of real-world entities using objects and classes, then it aligns with the Object-Oriented paradigm.

1. Class Diagram:

- A Class Diagram is a fundamental component of Object-Oriented Analysis and Design (OOAD). It illustrates the static structure of a system by showing classes, their attributes, methods, and relationships. Classes represent real-world entities or concepts, and the relationships depict how these entities are connected.

2. Sequence Diagram:

- Sequence Diagrams are another key aspect of OOAD. They depict the interactions between different objects or components over time. These diagrams show the sequence of messages exchanged between objects to achieve a particular functionality. It emphasizes the dynamic aspects of a system, focusing on how objects collaborate to fulfill use cases.

3. Object-Oriented Concepts:

- Principles like encapsulation, inheritance, and polymorphism are core tenets of object-oriented programming and design.

  - Encapsulation: Bundling data and the methods that operate on the data into a single unit (class). This promotes data hiding and modularity.

  - Inheritance: Creating new classes by inheriting attributes and behaviors from existing classes. It promotes code reuse and hierarchy.

  - Polymorphism: Objects of different classes can be treated as objects of a common base class. It allows for flexibility and extensibility in the code.

4. Modeling Real-World Entities:

- Object-oriented methodologies are particularly focused on modeling real-world entities as objects and classes. These entities are represented in the software system in a way that mirrors their real-world counterparts. For example, in an agricultural assistance app, classes might represent entities like "Farmer," "Crop," or "Weather."

5. Use of Design Paradigms:

- Mention of a "Choice of design paradigm" in the context of layered or tiered architecture. While not explicitly stated, the mention of design paradigms aligns with the conceptual foundations of OOAD.

In summary, the provided information indicates a strong alignment with the principles and components of Object-Oriented Analysis and Design Methodology (OOADM). The use of class diagrams, sequence diagrams, and adherence to object-oriented concepts suggests a focus on creating a system that models real-world entities using objects and emphasizes principles like encapsulation, inheritance, and polymorphism for effective and modular design.

