“Smart Agriculture (Agri-Tech) to improve Yield and profitability”

A

Project Report

submitted

in partial fulfillment

for the award of the Degree of

Bachelor of Technology

in Department of Information Technology

Project Mentor:
Mrs. Richa Rawal
Associate Professor

Submitted By :
Aditya Agarwal, 20ESKIT005
Akshat Jaiman, 20ESKIT007
Abhishek Pandey, 20ESKIT004

Department of Information Technology
Swami Keshvanand Institute of Technology, M & G, Jaipur
Rajasthan Technical University, Kota

Session 2023-2024

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Department of Information Technology

CERTIFICATE

This is to certify that Mr. Aditya Agarwal, a student of B.Tech(Information Technology) 8th semester has submitted his Project Report entitled “Smart Agriculture (Agri-Tech) to improve Yield and profitability” under my guidance.

Mentor	**Coordinator**
Mrs. Richa Rawal	Mrs. Sanju Choudhary
Associate Professor	Associate Professor
Signature............	Signature............

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Department of Information Technology

CERTIFICATE

This is to certify that Mr. Akshat Jaiman, a student of B.Tech(Information Technology) 8th semester has submitted his Project Report entitled "Smart Agriculture (Agri-Tech) to improve Yield and profitability" under my guidance.

Mentor	**Coordinator**
Mrs. Richa Rawal	Mrs. Sanju Choudhary
Associate Professor	Associate Professor
Signature............	Signature............

Swami Keshvanand Institute of Technology, Management & Gramothan, Jaipur

Department of Information Technology

CERTIFICATE

This is to certify that Mr. Abhishek Pandey, a student of B.Tech(Information Technology) 8^{th} semester has submitted his Project Report entitled "Smart Agriculture (Agri-Tech) to improve Yield and profitability" under my guidance.

Mentor	**Coordinator**
Mrs. Richa Rawal	Mrs. Sanju Choudhary
Associate Professor	Associate Professor
Signature............	Signature............

DECLARATION

We hereby declare that the report of the project entitled "Smart Agriculture (Agri-Tech) to improve Yield and profitability" is a record of an original work done by us at Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur under the mentorship of "Mrs. Richa Rawal"(Dept. of Information Technology) and coordination of "Mrs. Sanju Choudhary" (Dept.of Information Technology). This project report has been submitted as the proof of original work for the partial fulfillment of the requirement for the award of the degree of Bachelor of Technology (B.Tech) in the Department of Information Technology. It has not been submitted anywhere else, under any other program to the best of our knowledge and belief.

Team Members	**Signature**
Aditya Agarwal, 20ESKIT005	
Akshat Jaiman, 20ESKIT007	
Abhishek Pandey, 20ESKIT004	

Acknowledgement

A project of such a vast coverage cannot be realized without help from numerous sources and people in the organization.We take this opportunity to express our gratitude to all those who have been helping us in making this project successful.

We are highly indebted to our faculty mentor Mrs. Richa Rawal. She has been a guide, motivator source of inspiration for us to carry out the necessary proceedings for the project to be completed successfully. We also thank our project coordinator Mrs. Sanju Choudhary for her co-operation, encouragement, valuable suggestions and critical remarks that galvanized our efforts in the right direction.

We would also like to convey our sincere thanks to Dr. Anil Chaudhary, HOD, Department of Information Technology, for facilitating, motivating and supporting us during each phase of development of the project. Also, we pay our sincere gratitude to all the Faculty Members of Swami Keshvanand Institute of Technology, Management and Gramothan, Jaipur and all our Colleagues for their cooperation and support.

Last but not least we would like to thank all those who have directly or indirectly helped and cooperated in accomplishing this project.

Team Members:

Aditya Agarwal, 20ESKIT005

Akshat Jaiman, 20ESKIT007

Abhishek Pandey, 20ESKIT004

Table of Content

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Problem Statement and Objective

The agricultural sector faces significant challenges in effectively diagnosing and managing crop diseases, leading to substantial economic losses and food security concerns. Traditional methods of disease detection often lack accuracy, timeliness, and scalability, hindering farmers' ability to mitigate the impact of crop diseases on their yields. Additionally, the increasing complexity and diversity of crop diseases further exacerbate the challenge of timely and accurate diagnosis.
The objective of this project is to develop and implement innovative solutions for improving the accuracy, timeliness, and scalability of crop disease detection in the agricultural sector. By leveraging cutting-edge technologies such as artificial intelligence (AI), machine learning (ML), and computer vision, the project aims to empower farmers with accessible and user-friendly tools for early diagnosis and management of crop diseases. Through collaboration with researchers, agricultural experts, technology developers, and farmers, the project seeks to co-create and share knowledge, expertise, and best practices in crop disease management. Ultimately, the project aspires to enhance agricultural practices, increase crop yields, reduce economic losses, and ensure food security by addressing the pressing challenges associated with crop disease diagnosis and management.

1.2 Literature Survey/Market Survey/Investigation and Analysis

The literature survey conducted for the ”Kisan Vikas” involved a comprehensive review of existing research, industry practices, and advancements related to agricultural support systems, mobile applications in agriculture, and relevant technologies.

Objective

- **Agricultural Support Systems:**
  The survey encompassed a thorough examination of existing agricultural support systems, both digital and traditional, analyzing their functionalities, user interfaces, features, and the impact on farming practices. Notable systems reviewed include Plantix, AgriApp, AgriCentral.

- **Mobile Applications in Agriculture:**
  An analysis of mobile applications designed for the agricultural sector was performed to understand the prevalent features, user experiences, and adoption rates among farmers. Key aspects examined include information dissemination, crop management, weather forecasting, market analysis, and financial assistance.

- **Technological Trends:**
  A review of technological trends and innovations in agriculture, encompassing Internet of Things (IoT) devices for farming, AI-powered crop management systems, and remote sensing technologies, was conducted to identify emerging technologies that could augment the "Kisan Vikas" platform.

- **Research Papers and Publications:**
  Relevant research papers, publications, and articles in agricultural technology, mobile app development in agriculture, and AI integration in farming practices were surveyed to gather insights into the methodologies, best practices, and findings shaping the agricultural technology landscape.

- **Standards and Regulations:**
  An exploration of industry standards, government regulations, and policies related to agricultural data management, privacy, and information dissemination was conducted to ensure compliance and adherence to legal and ethical

considerations in the development of the ”Kisan Vikas” platform.

- **Findings and Insights:**
  The literature survey revealed insightful findings into the realm of agricultural support systems, emphasizing the importance of leveraging technological advancements in improving agricultural practices. It identified opportunities for integrating cutting-edge technologies, such as IoT and AI, to enhance decision-making processes and increase productivity. Additionally, the survey highlighted the need to focus on addressing the identified gap of accessibility and affordability for small-scale farmers, ensuring that these innovative solutions are inclusive and accessible to all farming communities. Overall, the literature underscores the transformative potential of technology in agriculture while emphasizing the necessity for a concerted effort to bridge existing gaps and ensure equitable access to these advancements.

1.3 Introduction to Project

In the ever-evolving landscape of agriculture, the integration of technology has become increasingly vital for modern farming practices. Recognizing this need, the Swami Keshvanand Institute of Technology, Management Gramothan, Jaipur, presents a visionary project titled ”Smart Agriculture (Agri-Tech) to Improve Yield and Profitability.” This initiative, undertaken by the Department of Information Technology, aims to revolutionize traditional farming methodologies by leveraging cutting-edge technologies.

At the heart of this endeavor lies ”KisanVikas,” a digital companion designed to empower farmers with knowledge, technology, and a supportive community. By seamlessly integrating modern tools such as IoT sensors and machine learning with age-old farming practices, KisanVikas equips farmers with the information they need to make informed decisions at every stage of the farming lifecycle.

The project encompasses a comprehensive suite of functionalities tailored to ad-

dress the diverse needs of farmers. From real-time weather updates and market insights, KisanVikas offers a holistic solution to enhance agricultural productivity and profitability.
Supported by a team of dedicated students and guided by a structured roadmap, this project embarks on a journey to transform the agricultural landscape. With its emphasis on innovation, sustainability, and community empowerment, Smart Agriculture endeavors to usher in a new era of prosperity for farmers, bridging the gap between tradition and technology for a brighter, more resilient agricultural future.

1.4 Proposed Logic / Algorithm / Business Plan / Solution / Device

In this part student group need to write logic that they have used in development of project logic, included any of the proposed algorithm.

