Smart Agriculture (Agri-Tech) to improve Yield and profitability
Software Requirements Specification

Version <1.1>

Submitted in Partial Fulfillment for the Award of Degree of Bachelor of Technology in Information Technology from Rajasthan Technical University, Kota

MENTOR:

Mrs. Richa Rawal

(Dept. of Information Technology)

COORDINATOR:

Mrs. Sanju Chaudhary

(Dept. of Information Technology)

SUBMITTED BY:

Aditya Agarwal (20ESKIT005)

Akshat Jaiman (20ESKIT007)

Abhishek Pandey (20ESKIT004)

DEPARTMENT OF INFORMATION TECHNOLOGY

SWAMI KESHWANAND INSTITUTE OF TECHNOLOGY, MANAGEMENT & GRAMOTHAN

Ramnagaria (Jagatpura), Jaipur – 302017

SESSION 2023-24

Revision History

Date	Version	Description	Author
20/11/2023	1.0	Contain Basic Layout and requirement Specification for this Project	Abhishek Pandey, Aditya Agarwal, Akshat Jaiman
02/12/2023	1.1	Added Module for Market Price, and Disease Detect using CNN Module	Abhishek Pandey, Aditya Agarwal, Akshat Jaiman

Table of Contents

1. Introduction

The Software Requirements Specification (SRS) document for the "Kisan Vikas" platform serves as a comprehensive guide delineating the functional and non-functional requirements essential for designing, developing, and deploying an innovative agricultural support system. This introduction provides an overview of the entire SRS, encompassing its purpose, scope, definitions, acronyms, abbreviations, references, technology used, and an overview of the SRS contents.

1.1 Purpose

The primary objective of this SRS is to establish a clear and unified understanding among stakeholders, including developers, designers, project managers, domain experts in agriculture, and end-users, regarding the functionalities, constraints, and objectives of the "Kisan Vikas" platform. By defining and documenting these requirements, this document aims to guide the development team in creating a robust, user-centric application that addresses the needs of farmers and enhances agricultural practices.

1.2 Scope

"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, community forums, resource procurement, governmental scheme access, market analysis, and optimization of agricultural practices. This SRS delineates the specific requirements for each module within the system, outlining their functionalities, interactions, and constraints.

1.3 Definitions, Acronyms and Abbreviations

- **SRS**: Software Requirements Specification
- **API**: Application Programming Interface
- **UI**: User Interface
- **DBMS**: Database Management System
- **IoT**: Internet of Things

1.4 References

This SRS references industry standards, agricultural domain knowledge, and any external documents or specifications used in defining the requirements for the "Kisan Vikas" platform.

1.5 Technologies to be used

The "Kisan Vikas" platform utilizes cutting-edge technologies and frameworks to deliver its functionalities, including:

- **Frontend**: React Native for cross-platform mobile application development.
- **Authentication**: Firebase Authentication for Mobile SMS Verification
- **Backend**: Spring Boot for the backend server application.
- **Database**: MongoDB for data storage and retrieval.
- **AI Integration**: TensorFlow.js for integrating AI models.

1.6 Overview

This SRS document is structured systematically into distinct sections, each detailing specific facets of the "Kisan Vikas" platform. It includes an executive summary, system overview, detailed descriptions of functional and non-functional requirements, user interfaces, system constraints,

assumptions, dependencies, and technical specifications. Additionally, it incorporates diagrams, use cases, and mockups to provide a comprehensive understanding of the system's architecture and functionality.

2. Literature Survey

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

2.1 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.

2.2 Research Paper

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*] This research paper proposes a computer-aided disease recognition model using deep convolutional networks, specifically VGG16 and ResNet34, for plant disease detection. It compares the performance of these models in terms of accuracy, sensitivity, and specificity, offering personalized recommendations to farmers based on soil features, temperature, and humidity.

*Maaz Patel, Anagha Rane, Vansh Patni '*Crop Recommendation System using Machine Learning*'*, [International Journal of Research in Engineering, Science and Management (IJRESM) 1, no.6 (2018)] This research paper focuses on recommending the best crops for Indian farmers based on factors like soil nutrients, humidity, and rainfall, utilizing various machine learning models including Decision Tree, Support Vector Machine, Logistic Regression, and Gaussian Naïve Bayes. It leveraging machine learning algorithms to optimize agricultural productivity and risks for farmers

Conclusion
The literature survey provided valuable insights and a broader understanding of existing agricultural support systems, technological advancements, and industry trends. It serves as a foundational reference guiding the development and feature enhancement of the "Kisan Vikas" platform, ensuring alignment with industry best practices and innovation.

3.Specific Requirements

3.1 Functional Requirement

Functional requirements outline the specific functionalities and features that the "Kisan Vikas" platform must provide to meet Farmer needs and expectations.

1. **Login Module**:
    - **Description**: The Login Module facilitates Farmer authentication through mobile number input with OTP verification. Additionally, it offers a guest login option for Farmer who do not wish to register.
    - **Key Features**: Secure mobile number authentication, OTP verification, guest login option.
2. **AskForLocation Module**:
    - **Description**: This module requests permission to access the device's location services and fetches latitude and longitude coordinates. It then displays the Farmer

current city and state information on the screen.
- **Key Features**: Location permission request, retrieval of latitude and longitude coordinates, display of city and state information.

3. **Select a Crops Module**:
   - **Description**: The Select a Crops Module enables Farmer to choose the fruits or vegetables they intend to grow. The selected crops are then visually displayed on the screen for easy reference.
   - **Key Features**: Crop selection functionality, visual display of selected crops.
4. **Add User Info Module**:
   - **Description**: This module collects essential Farmer information such as name, contact number, address, farm NPK value, and farm size. It serves to personalize the user experience and tailor recommendations accordingly.
   - **Key Features**: Farmer information collection, personalization of recommendations.
5. **Weather Module**:
   - **Description**: The Weather Module provides Farmer with up-to-date weather information, including current temperature, weather conditions, prediction, and today's date. This data aids in informed decision-making for farming activities.
   - **Key Features**: Current temperature display, weather condition overview, prediction forecast, date information.
6. **Market Value of Crop Module**:
   - **Description**: This module displays the current maximum and minimum prices of crops in the market for the selected crop. It offers valuable insights for farmers regarding potential earnings from their harvest.
   - **Key Features**: Real-time market price display, crop-specific pricing information.
7. **Disease Detection Module**:
   - **Description**: The Disease Detection Module offers information on common crop diseases and allows users to capture or select images of affected crops. These images are then sent to the backend for disease prediction and diagnosis.
   - **Key Features**: Crop disease information, image capture/upload for diagnosis.
8. **Ask AI Module**:
   - **Description**: With the Ask AI Module, users can interact with an AI-powered chatbot (LLM model GEMINI Pro 1.5) to ask questions and seek advice related to agriculture. This feature provides instant assistance and guidance.
   - **Key Features**: AI-powered chatbot interaction, instant agricultural advice.
9. **Crop Suggestion Module**:
   - **Description**: The Crop Suggestion Module utilizes data-driven algorithms to recommend suitable crops based on the NPK value provided by the user. It helps users make informed decisions about crop selection.
   - **Key Features**: Data-driven crop recommendations, personalized suggestions based on NPK value.
10. **Main Screen Module**:
    - **Description**: The Main Screen Module serves as the central hub for accessing essential agricultural information. It offers seamless navigation between the home screen and dashboard tabs, ensuring a smooth user experience.