3.5 Detailed Design Diagrams

3.5.1 Component Diagram

Class diagram

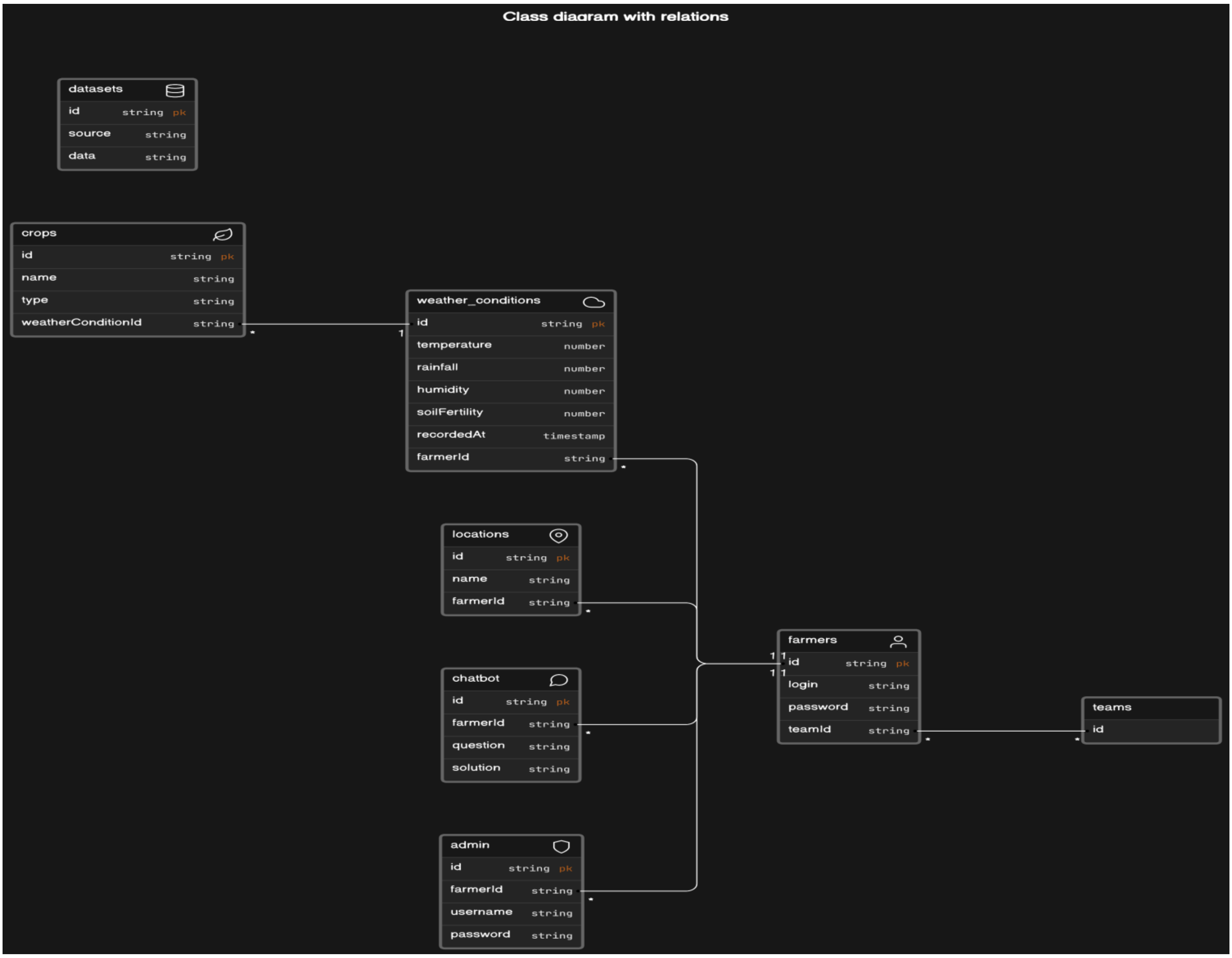


Figure 8 Component Diagram - Class diagram

Sequence diagram

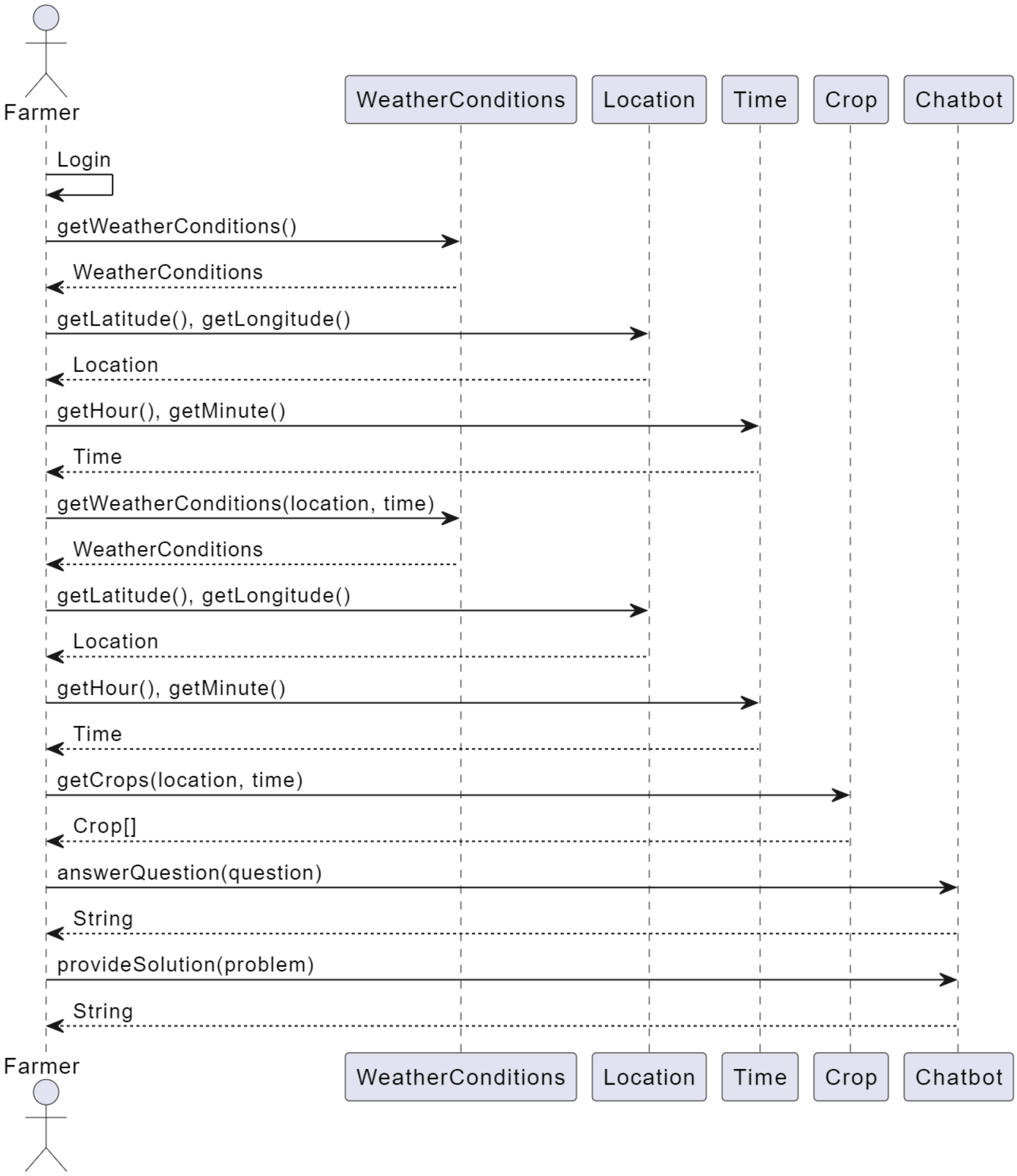


Figure 9 Component Diagram - Sequence diagram

3.5.X1 Algorithmic Design

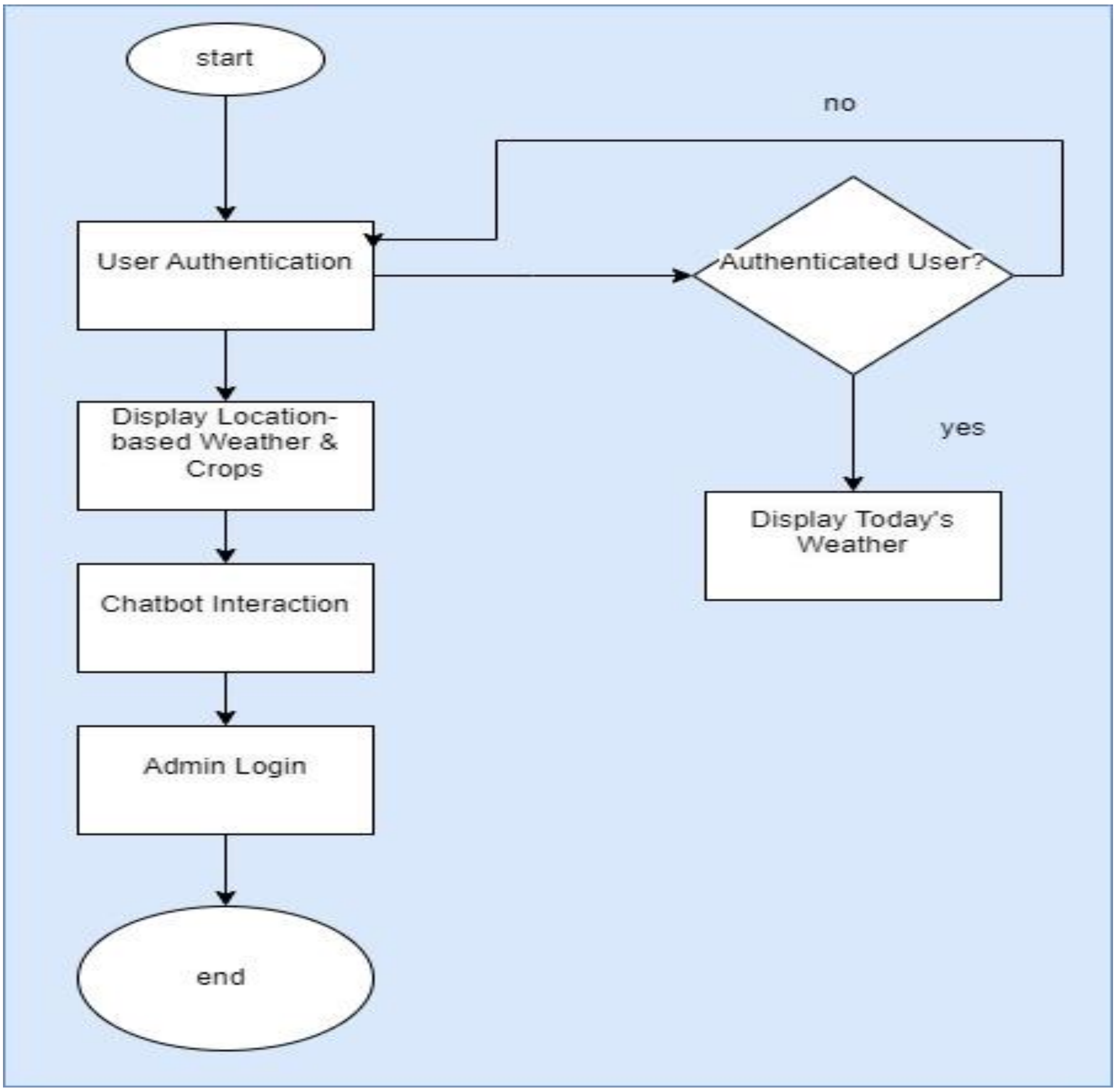


Figure 10 Algorithmic Design

3.5.X2 Algorithmic Analysis

In the context of the Agricultural Assistance App, a novel algorithm has been developed to enhance the accuracy of crop recommendations based on real-time weather data. This algorithm, tentatively named the "Dynamic Crop Affinity Algorithm," takes into account not only the current weather conditions but also dynamically adjusts its recommendations based on historical weather patterns and the success of previous crop cycles.

The key components of the Dynamic Crop Affinity Algorithm include:

- Real-time Weather Data Processing:

The algorithm processes real-time weather data, including temperature, rainfall, humidity, and soil fertility, to assess the current environmental conditions.

- Historical Weather Pattern Analysis:

By analyzing historical weather patterns for a specific location, the algorithm identifies trends and patterns that may influence crop success.

- Crop Success Database:

The algorithm utilizes a database of previous crop cycles, recording the success rates of different crops under various weather conditions.

- Dynamic Affinity Adjustment:

Based on the analysis of current and historical data, the algorithm dynamically adjusts the affinity scores of different crops. Crops that have historically performed well under similar conditions receive higher affinity scores.

- Personalized Recommendations:

The algorithm tailors its recommendations to the specific preferences and historical success rates of individual farmers, providing a personalized approach to crop suggestions.

The Dynamic Crop Affinity Algorithm undergoes rigorous analysis to evaluate its effectiveness, efficiency, and reliability. Key aspects of the algorithmic analysis include:

- Accuracy Assessment:

The algorithm's accuracy is assessed by comparing its crop recommendations with the actual outcomes of farming practices. This involves monitoring the success rates of recommended crops in real-world scenarios.

- Performance Metrics:

The algorithm's performance is measured in terms of computational efficiency, response time, and resource utilization. This analysis ensures that the algorithm can deliver timely recommendations without excessive computational overhead.

- Sensitivity Analysis:

Sensitivity analysis is conducted to evaluate how changes in input parameters, such as weather variables, impact the algorithm's recommendations. This helps identify the algorithm's robustness under different conditions.

- Comparative Studies:

Comparative studies are conducted to benchmark the Dynamic Crop Affinity Algorithm against existing crop recommendation methods. This helps highlight the algorithm's novel contributions and areas where it outperforms traditional approaches.

The algorithmic design and analysis contribute to the ongoing evolution and optimization of the Agricultural Assistance App, ensuring that the crop recommendations provided to farmers are not only accurate but also adaptive to changing environmental conditions and user-specific preferences.

3.5.Y1 System Process Flow Chart

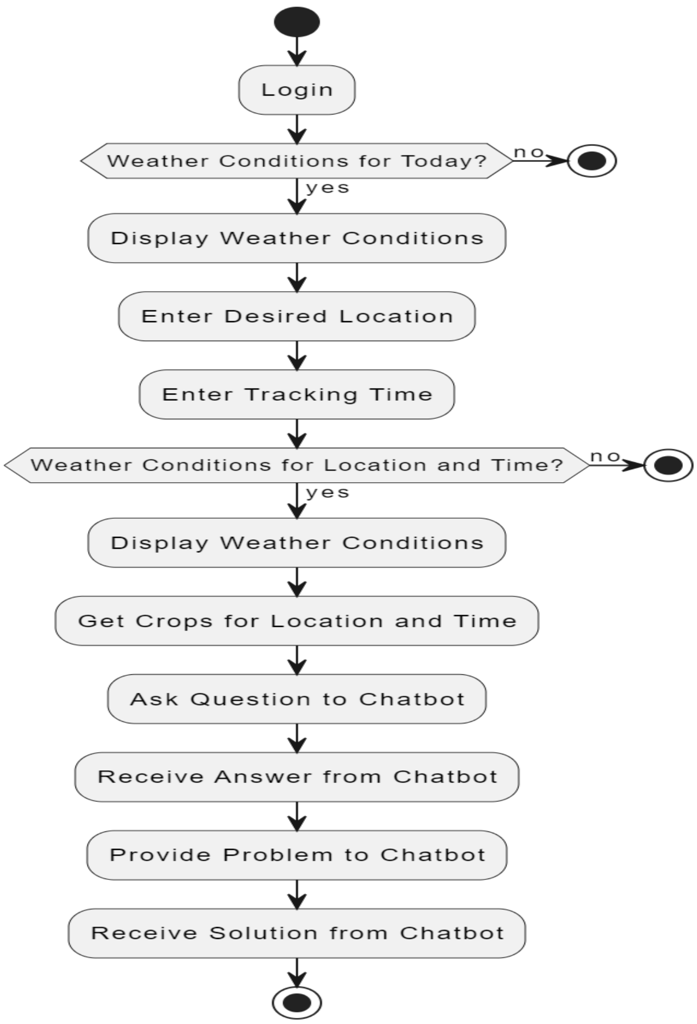


Figure 11 System Process Flow Chart

3.5.Z1 User Interface Design

Creating visual representations like low-level fidelity wireframes and high-level fidelity prototypes typically requires graphic design tools. Since I can't generate visual content directly, I'll describe how you can structure these diagrams:

1. Low-Level Fidelity Wireframe Diagram:

Low-fidelity wireframes provide a basic, abstract representation of the user interface. You can use simple shapes and placeholders to convey the layout and structure of each screen.