Proposed Logic :

- The platform will have a client-server architecture, with a mobile application (client) for farmers and a backend server for processing data and providing services.
- Upon login, the system will authenticate farmers using OTP verification.
- Farmers can access various modules such as crop selection, weather forecasts, market prices, disease detection, AI assistance, etc., based on their needs.
- Data collected from farmers, including crop selections, location, and personal information, will be stored securely in a database.
- The backend server will interact with external APIs for weather forecasts and market analysis to provide real-time information to farmers.

- Machine learning algorithms will be employed for disease detection, using input parameters such as soil nutrients and images of affected crops.
- The system will prioritize usability, security, and performance to ensure a seamless user experience for farmers.

Algorithm :

- Disease Detection Algorithm: Implement convolutional neural networks (CNNs) for image processing to detect common crop diseases based on images uploaded by farmers.
- Market Price Analysis Algorithm: Use statistical analysis techniques to process market data and provide insights into crop pricing trends.

Business Plan :

- Revenue Model: The platform could generate revenue through subscription plans for premium features, targeted advertising, or commission-based services such as connecting farmers with buyers or suppliers.
- Market Strategy: Targeting small-scale farmers initially, the platform can focus on regions with high agricultural activity and offer localized support and services. Collaborations with agricultural organizations, government agencies, and NGOs can help in market penetration and user acquisition.
- Sustainability: Continuous engagement with farmers through community forums, educational content, and regular updates will foster user loyalty and retention. Partnerships with agricultural product manufacturers or suppliers can also create additional revenue streams.

Solution :

- The "Kisan Vikas" platform will serve as a comprehensive agricultural support system, providing farmers with tools and resources to enhance productivity and profitability.

- By leveraging technology such as AI, machine learning, and real-time data analysis, the platform aims to address key challenges faced by farmers, including crop selection, weather variability, market fluctuations, and disease management.
- The solution will be scalable and adaptable, catering to the diverse needs of farmers across different regions and crop types.

Device

- Mobile Application: The primary interface for farmers to access the platform, available on both Android and iOS devices. The app will be optimized for performance and usability, with intuitive navigation and responsive design.
- Web Interface: A web-based dashboard for administrators and stakeholders to monitor system performance, manage user accounts, and analyze data trends.

1.5 Scope of the Project

“Kisan Vikas” is envisaged as a comprehensive, integrated platform offering a plethora of functionalities catering to various stages of farming, including crop suggestion, weather forecasting, Disease Detection, market analysis, and optimization of agricultural practices.

Functional Scope :

- User Authentication and Profile Management: Allow farmers to register, login securely, and manage their profiles with essential information such as contact details, farm size, and crop preferences.
- Crop Selection : Provide farmers with tools to select suitable crops based on soil nutrients, climate conditions, and market demand.

- Weather Forecasting and Advisory: Integrate weather APIs to deliver real-time weather updates, forecasts, and advisory services to help farmers plan their farming activities effectively.
- Market Analysis and Pricing: Fetch market data and analyze trends to provide farmers with insights into crop pricing, demand-supply dynamics, and market opportunities.
- Disease Detection and Management: Utilize AI-powered image processing to identify crop diseases from images uploaded by farmers and offer recommendations for disease management and prevention.
- AI Chatbot Assistance: Implement an AI-powered chatbot to provide instant assistance, answer queries related to farming practices, pest control, fertilization, etc.
- Notification and Alert System: Send timely notifications and alerts to farmers about weather emergencies, market fluctuations, government schemes, and relevant updates.

Non-Functional Scope :

- Performance: Ensure the system is responsive, with quick loading times and minimal downtime, even under heavy user loads.
- Security: Implement robust security measures to protect user data, including encryption of sensitive information and role-based access control.
- Usability: Design an intuitive and user-friendly interface across both mobile and web platforms, considering the diverse literacy levels and technological familiarity of farmers.
- Scalability: Architect the system to scale seamlessly as the user base grows, accommodating increased data processing and user interactions.

- Reliability: Establish mechanisms for data backup, disaster recovery, and system monitoring to maintain high availability and data integrity.

Future Enhancements :

- Advanced Analytics: Incorporate advanced data analytics techniques to derive actionable insights from agricultural data, enabling predictive modeling and decision support.
- Geospatial Analysis: Leverage geospatial data and mapping technologies to offer location-specific recommendations and precision agriculture solutions.
- Partnerships and Integration: Collaborate with agricultural research institutions, government agencies, agribusinesses, and e-commerce platforms to enrich the platform's offerings and services.
- Mobile App Enhancements: Continuously improve the mobile app with new features, optimizations, and compatibility updates to enhance user experience and engagement.

Geographical Scope :

- Initially target regions with high agricultural activity and a significant population of small-scale farmers, such as rural areas and agricultural belts.
- Plan for expansion to cover additional geographical regions based on demand, user feedback, and strategic partnerships with local agricultural stakeholders.

Chapter 2
Software Requirement Specification

2.1 Overall Description

The “Kisan Vikas” platform aims to revolutionize the agricultural sector by leveraging advanced technology to enhance farming practices, increase productivity, and improve the livelihoods of farmers. This section provides a comprehensive overview of the system, including its purpose, scope, functionalities, architecture, and future prospects.

- Introduces the “Kisan Vikas” project, highlighting its significance in enhancing agricultural practices and supporting farmers.
- Outlines the purpose, scope, definitions, technologies used, and references guiding the project’s development.

2.1.1 Product Perspective

The Product Perspective section provides an overview of the system from various dimensions, including its interfaces, operations, constraints, and dependencies. Understanding the product’s perspective is essential for comprehending its design, functionality, and interactions within its environment. This section delves into the system’s interfaces, both internal and external, highlighting their significance in enabling communication, data exchange, and seamless operation. Now, let’s delve into the specifics:

2.1.1.1 System Interfaces

System interfaces refer to the points of interaction between different components or subsystems within a software system. These interfaces facilitate communication, data exchange, and interaction between various elements of the system, enabling it to function as intended. In the context of the “Kisan Vikas” project, system interfaces play a crucial role in ensuring the seamless operation and integration of different modules and components.

2.1.1.2 User Interfaces

The user interfaces in the “Kisan Vikas” project include the mobile application interface developed using React Native and the web interface. These interfaces provide intuitive screens and features for farmers to interact with the platform, such as authentication screens, crop selection menus, weather forecasts display, market analysis screens, and chatbot interaction windows. The design prioritizes ease of use and accessibility, ensuring that farmers can navigate the application effortlessly to access relevant agricultural information and services.

2.1.1.3 Hardware Interfaces

Hardware interfaces in the project primarily involve communication between the software system and external hardware devices, such as IoT sensors deployed on farms. These interfaces enable data exchange and interaction with devices like soil moisture sensors, temperature sensors, and other IoT devices. The system integrates with these devices to gather real-time data on environmental conditions, crop health, and other relevant parameters, facilitating data-driven decision-making for farmers.

2.1.1.4 Software Interfaces

Software interfaces within the “Kisan Vikas” project facilitate communication and interaction between different software components. This includes interfaces between the frontend and backend components, API endpoints for data exchange, integration with third-party services for weather forecasting and market analysis, and database connectivity for storing and retrieving information. The use of technologies such as Spring Boot, Express, and MongoDB ensures seamless integration and interoperability between various system modules and external services.

2.1.1.5 Communications Interfaces

- Description: Communication interfaces refer to the mechanisms through which different components of the system interact and exchange data. This includes APIs, protocols, and channels used for communication.
- Memory Constraints: Communication interfaces may have memory constraints depending on the data being transmitted and the devices involved. Efficient data serialization and transmission protocols may be necessary to optimize memory usage.
- Operations: Operations related to communication interfaces include data transmission, reception, error handling, and protocol negotiation.
- Project Functions: Communication interfaces enable the exchange of information between different modules and external systems, facilitating functionalities such as data retrieval from APIs, database access, and real-time updates.
- User Characteristics: Users interact indirectly with communication interfaces through the system's user interface. They expect seamless data exchange and responsiveness, with minimal latency.
- Constraints: Constraints on communication interfaces may include bandwidth limitations, network latency, and security considerations. The system must ensure data integrity, confidentiality, and availability during communication.
- Assumptions and Dependencies: Assumptions regarding communication interfaces include the availability of network connectivity, compatibility with

external systems, and adherence to communication protocols. Dependencies include third-party APIs, network infrastructure, and security mechanisms.