- **Key Features**: Centralized access to agricultural information, intuitive navigation.

11. **Notification Module**:
    - **Description**: The Notification Module displays notifications sent by the server, keeping users informed about important updates, alerts, and informational messages. It also includes options to filter notifications by type.
    - **Key Features**: Notification display, filtering options by offer, alert, and info.
12. **Home Screen Module**:
    - **Description**: The Home Screen Module presents users with key information such as weather updates, selected crops, current location, and navigation buttons to access other features like disease detection and AI assistance.
    - **Key Features**: Weather display, selected crop information, location details, navigation buttons.
13. **Dashboard Module**:
    - **Description**: The Dashboard Module provides users with personalized information such as user details, access to personal and farm information, and a logout button for account management.
    - **Key Features**: User information display, access to personal and farm info, account management options.

3.2Non-Functional Requirements

Non-functional requirements define the qualities and characteristics that the system must possess, such as performance, security, and usability.

1. Performance
    - The system response time for generating crop recommendations shall be within 3 seconds.
    - The platform shall handle a concurrent Farmer load of at least 1000 Farmer.
2. Security
    - Farmer passwords shall be securely stored using industry-standard encryption algorithms.
    - The system shall implement role-based access control to ensure data privacy
3. Usability
    - The user interface shall follow industry best practices for a user-friendly experience.
    - The platform shall support multiple languages for enhanced accessibility
4. Reliability
    - The system shall have a minimum uptime of 99.9%.
    - Data backup and recovery mechanisms shall be in place to prevent data loss.

3.3 Hardware Requirements

The hardware requirements outline the necessary infrastructure for deploying and running the "Kisan Vikas" platform.

1. Server Infrastructure:
    - Minimum of dual-core processors with 8 GB RAM for the backend server.
    - Adequate storage capacity for database and system files.

2. Mobile Devices
    - The mobile application shall be compatible with Android devices (version 7.0 and above)

3.4 Software Requirements

Software requirements detail the necessary software components and dependencies for the "Kisan Vikas" platform.

1. Backend Technologies:
    - Spring Boot framework for the backend server application.
    - Express to fetch and use AI Prediction
    - MongoDB database for data storage.
2. Frontend Technologies:
    - React Native for cross-platform mobile application development.
3. Third-Party Integrations:
    - Integration with weather forecasting APIs.
      Integration with market analysis APIs.

3.5 Agile Methodology

The "Kisan Vikas" project embraces the Agile development methodology, recognizing its effectiveness in managing dynamic project requirements and ensuring continuous improvements. The Agile approach prioritizes flexibility, collaboration, and iterative development cycles to meet evolving Farmer needs.

The Agile methodology is implemented in the following steps:

1. **Project Initiation:**
    - Define project objectives, goals, and scope.
    - Assemble the project team comprising developers, designers, and domain experts.
2. **Product Backlog Creation:**
    - Identify and list all desired features and functionalities.
    - Prioritize these features based on their importance and potential impact on Farmers.
3. **Sprint Planning:**
    - Break down the prioritized features into smaller tasks for implementation within sprints (short development cycles).
    - Estimate the effort required for each task and allocate them to the upcoming sprints.
4. **Sprint Execution:**
    - Development teams work on the tasks assigned for the sprint.
    - Daily stand-up meetings are held to discuss progress, challenges, and plan adjustments.
5. **Continuous Integration and Testing:**
    - Developers continuously integrate their code into the shared repository.
    - Automated and manual testing is performed to ensure the developed features meet quality standards.
6. **Sprint Review:**
    - At the end of each sprint, a review meeting is held to showcase the completed work to stakeholders.

- Feedback is collected, and adjustments are made to improve the next sprint.

7. **Sprint Retrospective:**
   - The team reflects on the sprint's successes and challenges.
   - Identify areas for improvement and implement changes to enhance team efficiency and product quality.
8. **Incremental Deployment:**
   - Completed and tested features are deployed incrementally to the live environment.
   - Farmers can start using and providing feedback on the new functionalities.
9. **Continuous Feedback and Adaptation:**
   - Gather feedback from Farmers and stakeholders regularly.
   - Adapt the project scope, features, or priorities based on feedback and changing requirements.
10. **Iterative Development:**
    - Repeat the cycle, starting from sprint planning, to incorporate new features and improvements.

3.6 Business Process Model

1. **Login Module**:
   - **Input**: Farmer's mobile number and OTP.
   - **Process**: Verify the mobile number and OTP for authentication.
   - **Output**: Successful login or error message.
2. **AskForLocation Module**:
   - **Input**: Farmer's permission to access device location.
   - **Process**: Request permission to access device location services and fetch latitude and longitude coordinates.
   - **Output**: Display Farmer's current city and state information.
3. **Select a Crops Module**:
   - **Input**: Farmer's selection of fruits or vegetables.
   - **Process**: Capture and store the Farmer's crop selection.
   - **Output**: Visual display of selected crops.
4. **Add User Info Module**:
   - **Input**: Farmer's personal information such as name, contact number, and address.
   - **Process**: Collect and store Farmer information in the database.
   - **Output**: Personalized Farmer experience based on stored information.
5. **Weather Module**:
   - **Input**: Current weather data and forecast.
   - **Process**: Retrieve and display weather information.
   - **Output**: Real-time weather updates for informed decision-making.
6. **Market Value of Crop Module**:
   - **Input**: Selected crop by the Farmer.
   - **Process**: Retrieve market data for the selected crop.
   - **Output**: Display current market prices for the selected crop.
7. **Disease Detection Module**:
   - **Input**: Farmer-captured images of affected crops.

- **Process**: Analyze the images using AI algorithms for disease detection.
- **Output**: Disease diagnosis and recommended actions.

8. **Ask AI Module**:
    - **Input**: Farmer's queries related to agriculture.
    - **Process**: Process Farmer's queries using AI chatbot.
    - **Output**: Instant responses and agricultural advice.
9. **Crop Suggestion Module**:
    - **Input**: Farmer's farm NPK value.
    - **Process**: Utilize data-driven algorithms to recommend suitable crops.
    - **Output**: Crop suggestions based on specific farming conditions.
10. **Main Screen Module**:
    - **Input**: Farmer's interaction with navigation buttons.
    - **Process**: Navigate between home screen and dashboard tabs.
    - **Output**: Seamless access to essential agricultural information.
11. **Notification Module**:
    - **Input**: Server-generated notifications.
    - **Process**: Filter notifications by type (offer, alert, info).
    - **Output**: Display filtered notifications to Farmer.
12. **Home Screen Module**:
    - **Input**: Farmer's interaction with the home screen.
    - **Process**: Display weather updates, selected crops, and navigation buttons.
    - **Output**: Key agricultural information for Farmer's reference.
13. **Dashboard Module**:
    - **Input**: Farmer's access to personal and farm information.
    - **Process**: Provide access to Farmer's details, personal info, and account management options.
    - **Output**: Personalized dashboard with Farmer-specific information

This business process model outlines key processes within the "Kisan Vikas" system, detailing inputs, processes, and resulting outputs. It illustrates how various functionalities interact and contribute to the overall functioning of the platform, aiding in streamlining agricultural practices and empowering farmers.

3.7 Supplementary Requirements

Supplementary requirements include any additional requirements that are not covered by the previous sections but are essential for the success of the project.