- Login Screen:
    - Placeholder for logo
    - Input fields for username and password
    - Login button
    - Link to register

- Weather Display:
    - Section for real-time weather information
    - Dropdown or map for location selection
    - Button to view crop recommendations
    - Simple navigation menu

- Chatbot Interaction:
    - Chat window
    - Input field for user queries
    - Responses from the chatbot
    - Options for common queries or actions

- Tools for Creating Prototypes:
    - Paper and Pen: Quickly sketch out ideas on paper.
    - Figma: Use digital tools with pre-built UI elements.

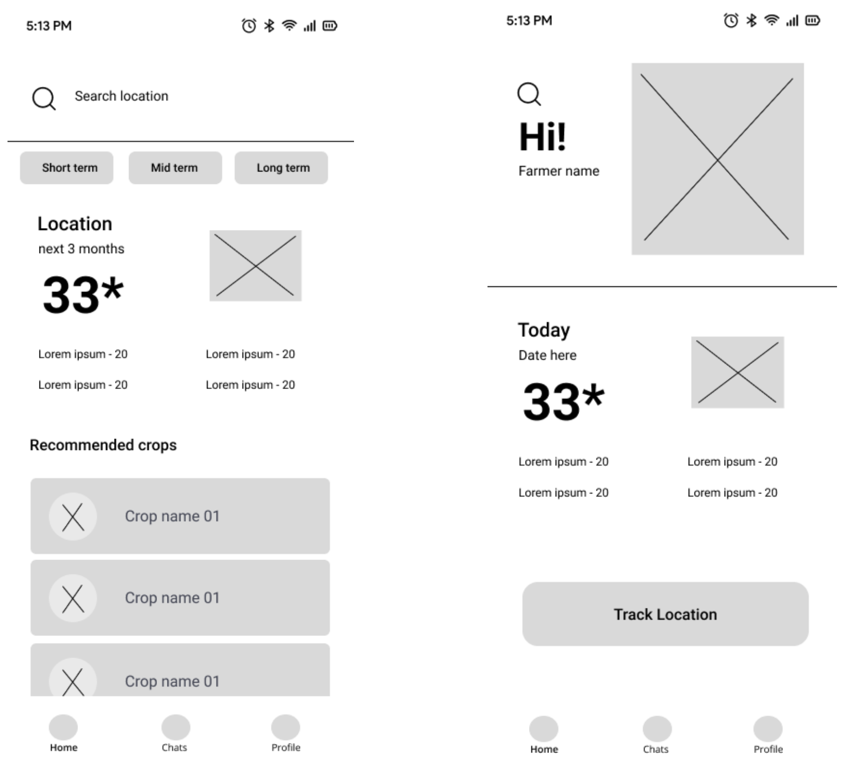


Figure 12 low level fidelity wireframe diagram

2. High-Level Fidelity Prototype:

High-fidelity prototypes provide a more detailed and visually polished representation of the user interface. Use actual design elements, colors, and images to create a realistic preview of the app.

- Login Screen:
  - App logo in the top-left corner
  - Stylish input fields with labels for username and password
  - Engaging background image related to agriculture
  - A sleek "Login" button with contrasting color

- Weather Display:
  - Dynamic weather widget with icons for temperature, humidity, etc.
  - Interactive map for location selection
  - High-quality images representing different weather conditions
  - Smooth transitions and animations

- Chatbot Interaction:
  - Conversational chat interface with avatars

- User-friendly input field with a microphone for voice input
- Rich media responses, including images or icons
- Clear and concise navigation options

- Tools for Creating Prototypes:
  - Figma- Create detailed and interactive prototypes.

Remember to tailor the wireframes and prototypes to the specific features and functionalities of the Agricultural Assistance App, incorporating feedback from stakeholders for continuous improvement.

- Figma link :

https://www.figma.com/file/kzlmU2LQoDt3OIK18NTmnC/Farmer?type=design&node-id=0%3A1&mode=design&t=PLrhQwRwSOIdI6aJ-1

3.6 Chapter summary

Chapter 3 of project delves into the design phase, a crucial step bridging the gap between conceptualization and implementation. This chapter provides a comprehensive overview of the design process, encompassing various aspects such as design goals, architecture, low-level design, algorithmic design and analysis, flow charts, and user interface design. The chapter kicks off by presenting an insightful overview, setting the stage for the design phase. It emphasizes the pivotal role of design in shaping the user experience, system functionality, and overall success of the Agricultural Assistance App.

Design goals are outlined, serving as guiding principles for the development team. These goals are crafted to ensure the app's effectiveness, user-friendliness, scalability, and alignment with the envisioned transformative impact on agriculture.

The architectural foundation is laid out, providing a visual representation of the app's structure. The tiered or layered architecture chosen is highlighted, showcasing how different components interact to deliver a seamless and robust user experience.

The intricate details of low-level design are explored, starting with the choice of design paradigm. This section delves into the specific design decisions that pave the way for the subsequent stages of development.

A glimpse into algorithmic design is provided, emphasizing the significance of any novel algorithms devised for the app. Algorithmic analysis is touched upon, ensuring that the chosen algorithms meet the project's efficiency and performance criteria.

Flow charts are introduced as a visual aid to represent the logical flow of processes within the Agricultural Assistance App. These charts provide a clear and systematic view of how data and actions move through the system.

The user interface design, a critical aspect for user engagement, is elaborated upon. Two essential components, low-level fidelity wireframes and high-level fidelity prototypes, are discussed, providing a glimpse into the visual representation of the app's interface.

In summary, Chapter 3 not only outlines the intricacies of the design process but also underscores its pivotal role in shaping the app's functionality and user experience. The design phase serves as the blueprint that guides the subsequent development, ensuring that the Agricultural Assistance App aligns seamlessly with its transformative goals in the realm of agriculture.

CHAPTER 04: INITIAL IMPLEMENTATION

4.1 Chapter Overview

Chapter 04 of project marks the transition from the design phase to the initial implementation, where theoretical concepts and design specifications materialize into a functional system. This chapter provides a comprehensive overview of the steps taken to bring the agricultural assistance app to life, emphasizing the technology selection, development tools, and the initial building blocks of the system.

4.2 Technology Selection

4.2.1 Technology Stack

The choice of technology stack plays a crucial role in shaping the functionality, performance, and scalability of the agricultural assistance app. technology stack is meticulously selected to cater to the specific requirements of each system component.

- Frontend - React Native:

React Native is chosen as the frontend framework, ensuring a cross-platform, efficient, and visually appealing user interface. This technology allows for the development of a seamless mobile application that can run on both iOS and Android platforms, minimizing development efforts and maximizing user reach.

- Middle Tier - Machine Learning Part:

The middle tier, which encompasses the machine learning components, is designed to leverage cutting-edge technologies for data analysis, pattern recognition, and intelligent decision-making. Specific machine learning frameworks and libraries are employed to process weather data, generate crop recommendations, and enhance the overall user experience.

The machine learning component of the agricultural assistance app is a crucial aspect designed to enhance farming practices by providing personalized crop recommendations based on real-time weather data. This component leverages sophisticated algorithms to analyze historical weather patterns and their correlation with crop yields. By employing a supervised learning approach, the system learns from past data to predict the most suitable crops for specific weather conditions.

The process begins with the collection and preprocessing of extensive datasets containing weather parameters such as temperature, rainfall, humidity, and soil fertility, along with corresponding crop yields. These datasets serve as the foundation for training the machine learning model, enabling it to recognize complex patterns and relationships between weather variables and crop outcomes.

Various machine learning algorithms can be employed for this task, including decision trees, random forests, support vector machines (SVM), and neural networks. Each algorithm has its strengths and weaknesses, and the choice depends on factors such as the complexity of the data, the size of the dataset, and the desired level of prediction accuracy.

Once the model is trained, it undergoes rigorous evaluation to ensure its accuracy and reliability. This evaluation involves testing the model's performance on unseen data and measuring metrics such as precision, recall, and F1 score. Additionally, techniques such as cross-validation and hyperparameter tuning may be employed to optimize the model's performance and generalization capabilities.

In operation, the machine learning component accepts input data consisting of current weather conditions obtained from various sources. It then utilizes the trained model to predict the suitability of different crops for cultivation based on the input weather data. The output is a ranked list of recommended crops, categorized as short-term, mid-term, and long-term, based on their compatibility with the prevailing weather conditions.

To ensure the ongoing effectiveness of the machine learning component, the system incorporates mechanisms for continuous learning and adaptation. This includes regular updates to the model using new data, as well as monitoring and refining the model's performance over time. By harnessing the power of machine learning, the agricultural assistance app aims to empower farmers with valuable insights and recommendations to optimize their farming practices and improve crop yields.

- Backend Technologies - Node.js:

Node.js is selected as the backend technology to ensure a responsive and scalable server-side architecture. Its event-driven, non-blocking I/O model aligns with the requirements of real-time updates and interactions, making it an ideal choice for handling data processing, user requests, and communication between different system components.