2.1.1.6 Memory Constraints

- Description: Memory constraints refer to limitations on the amount of memory available for storing data and executing processes within the system. Communication Interfaces: Memory constraints impact the design of communication protocols and data serialization methods to optimize memory usage during data exchange.
- Operations: Operations affected by memory constraints include data storage, retrieval, processing, and caching. Efficient memory management techniques are essential to ensure optimal system performance.
- Project Functions: Memory constraints influence the design and implementation of system functionalities, particularly those involving large datasets, complex algorithms, or real-time processing.
- User Characteristics: Users may experience performance issues or system slowdowns if memory constraints are not adequately managed. They expect responsive and efficient system behavior despite memory limitations.
- Constraints: Memory constraints may arise from hardware limitations, platform specifications, or resource allocation policies. The system must prioritize memory usage and allocate resources judiciously to meet performance requirements.
- Assumptions and Dependencies: Assumptions regarding memory constraints include the availability of sufficient memory for system operation and the implementation of memory optimization techniques. Dependencies may include hardware specifications, operating system capabilities, and runtime environments.

2.1.1.7 Operations

- Description: Operations refer to the actions and processes performed by the system to achieve its functionalities and objectives.
- Communication Interfaces: Operations related to communication interfaces include data transmission, reception, validation, and error handling. Efficient and reliable communication is essential for seamless system operation.
- Memory Constraints: Operations affected by memory constraints include data storage, retrieval, manipulation, and caching. Memory management techniques optimize resource usage and prevent performance degradation.
- Project Functions: Operations encompass various functionalities provided by the system, such as user authentication, data processing, analysis, reporting, and system maintenance.
- User Characteristics: Users interact with the system through its user interface, expecting intuitive operation, responsiveness, and reliability. Operations should be transparent and predictable from the user's perspective.
- Constraints: Operational constraints may include time limitations, resource availability, regulatory requirements, and system dependencies. The system must adhere to operational constraints to ensure compliance and functionality.
- Assumptions and Dependencies: Assumptions regarding operations include the availability of necessary resources, adherence to operational guidelines, and the reliability of underlying systems. Dependencies may include hardware components, software libraries, and external services.

2.1.1.8 Project Functions

- Description: Project functions refer to the functionalities provided by the system to fulfill user requirements and objectives.

- Communication Interfaces: Project functions rely on communication interfaces to exchange data with external systems, access APIs, and interact with users. Seamless communication is essential for efficient function execution.
- Memory Constraints: Project functions may be impacted by memory constraints, particularly those involving large datasets, complex computations, or real-time processing. Memory optimization techniques ensure smooth function execution within resource constraints.
- Operations: Project functions encompass various operations performed by the system to achieve specific objectives, including data processing, analysis, storage, retrieval, and presentation. User Characteristics: Project functions are designed to meet the needs and expectations of system users, providing intuitive, responsive, and reliable functionality.
- Constraints: Functional constraints may arise from technical limitations, regulatory requirements, user preferences, or system dependencies. The system must accommodate these constraints while delivering required functionalities.
- Assumptions and Dependencies: Assumptions regarding project functions include user acceptance, functional correctness, and compliance with requirements. Dependencies may include external systems, third-party services, and technology platforms.

2.1.1.9 User Characteristics

- Description: User characteristics refer to the traits, preferences, and behaviors of individuals interacting with the system.
- Communication Interfaces: Users expect intuitive, responsive, and reliable communication interfaces that facilitate seamless interaction with the system. Clear communication channels and feedback mechanisms enhance user experience.

- Memory Constraints: User characteristics influence system design and operation, particularly in terms of responsiveness, efficiency, and usability. Memory optimization techniques ensure optimal performance and user satisfaction.
- Operations: User characteristics impact the design and implementation of system operations, including user authentication, data entry, navigation, and feedback mechanisms.
- Project Functions: Project functions are designed to meet user needs and preferences, providing relevant, accessible, and reliable functionality. User characteristics inform feature prioritization and design decisions.
- Constraints: User characteristics may include technical proficiency, accessibility requirements, language preferences, and cultural norms. The system must accommodate diverse user characteristics to ensure inclusivity and usability.
- Assumptions and Dependencies: Assumptions regarding user characteristics include user acceptance, engagement, and satisfaction. Dependencies may include user feedback, usability studies, and user experience research.

2.1.1.10 Constraints

- Description: Constraints refer to limitations or restrictions imposed on the system's design, implementation, or operation.
- Communication Interfaces: Constraints on communication interfaces may include bandwidth limitations, network latency, security protocols, and compatibility requirements. The system must adhere to these constraints to ensure reliable data exchange.
- Memory Constraints: Constraints on memory usage may arise from hardware limitations, platform specifications, or resource allocation policies. Mem-

ory optimization techniques mitigate performance degradation and ensure efficient resource utilization.

- Operations: Constraints on system operations may include time limitations, resource availability, regulatory requirements, and interoperability constraints. The system must comply with operational constraints to meet functional objectives.
- Project Functions: Functional constraints may arise from technical limitations, regulatory requirements, user preferences, or system dependencies. The system must accommodate these constraints while delivering required functionalities.
- User Characteristics: Constraints related to user characteristics may include technical proficiency, accessibility requirements, language preferences, and cultural norms. The system must accommodate diverse user characteristics to ensure inclusivity and usability.
- Assumptions and Dependencies: Assumptions regarding constraints include adherence to technical specifications, regulatory compliance, and user expectations. Dependencies may include hardware components, software libraries, and external services.

2.1.1.11 Assumption and Dependencies

- Description: Assumptions and dependencies refer to underlying conditions, requirements, or relationships that affect the system's design, implementation, or operation.
- Communication Interfaces: Assumptions and dependencies related to communication interfaces include the availability of network connectivity, compatibility with external systems, and adherence to communication protocols.

- Memory Constraints: Assumptions and dependencies regarding memory constraints include the availability of sufficient memory for system operation and the implementation of memory optimization techniques.
- Operations: Assumptions and dependencies regarding operations include the availability of necessary resources, adherence to operational guidelines, and the reliability of underlying systems.
- Project Functions: Assumptions and dependencies related to project functions include user acceptance, functional correctness, and compliance with requirements.
- User Characteristics: Assumptions and dependencies regarding user characteristics include user acceptance, engagement, and satisfaction.

Chapter 3
System Design Specification

3.1 System Architecture

System architecture refers to the high-level structure of a software system, defining its components, their interactions, and the principles governing their design and evolution. It provides a blueprint for organizing and implementing the system, guiding developers in making design decisions and ensuring that the system meets its functional and non-functional requirements.

1. **Key Components:**
   - Components: These are the building blocks of the system, representing distinct units of functionality or data processing. Components may include modules, libraries, services, databases, and external interfaces.
   - Interactions: System architecture defines how components interact with each other, including communication protocols, data exchange formats, and interface specifications. Interactions ensure seamless coordination and collaboration among system elements.
   - Principles: System architecture is guided by design principles such as modularity, encapsulation, abstraction, and separation of concerns. These principles promote maintainability, scalability, and extensibility, enabling the system to evolve over time.

2. **Types of Architecture:**
   - Monolithic Architecture: In a monolithic architecture, all components of the system are tightly integrated into a single executable or deployment

unit. This architecture simplifies development and deployment but may lack flexibility and scalability.

- Microservices Architecture: In a microservices architecture, the system is decomposed into small, independently deployable services, each responsible for a specific functionality. This architecture promotes scalability, flexibility, and resilience but introduces complexity in managing distributed systems.
- Layered Architecture: In a layered architecture, components are organized into horizontal layers, each representing a different level of abstraction or functionality. This architecture promotes separation of concerns and modularity but may introduce dependencies between layers.
- Service-Oriented Architecture (SOA): SOA involves organizing the system into loosely coupled services that communicate via standardized interfaces. This architecture promotes reusability, interoperability, and agility but requires careful design of service contracts and governance.

