CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

## IV Trimester

## MCA B

## SPECIALIZATION Project Documentation

**Department of Computer Science**

# MittiSe – Your All-in-One Farming Assistant

**By**

**Autade Dhiraj Laxman (2447216)**

**Prashanth Singh (2447239)**

**Nitish Churiwala (2447261)**

Under the guidance of

**Dr. Thirunavukkarasu V**

**JUNE 2025**

# Table of Contents

# 1. Synopsis

India's agricultural sector supplies livelihood to over 100 million farmers, the majority of whom have small or marginal landholdings. However, despite the government's multiple initiatives and technological advances, the farmers continue to struggle to access critically needed resources such as real-time market prices, precise weather forecasts, crop planning tools, financial services, welfare schemes, etc. These issues become manifold with literacy barriers and language diversity from one region to another, coupled with unsatisfactory internet connectivity, thus widening the base where policies intend to implement their guidelines on the ground.

The rapid spread of smartphones and the Digital India initiative present a unique opportunity to deliver agricultural intelligence and services directly into the hands of farmers. However, most existing agri-tech applications either focus on a single service or fail to address the unique user experience needs of rural communities.

MittiSe aims to overcome these barriers by operating as a one-stop digital farming assistant. The platform packs government services, AI-driven advisories, real-time data, and community support all into a single, multilingual application that can run in offline mode. With the primary services offered by MittiSe, farmers can communicate and collaborate with nearby agricultural agents, buyers, or service providers who offer various forms of local support. This, in turn, helps with local engagement to fulfil direct transactions, logistics, or advisory needs. The application has an AI-based crop disease prediction feature offering timely warnings to mitigate crop damage sooner so that crop health and yield can be improved. Prioritizing accessibility, inter-activeness, and empowerment to remain with the farmer through the entire agricultural life cycle and realizing the eventual farmer's family needs to make some informed decisions for a better livelihood.

# 2. Introduction

## 2.1 Problem Statement

Despite being the backbone of India’s economy, the agricultural sector—especially for small and marginal farmers—continues to face persistent challenges in productivity, income stability, and access to critical services. Farmers often lack real-time information related to mandi (market) prices, localized weather forecasts, government welfare schemes, and agricultural best practices. This information gap is further exacerbated by barriers such as low digital literacy, lack of internet connectivity in rural areas, limited access to financial and advisory services, and a general dependence on middlemen for selling produce.

Existing agri-tech solutions are often fragmented, language-restrictive, or require continuous internet access, making them less effective or even inaccessible for a large segment of the rural farming population. Additionally, farmers struggle to connect directly with local agents, buyers, or service providers due to the absence of a reliable, location-aware discovery system. Moreover, early detection of crop diseases, which could significantly reduce yield loss, is rarely available in user-friendly or offline formats for the average farmer.

There is a critical need for a comprehensive, inclusive, and intelligent platform that brings together multiple agri-services—market data, weather, government schemes, disease prediction, agent discovery, and community support—into a single, easy-to-use mobile solution. Such a platform must prioritize regional language support, voice-based interaction, offline accessibility, and AI-driven assistance to meet the real-world constraints and needs of farmers in India’s heartland.

MittiSe aims to address this need by offering a unified, vernacular-first, AI-powered farming assistant that empowers farmers with information, connectivity, and autonomy—helping them make better decisions, improve productivity, and build resilient livelihoods.

### 2.2 Objective

The objective of the MittiSe project is to develop an AI-powered, vernacular-first mobile application that empowers small and marginal farmers in India by bridging gaps in access to agricultural information, markets, government schemes, and advisory services. Agriculture remains the backbone of India’s economy, yet rural farmers face persistent challenges such as language barriers, digital illiteracy, unreliable internet connectivity, and lack of timely information. MittiSe addresses these issues through an inclusive, user-centric platform that works both online and offline.

The app delivers real-time mandi price updates and hyper-local weather forecasts by integrating trusted APIs such as Agmarknet and OpenWeatherMap/IMD. It features a government scheme discovery tool powered by an AI-based eligibility calculator that recommends relevant schemes based on user profile, location, and landholding data. A built-in document management system enables users to securely upload and store essential records like Aadhaar, land documents, and bank details.