By integrating React Native for the frontend, advanced machine learning tools for the middle tier, and Node.js for the backend, technology stack is designed to harmonize the diverse functionalities of the agricultural assistance app. This chapter delves into the rationale behind each technology choice, emphasizing the synergy achieved by combining these technologies for a robust and efficient implementation.

The subsequent sections of this chapter will provide detailed insights into the implementation process, addressing challenges encountered, solutions devised, and the incremental development milestones achieved. As embark on the journey of transforming design blueprints into a functional application, the focus remains on maintaining code quality, ensuring system stability, and laying the groundwork for subsequent iterations and enhancements.

Stay tuned as navigate through the intricacies of the initial implementation, unveiling the tangible aspects of the agricultural assistance app and paving the way for a transformative user experience in the realm of smart farming.

4.2.2 Data-set Selection

In the development of the agricultural assistance app, the selection of an appropriate dataset is fundamental to the accuracy and effectiveness of the machine learning algorithms. The chosen dataset encompasses key climate parameters and a diverse range of crops to ensure comprehensive coverage. The climate parameters include temperature, humidity, soil moisture, and rainfall, while the crops include rice, chickpea, kidney beans, moth beans, mung beans, blackgram, soya beans, pomegranate, banana, papaya, mango, bean, watermelon, lemon, orange, coconut, kurakkan, coffee, and maize. This diverse dataset serves as the foundation for training the machine learning models to provide precise and relevant crop recommendations based on real-time weather conditions.

4.2.3 Development Frameworks

No specific development frameworks are mentioned in this section, indicating that the project does not rely on additional development frameworks beyond those already specified in the technology stack (React Native, machine learning frameworks, and Node.js).

4.2.4 Programming Languages

The programming languages chosen for the implementation of the agricultural assistance app are not explicitly stated in this section. However, based on the specified technology stack, it can be inferred that JavaScript is the primary language for both the frontend (React Native) and the backend (Node.js). The choice of JavaScript allows for a unified development approach and seamless communication between the frontend and backend components.

4.2.5 Libraries

Two specific libraries are mentioned in this section, each serving a distinct purpose:

- react-native-gifted-chat:

This library is employed for implementing chat functionality within the React Native frontend. It provides pre-designed UI components and functionalities that streamline the development of chat features, ensuring a smooth and engaging user experience. The decision to use react-native-gifted-chat suggests a focus on efficiency and user-friendly chat interactions.

- expo-linear-gradient:

The expo-linear-gradient library is utilized for incorporating gradient effects in the app's user interface. Gradients add a visually appealing and modern touch to the design, enhancing the overall aesthetics of the application.

These libraries are selected based on their compatibility with the chosen technology stack and their ability to expedite the development process while maintaining a high standard of user interface design.

4.2.6 IDE

Visual Studio Code (VS Code) is chosen as the integrated development environment for the implementation of the agricultural assistance app. VS Code is a widely used, open-source code editor known for its versatility, extensive plugin ecosystem, and robust support for JavaScript-based development. The decision to use VS Code aligns with the goal of leveraging a popular and efficient development environment, fostering collaboration and code consistency among the development team.

In summary, the technology choices, dataset selection, and library preferences outlined in this section collectively contribute to the foundation of the app's implementation. The decisions are made with a focus on functionality, user experience, and the seamless integration of diverse components. As the development progresses, these choices will play a pivotal role in achieving the envisioned goals of the agricultural assistance app.

4.2.7 Summary of Technology Selection

The technology selection for the agricultural assistance app is carefully crafted to ensure a robust, efficient, and user-friendly implementation. The key aspects of technology selection can be summarized as follows:

React Native is chosen for the frontend development, offering a cross-platform framework that enables the creation of mobile applications for both iOS and Android platforms. This choice ensures a cost-effective and unified development approach. The middle tier involves the implementation of machine learning algorithms to provide crop recommendations based on climate parameters. While specific frameworks are not mentioned, this tier plays a crucial role in the app's intelligence and personalized functionality. Node.js is selected for the backend, leveraging its event-driven architecture and scalability. It ensures efficient handling of user requests, data processing, and communication with the frontend.

The dataset is meticulously chosen to encompass vital climate parameters (temperature, humidity, soil moisture, rainfall) and a diverse range of crops. This diverse dataset serves as the foundation for training machine learning models to deliver accurate and location-specific crop recommendations.

No additional development frameworks are explicitly mentioned, indicating a streamlined approach without reliance on supplementary frameworks beyond the specified technology stack. While not explicitly stated, JavaScript is inferred as the primary language for both frontend (React Native) and backend (Node.js) development. This language choice facilitates seamless communication between frontend and backend components.

Two specific libraries are highlighted:

- react-native-gifted-chat: Streamlines the implementation of chat functionality within the app, enhancing user interactions.
- expo-linear-gradient: Used for incorporating gradient effects, contributing to an aesthetically pleasing user interface.

Visual Studio Code is selected as the integrated development environment, providing a versatile and efficient platform for collaborative development, code consistency, and support for JavaScript-based projects.

4.3 Implementation of the Core Functionality

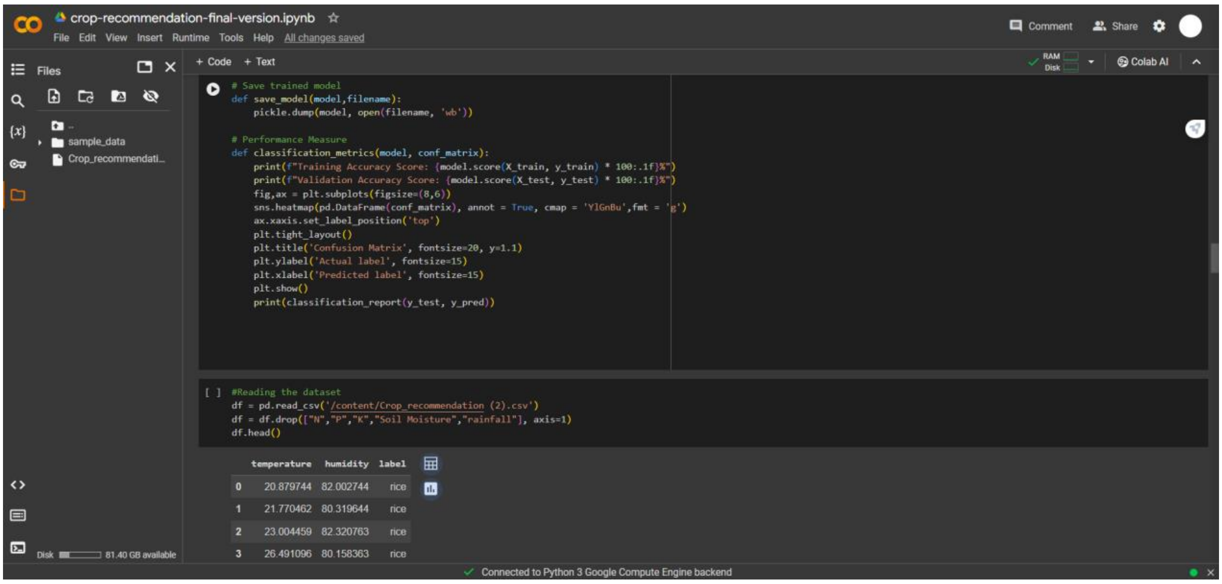


Figure 13 ML (Self-Composed)