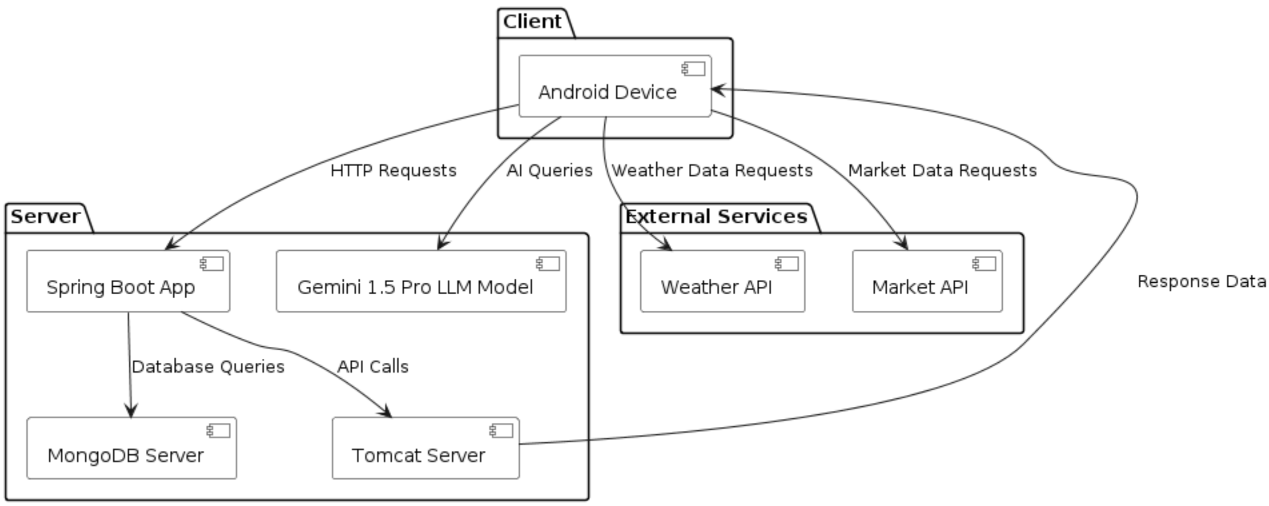

Figure 3.1: Client Serve Diagram

3.2 Module Decomposition Description

Module decomposition involves breaking down the system into smaller, manageable units called modules, each responsible for a specific set of functionalities or data processing tasks. Module decomposition facilitates modular design, code reuse, and maintenance, enabling developers to focus on individual components without being overwhelmed by the complexity of the entire system.

1. **Module Decomposition Process:**

   - Identifying Functionalities: The module decomposition process begins by identifying the functionalities or features that the system needs to support. These functionalities are typically derived from the system requirements and user needs.
   - Grouping Related Functionality: Similar or related functionalities are grouped together to form cohesive modules. Grouping related functionality promotes modularity and encapsulation, allowing modules to be developed, tested, and maintained independently.
   - Defining Interfaces: Modules communicate with each other via well-defined interfaces, specifying how data and control flow between modules. Interface definitions include method signatures, data structures, communication protocols, and error handling mechanisms.
   - Establishing Dependencies: Modules may depend on each other to fulfill their respective responsibilities. Dependency management involves identifying and managing dependencies between modules to minimize coupling
   - Refining Decomposition: Module decomposition is an iterative process that may evolve over time as the system requirements change or new insights are gained. Refining decomposition involves revisiting module boundaries, interfaces, and dependencies to ensure that the system remains flexible, scalable, and maintainable.

2. **Benefits of Module Decomposition:**

   - Modularity: Module decomposition promotes modularity by breaking down the system into smaller, cohesive units. Modularity facilitates code reuse, maintenance, and testing, allowing developers to focus on individual modules without being overwhelmed by the complexity of the entire system.

- Encapsulation: Modules encapsulate their internal implementation details, exposing only well-defined interfaces to other modules. Encapsulation promotes information hiding, abstraction, and separation of concerns, reducing dependencies and minimizing the impact of changes.
- Scalability: Module decomposition enables the system to scale horizontally by adding or removing modules as needed. Scalability is achieved through loose coupling, service-oriented design, and distributed architectures that allow modules to be deployed and scaled independently.
- Maintainability: Module decomposition improves maintainability by isolating changes within individual modules. Changes to one module have minimal impact on other modules, reducing the risk of unintended side effects and making the system easier to evolve and maintain over time.