A key innovation is the integration of a voice-enabled AI assistant using Ollama and Whisper models to provide natural language support in regional languages—making the platform accessible even to low-literate farmers. In addition, MittiSe includes an AI-based crop disease prediction feature, which allows farmers to detect crop issues early by uploading images or describing symptoms, receiving timely preventive advice.

MittiSe also focuses on local empowerment by enabling farmers to discover and connect with nearby agricultural agents, buyers, transporters, and service providers, fostering direct transactions and logistical collaboration. To strengthen community engagement, the platform includes a peer-support forum where users can post queries, share best practices, and learn from each other.

Built on a modular and scalable architecture, MittiSe supports offline functionality through local caching and is designed for future integration with IoT sensors, satellite data, and advanced crop analytics.

In essence, MittiSe aims to serve as a comprehensive digital farming assistant that democratizes access to agricultural intelligence, promotes self-reliance, and enhances the productivity and well-being of farmers across rural India.

## 2.2 Target Audience

- Small and Marginal Farmers
  MittiSe is primarily designed for small and marginal landholders (owning less than 2 hectares of land), who make up over 85% of India's farming population. These farmers often face barriers in accessing real-time information, agricultural services, and government benefits due to low digital literacy, limited resources, and rural isolation.

- Women Farmers and Rural SHG Members
  The app supports voice-based navigation and vernacular language interfaces, making it inclusive and accessible to women farmers and members of rural self-help groups (SHGs), many of whom are increasingly taking leadership in agricultural activities.

- Agricultural Agents and Input Suppliers
  MittiSe facilitates agent discovery and transaction support, enabling agri-input dealers, machinery service providers, transporters, and mandi agents to connect directly with farmers for efficient service delivery and market access.

- Agri-Entrepreneurs and FPOs (Farmer Producer Organizations)
  The platform can be used by progressive farmers, agri-startups, and FPOs to expand their reach, share advisories, list services, or sell produce in bulk through local or nearby networks.

- Rural Youth and First-Time Smartphone Users
  Designed with offline functionality and simplified UX, MittiSe also targets rural youth who are early adopters of technology and can act as digital enablers for their communities.

# 3. Software Requirements Specification (SRS)

The Software Requirements Specification (SRS) outlines the functional and non-functional requirements, system interfaces, and constraints for the MittiSe application.

## 3.1 Purpose

The purpose of this SRS is to provide a detailed description of MittiSe's expected behavior, system features, and performance requirements. It serves as a foundation for design, development, testing, and maintenance.

## 3.2 Scope

MittiSe is a mobile-first farming assistant that integrates:

- Provide real-time market and weather data using APIs.
- Offer multilingual AI-powered advisory through voice/text interface.
- Help users discover and apply for government schemes using an eligibility engine.
- Enable upload and storage of essential documents securely.
- Allow farmers to discover and connect with nearby agents, buyers, and service providers.
- Include crop disease detection using AI and computer vision.
- Facilitate peer-to-peer knowledge sharing through a built-in community forum.

## 3.3 Definitions, Acronyms, and Abbreviations

- AI – Artificial Intelligence: Used for advisory, crop suggestions, and disease detection
- API – Application Programming Interface: Enables integration with external data sources
- IMD – India Meteorological Department: Source for weather data
- LLM – Large Language Model: Powers the offline AI assistant
- PWA – Progressive Web App: Supports offline access and app-like experience
- UI/UX – User Interface / User Experience: Focused on simplicity and vernacular usability

## 3.4 Functional Requirements

These define the essential functionalities that the system must support:

**User Management**

- OTP-based or password-based registration/login for farmers and agents
- Profile creation including name, language, location, crops grown, land size, etc.
- Role-based access: Farmer or Agent Dashboard with personalized features

**Marketplace Module**

- Post, view, and search agricultural products and services (e.g., seeds, tools, transport)
- Real-time stock availability and pricing updates by sellers/agents
- In-app messaging or calling to initiate deals between buyers and sellers