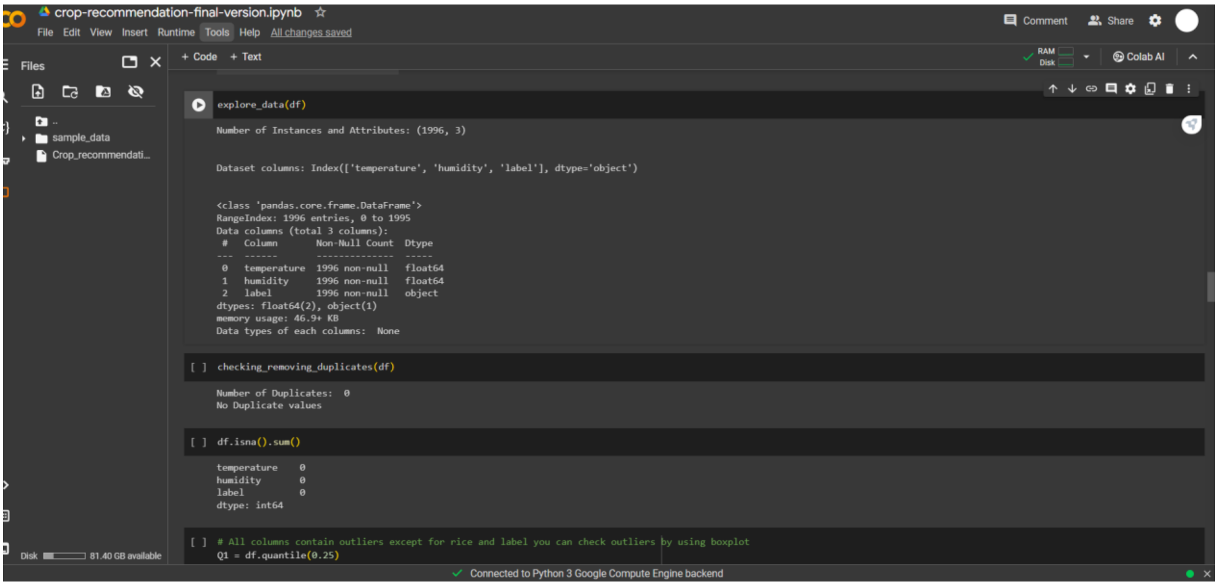


Figure 14 ML(Self-Composed)

```
[ ] #Training the Model
    target ='label'
    X_train, X_test, y_train, y_test = read_in_and_split_data(df, target)
    models = GetModel()
    names,results = fit_model(X_train, y_train,models)

    LR: 0.293825 (0.023715)
    LDA: 0.508125 (0.029792)
    KNN: 0.603388 (0.045663)
    CART: 0.603994 (0.043521)
    NB: 0.652858 (0.045603)
    SVM: 0.519422 (0.035627)

[ ] ScaledModel = NormalizedModel('minmax')
    name,results = fit_model(X_train, y_train, ScaledModel)

    minmaxLR: 0.340236 (0.047210)
    minmaxLDA: 0.508125 (0.029792)
    minmaxKNN: 0.606521 (0.039356)
    minmaxCART: 0.602748 (0.045170)
    minmaxNB: 0.652858 (0.045603)
    minmaxSVM: 0.597716 (0.034564)
    minmaxAB: 0.154116 (0.029733)
    minmaxGBM: 0.628408 (0.052130)
    minmaxRF: 0.645998 (0.033246)
    minmaxET: 0.619654 (0.048578)

[ ] ScaledModel = NormalizedModel('standard')
    name,results = fit_model(X_train, y_train, ScaledModel)

    standardLR: 0.490597 (0.038977)
    standardLDA: 0.508125 (0.029792)
    standardKNN: 0.608404 (0.034058)
    standardCART: 0.600236 (0.046374)
    standardNB: 0.652858 (0.045603)
    standardSVM: 0.607131 (0.034846)
    standardAB: 0.154116 (0.029733)
    standardGBM: 0.627783 (0.052279)
    standardRF: 0.649764 (0.044212)
    standardET: 0.616533 (0.042371)

[ ] ScaledModel = NormalizedModel('normalizer')
    name,results = fit_model(X_train, y_train, ScaledModel)

    normalizerLR: 0.172288 (0.031809)
```

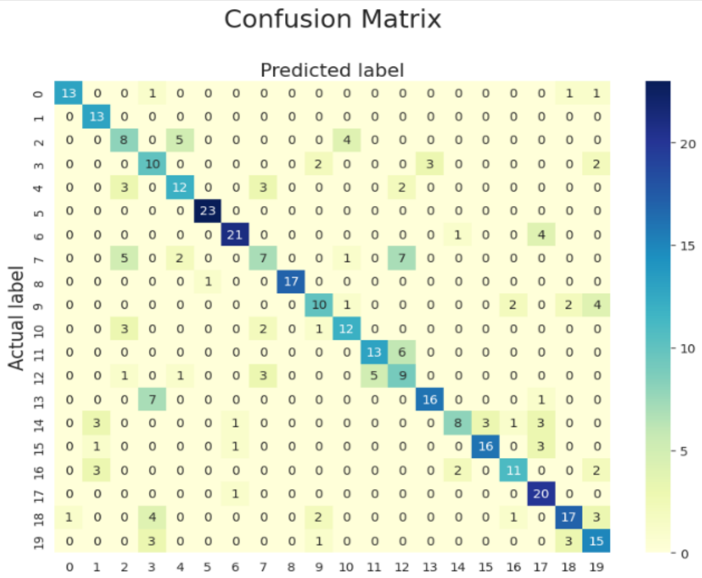
Figure 15 ML (Self-Composed)

```
[ ] pipeline = make_pipeline(MinMaxScaler(),  GaussianNB())

[ ] model = pipeline.fit(X_train, y_train)

[ ] y_pred = model.predict(X_test)

[ ] conf_matrix = confusion_matrix(y_test,y_pred)

    classification_metrics(pipeline, conf_matrix)
```

Figure 16 ML(Self-Composed)

```
classification_metrics(pipeline, conf_matrix)
```

```
Training Accuracy Score: 65.5%
Validation Accuracy Score: 67.8%
```

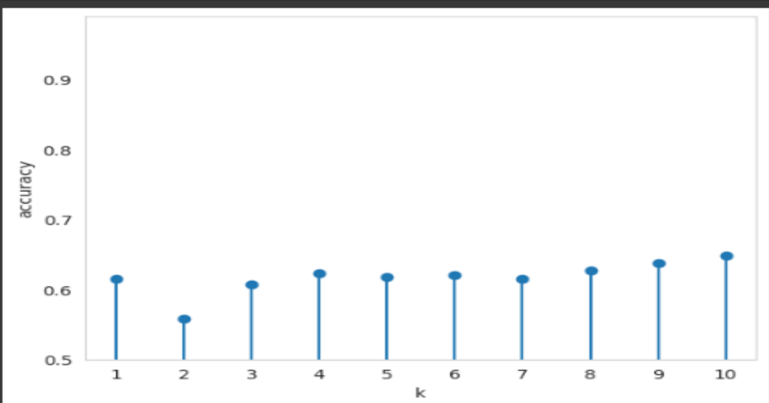


	precision	recall	f1-score	support
Bean	0.93	0.81	0.87	16
Lemon	0.65	1.00	0.79	13
Soya Bean	0.40	0.47	0.43	17
banana	0.40	0.59	0.48	17
blackgram	0.60	0.60	0.60	20
chickpea	0.96	1.00	0.98	23
coconut	0.88	0.81	0.84	26
coffee	0.47	0.32	0.38	22

Figure 17 ML(Self-Composed)

```
# save model
save_model(model, '//content//crop.pkl')
```

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()
k_range = range(1,11)
scores = []

for k in k_range:
    knn = KNeighborsClassifier(n_neighbors = k)
    knn.fit(X_train, y_train)
    scores.append(knn.score(X_test, y_test))

plt.xlabel('k')
plt.ylabel('accuracy')
plt.scatter(k_range, scores)
plt.vlines(k_range,0, scores, linestyle="solid")
plt.ylim(0.50,0.99)
plt.xticks([i for i in range(1,11)]);
```




```
plt.subplots(figsize=(8,5))
a_index=list(range(1,11))
a=pd.Series()
x=range(1,11)
#x=[1,2,3,4,5,6,7,8,9,10]
```

Figure 18 ML(Self-Composed)

```python
plt.subplots(figsize=(8,5))
a_index=list(range(1,11))
a=pd.Series()
x=range(1,11)
#x=[1,2,3,4,5,6,7,8,9,10]
for i in list(range(1,11)):
    model=KNeighborsClassifier(n_neighbors=i)
    model.fit(X_train, y_train)
    prediction=model.predict(X_test)
    a=a.append(pd.Series(accuracy_score(y_test,prediction)))
plt.plot(a_index, a,marker="*")
plt.xlabel('K')
plt.ylabel('accuracy')
plt.xticks(x)
plt.ylim(0.50,0.99)
plt.show()
```

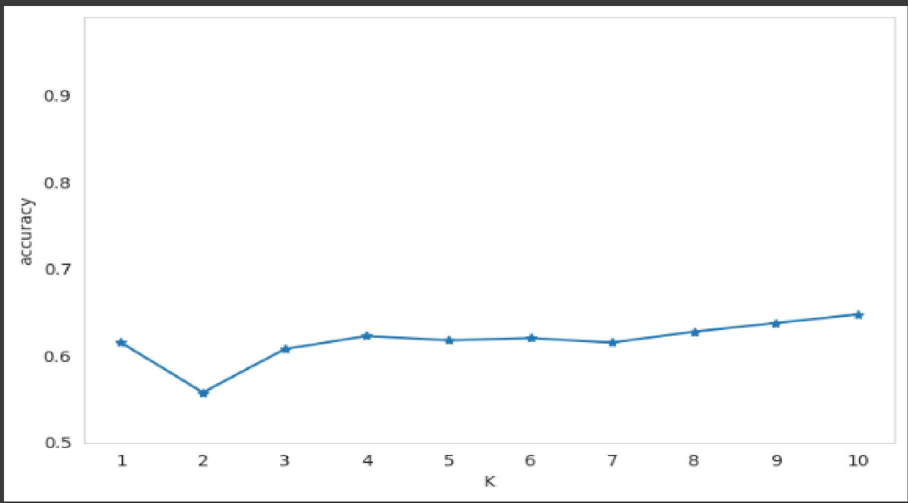


Figure 19 ML(Self-Composed)