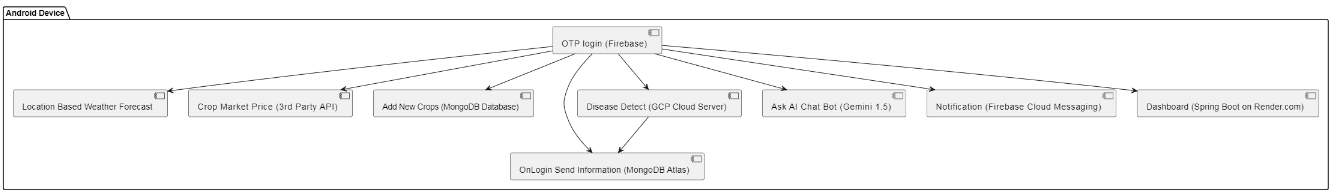

Figure 3.2: Component Diagram

3.3 High Level Design Diagrams

3.3.1 Use Case Diagram

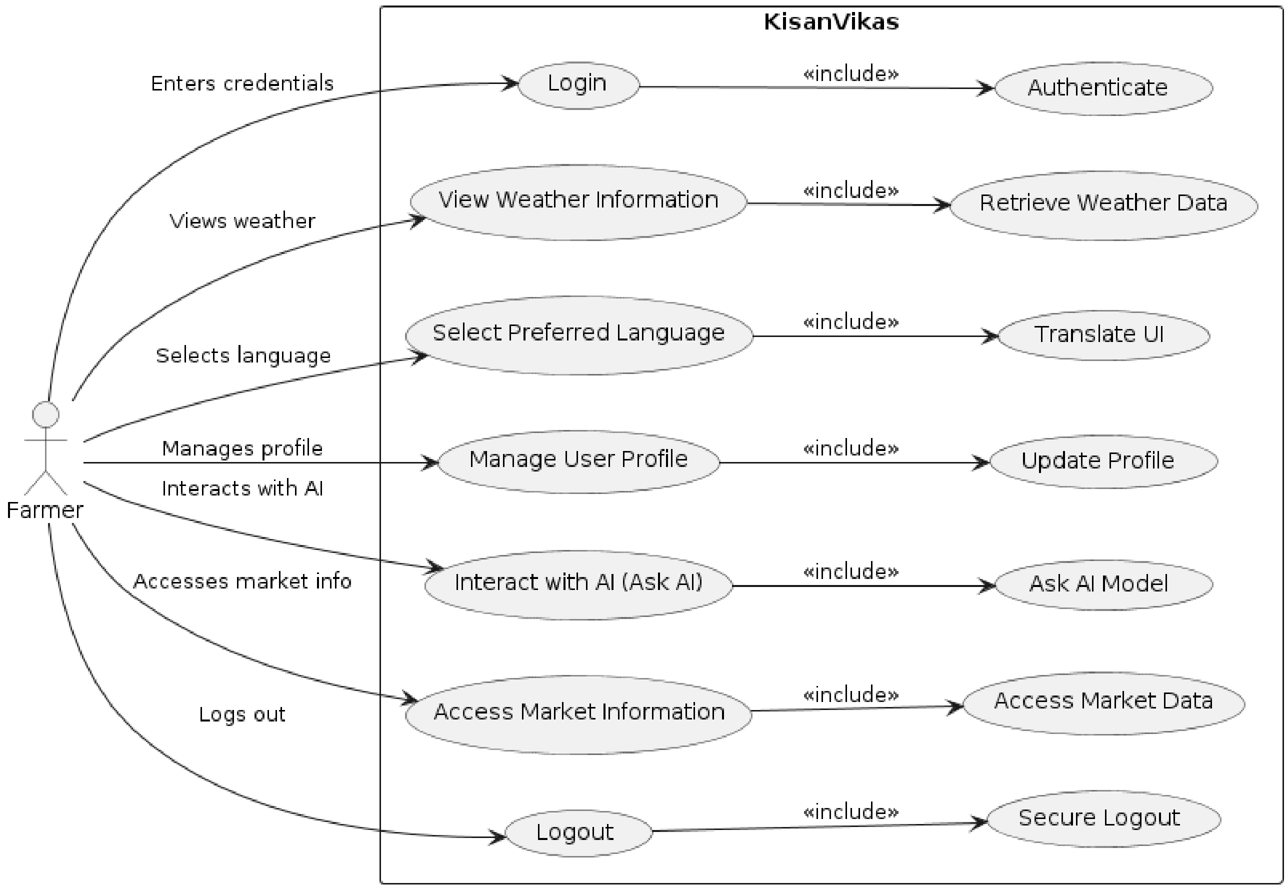

Figure 3.3: Use Case Diagram

3.3.2 Activity Diagram

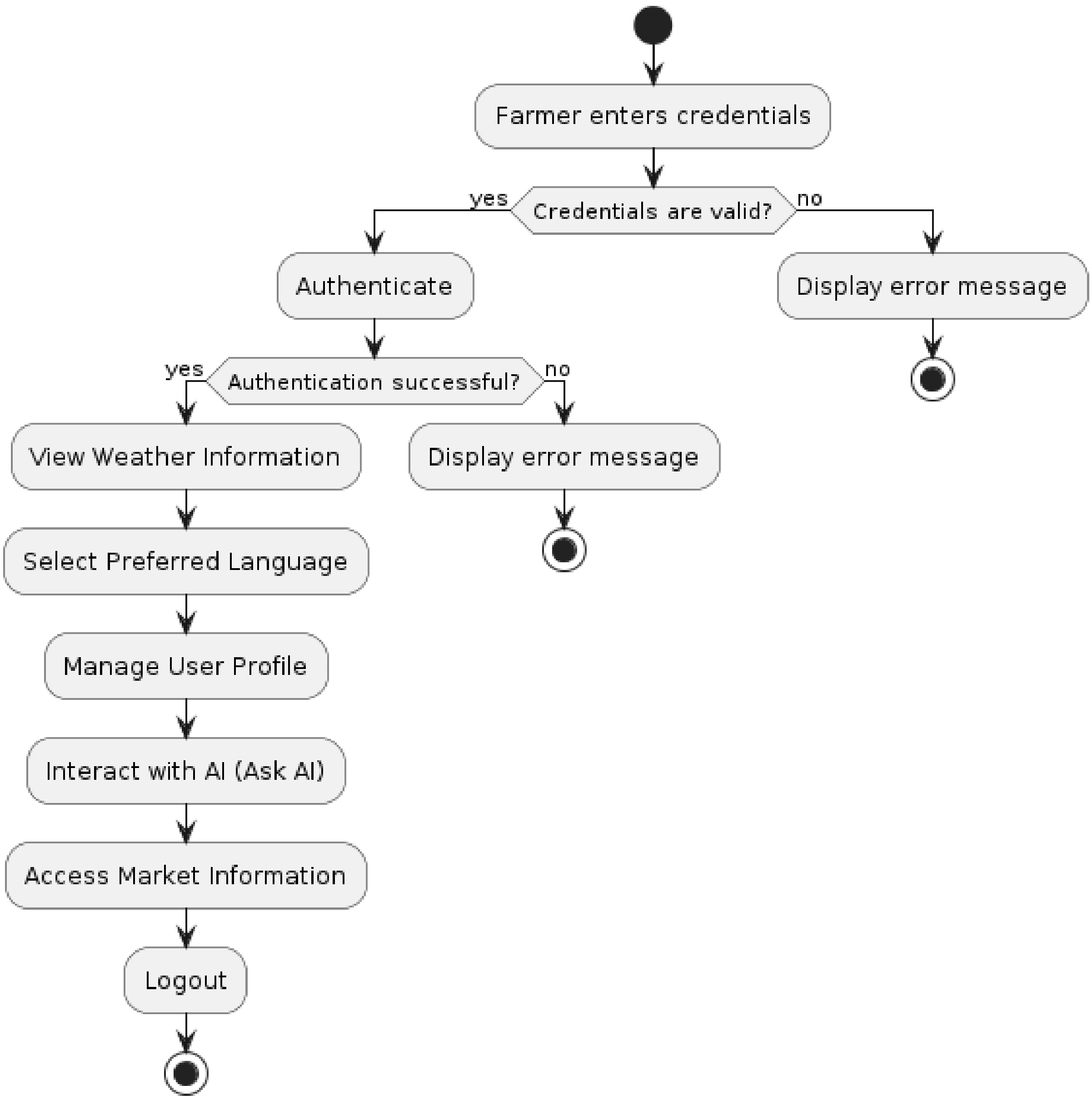

Figure 3.4: Activity Diagram

3.3.3 Data-Flow Diagram

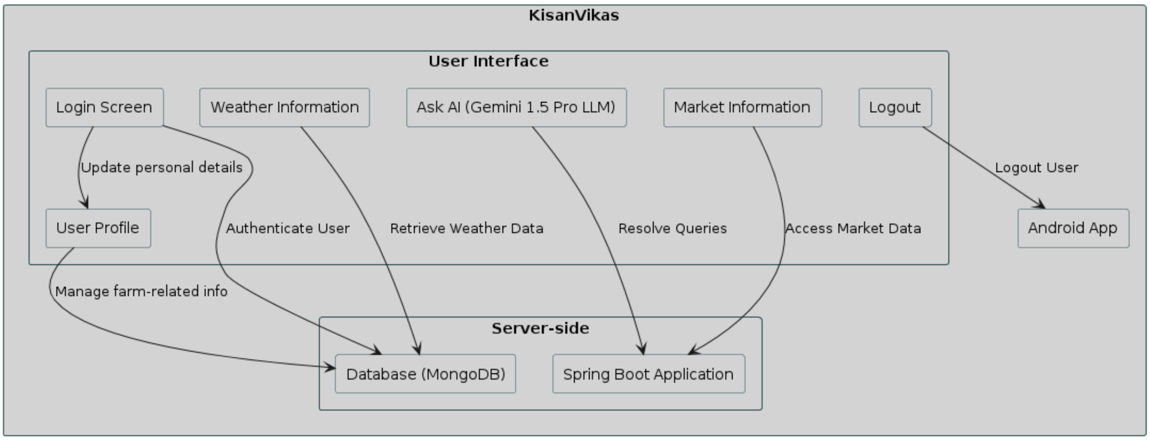

Figure 3.5: Data-Flow Diagram(level-0)

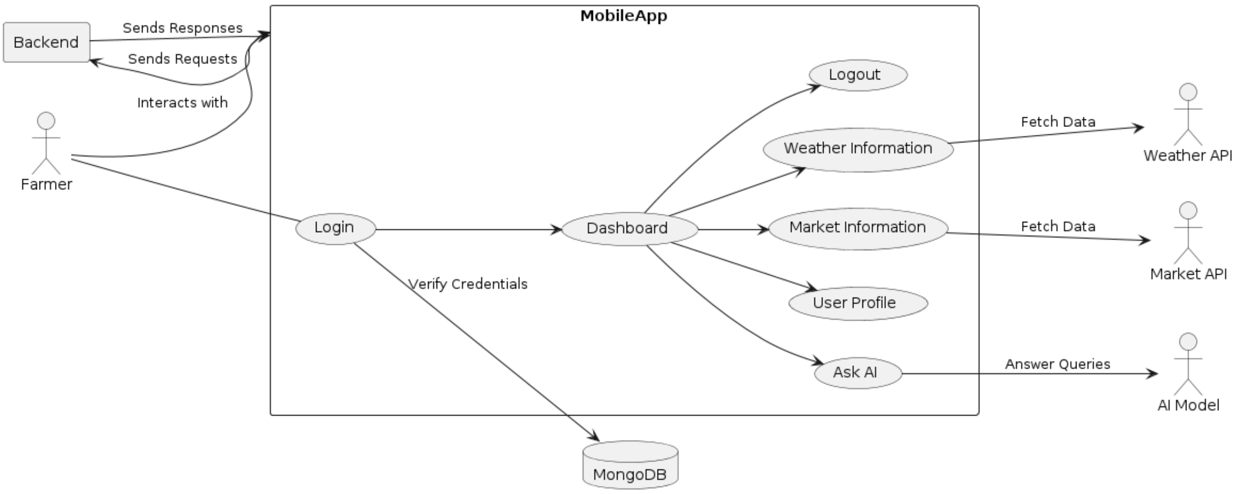

Figure 3.6: Data-Flow Diagram(level-1)

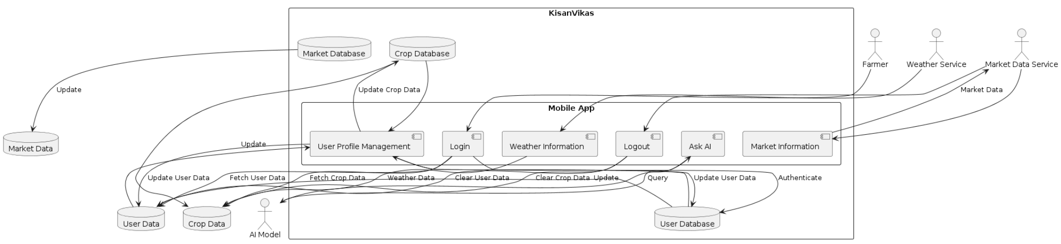

Figure 3.7: Data-Flow Diagram(level-2)