**Mandi Prices & Weather Updates**

- Display real-time mandi (market) prices using Agmarknet or similar APIs
- Hyper-local weather forecasts via OpenWeatherMap or IMD APIs
- Alert system for price dips, extreme weather, or favorable conditions

**Government Schemes & Eligibility Calculator**

- List of central/state agricultural and welfare schemes
- AI-based eligibility checker using user profile (e.g., landholding, crop type, location)
- Display application steps, documents required, and relevant URLs

**Voice & Vernacular AI Assistant**

- Voice-to-text input support in multiple Indian languages
- Conversational AI for queries related to crops, schemes, market, weather, etc.
- Runs on-device (Ollama) with fallback to cloud model when online

**Document Management**

- Upload, store, and manage Aadhaar, land records, passbooks, certificates
- Offline access and cloud sync for important documents
- Share documents securely with agents or for scheme applications

**Peer Support & Community**

- Join regional/topic-based forums for crop issues, techniques, and discussions
- Post questions, images, or voice messages to receive responses from peers or AI
- Community moderation and flagging for safety

**Agent Discovery & Communication**

- Search and discover nearby agents, suppliers, buyers, or transporters
- Initiate voice calls, text chats, or consultation bookings
- View agent profiles, reviews, and service areas

**Disease Prediction & Crop Health Advisory**

- Upload crop images or describe symptoms via voice
- Detect crop diseases using AI (image analysis + symptom matching)
- Recommend preventive or corrective measures with local language instructions

**Offline Functionality**

- Cache mandi data, schemes, crop images, and user interactions for offline access
- Queue transactions, messages, and advisory queries to sync automatically when online
- Provide essential functionality even in low-connectivity zones

### 3.5 User Interactions and Use Cases

**Use Case 1: Register and Create Profile**

Actor: Farmer or Agent
Precondition: App is installed
Flow:

- User selects role (Farmer or Agent)
- Enters mobile number and OTP

- Fills in profile details (name, region, language, crops)
- Dashboard loads based on role

**Use Case 2: Check Mandi Prices and Weather**
Actor: Farmer
Precondition: User is logged in
Flow:

- User opens "Market Info"
- App shows local mandi prices and weather forecast
- User can view more details if needed

**Use Case 3: Find and Apply for Government Schemes**
Actor: Farmer
Precondition: Profile is filled
Flow:

- User opens "Schemes" section
- App shows eligible schemes based on user data
- User views details and follows steps to apply

**Use Case 4: Use AI Voice Assistant**
Actor: Farmer
Precondition: Mic access is given
Flow:

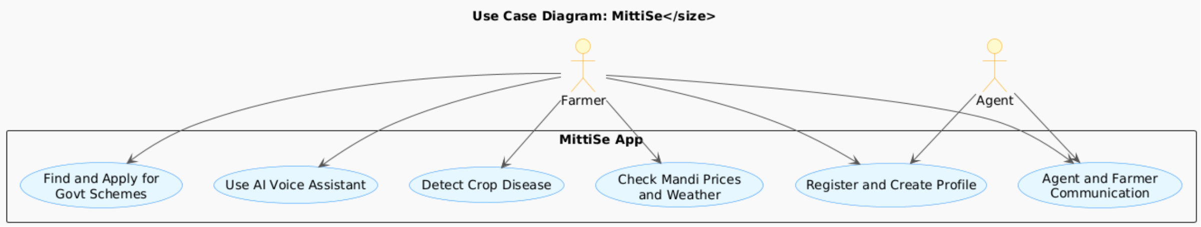
- User taps mic and asks a question
- App converts voice to text
- AI gives response in selected language

**Use Case 5: Detect Crop Disease**
Actor: Farmer
Precondition: Camera is enabled
Flow:

- User opens "Crop Health"
- Takes or uploads photo of crop
- App checks for disease and gives advice

**Use Case 6: Agent and Farmer communication**
Actor: Farmer and Agent
Flow:

- Agent can view nearby farmers and what they have listed on sale
- Farmer can view the agents and contact them
- Both can communicate through shared communication medium