1. Scalability:
    - The system architecture shall be designed to scale horizontally to accommodate an increasing user base.
2. Documentation:
    - Comprehensive user manuals and developer documentation shall be provided for system understanding and maintenance.
3. Security Measures:
    - Data Encryption: Ensure sensitive Farmer data is encrypted to maintain

confidentiality.

4. Compatibility and Integration:
    - API Integration: Provide well-documented APIs to allow seamless integration with third-party applications or services.
5. Performance Optimization:
    - Load Balancing: Implement load balancing techniques to ensure optimal performance, especially during high traffic periods.

4.Overall Description

4.1 Use-Case Model Survey

Actors:

Farmer: Primary user who interacts with the system.

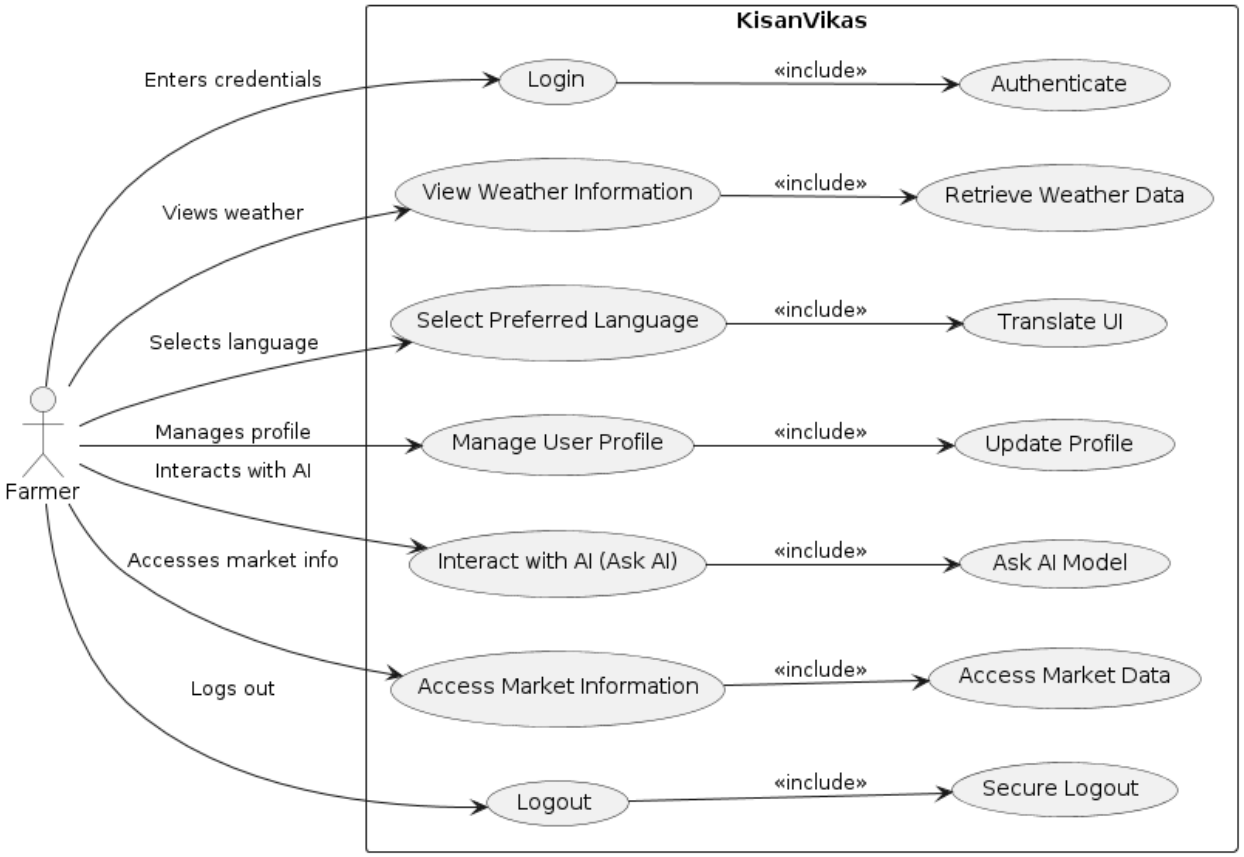

Figure 1: Kisan Vikas Platform Use Case Diagram

4.2 Behaviors Diagrams

- Activity Diagram

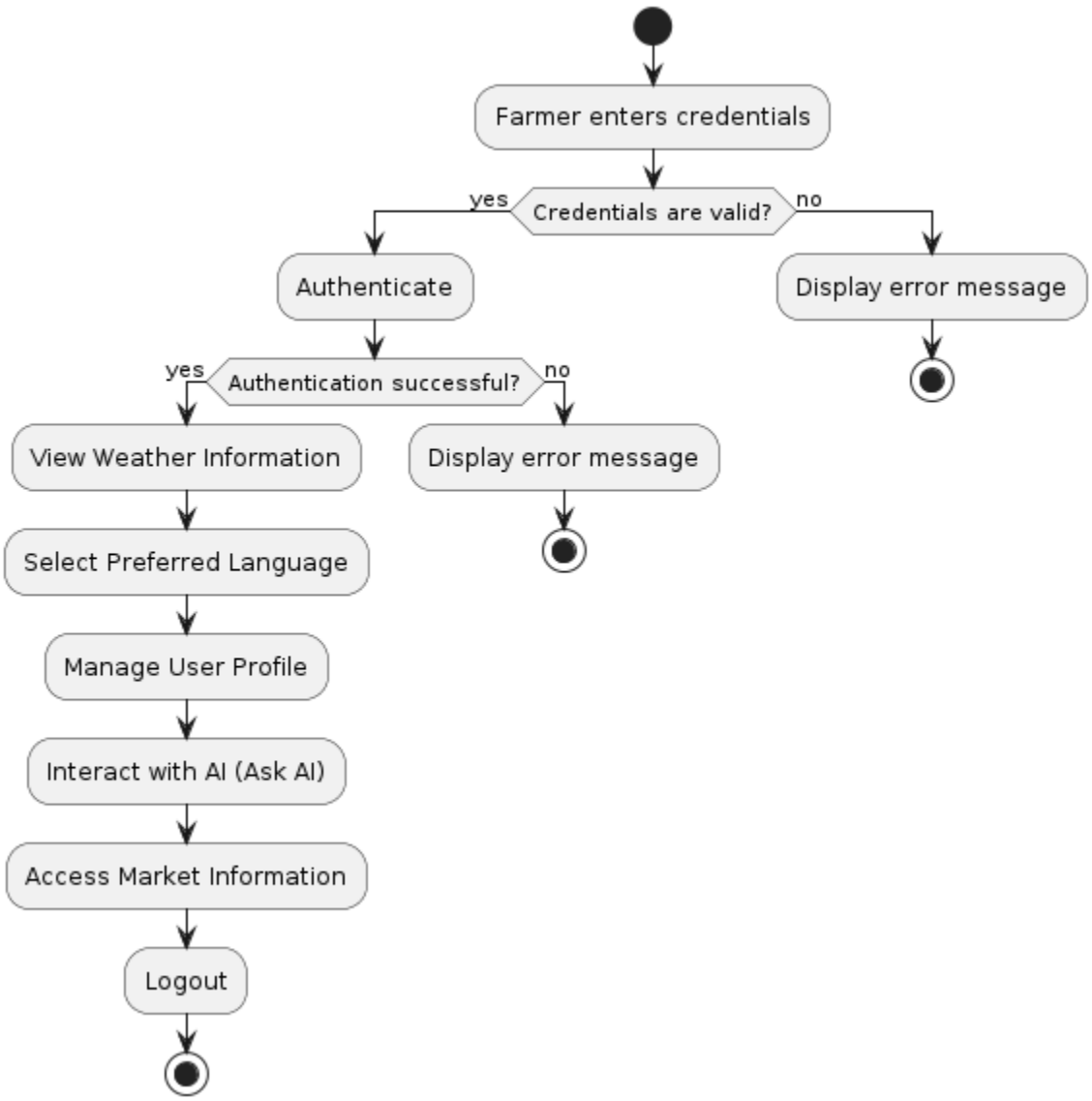

Figure 2: Kisan Vikas Platform Activity Diagram

- Sequence Diagram

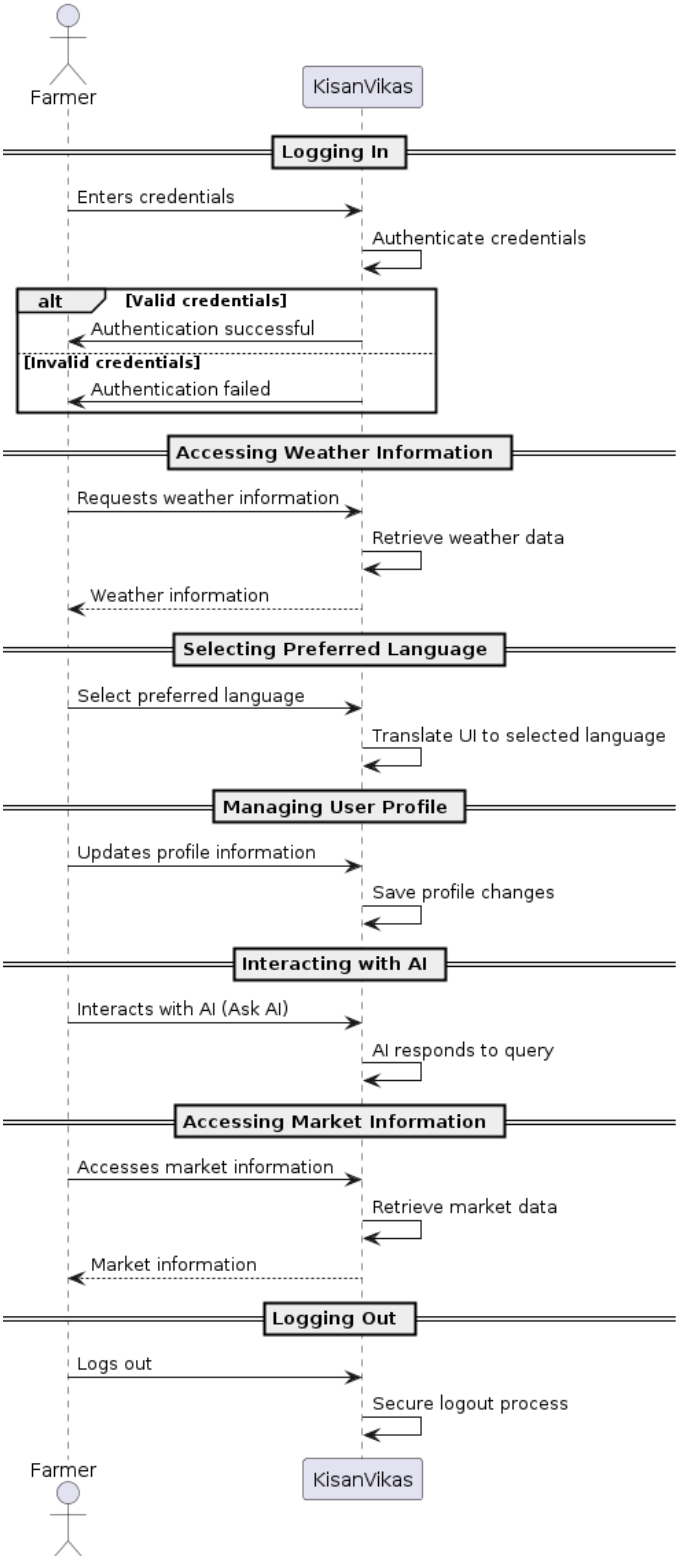

Figure 3: Sequence Diagram for Farmer all Modules

- Data Flow Diagram – Level 0

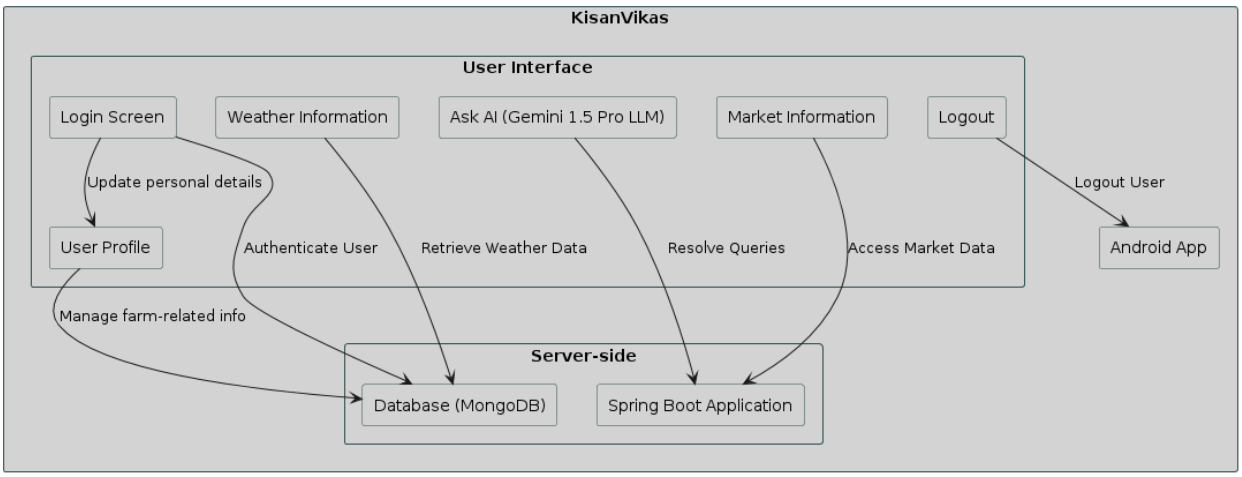

Figure 4: DFD Level 0

- Data Flow Diagram – Level 1

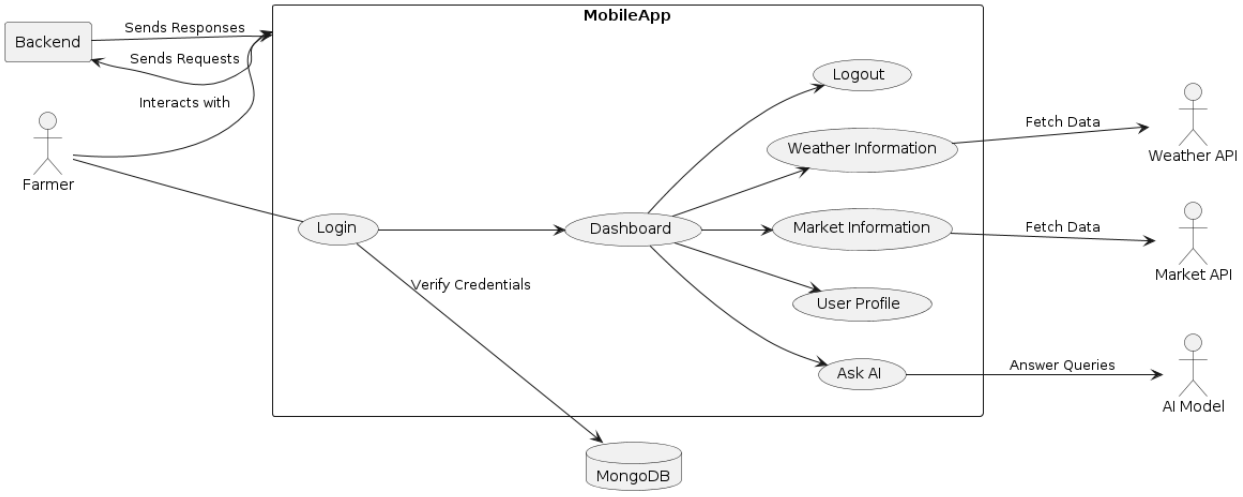

Figure 5:DFD Level 1

- Data Flow Diagram – Level 2

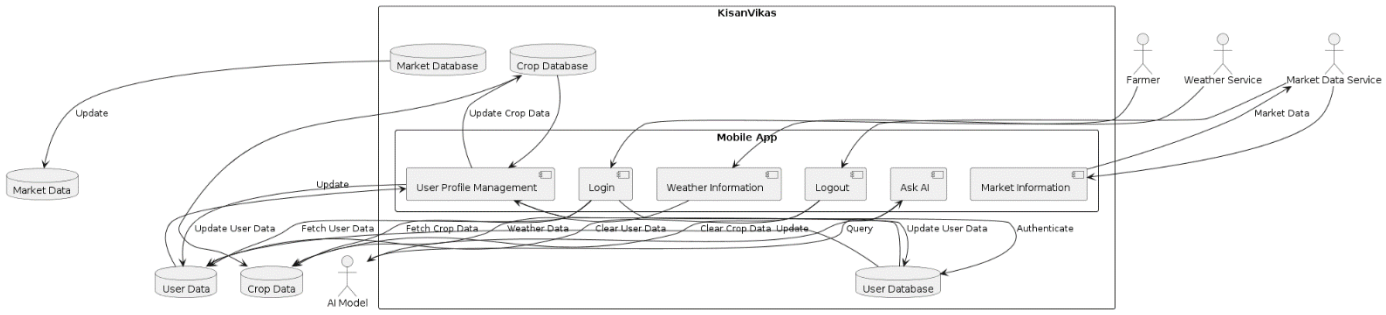

Figure 6: DFD Level 2

- Communication Diagram