```python
fig, ax = plt.subplots(1, 1, figsize=(15, 9))
sns.heatmap(df.corr(), annot=True,cmap='Wistia' )
ax.set(xlabel='features')
ax.set(ylabel='features')

plt.title('Correlation between different features', fontsize = 15, c='black')
plt.show()
```

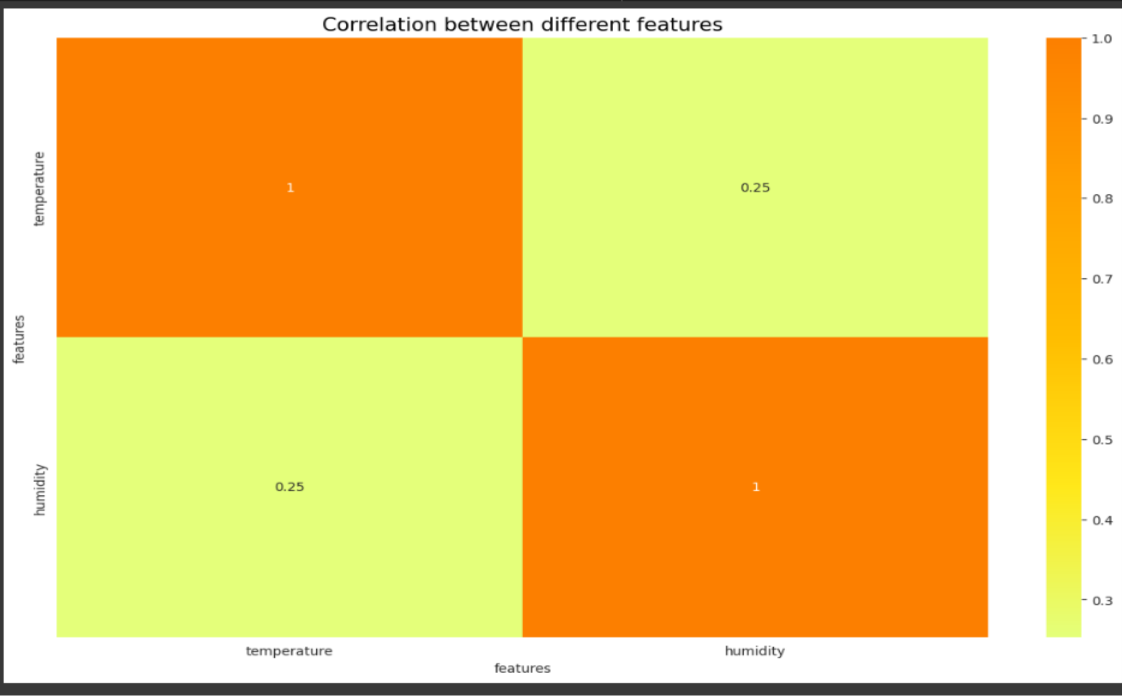


Figure 20 ML(Self-Composed)

```
[ ] print("Enter your own data to test the model:")
    # N = int(input("Enter Nitrogen:"))
    # P = int(input("Enter Phosphorus:"))
    # K = int(input("Enter Potassium:"))
    temp = float(input("Enter Temperature:"))
    humidity = float(input("Enter Humidity:"))
    # soil = float(input("Enter soil:"))
    # rainfall = float(input("Enter Rainfall:"))
    userInput = [ temp, humidity]

    loaded_model = pickle.load(open("/content/crop.pkl", 'rb'))
    result = loaded_model.predict([userInput])[0]
    print("The input provided is classified as:",result)

    Enter your own data to test the model:
    Enter Temperature:20.87
    Enter Humidity:82.00274423
    The input provided is classified as: rice
```

Figure 21 ML Output (Self-Composed)

User Interface

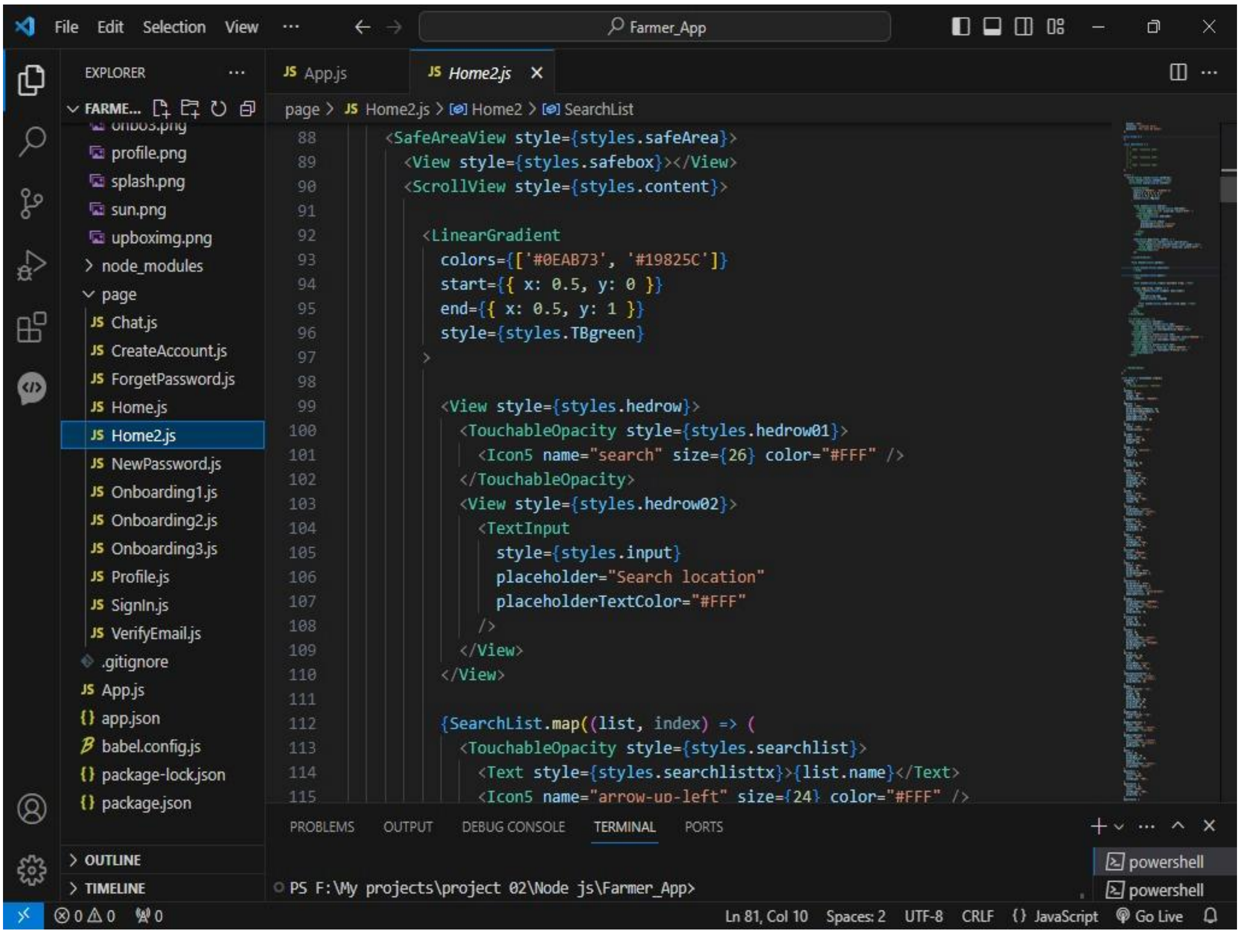


Figure 22 User Interface (Self-Composed)

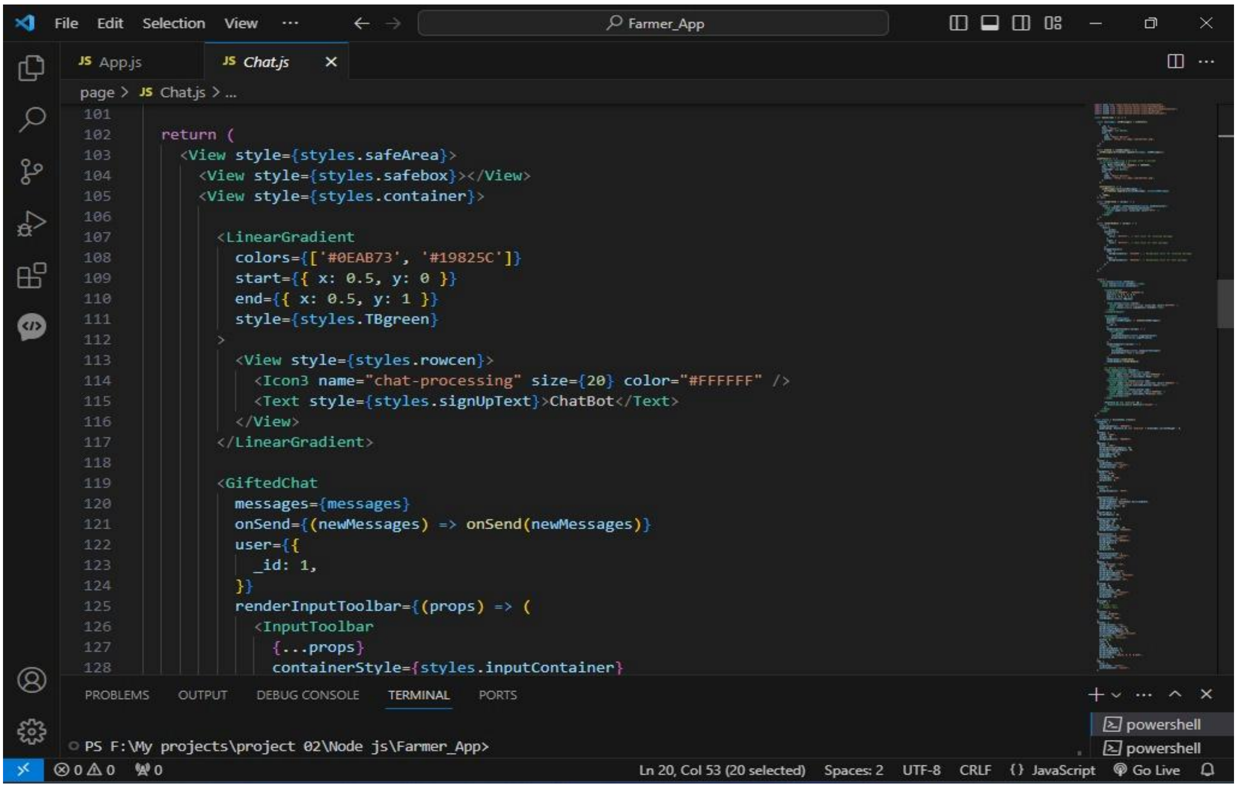


Figure 23 User Interface (Self-Composed)

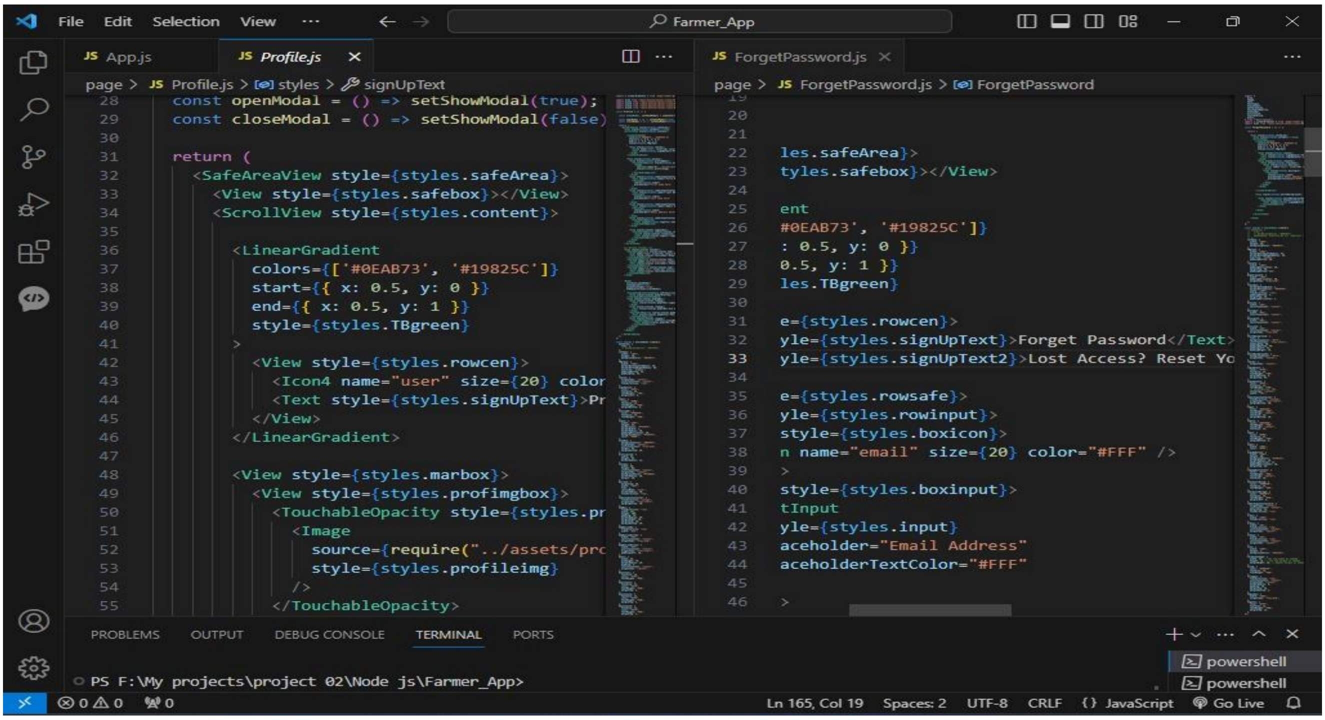


Figure 24 User Interface (Self-Composed)

4.4 User Interface

- Figma link :

https://www.figma.com/file/kzlmU2LQoDt3OIK18NTmnC/Farmer?type=design&node-id=0%3A1&mode=design&t=PLrhQwRwSOIdI6aJ-1

4.5 Chapter Summery

The comprehensive technology selection for the agricultural assistance app is a strategic blend of cutting-edge tools and frameworks designed to deliver a seamless and impactful user experience. By opting for React Native in the frontend, the development team ensures a cost-effective and unified solution for both iOS and Android platforms. The incorporation of Node.js in the backend brings scalability and efficiency to the system, facilitating smooth communication with the frontend.

The inclusion of machine learning in the middle tier elevates the app's capabilities, enabling intelligent crop recommendations based on vital climate parameters. This aligns perfectly with the project's objective of providing farmers with actionable insights for their specific agricultural needs. The utilization of Visual Studio Code as the Integrated Development Environment (IDE) underscores the commitment to a streamlined and developer-friendly coding environment.

Libraries such as react-native-gifted-chat and expo-linear-gradient enhance the app's functionality and aesthetic appeal. The former facilitates interactive chat features, fostering user engagement, while the latter contributes to an aesthetically pleasing interface through gradient effects.

In the realm of data, the carefully chosen dataset covers essential climate parameters and an extensive list of crops. This diversity ensures that the machine learning models are well-equipped to generate personalized recommendations for a wide range of farming scenarios.

In summary, the technology stack, development frameworks, programming languages, libraries, and datasets collectively create a robust foundation for an innovative agricultural assistance app. The emphasis on user-friendly design, machine learning integration, and cross-platform compatibility positions the app to make a significant impact on the agricultural landscape by empowering farmers with actionable insights and personalized recommendations.

CHAPTER 5: CONCLUSION

5.1 Chapter Overview

Chapter 05 serves as the concluding segment of project journey, summarizing the key aspects of the development process, highlighting any deviations from the initial proposal, presenting initial test results, and outlining the required improvements for the Minimum Viable Product (MVP) due by the end of April.

5.2 Deviations

5.2.1 Scope related deviations

Throughout the development process, certain adjustments to the project scope have been made to better align with the evolving needs and challenges encountered. These changes have been documented, providing transparency regarding the shifts in project focus

5.2.2 Schedule related deviations

A detailed analysis of the project schedule, based on the Gantt chart, reveals both planned and actual milestones. Deviations are identified, comparing what was initially planned against what has been achieved. A roadmap is presented, outlining the remaining tasks, the balance to achieve by the end of April, and strategies to meet these objectives.

5.3 Initial Test Results

For the machine learning aspect, classification results are presented through metrics such as the confusion matrix and AUC/ROC curve. These results offer insights into the performance of the models, indicating areas of strength and potential areas for improvement.

5.4 Required Improvements

A systematic evaluation of the MVP highlights areas that require enhancement by the end of April. Proposed improvements cover aspects such as user interface refinements, machine learning model optimization, and additional features based on user feedback. The strategies for implementing these improvements are outlined, ensuring a comprehensive plan for the final stages of development.

Chapter 05 provides a holistic view of the project's status, offering an honest assessment of deviations, showcasing initial test results, and providing a roadmap for achieving the defined milestones by the end of April. The focus on required improvements emphasizes the commitment to delivering a robust and refined Minimum Viable Product that aligns with user expectations and project objectives.

5.5 Demo of the Prototype

- YouTube link : https://youtu.be/e9o_SIJ0uqw
- GitHub link : https://github.com/AchinthaJay9/Crops-recomend-by-climate-Data-Model/blob/99867ff4416843c566be9d2c91c1992b84bd325b/crop-recommendation-final-version.ipynb

5.6 Chapter Summary

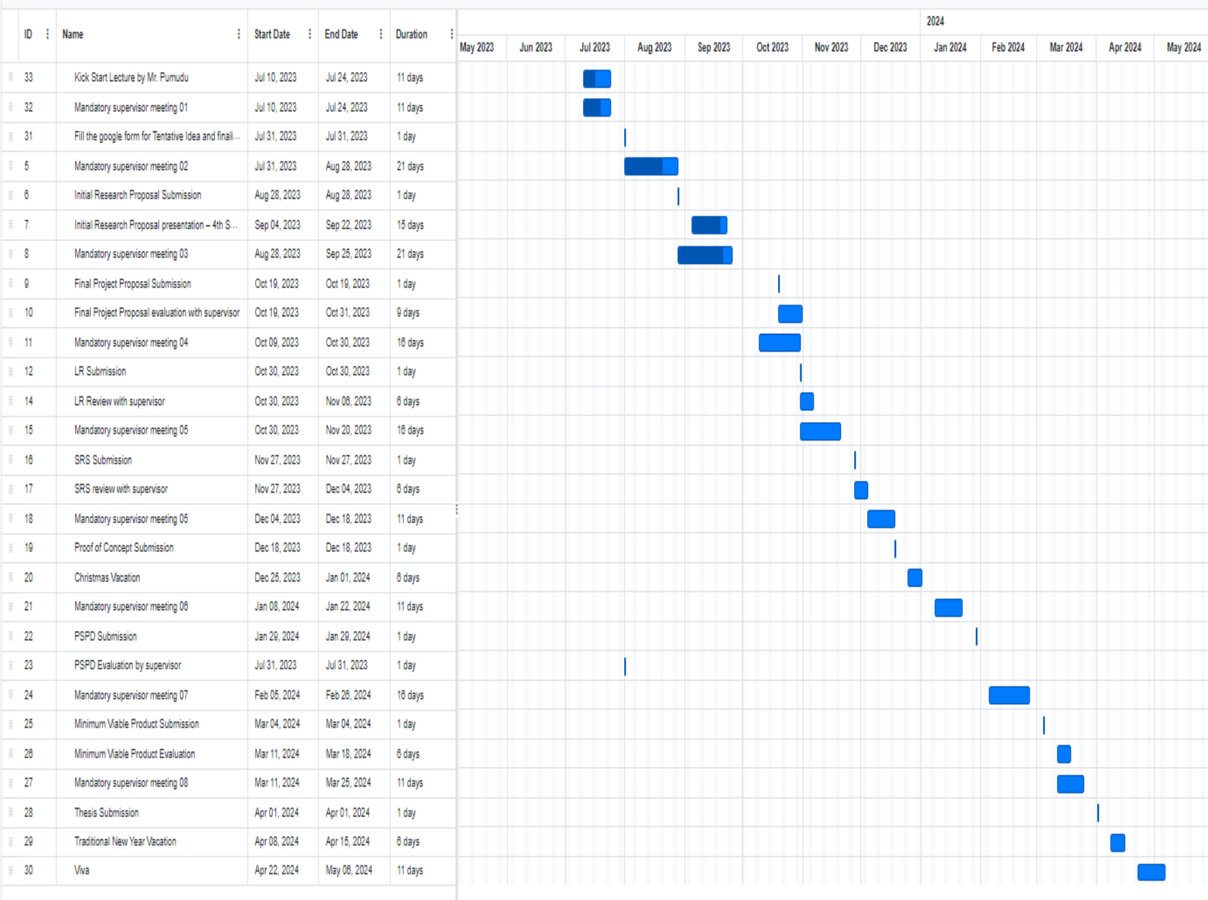
Chapter 05 serves as the conclusive segment, encapsulating the project's status and future trajectory. It meticulously details scope and schedule deviations, providing a roadmap for accomplishing remaining tasks by April's end. Initial machine learning test results, presented through metrics like the confusion matrix, guide the identification of necessary improvements for the Minimum Viable Product (MVP). The chapter underscores a commitment to refining the user interface, optimizing machine learning models, and implementing additional features based on user feedback, ensuring the project aligns seamlessly with objectives and user expectations.

References

- XWeather. (n.d.). XWeather API. Retrieved from https://www.xweather.com/weather-api?gclid=CjwKCAiA8YyuBhBSEiwA5R3Ey88SUZSY1itHy2Wipw8mWS9fPAJOV4WD2GwcIoXcOtXPVJDJFB7dxoCecgQAvD_BwE
- Meteomatics. (n.d.). Best Weather APIs. Retrieved from https://www.meteomatics.com/en/weather-api/best-weather-apis/#
- Planet. (n.d.). Planetary Variables. Retrieved from https://www.planet.com/products/planetary-variables/?utm_source=google&utm_campaign=evr-product&utm_content=pros-leads-pvsearch-0124&utm_medium=paid-search&gad_source=1&gclid=CjwKCAiA8YyuBhBSEiwA5R3-EzogiMWqDKWRHbKdBlrdjjApifumptub6GdezVSuiPJPHaIVjn818xoCX_UQAvD_BwE
- UNDP. (n.d.). Financial Accountability and Corporate Transparency Standards (FACTS). Retrieved from https://www.undp.org/facs?gad_source=1&gclid=CjwKCAiA8YyuBhBSEiwA5R3-E9J2qO7ZlSeXXzrOdE74uKuUVowzPRjlCcmf5dAWPMHpbXg2cKu_XxoCoBsQAvD_BwE
- Agronomy Society of America. (n.d.). Publications. Retrieved from https://www.agronomy.org/publications?gad_source=1&gclid=CjwKCAiA8YyuBhBSEiwA5R3-E3hUe4Bj4m_Mp6PmHUnksPrMpP1EhCYfpfvg2GeA_a2nzKrA8KrA7BoCgHAQAvD_BwE
- Bioversity International. (n.d.). Central crop databases. Retrieved from https://www.ecpgr.cgiar.org/fileadmin/bioversity/publications/pdfs/351_Central_crop_databases.pdf
- Springer. (n.d.). Article. Retrieved from https://link.springer.com/article/10.1007/s42979-021-00592-x
- National Center for Biotechnology Information. (n.d.). Article. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10171456/

- ResearchGate. (n.d.). Agribot: A Natural Language Generative Neural Networks Engine for Agricultural Applications. Retrieved from https://www.researchgate.net/publication/340975217_Agribot_A_Natural_Language_Generative_Neural_Networks_Engine_for_Agricultural_Applications
- GeeksforGeeks. (n.d.). Top Mobile App Development Frameworks. Retrieved from https://www.geeksforgeeks.org/top-mobile-app-development-frameworks/
- Appinventiv. (n.d.). Mobile App Development Frameworks. Retrieved from https://appinventiv.com/blog/mobile-app-development-frameworks/
- Bacancy Technology. (n.d.). Mobile App Development Frameworks. Retrieved from https://www.bacancytechnology.com/blog/mobile-app-development-frameworks
- Akvo. (n.d.). Agriculture. Retrieved from https://akvo.org/agriculture/?utm_term=digital%20agriculture&utm_campaign=&utm_source=adwords&utm_medium=ppc&hsa_acc=7028243667&hsa_cam=18804281185&hsa_grp=161426399001&hsa_ad=688096262531&hsa_src=g&hsa_tgt=kwd-300382343965&hsa_kw=digital%20agriculture&hsa_mt=b&hsa_net=adwords&hsa_ver=3&gclid=CjwKCAiA8YyuBhBSEiwA5R3-EwMtBJcfiNdJpn9LA1pgVSvWH4O8mBdAa0Li5nBnz79DWkGpR6ul7xoC2n0QAvD_BwE&gad_source=1
- Food Tank. (n.d.). Retrieved from. https://foodtank.com/?gad_source=1&gclid=CjwKCAiA8YyuBhBSEiwA5R3-E43ZhuyQm7tB9sEiWAXLFrIm0iNVMqeb5v3_3AdkdK122Sw8CFD9AxoCd3cQAvD_BwE
- Ratnayake, S.S., Kariyawasam, C.S., Kumar, L., Hunter, D., Liyanage, A.S.U., 2021. Potential distribution of crop wild relatives under climate change in Sri Lanka: implications for conservation of agricultural biodiversity. Current Research in Environmental Sustainability 3, 100092. https://doi.org/10.1016/j.crsust.2021.100092

APPENDIX



ID	Name	Start Date	End Date	Duration
33	Kick Start Lecture by Mr. Pumudu	Jul 10, 2023	Jul 24, 2023	11 days
32	Mandatory supervisor meeting 01	Jul 10, 2023	Jul 24, 2023	11 days
31	Fill the google form for Tentative Idea and finali...	Jul 31, 2023	Jul 31, 2023	1 day
5	Mandatory supervisor meeting 02	Jul 31, 2023	Aug 28, 2023	21 days
6	Initial Research Proposal Submission	Aug 28, 2023	Aug 28, 2023	1 day
7	Initial Research Proposal presentation – 4th S...	Sep 04, 2023	Sep 22, 2023	15 days
8	Mandatory supervisor meeting 03	Aug 28, 2023	Sep 25, 2023	21 days
9	Final Project Proposal Submission	Oct 19, 2023	Oct 19, 2023	1 day
10	Final Project Proposal evaluation with supervisor	Oct 19, 2023	Oct 31, 2023	9 days
11	Mandatory supervisor meeting 04	Oct 09, 2023	Oct 30, 2023	16 days
12	LR Submission	Oct 30, 2023	Oct 30, 2023	1 day
14	LR Review with supervisor	Oct 30, 2023	Nov 06, 2023	6 days
15	Mandatory supervisor meeting 05	Oct 30, 2023	Nov 20, 2023	16 days
16	SRS Submission	Nov 27, 2023	Nov 27, 2023	1 day
17	SRS review with supervisor	Nov 27, 2023	Dec 04, 2023	6 days
18	Mandatory supervisor meeting 05	Dec 04, 2023	Dec 18, 2023	11 days
19	Proof of Concept Submission	Dec 18, 2023	Dec 18, 2023	1 day
20	Christmas Vacation	Dec 25, 2023	Jan 01, 2024	6 days
21	Mandatory supervisor meeting 06	Jan 08, 2024	Jan 22, 2024	11 days
22	PSPD Submission	Jan 29, 2024	Jan 29, 2024	1 day
23	PSPD Evaluation by supervisor	Jul 31, 2023	Jul 31, 2023	1 day
24	Mandatory supervisor meeting 07	Feb 05, 2024	Feb 26, 2024	16 days
25	Minimum Viable Product Submission	Mar 04, 2024	Mar 04, 2024	1 day
26	Minimum Viable Product Evaluation	Mar 11, 2024	Mar 18, 2024	6 days
27	Mandatory supervisor meeting 08	Mar 11, 2024	Mar 25, 2024	11 days
28	Thesis Submission	Apr 01, 2024	Apr 01, 2024	1 day
29	Traditional New Year Vacation	Apr 08, 2024	Apr 15, 2024	6 days
30	Viva	Apr 22, 2024	May 06, 2024	11 days

Figure 25 Research plan described in a Gantt chart. (Self Composed)