Use Case Diagram: MittiSe</size>
Farmer
Agent
MittiSe App
Find and Apply for Govt Schemes
Use AI Voice Assistant
Detect Crop Disease
Check Mandi Prices and Weather
Register and Create Profile
Agent and Farmer Communication

## 3.6 Non-Functional Requirements

These define the quality and constraints of the system:

**Performance**

- Response time < 2 seconds for major functions
- AI voice query response time < 3 seconds (if local LLM)

**Scalability**

- Modular architecture for adding new features (IoT sensors, drone mapping)
- Horizontal scalability for marketplace traffic

**Reliability & Fault Tolerance**

- Local data caching and background syncing for offline support
- Graceful fallback if APIs (e.g., Agmarknet) fail

**Security**

- AES encryption for documents
- Role-based access control (RBAC)
- OTP or 2FA for critical actions

**Usability**

- Vernacular UI (Hindi, Kannada, Tamil, etc.)
- Large buttons, voice-first UX
- Intuitive navigation (bottom tab, side drawer)

**Maintainability**

- Clear modular codebase using MVC/MVVM
- RESTful APIs or GraphQL with clean documentation
- Dockerized backend services for easy updates

**Compliance**

- Adherence to Indian data privacy norms (DPDP Bill)
- Integration-ready for DigiLocker, DBT, and PM-Kisan databases

### 3.7 Data Management and Storage

1. User Data

Stored in a relational database with encryption for sensitive fields.

2. Marketplace Listings

Product details and images stored in database and cloud storage respectively.

3. Government Schemes Database

Stored as structured JSON or in a relational table for quick eligibility checks.

4. Weather and Mandi Price Data

Cached locally for offline access and periodically refreshed via APIs.

5. Voice and AI Interaction Logs

Stored as lightweight logs with optional user consent for advisory tracking.

6. Document Uploads

Encrypted documents stored securely in cloud storage (e.g., Firebase or AWS S3).

7. Offline Cache (Mobile Side)

Data like schemes, profile, and advisories stored in local storage for offline use.

## 4. System Design

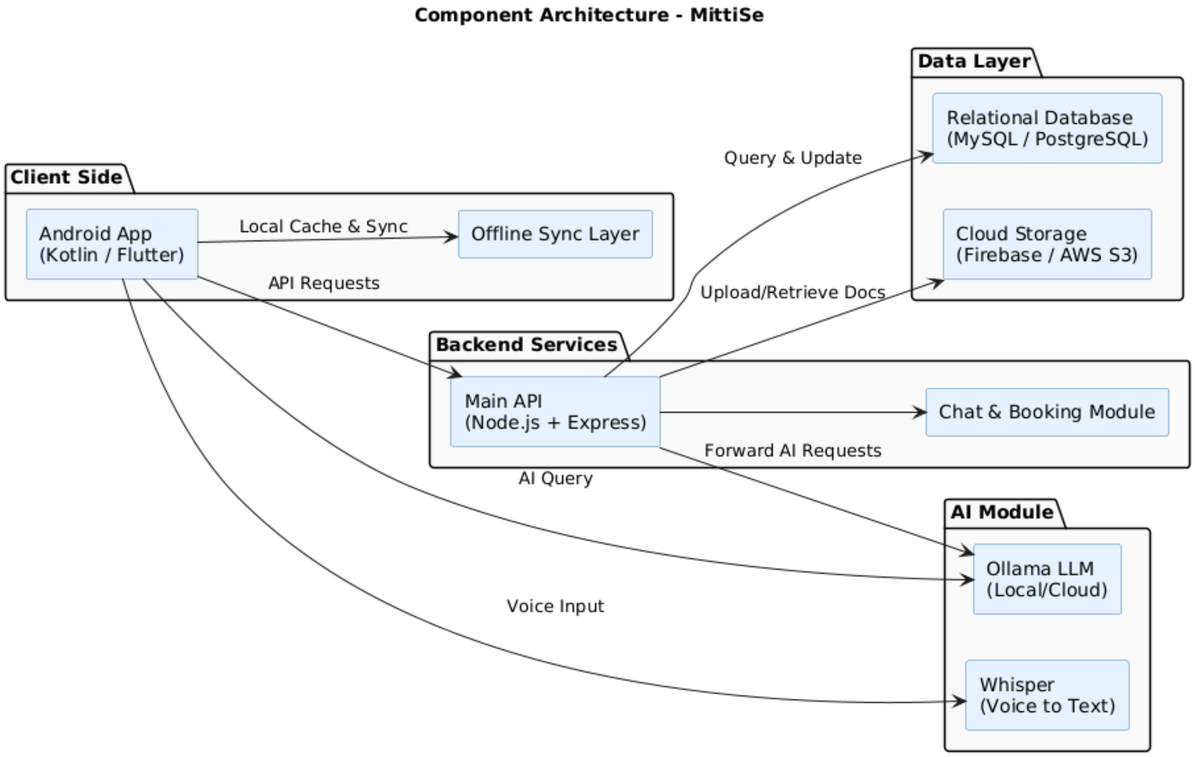
This section provides a comprehensive breakdown of MittiSe's architecture, workflows, component interactions, and deployment considerations.