3.3.4 Class Diagram

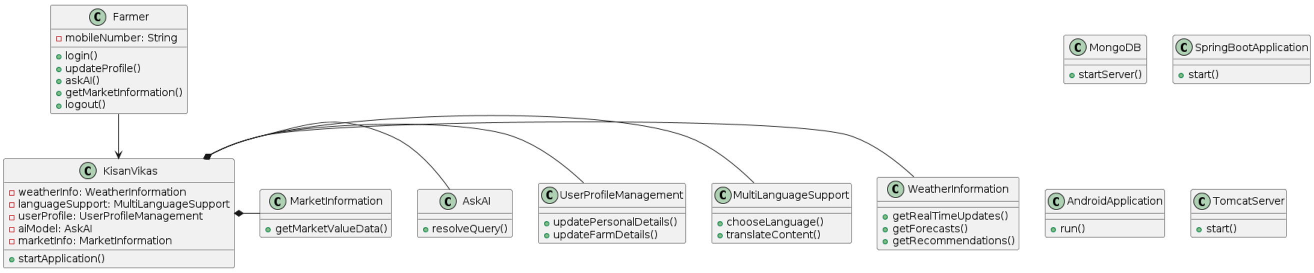

Figure 3.8: Class Diagram

Chapter 4
Methodology and Team

4.1 Introduction to Agile Methodology

Agile methodology is a dynamic and collaborative approach to project management that prioritizes flexibility, customer satisfaction, and continuous improvement. Unlike traditional project management methods, which follow a linear and sequential process, Agile embraces an iterative cycle of planning, development, testing, and review. This allows teams to adapt quickly to changes, incorporate feedback, and deliver incremental value throughout the project lifecycle.

The Kisan Vikas project aims to develop a comprehensive mobile application tailored for farmers. This app provides various functionalities, including user authentication, crop disease detection, real-time weather updates, community forums, farming guidance, financial assistance, and market insights. Given the complexity and the diverse range of features, Agile methodology is particularly well-suited for this project. It enables the development team to remain responsive to the evolving needs of farmers, integrate new technologies seamlessly, and ensure high-quality deliverables.

Adopting Agile methodology for the Kisan Vikas project ensures that we can deliver a user-centric product that meets the real-world needs of farmers. By involving stakeholders and users in the development process, we can gather valuable feedback and make informed decisions. This collaborative approach not only enhances the quality of the final product but also fosters a sense of ownership and satisfaction among all participants.

Following is a diagrammatic representation of different phases of Agile Methodology.

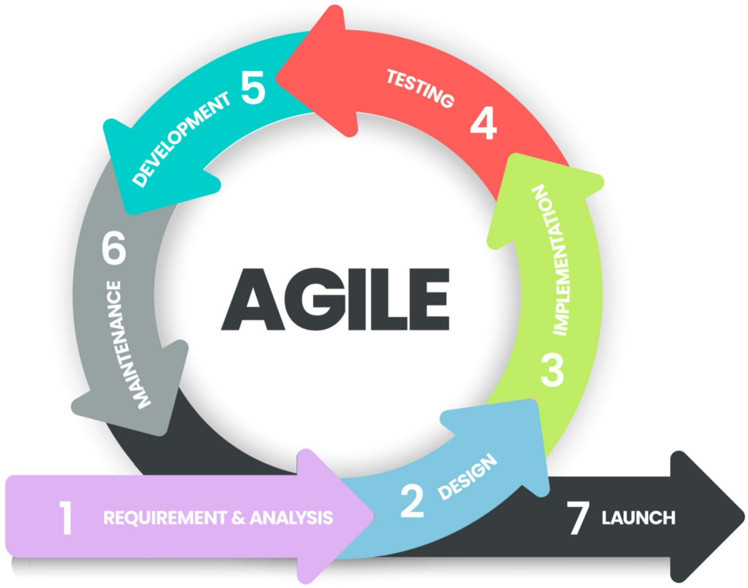

Figure 4.1: Agile Methodology

The Agile methodology is implemented in the following steps-

1. **Initiation and Planning:** Define the project scope, objectives, and stakeholders. Create an initial product backlog with prioritized features and functionalities. Assemble the project team and assign roles.

2. **Sprint Planning:** Break down the project into small, manageable iterations called sprints (typically 2-4 weeks). Select the highest-priority items from the product backlog to include in the sprint backlog. Define sprint goals and plan tasks for the team members.

3. **Development and Implementation:** Develop the selected features and functionalities in incremental steps. Conduct daily stand-up meetings to monitor progress and address any issues. Collaborate closely with the team to ensure alignment and integration of work.

4. **Testing and Quality Assurance:** Perform unit testing, integration testing, and system testing on developed features. Identify and fix bugs and issues.

Ensure the functionality meets the acceptance criteria.

5. **Review and Retrospective:** Conduct a sprint review meeting to showcase the completed work to stakeholders. Gather feedback and evaluate the sprint's outcomes. Hold a sprint retrospective meeting to reflect on the process and identify areas for improvement.

6. **Release and Deployment:** Prepare the application for release to users. Deploy the developed features and functionalities to the production environment. Monitor the deployment and ensure it operates smoothly.

The Agile methodology's sequential phases provide a structured yet flexible approach to project management. Each phase ensures that the project remains aligned with user needs and stakeholder expectations, allowing for continuous improvement and adaptation. The iterative nature of Agile allows the Kisan Vikas project to respond to feedback quickly, ultimately delivering a high-quality, user-centric mobile application for farmers.

Agile methodology Pros & Cons

Advantages

- Flexibility and Adaptability: Agile allows the team to respond quickly to changes in requirements, ensuring that the project remains relevant to user needs.

- Customer Collaboration: Frequent feedback from stakeholders and users ensures that the product meets their expectations and requirements.

- Incremental Delivery: The project delivers small, usable pieces of the application regularly, providing early value to users and stakeholders.

- Improved Quality: Continuous testing and review processes help in identifying and fixing issues early, resulting in a more reliable and robust application.

Enhanced Team Collaboration: Agile promotes close collaboration among team members, fostering a sense of ownership and shared responsibility.

Disadvantages

- Requires Cultural Change: Adopting Agile often requires a shift in organizational culture, which can be challenging for teams used to traditional project management approaches.
- Resource Intensive: Agile can be resource-intensive, requiring dedicated team members and frequent meetings, which may not be feasible for all projects
- Scope Creep Risk: The flexibility of Agile can sometimes lead to scope creep if changes are not carefully managed and prioritized.
- Less Predictable: Due to its iterative nature, Agile projects can be less predictable in terms of timelines and budgets compared to traditional methodologies.
- Dependence on Team Dynamics: Agile relies heavily on the skills and collaboration of the team, which means that the success of the project can be significantly impacted by team dynamics and individual performance.

4.2 Team Members, Roles & Responsibilities

Aditya Agarwal- Project Lead, Full Stack Development
Askhat Jaiman - Front End Development
Abhishek Pandey - AI Model Development and Integration

Chapter 5
Centering System Testing

The designed system has been testing through following test parameters.

5.1 Functionality Testing

In testing the functionality of the web sites the following features were tested:

1. Links

    (a) Internal Links: All internal links of the website were checked by clicking each link individually and providing the appropriate input to reach the other links within.

    (b) External Links: Till now no external links are provided on our website but for future enhancement we will provide the links to the candidate's actual profile available online and link up with the elections updates online etc.

    (c) Broken Links : Broken links are those links which so not divert the page to specific page or any page at all. By testing the links on our website, there was no link found on clicking which we did not find any page.

2. Forms

    (a) Error message for wrong input : Error messages have been displayed as and when we enter the wrong details (eg. Dates), and when we do not enter any details in the mandatory fields. For example: when we enter wrong password we get error message for acknowledging us that we have entered it wrong and when we do not enter the username and/or password we get the messages displaying the respective errors.

(b) Optional and Mandatory fields : All the mandatory fields have been marked with a red asterisk (*) and apart from that there is a display of error messages when we do not enter the mandatory fields. For example: As the first name is a compulsory field in all our forms so when we do not enter that in our form and submit the form we get an error message asking for us to enter details in that particular field.