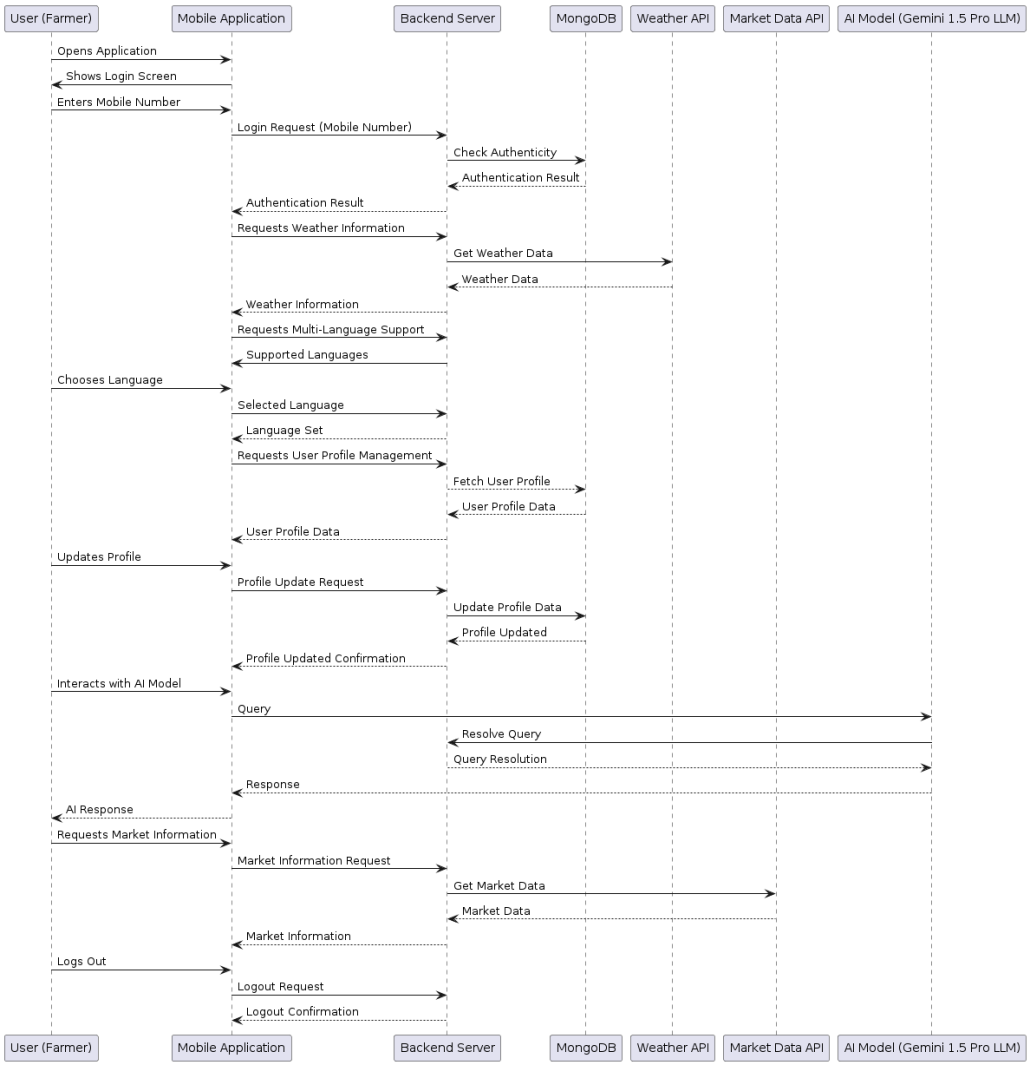

Figure 7: Kisan Vikas Platform Communication Diagram

4.3 Structural Diagrams

- Deployment Diagram

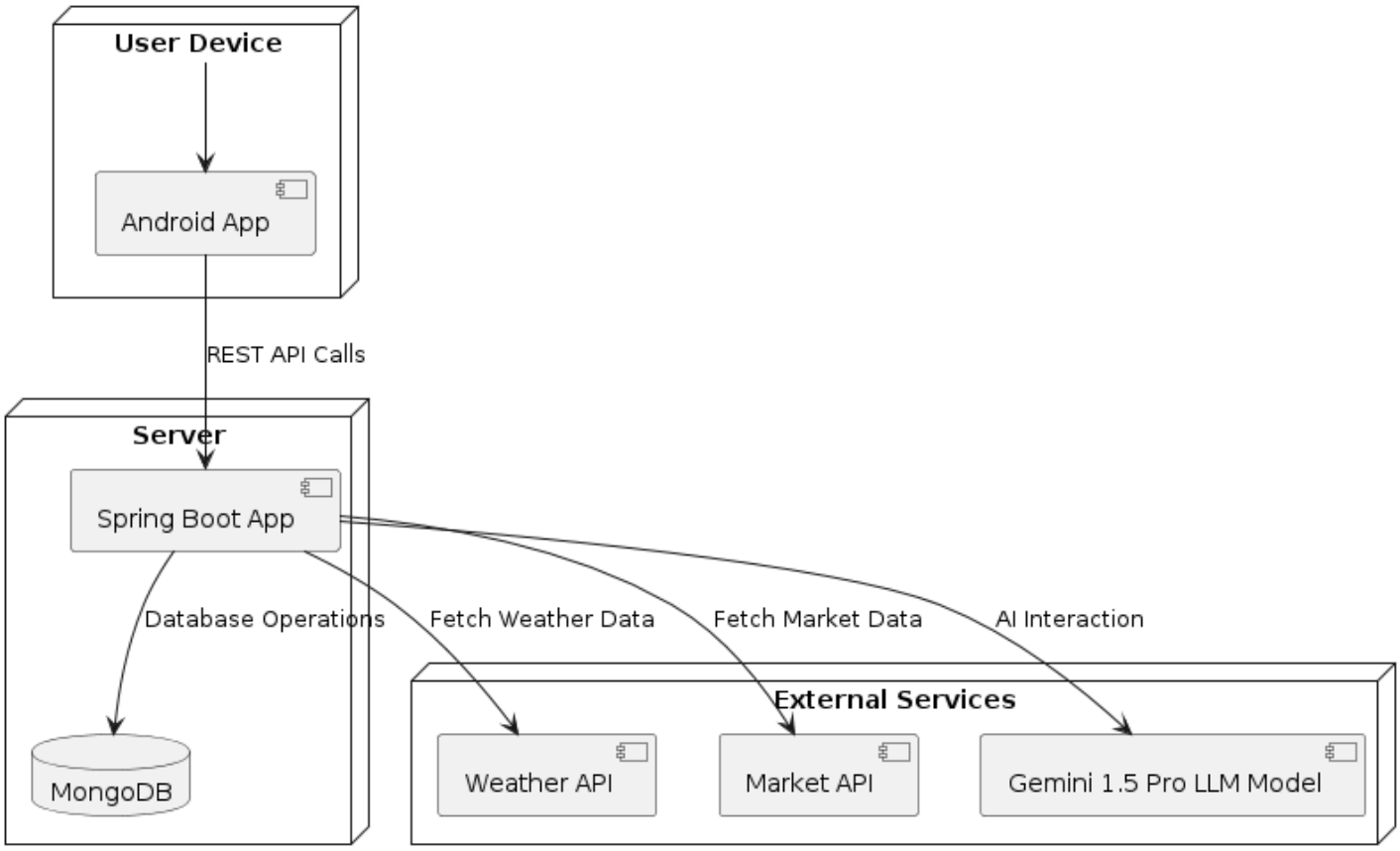

Figure 8: Deployment Diagram for Project

- Component Diagram

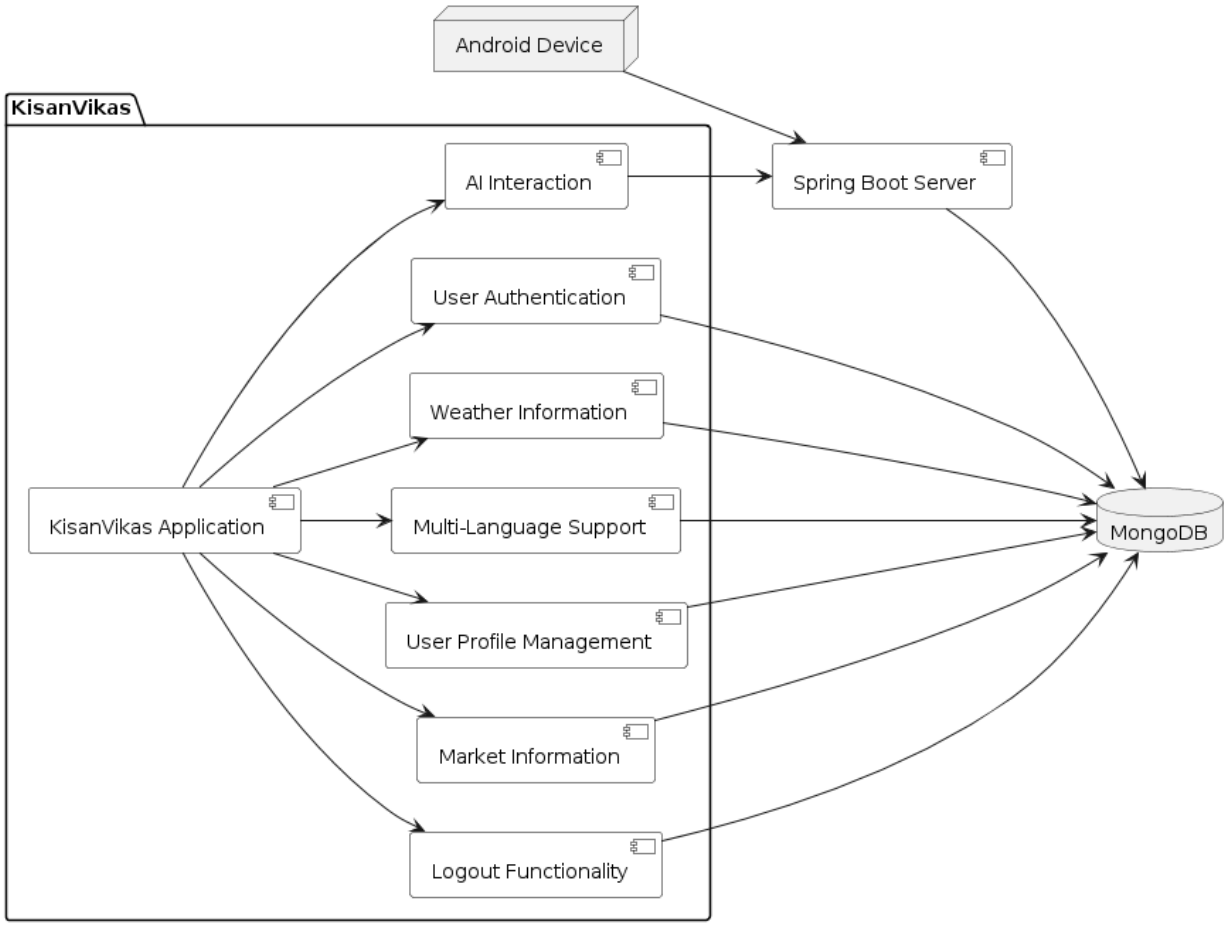

Figure 9: Component Diagram

- Class Diagram

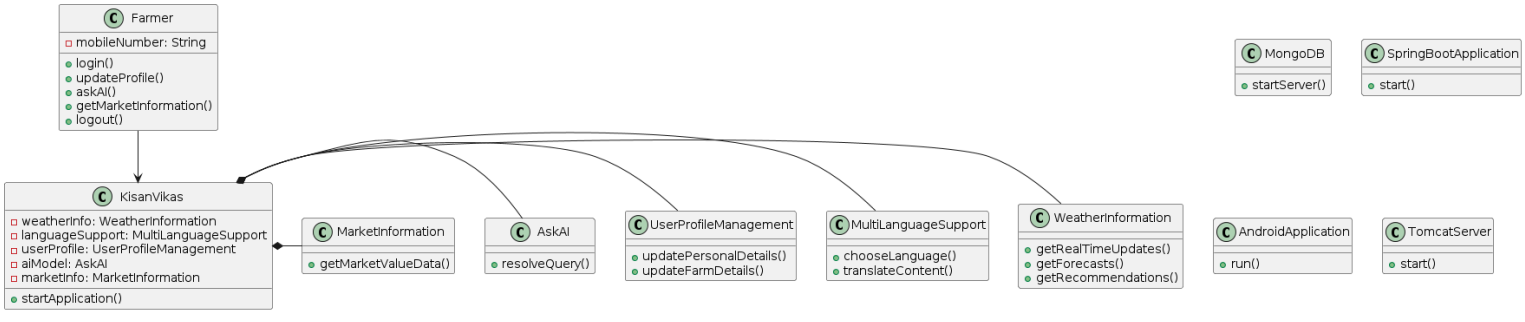

Figure 10: Class Diagram

4.4 Database Diagram

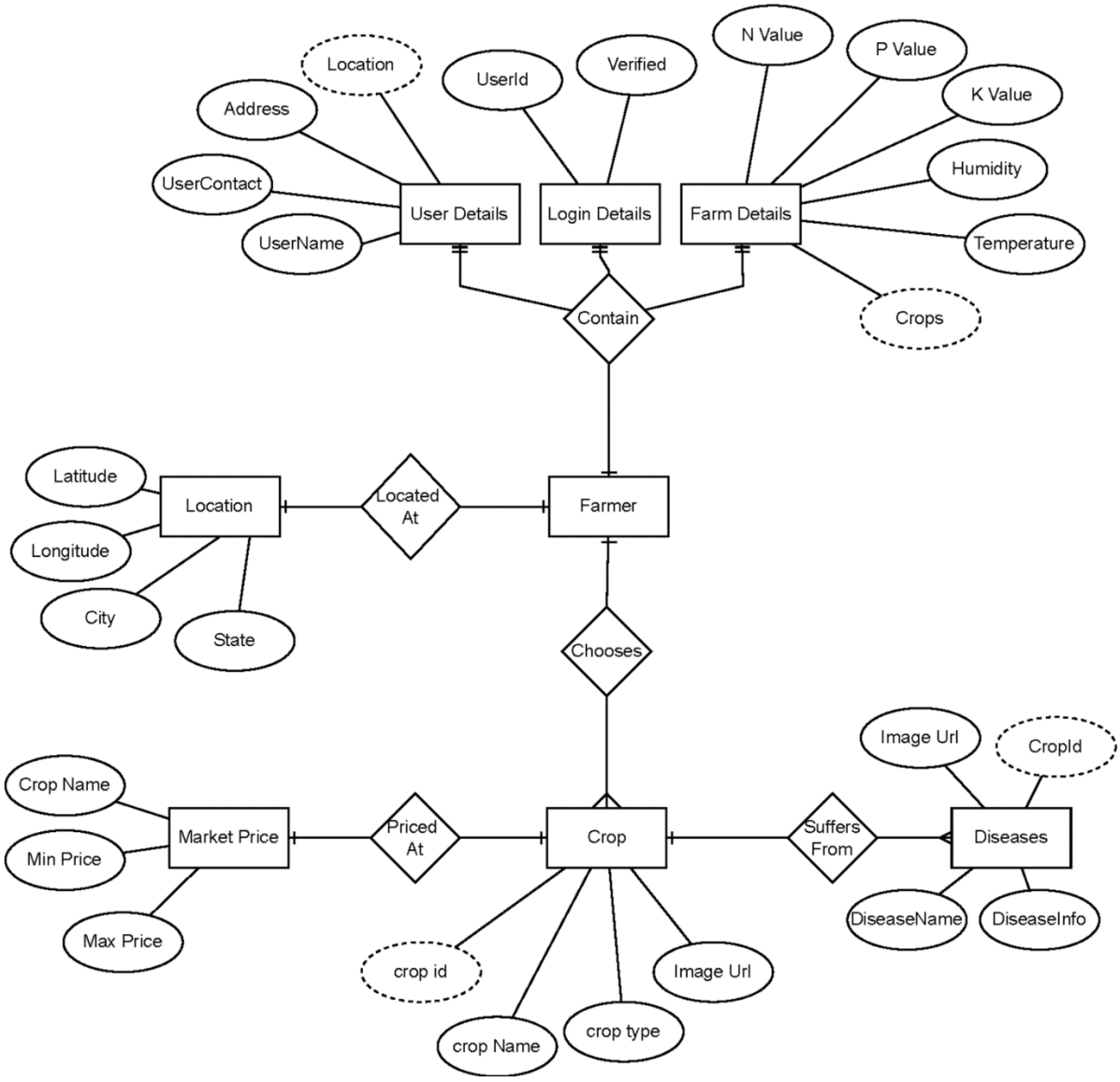

Figure 11: Entity Relationship Diagram

4.5 Assumptions and Dependencies

1. **Technical Feasibility Assumptions:**
   - **Hardware Infrastructure:** The assumption is that the required hardware infrastructure, including servers, network components, and IoT devices, will be available and capable of supporting the system's functionalities.
   - **Software Dependencies:** The project assumes access to necessary software

components like Node.js, Spring Boot, React Native, MySQL, etc., with compatible versions and configurations to run the system smoothly.

- **IoT Device Integration:** Assumes the successful integration and functionality of IoT devices (e.g., soil moisture sensors, temperature sensors) within the system architecture.

2. **Subsystems or Component Availability:**
   - **Availability of APIs:** The project relies on third-party APIs for weather forecasts, market data, etc., assuming their consistent availability and access for the system's functionality.
   - **Database System:** Assumes uninterrupted access to the MySQL database system for data storage and retrieval.

3. **Project-Related Assumptions:**
   - **User Adoption:** Assumes an adoption rate among the target user base (farmers and buyers) for the mobile and web platforms, leading to active system usage.
   - **Data Accuracy:** Assumes the accuracy and reliability of data received from IoT devices, external APIs, and user inputs.

4. **Dependencies on External Factors:**
   - **Internet Connectivity:** Dependencies exist on stable internet connectivity for Farmer to access the system's online features and real-time data.
   - **Regulatory Compliance:** The project assumes adherence to agricultural regulations and standards governing farming practices and data privacy laws.