### 4.1 Component Architecture

- Frontend: Android app built with Kotlin or Flutter; offers vernacular UI, voice input, offline caching, and farmer-friendly navigation.
- AI Module: Embedded Ollama-based LLM running locally (or fallback cloud API) for voice advisory, crop suggestions, and disease detection.
- Backend API: Node.js with Express.js; handles authentication, scheme eligibility logic, crop advisory requests, and agent-farmer communication.
- Database: Cloud-hosted MySQL or PostgreSQL for managing user profiles, listings,

documents, and chat records.

- CDN/Storage: Firebase Storage or AWS S3 for storing KYC documents, crop images, and product photos with secure access controls.
- Offline Sync Layer: Local storage and background service to queue actions, sync data, and ensure app functionality in low-connectivity areas.



## 4.2 Sequence Diagram

1. App Launch & User Authentication

- User opens MittiSe app → UI displays login screen → User enters phone number → App sends POST /auth/otp → Backend verifies and returns token → App stores token locally.

2. Profile Setup

- On first login, app prompts profile setup → User selects role (Farmer/Agent), language, crops → App sends POST /user/profile → Backend saves profile.

3. Mandi & Weather Fetch

- User taps “Market Info” → App sends GET /mandi-prices and /weather?region=X → Backend fetches data via Agmarknet & Weather API → Returns mandi rates and forecasts → App displays info.

4. Government Scheme Discovery

- User taps “Schemes” → App sends GET /schemes?region=X → Backend filters from DB → Returns list of applicable schemes → App displays with apply links or steps.

5. Voice Advisory & Disease Detection

- User taps mic or advisory → App records voice → Sends audio to Whisper for STT → Text sent to /ai/advice → Ollama LLM responds with guidance → App displays message
- For crop issues, user uploads image → App sends image to /ai/disease-check → AI module returns disease info and solution → App displays advice.

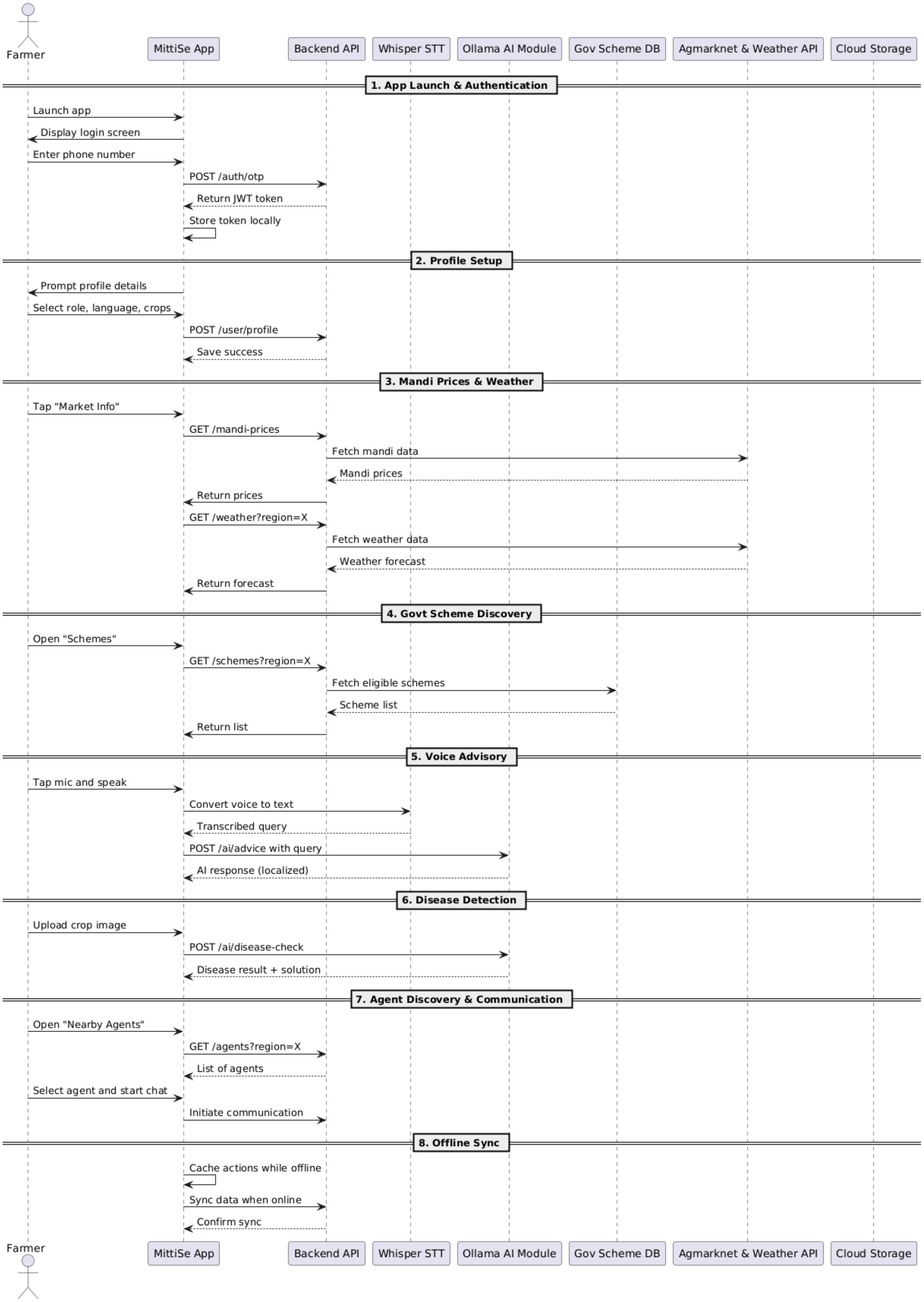
6. Agent Discovery & Communication

- User opens “Nearby Agents” → App sends GET /agents?region=X → App shows list → User selects one → Initiates chat or call via communication module.

7. Offline Sync

- If offline, data requests (e.g. advisory, scheme checks) are queued → On reconnect, background sync sends data to backend → Sync module confirms delivery and updates UI.