3. Database Testing is done on the database connectivity.

5.2 Performance Testing

Performance testing evaluates the speed, responsiveness, and stability of a software system under various conditions, including normal and peak loads. For the "Kisan Vikas" platform, performance testing ensures that the application can handle the expected number of concurrent users, data processing tasks, and interactions without degradation in performance.

1. **Types of Performance Testing:**

   - Load Testing: Determines how the system behaves under normal and peak loads by simulating multiple users accessing the application simultaneously. Load testing assesses response times, throughput, and resource utilization to identify bottlenecks and scalability issues.
   - Stress Testing: Pushes the system beyond its normal operating limits to determine its breaking point. Stress testing assesses the system's ability to recover from extreme loads, identify performance bottlenecks, and ensure resilience under adverse conditions.
   - Scalability Testing: Evaluates the system's ability to scale horizontally and vertically to accommodate increasing loads. Scalability testing assesses how the system performs as the user base grows, ensuring that additional resources can be provisioned to handle increased demand.

- Endurance Testing: Validates the system's stability and performance over prolonged periods of continuous operation. Endurance testing assesses memory leaks, resource exhaustion, and degradation in performance over time, ensuring that the system remains reliable under sustained loads. Allow System stability and performance limitations known.

2. **Performance Testing Scenarios for "Kisan Vikas":**

   - Simulating Concurrent Users: Test scenarios simulate various numbers of concurrent users accessing the platform to assess its response times and throughput under different load conditions.
   - Weather Data Processing: Test scenarios simulate large volumes of weather data being processed by the system to evaluate its performance in handling data-intensive tasks.
   - Image Processing for Disease Detection: Test scenarios simulate the processing of images for disease detection to assess the system's performance in handling computationally intensive tasks.
   - Market Data Retrieval: Test scenarios simulate real-time retrieval of market data for crop pricing to evaluate the system's responsiveness in fetching external data.

3. **Performance Testing Tools:**

   - Apache JMeter: Used for load testing, stress testing, and performance measurement of web applications.
   - Gatling: A highly scalable performance testing tool for web applications, capable of simulating thousands of concurrent users.
   - LoadRunner: Provides comprehensive performance testing capabilities for web, mobile, and enterprise applications, including load testing, stress testing, and scalability testing.

5.3 Usability Testing

Usability testing evaluates the ease of use, intuitiveness, and user satisfaction with the "Kisan Vikas" platform's user interface and functionalities. Usability testing involves observing users as they interact with the system, gathering feedback, and identifying usability issues to enhance the overall user experience.

1. **Types of Usability Testing:**

   - Navigation Testing: Assesses the ease of navigation and discoverability of features within the application. Navigation testing evaluates the clarity of menu structures, labels, and navigation paths to ensure that users can easily find and access desired functionalities.
   - Task Completion Testing: Measures the efficiency and effectiveness of users in completing common tasks within the application. Task completion testing assesses the time taken, errors encountered, and user satisfaction in accomplishing tasks such as crop selection, weather forecasting, and market analysis.
   - Feedback Collection: Solicits feedback from users regarding their overall experience, satisfaction levels, and suggestions for improvement. Feedback collection methods include surveys, interviews, and usability questionnaires to gather qualitative insights into user preferences and pain points.
   - Accessibility Testing: Evaluates the accessibility of the application for users with disabilities, ensuring compliance with accessibility standards such as WCAG (Web Content Accessibility Guidelines). Accessibility testing assesses the usability of features such as screen readers, keyboard navigation, and alternative text for images.

2. **Usability Testing Scenarios for “Kisan Vikas”:**

   - Mobile Application Navigation: Test scenarios involve users navigating through the mobile application to perform common tasks such as crop selection, weather checking, and market analysis. Observations are made regarding the ease of navigation, clarity of labels, and intuitiveness of the user interface.
   - Feedback Collection Sessions: Users are invited to participate in feedback collection sessions where they interact with the application and provide feedback on their experience. Feedback is gathered through surveys, interviews, and usability questionnaires to identify areas for improvement.
   - Task Completion Testing: Test scenarios simulate common user tasks such as adding crop information, checking weather forecasts, and analyzing market prices. Observations are made regarding the time taken to complete tasks, errors encountered, and user satisfaction levels.

3. **Usability Testing Tools:**

   - UsabilityHub: Provides online usability testing tools for gathering feedback, conducting user surveys, and performing remote usability testing.
   - UserTesting: Offers remote usability testing services, allowing testers to observe real users interacting with the application and gather feedback on usability issues.
   - Morae: A usability testing tool that enables testers to record user interactions, capture user feedback, and analyze usability issues through screen recordings and session replays.

Chapter 6
Test Execution Summary

Execution Test Summary Report is an overall view of Testing Process from start to end. Test Plan comes at the starting of project while Test Summary Report comes at the end of the testing process. This report is given to the client for his understanding purpose. The Test Summary Report contents are :

1. Test Case ID generated
2. Total number of resources consumed
3. Passed Test Cases
4. Failed Test Cases
5. Status of Test Cases

S No.	Test Case Id	Test Case Description	Test Case Status	No. of Resources Consumed
1	1111	Test the system's login functionality.	Pass	
2	1112	Verify if users can add crops to their profile.	Pass	
3	1113	Check if users can select their stage of farming.	Pass	
4	1114	Validate the system's weather prediction.	Pass	
7	1115	Verify support for multi-language input/output.	Pass	
8	1116	Test the ability to navigate between components.	Pass	
9	1117	Check if users can update personal and farm details.	Pass	
11	1118	Validate push notifications functionality.	Pass	
12	1119	Test the ability to access market values.	Pass	
13	1120	Ensure the availability of the logout option.	Pass	

Table 6.1: Test Case Summary

Chapter 7
Project Screen Shots

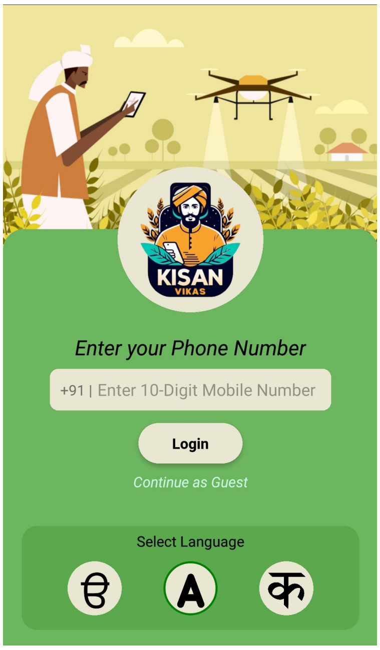

Figure 7.1: Login Screen (English)

Figure 7.2: Login Screen (Hindi)