5. System Architecture

5.1 Client-Server Architecture

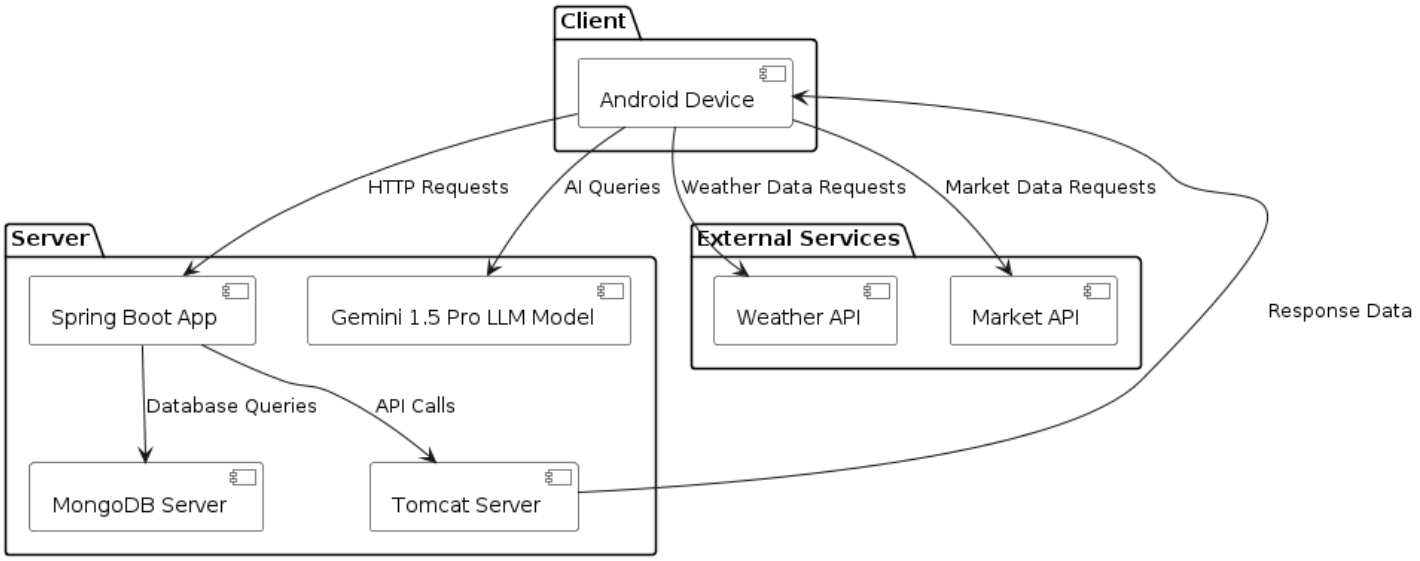

Figure 13: Client Server Architecture

5.2 Communication Interface

Communication interfaces refer to the channels or mechanisms through which various system components, devices, or modules interact and exchange data. Here's an outline of communication interfaces in the project:

API Endpoints:

- **RESTful APIs:** Utilized for communication between the client-side applications (React Native mobile app and web interface) and the server-side (Express and Spring Boot).
- **Endpoints for Data Exchange**: Different API endpoints are defined for functionalities like Farmer's authentication, crop management, weather data retrieval, forum interaction, financial services, and IoT device integration.

External Service Integration:

- **Weather Forecast API:** An interface integrating external weather forecast services, allowing the system to retrieve real-time weather data for farming recommendations.
- **Market Data API**: Integration point for obtaining market-related information and updates for crop selling decisions.

Database Interaction:

- **Database Connectivity:** Interfaces responsible for connecting the backend server (Node.js and Spring Boot) to the MySQL database for data storage and retrieval.
- **Query Interfaces:** Mechanisms facilitating the execution of database queries to manage and retrieve information from the database.

Networking Interfaces:

- **Internet Connectivity:** Ensures connectivity between the client-side applications, server-side components, and external services via the internet.
- **Secure Communication Protocols:** Integration of secure communication protocols (HTTPS, SSL/TLS) for encrypted data transmission between the clients and the server.

These communication interfaces define the pathways and methods through which different system components and external entities interact, ensuring effective data exchange and system functionality within the "Kisan Vikas" agricultural platform.

6. Supporting Information

6.1 Table of Contents

6.2 User Interface Prototypes:

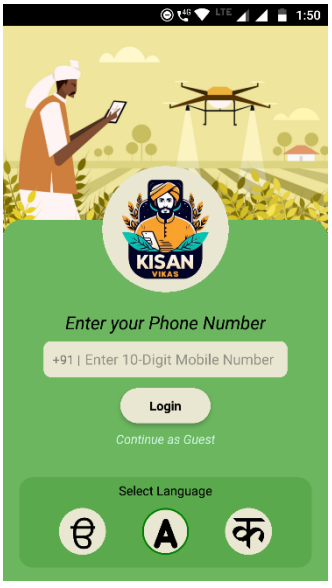

Figure 14: Login Screen

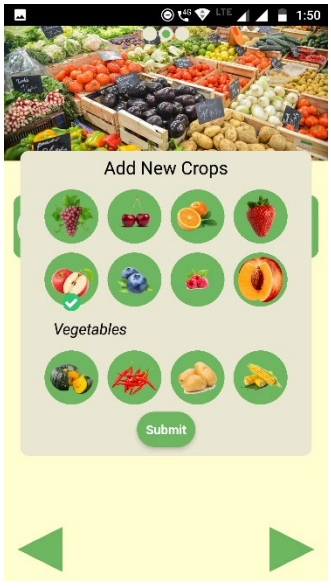

Figure 15: AddNewCrop Screen

Figure 16: Home Screen

7. Conclusion & Future scope

7.1 Conclusion

Summary of Achievements:

- **Accomplishments**: Summarize the achievements and successful implementations within the "Kisan Vikas" project.
- **Key Objectives:** Highlight how the project has addressed its primary objectives, such as providing agricultural support, improving farming practices, etc.
- **Challenges Overcome:** Discuss any challenges faced during the project and how they were mitigated.

Impact:

- **Benefits to Farmer**: Explain how the project's functionalities positively impact farmers' lives by aiding in decision-making, increasing productivity, or providing valuable insights.
- **Contribution to Agriculture:** Discuss the project's role in advancing agricultural practices, promoting better crop yields, and fostering a supportive farming community.

7.2 Future Scope

Potential Enhancements:

- **New Features:** Outline potential additional functionalities that could be integrated, such as advanced analytics, enhanced community engagement features, expanded market analysis, etc.
- **Technological Upgrades:** Discuss opportunities for technological advancements or upgrades that could improve the platform's efficiency or user experience.
- **Scalability Considerations:** Address plans for scalability, catering to an increased user base or expanding services to cover more geographical regions.

Research and Development:

- **AI Integration:** Explore the possibility of integrating more sophisticated AI algorithms for improved crop suggestion, disease detection, or predictive analytics.
- **IoT Integration:** Investigate the integration of IoT sensors for real-time data collection on farms, aiding in precise decision-making.
- **Mobile App Enhancement:** Consider enhancements for the mobile application, optimizing it for various devices and operating systems.

Community Engagement and Partnerships:

- **Collaborations:** Discuss potential partnerships with agricultural experts, government bodies, or NGOs to expand the project's reach and impact.
- **Community Growth:** Highlight strategies to foster a more vibrant and engaged farming community within the platform.

Conclusion of Future Scope:

- **Vision and Direction:** Sum up the future direction of the "Kisan Vikas" project, expressing the vision for its growth, improvements, and continued support for the agricultural community.

8. Concerns / Queries / Doubts if any:

Project-related Queries:

- How can we effectively integrate soil moisture and temperature sensors into our system?
- What methods can be employed to ensure seamless multi-language support in the platform?

Technological Queries:

- Are there any known challenges in implementing specific functionalities using Spring Boot?
- How can we optimize UI development using React Native for mobile applications?
- How can we ensure compatibility and reliability while connecting various IoT sensors to our platform?
- Any key pointers for effectively implementing Agile methodology for iterative development?