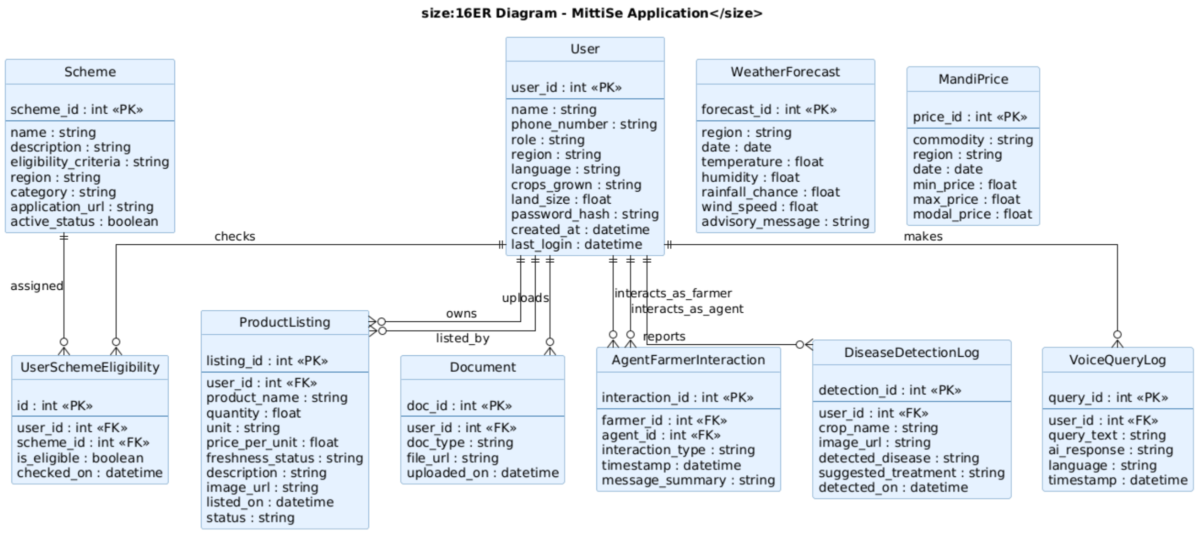
Sequence Diagram: MittiSe



## 4.3 Deployment Architecture

- Frontend builds are deployed via Google Play Store for farmers and agents, with support for PWA (Progressive Web App) for browser-based offline access.
- AI Module (Ollama + Whisper) is hosted on a dedicated on-premise edge server or fallback to cloud containers (e.g., Dockerized service on Google Cloud Run) with API key authentication.
- Backend API is hosted on scalable Node.js containers using services like AWS ECS or Google Cloud Run, with auto-restart and horizontal scaling enabled.
- Database is provisioned via managed PostgreSQL or MySQL on Cloud SQL, with daily backups, geo-redundancy, and encrypted connections.
- Media and document storage is handled using Firebase Storage or AWS S3 with access control via signed URLs.
- CDN (e.g., Cloudflare or Firebase Hosting) is used for fast delivery of images, scheme data, advisory content, and static assets to remote regions.

## 4.4 Entity Relationship Diagram



### Entities & Attributes

1. **Users**

- user_id (PK): Unique identifier for each registered user.
- name, phone_number, role (Farmer/Agent), region, language, crops_grown, land_size: Core profile attributes.
- created_at, last_login: Timestamps to monitor registration and activity.
- Styling: Soft green background (#F0FFF0) highlights "Users" as the central actor entity in MittiSe.

2. **ProductListings**

- listing_id (PK): Unique identifier for each product listed.
- user_id (FK): Linked to the user who created the listing.
- product_name, quantity, unit, price_per_unit, freshness_status: Essential marketable information.
- description, image_url, listed_on, status: Display content and listing lifecycle metadata.
- Styling: Beige background (#FFFACD) represents active marketplace data.

3. **Schemes**

- scheme_id (PK): Unique ID for each government welfare scheme.
- name, description, eligibility_criteria, region, category: Informational fields to filter applicability.
- application_url, active_status: Actionable attributes for enrollment.
- Styling: Pale blue background (#E0FFFF) reflects trusted external programs and assistance.

4. **UserSchemeEligibility**

- id (PK): Unique record for each scheme eligibility check.
- user_id (FK), scheme_id (FK): Cross-references between users and schemes.
- is_eligible: Boolean status.
- checked_on: Timestamp of verification.
- Styling: Light steel blue background (#B0C4DE) visually connects user-scheme relationships.

5. **VoiceQueryLogs**

- query_id (PK): Unique record of each voice advisory query.
- user_id (FK), query_text, ai_response, language, timestamp: Log of each interaction with AI.
- Styling: Light lavender (#E6E6FA) denotes conversational features.

6. **DiseaseDetectionLogs**

- detection_id (PK): Unique log of each disease detection event.
- user_id (FK), crop_name, image_url, detected_disease, suggested_treatment, detected_on: Diagnosis record.
- Styling: Misty rose (#FFE4E1) to signify health-related analysis.

7. **Documents**

- doc_id (PK): Unique ID per uploaded document.
- user_id (FK), doc_type, file_url, uploaded_on: Stores Aadhaar, bank passbook, land record etc.

8. **MandiPrices**

- price_id (PK): Record per commodity per region per day.
- commodity, region, date, min_price, max_price, modal_price: Structured market price reference.
- Styling: Light goldenrod (#FAFAD2) for market-facing values.

**9. WeatherForecasts**

- forecast_id (PK): Daily forecast per region.
- region, date, temperature, humidity, wind_speed, rainfall_chance, advisory_message.
- Styling: Alice blue (#F0F8FF) aligns with environmental data.

**10. AgentFarmerInteractions**

- interaction_id (PK): Communication or booking instance.
- farmer_id (FK), agent_id (FK), interaction_type (Chat/Call/Booking), message_summary, timestamp.
- Styling: Cornsilk (#FFF8DC) emphasizes collaboration and communication history.

Relationships

- Users → ProductListings (1-to-many): Each user can post multiple agri-product listings.
- Users → VoiceQueryLogs (1-to-many): Each user can engage in multiple AI advisory interactions.
- Users → DiseaseDetectionLogs (1-to-many): Users can diagnose different crops over time.
- Users → Documents (1-to-many): Every user may upload various legal/agricultural documents.
- Users → AgentFarmerInteractions (1-to-many as both farmer_id and agent_id): Supports bidirectional local collaboration.
- Users → UserSchemeEligibility (1-to-many): Scheme access logs linked per user.
- Schemes → UserSchemeEligibility (1-to-many): Tracks multiple eligible users per scheme.
- ProductListings → Users (many-to-1): Listings belong to the user who posted them.

## 4.5 Performance & Scalability Considerations

- Leverage offline caching and local storage for marketplace and scheme data using PWA principles.
- Use Redis or similar in-memory caching for frequent data (mandi prices, weather).
- Implement async queuing for heavy AI tasks (disease detection, voice queries).
- Optimize database indexes and enable read replicas for scale.
- Integrate load balancing and autoscaling for backend containers during crop seasons.

### 4.6 Monitoring & Maintenance

- Logging: Central logs via Google Cloud Logging or ELK Stack for backend + AI module observability.
- Monitoring: Prometheus & Grafana dashboards for resource usage, latency, and traffic patterns.
- Alerts: Triggered for AI timeout, API failures, and regional connectivity issues.
- CI/CD: GitHub Actions or GitLab CI pipelines for auto-deployment and test workflows.

### 4.7 Future Enhancements

- Satellite and IoT Integration: Enable NDVI-based crop health and soil sensor input.
- Smart Recommendations: Dynamic pricing tips and crop planning based on AI analysis.
- Voice Agent in Native Dialects: Extend LLM prompts to regional dialects via Whisper + TTS.
- Blockchain Traceability: For produce sold via the marketplace to track origin and quality.
- Local Agent Onboarding Portal: Agents can register and manage logistics/advice centers.

## 5. Conclusion

MittiSe represents a transformative step toward bridging the digital divide in Indian agriculture by delivering an inclusive, AI-powered mobile platform tailored to the real-world needs of small and marginal farmers. By integrating key agricultural services such as real-time mandi prices, localized weather updates, welfare scheme discovery, document management, and voice-enabled AI assistance into a single vernacular-first application, MittiSe empowers farmers with knowledge, access, and autonomy. The platform's offline functionality, agent discovery features, and peer-support community further ensure that farmers are not just passive recipients of information but active participants in their agricultural journey. Additionally, features like AI-driven crop disease prediction enhance decision-making and preventive care at the grassroots level. With a modular and scalable technical architecture, MittiSe is not only a solution for present-day challenges but also a foundation for future innovations in agri-tech. It redefines how technology can serve the soil ensuring sustainable growth, digital inclusion, and dignified livelihoods for the farmers of Bharat.