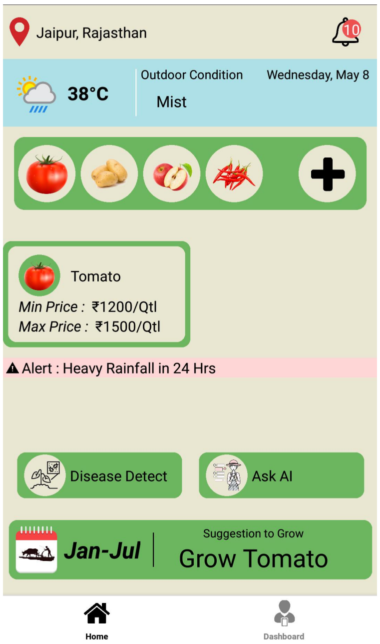

Figure 7.3: Home Screen

Figure 7.4: Select Crops Pop-up

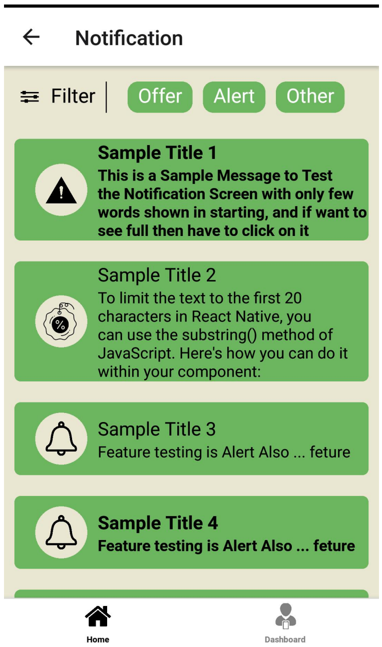

Figure 7.5: Notification Screen

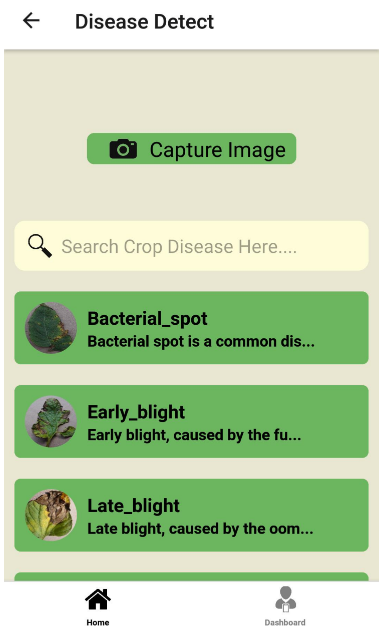

Figure 7.6: Disease Detection

Figure 7.7: Dashboard

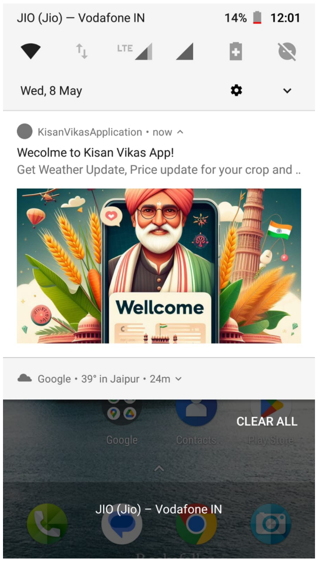

Figure 7.8: Push Notification

Chapter 8
Project Summary and Conclusions

8.1 Conclusion

The development of the “Kisan Vikas” platform marks a significant milestone in leveraging technology to empower farmers, improve agricultural practices, and foster sustainable rural development. Through a collaborative effort involving developers, domain experts, and stakeholders, the project has successfully addressed key objectives and delivered a comprehensive agricultural support system tailored to the needs of farmers.

Summary of Achievements : The “Kisan Vikas” platform has achieved several milestones and notable accomplishments throughout its development lifecycle:

- Comprehensive Functionality: The platform offers a wide range of functionalities, including crop selection, weather forecasting, market analysis, disease detection, and AI-powered assistance, providing farmers with valuable tools and insights to enhance their farming practices.
- Technological Innovation: Leveraging cutting-edge technologies such as AI, IoT, and mobile application development frameworks, the platform integrates advanced features to optimize decision-making processes, improve crop yields, and mitigate risks for farmers.
- User-Centric Design: The platform prioritizes user experience, with an intuitive interface, seamless navigation, and personalized recommendations, ensuring accessibility and usability for farmers across diverse geographical regions and backgrounds.
- Agile Development Methodology: Embracing the Agile methodology, the

project adopted an iterative approach to development, enabling continuous improvements, rapid iterations, and responsiveness to changing requirements and user feedback.

Key Objectives Achieved : The "Kisan Vikas" platform has successfully addressed its primary objectives, including:

- Enhancing Agricultural Support: By providing real-time weather forecasts, market analysis, and disease detection capabilities, the platform facilitates informed decision-making and supports farmers in optimizing their agricultural practices.
- Improving Farmer Productivity: Through personalized recommendations, AI-powered assistance, and access to relevant agricultural information, the platform enables farmers to increase productivity, minimize risks, and maximize returns on their investments.
- Fostering Community Engagement: The platform serves as a hub for farmers to connect, share knowledge, and access resources, fostering a collaborative farming community and promoting knowledge exchange and peer support.

Challenges Overcome : Throughout the project journey, various challenges were encountered and successfully addressed:

- Technical Complexity: Managing the integration of diverse technologies, handling large volumes of data, and ensuring system scalability posed technical challenges that were mitigated through careful design, optimization, and testing.
- User Adoption: Encouraging user adoption and engagement among farmers, particularly in remote or underserved areas, required strategic outreach efforts, user education, and ongoing support to overcome barriers and promote platform usage.

- Data Accuracy and Reliability: Ensuring the accuracy, reliability, and timeliness of data sources such as weather forecasts, market prices, and disease detection models necessitated rigorous validation, quality assurance processes, and collaboration with reliable data providers.

Impact : The "Kisan Vikas" platform has had a profound impact on farmers and agricultural communities, contributing to:

- Empowerment: By providing access to information, resources, and tools, the platform empowers farmers to make informed decisions, adopt best practices, and improve their livelihoods.
- Sustainability: Through optimized agricultural practices, resource management, and risk mitigation strategies, the platform promotes sustainable farming practices and environmental stewardship, contributing to long-term resilience and sustainability.
- Inclusivity: By prioritizing accessibility, usability, and multi-language support, the platform ensures inclusivity and equitable access to agricultural support services, reaching farmers across diverse socio-economic backgrounds and geographic regions.

Vision and Direction : In conclusion, the "Kisan Vikas" platform stands as a testament to the transformative potential of technology in agriculture, embodying the principles of innovation, inclusivity, and sustainability. With a clear vision and ongoing commitment to excellence, the project aims to continue its journey of empowering farmers, advancing agricultural practices, and building resilient rural communities for a brighter and more prosperous future.

Chapter 9
Future Scope

The ”Kisan Vikas” platform has immense potential for further development and expansion, offering numerous opportunities to enhance its capabilities, reach, and impact. Below are detailed aspects of the future scope for the project:

Advanced Analytics and Predictive Modeling :

- Integrate advanced data analytics and predictive modeling techniques to forecast crop yields, identify emerging trends, and optimize resource allocation for enhanced productivity and profitability.
- Implement machine learning algorithms to analyze historical data, weather patterns, soil conditions, and market trends to provide personalized insights and recommendations for farmers.

Precision Agriculture and IoT Integration :

- Explore the integration of IoT sensors, drones, and satellite imagery for real-time monitoring of crop health, soil moisture, and environmental conditions, enabling precision agriculture practices and proactive decision-making.
- Develop IoT-based solutions for smart irrigation, automated pest control, and crop management, leveraging sensor data and automation to optimize resource usage and minimize environmental impact.

Blockchain Technology for Traceability and Transparency :

- Implement blockchain technology to establish a secure and transparent supply chain, enabling traceability of agricultural products from farm to fork.

- Create a decentralized marketplace for farmers to sell their produce directly to consumers, bypassing intermediaries and ensuring fair prices and transparent transactions.

Enhanced Community Engagement and Knowledge Sharing :

- Expand community engagement features such as discussion forums, knowledge sharing platforms, and virtual farmer networks to facilitate peer-to-peer learning, collaboration, and information exchange.
- Organize virtual workshops, training sessions, and expert consultations to empower farmers with the latest agricultural techniques, best practices, and technological innovations.

Mobile Application Enhancements :

- Continuously improve the mobile application interface, functionality, and performance to ensure seamless user experience across different devices, operating systems, and network conditions.
- Introduce gamification elements, interactive tutorials, and personalized recommendations to increase user engagement and retention among farmers.

Partnerships and Collaborations :

- Forge strategic partnerships with government agencies, agricultural organizations, research institutions, and technology providers to expand the platform's reach, access additional resources, and leverage domain expertise.
- Collaborate with financial institutions to offer microfinance, insurance, and credit facilities tailored to the needs of smallholder farmers, enabling access to capital and risk mitigation solutions.

Integration with Emerging Technologies :

- Explore the integration of emerging technologies such as augmented reality (AR), virtual reality (VR), and natural language processing (NLP) to enhance user interactions, visualize agricultural data, and provide immersive learning experiences.

- Investigate the potential applications of quantum computing, edge computing, and 5G connectivity in agriculture to enable real-time data processing, edge analytics, and high-speed communication for remote farming operations.

Scalability and Global Expansion :

- Design the platform architecture with scalability in mind to accommodate growing user demand, increased data volume, and geographic expansion to new regions and markets.
- Localize the platform for different languages, cultures, and regulatory environments to make it accessible and relevant to farmers worldwide, addressing the diverse needs and preferences of agricultural communities.

References

[1] *Hema M S, Niteesha Sharma, Y Sowjanya, Ch. Santoshini ,R Sri Durga , V. Akhila, "Plant disease prediction using convolutional neural network", EMITTER International Journal of Engineering Technology Vol. 9, No. 2, December 2021.*

[2] *Akshatha, K. R., and K. S. Shreedhara. "Implementationof machine learning algorithms for crop recommendationusing precision agriculture." International Journal ofResearch in Engineering, Science and Management(IJRESM) 1, no. 6 (2018)*

[3] *Dataset Link -*
Disease Detection: `https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset`