Riphah International University Lahore, Pakistan

Riphah School of Computing & Innovation

Final Year Project

PROJECT REPORT

Harvest Hub

Project Team

Student Name	Student ID	Program	Contact Number	Email Address
Laiba Ahmad	39102	BSCS		
Areesha Mushtaq	38617	BSCS		

Dr. Fawad

APPROVAL

_____

PROJECT SUPERVISOR

Comments: _____

_____

Name:_____

Date:_____ Signature:_____

_____

PROJECT MANAGER

Comments: _____

_____

Date:_____ Signature:_____ **HEAD OF THE DEPARTMENT**

Comments: _____

_____

Date:_____ Signature:_____

Executive Summary

Harvest Hub is a digital platform created to tackle key challenges in Pakistan's agricultural sector, such as exploitation by middlemen, limited access to tools, labor shortages, and outdated practices. By offering a user-friendly solution, it empowers both small- and large-scale farmers.

The platform combines essential features into one system: a direct marketplace, equipment rentals, labor hiring, and secure digital payments. It also includes real-time weather updates, crop advisory services, and multilingual support. Built with scalable technologies like Flutter and Firebase, Harvest Hub ensures accessibility, security, and ease of use for rural communities.

With features like user ratings, admin oversight, and Agile-based development, Harvest Hub aims to increase farmer productivity, improve income, and modernize agriculture. It stands as a powerful step toward a more sustainable and equitable farming ecosystem in Pakistan.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Pakistan, despite being an agricultural powerhouse, sees its farmers grapple with numerous daily challenges. Unfair pricing practices and the dominance of middlemen in the market significantly reduce farmers' profits. They face limited access to affordable equipment and essential resources, while outdated irrigation systems and expensive tools further hinder productivity. Rising input costs and a lack of modern techniques make farming increasingly unsustainable, especially for small-scale farmers. Many struggle to find reliable labor and face barriers in adopting innovative solutions. These challenges create a cycle of low yield, poor income, and limited growth opportunities for the agricultural community.

1.1. Background

Pakistan's agricultural sector is a cornerstone of the economy, yet the majority of small-scale farmers face significant systemic challenges. Unfair pricing structures driven by middlemen, inadequate access to affordable tools, and inefficient water management systems have led to decreased productivity and profit margins. The reliance on outdated equipment and a lack of direct marketplaces have made it increasingly difficult for farmers to sustain their livelihoods. Harvest Hub seeks to transform this landscape by leveraging technology to create a more equitable and efficient agricultural ecosystem.

1.2. Motivations and Challenges

The motivation behind Harvest Hub stems from the urgent need to address the inefficiencies and inequities in Pakistan's farming practices. Farmers are often exploited by middlemen, lack access to expert guidance, and struggle with high input costs and poor infrastructure. The challenges include bridging the digital divide in rural areas, building trust in technology-based solutions, and ensuring affordability and usability for non-tech-savvy users. Despite these challenges, the potential impact of a well-designed digital platform can be transformative

1.3. Goals and Objectives

Goals:

To digitally empower farmers by providing an integrated platform that connects them directly to marketplaces, labor, and essential resources.

Objectives:

- Eliminate middlemen to ensure fair pricing.
- Enable affordable equipment rentals.
- Provide access to a labor network for seasonal and daily tasks.
- Ensure secure and user-friendly transactions via web and mobile apps.

1.4. Literature Review/Existing Solutions

Feature Category	HarvestHub	Khaity.pk	Kheti-Badi	AgriMart	Kissan Ghar	Agri-Route	GearTap	KissanKraft	Zaraee.pk
User Management									
Registration (Farmer/Buyer)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Email Verification	✓	✓	✓	✓	✓	✓	✓	✓	✓
Profile Management	✓	✓	✓	✓	✓	✓	✓	✓	✓
Resource & Equipment Management									
Equipment Rental System	✓	✗	✗	✗	✗	✗	✓	✗	✗
Manage Rentals	✓	✗	✗	✗	✗	✗	✓	✗	✗
Labor Management									
Laborer Registration	✓	✗	✗	✗	✗	✗	✗	✗	✗
View Labor Listings	✓	✗	✗	✗	✗	✗	✗	✗	✗
Crop Information & Advisory									
Crop Details & Specs	✓	✓	✓	✓	✓	✓	✓	✓	✓
Real-time Updates	✓	✗	✓	✗	✗	✗	✗	✗	✗
Weather Forecasts	✓	✗	✓	✗	✗	✗	✗	✗	✗
Inventory & Notifications									
Stock Tracking	✓	✗	✗	✗	✗	✗	✗	✗	✗
Real-time Alerts	✓	✓	✓	✓	✓	✓	✓	✓	✓
Transaction & Payments									
Secure Payment Gateway	✓	✓	✓	✓	✓	✓	✓	✓	✓
Multiple Payment Options	✓	✓	✓	✓	✓	✓	✓	✓	✓
Reporting & Feedback									
Report Another User	✓	✗	✗	✗	✗	✗	✗	✗	✗
Ratings & Feedback	✓	✓	✓	✓	✓	✓	✓	✓	✓

1.5. Gap Analysis

- Harvest Hub addresses critical gaps left by existing solutions.
- Comprehensive Marketplace: Unlike competitors, HarvestHub integrates buying, selling, rentals, and labor services in one platform.
- Scalability: Designed to cater to small and large-scale farmers alike.
- User-Centric Design: Focuses on accessibility and ease of use for farmers with limited digital literacy.

1.6. Proposed Solution

Harvest Hub is a one-stop digital platform that combines marketplace functionalities, equipment rentals and labor connectivity. Key features include:

Direct Marketplace: Farmers can trade seeds, fertilizers, and tools without intermediaries.

Rental System: Affordable access to tractors, harvesters, and other machinery.

Labor Network: Connects farmers with skilled laborers for timely assistance.

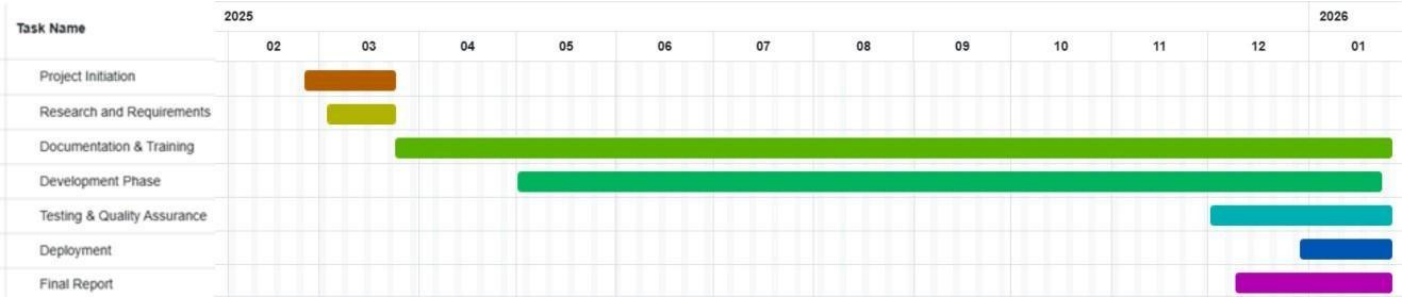
Secure Transactions: Ensures safe and transparent financial operations.

By integrating these features, Harvest Hub aims to transform agriculture into a more profitable and sustainable sector.

1.7. Project Plan

The project will be executed in a phased manner following Agile-Scrum methodology. The initial phase includes requirement analysis and system design, followed by iterative development and testing cycles, and concludes with deployment and feedback incorporation. The project uses tools like Flutter, Dart, Firebase, and Android Studio for development.

1.7.1. Gantt Chart



Chapter 2

1.1. Introduction

This document outlines the Software Requirements Specification (SRS) for HarvestHub(digital platform) as we know Punjab is the heart of agriculture in Pakistan, with vast fertile lands and hardworking farmers.

Despite its agricultural strength, endless golden fields feeding millions yet the hands that sow these seeds struggle daily. Broken systems, dry canals, and middlemen squeezing profits leave farmers fighting just to survive. A digital lifeline for farmers to sell seeds, fertilizers directly, cutting out exploitative middlemen. Rent and leace agricultural tools by the hour or more as per need. Connect laborers to farms with a single tap. The SRS serves as a blueprint for development, ensuring all stakeholders, developers, testers, project managers, and end-users also have a clear understanding of the system's functionalities, constraints, and goals.

1.2. Purpose

The purpose of this SRS is to:

- Define the functional and non-functional requirements of HarvestHub.
- Establish a common understanding between developers and stakeholders.
- Guide the design, development, and testing phases.
- Ensure compliance with agricultural and digital transaction standards.
- Additionally, the SRS serves as a valuable resource for the development team, guiding them throughout the development process. Furthermore, it provides a basis for validation and verification activities to ensure that the software meets the specified requirements and aligns with the user's expectations.

1.2.1. Document Conventions

- **Font size:** 12
- **Font Style:** Calibri
- **Bullets:** Number format
- **Headers:** Bold with hierarchical numbering (e.g., 1.1, 1.1.1) **Priority Levels:**

High (H): Core features (e.g., user authentication, marketplace)

Medium (M): Secondary features (e.g., weather integration)

Low (L): Enhancements (e.g., multi-language support)

1.2.2. Intended Audience

Developers: For system implementation.

Testers: To validate functionality against requirements.

Farmers & Laborers: End-users who will interact with the platform.

Project Managers: To track progress and deliverables.

1.3. Product Scope

HarvestHub is designed to revolutionize the agricultural ecosystem by providing a comprehensive digital marketplace for farmers. It enables them to directly buy and sell essential items such as seeds, fertilizers, and farming tools, eliminating the need for middlemen. The platform also offers affordable equipment rentals, including tractors and harvesters, making advanced machinery accessible to small-scale farmers. Additionally, It connects farmers with laborers for seasonal work, streamlining the hiring process. To ensure safety and convenience, all transactions are secured through an integrated payment gateway.

1.3.1. References

This **SRS** is referred to as IEEE Std. 830-1998 (SRS Guidelines)

- Punjab Agriculture Department Reports (2023)
- World Bank Agri-Report (2022)

1.3.2. Product Perspective

Admin can enroll different person so they can easily use this system.

HarvestHub is a standalone web and mobile application integrating:

Frontend: Flutter (cross-platform compatibility).

Backend: Firebase (real-time database, authentication).

External Systems: Weather APIs, payment gateways.

1.3.3. Product Functions

HarvestHub is structured into several key modules, each offering essential features tailored to the needs of the agricultural community. The User Management module allows both farmers and laborers to register, log in, reset passwords, and manage their profiles. Through the Marketplace, users can buy or sell agricultural inputs such as seeds, fertilizers, and tools, with the ability to create and edit their listings. The Equipment Rental module enables farmers to rent machinery like tractors and harvesters and manage their rental schedules efficiently. In the Labor Network, farmers can hire laborers, browse through their profiles, and view ratings to make informed decisions. Finally, the Notifications module keeps users informed with timely alerts about transactions, weather updates, and labor matches, ensuring they stay connected and prepared., weather updates, and labor matches, ensuring they stay connected and prepared.

1.3.4. User Classes

User Type	Characteristics
Farmers	Small/large-scale; need tools, labor, and fair pricing.
Laborers	Daily wage workers; seek job opportunities.
Buyers	Purchase farm produce directly from farmers.
Admins	Manage disputes, user reports, and system analytics.

1.3.5. Operating Environment

Mobile: Android/iOS (Flutter).

Web: Chrome, Firefox.

This system will be operated in any Environment, you just need internet and web browsers.

1.3.6. Constraints

Internet dependency for real-time features.

Regional language support (Urdu, English).

The platform, as e-commerce system, must prioritize the safety and security of user-provided information. Firstly, HarvetsHub provides its user with security to manage who can view and edit their data. Secondly, it has encrypted database, So the admins of system have no access to their data. Thirdly, HarvestHub strictly prohibits the unauthorized sharing or selling of users' private information to any third party.

1.3.7. User Documentation

User Documentation will be mentioned once the project is ready to be delivered.

1.5. External Interface Requirements

1.5.1. User Interfaces

Login Screen: Email/phone, password.

Marketplace Dashboard: Product listings with filters (price, location).

Rental Portal: Equipment availability calendar.

1.5.2. Hardware Interfaces

GPS: For location-based services.

Camera: Upload product/laborer photos.

1.5.3. Software Interfaces

Firebase Auth: Secure login.

REST APIs: Weather data, payment processing.

1.5.4. Communications Interfaces

SMS/Email: Notifications.

HTTPS: Secure data transmission.

1.6. System Features

1.6.1. User Registration (Priority: High)

1. Registration & Login

- Farmer/Buyer Registration:

Users can register as either farmers (to sell produce/resources) or buyers (to purchase agricultural goods).

- Email Verification:

Ensures secure account creation by validating users via email.

2. User Profile

Profile editing:

Users can update personal details, contact information, and preferences.

View Profile: Displays transaction history, ratings, and activity logs.

3. Authentication

Password reset:

A secure recovery process for forgotten passwords.

1.6.2. Resource Management

1. Crop Upload & Management

- Fertilizers, Seeds, and Tools:

Farmers can list agricultural inputs for sale.

- Edit Listings:

Farmers can modify or remove items from their inventory.

Farmers can track their inventory of seeds, fertilizers, and tools through an intuitive stock management system. This feature helps maintain adequate stock levels, preventing shortages or overstocking, and ensures resources are available when needed.

1.6.3. Rental System Management

- Equipment Rental
- Manage Rentals

Farmers can rent/lease machinery (e.g., tractors, harvesters). Track rental history and active leases. HarvestHub empowers farmers by providing tools to manage agricultural resources efficiently. Farmers can upload and manage listings for fertilizers, seeds, and tools, with options to edit or remove items as needed. The platform also features a rental system for agricultural equipment, allowing farmers to lease machinery temporarily. This functionality includes rental management tools, enabling users to track rentals, view histories, and update listings, ensuring optimal resource utilization and cost-effectiveness.

1.6.4. Labor Management

- Labor Registration
- View Labor Listings

To address labor shortages, HarvestHub includes a dedicated labor management module. Laborers can register on the platform to offer their services, while farmers can browse listings to find suitable workers. This feature simplifies the hiring process by providing detailed laborer

profiles, ensuring farmers can make informed decisions based on skills, experience, and availability.

1.6.5. Crop Information & Advisory

Crop specification:

Detailed guides on crop cultivation, soil requirements, and harvesting techniques. Farmers gain access to a comprehensive crop advisory section, offering detailed specifications for various crops. This includes growth requirements, ideal soil conditions, and harvesting techniques. By leveraging this knowledge base, farmers can optimize crop yields and adopt best practices, enhancing productivity and sustainability.

1.6.7. Notifications

Real-time Alerts

Notifications for transactions, weather changes, or rental updates. HarvestHub keeps users informed with notifications for transactions, payments, and other critical activities. This ensures timely communication and enhances user engagement by alerting them to important updates.

1.6.8. Multi-Language Support

Regional Languages

Ensures accessibility for users across diverse linguistic backgrounds. To cater to a diverse user base, the platform supports multiple regional languages(English, Urdu). This inclusivity ensures farmers from different linguistic backgrounds can navigate the system effortlessly, breaking down barriers to access.

1.6.9. Transaction & Payment

Secure Payment Gateway

Supports multiple methods. A secure payment gateway supports multiple payment methods, including bank transfers, credit/debit cards, and mobile wallets. Through this feature we are simplifying financial management.

1.6.10. Reporting & Feedback

- Report Users
- Rating System

Flag suspicious or fraudulent activities. Farmers and buyers can rate each other to build trust. After transactions, farmers and buyers can rate and review each other, fostering accountability within the community. This feature encourages positive interactions and helps users identify reliable partners.

Users can report suspicious or inappropriate behavior, ensuring a safe and respectful environment. This moderation tool underscores HarvestHub's commitment to maintaining platform integrity.

1.6.11. Technical & Security Features

- Data Encryption
- Protects sensitive user and transaction data.
- Scalability
- Designed to handle growing user bases and data loads.
- Cross-Platform Compatibility
- Accessible via web and mobile devices.

2.1. Non-Functional Requirements

2.1.1. Performance

Response Time: <2 seconds for critical actions (e.g., login).

2.1.2. Security

Data Encryption: AES-256 for sensitive data.

Role-Based Access: Farmers/laborers/admins have distinct permissions.

2.1.3. Usability

UI/UX: Intuitive for low-literacy users.

2.2. Other Requirements

2.2.1. Legal Compliance

Adherence to Punjab's agricultural laws.

2.2.2. Database

Firebase Firestore: NoSQL structure for scalability.

Conclusion

This SRS ensures HarvestHub meets farmers' needs through a secure, scalable, and user-friendly platform. Subsequent phases (design, testing) will reference this document for alignment with stakeholder expectations.

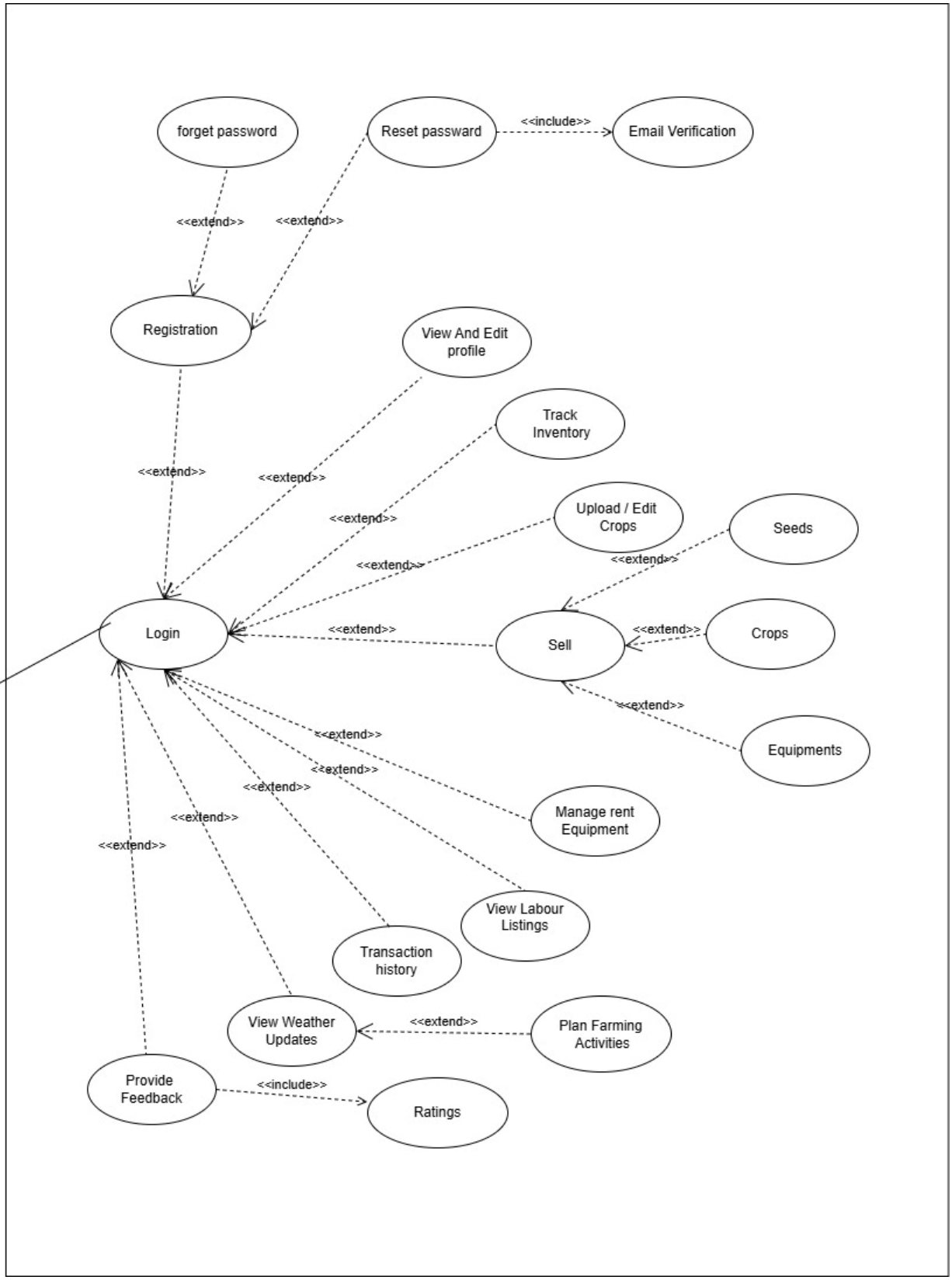
HarvestHub's integration of modern technologies (Flutter for cross-platform compatibility, Firebase for real-time data management, and REST APIs for external services like weather and payments) underscores its scalability and adaptability. Furthermore, the emphasis on security (AES-256 encryption, role-based access) and usability (intuitive UI for low-literacy users) ensures the platform is both robust and accessible. Compliance with regional agricultural laws and

support for local languages (Urdu, Punjabi) further solidify its relevance in target markets like Punjab

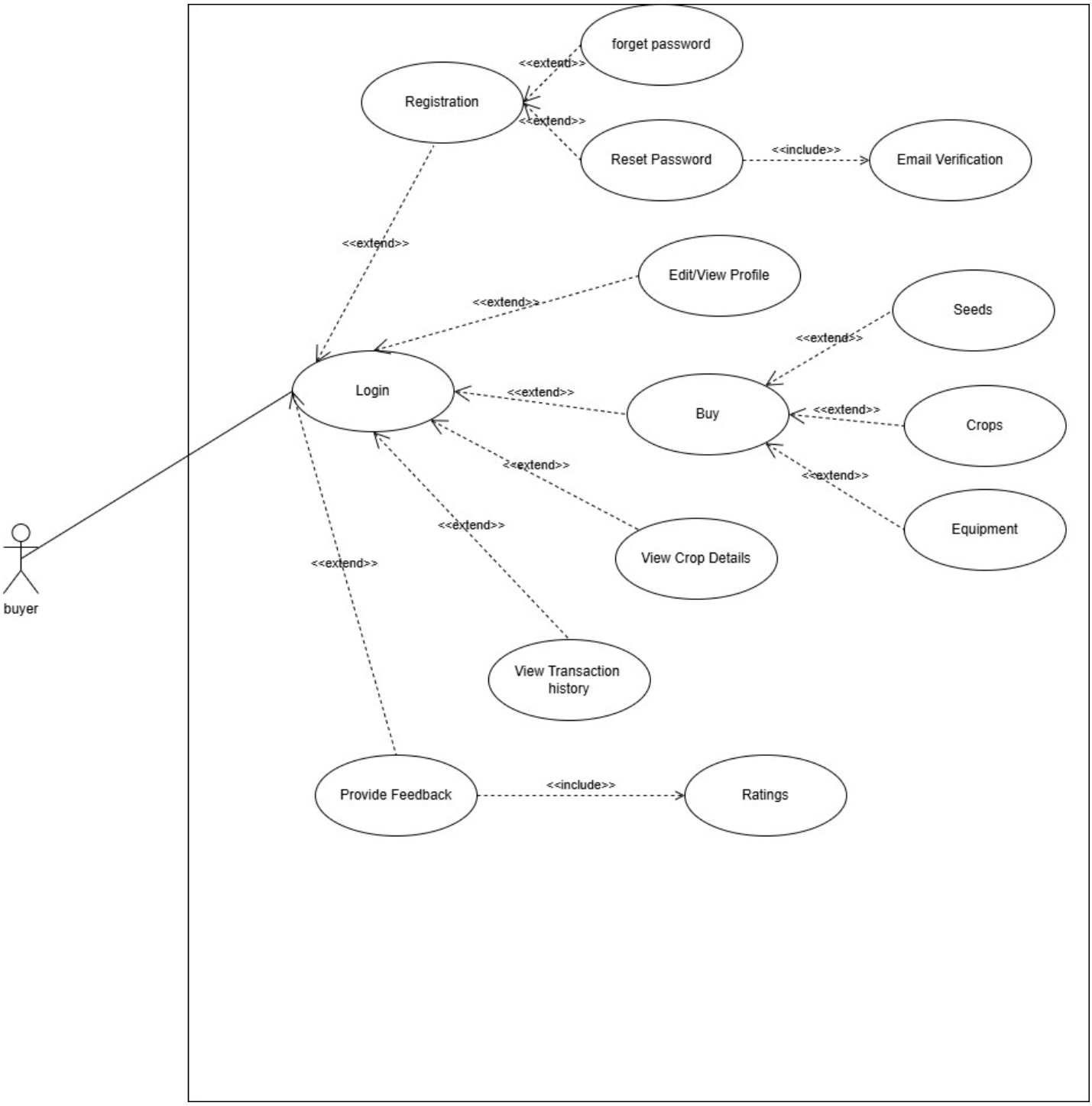
Chapter 3

System Design:

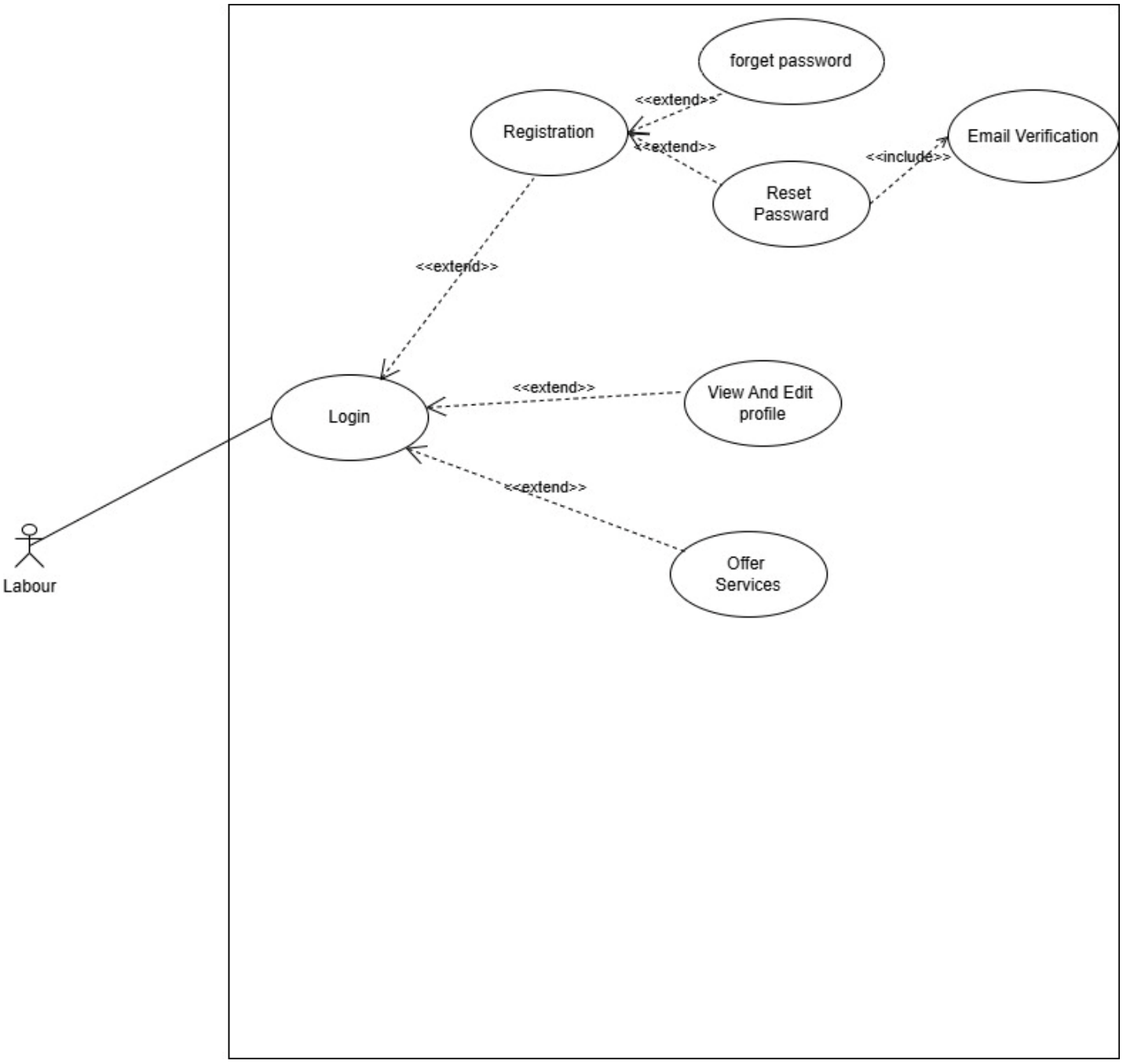
3.1.1 Farmer Usecase :



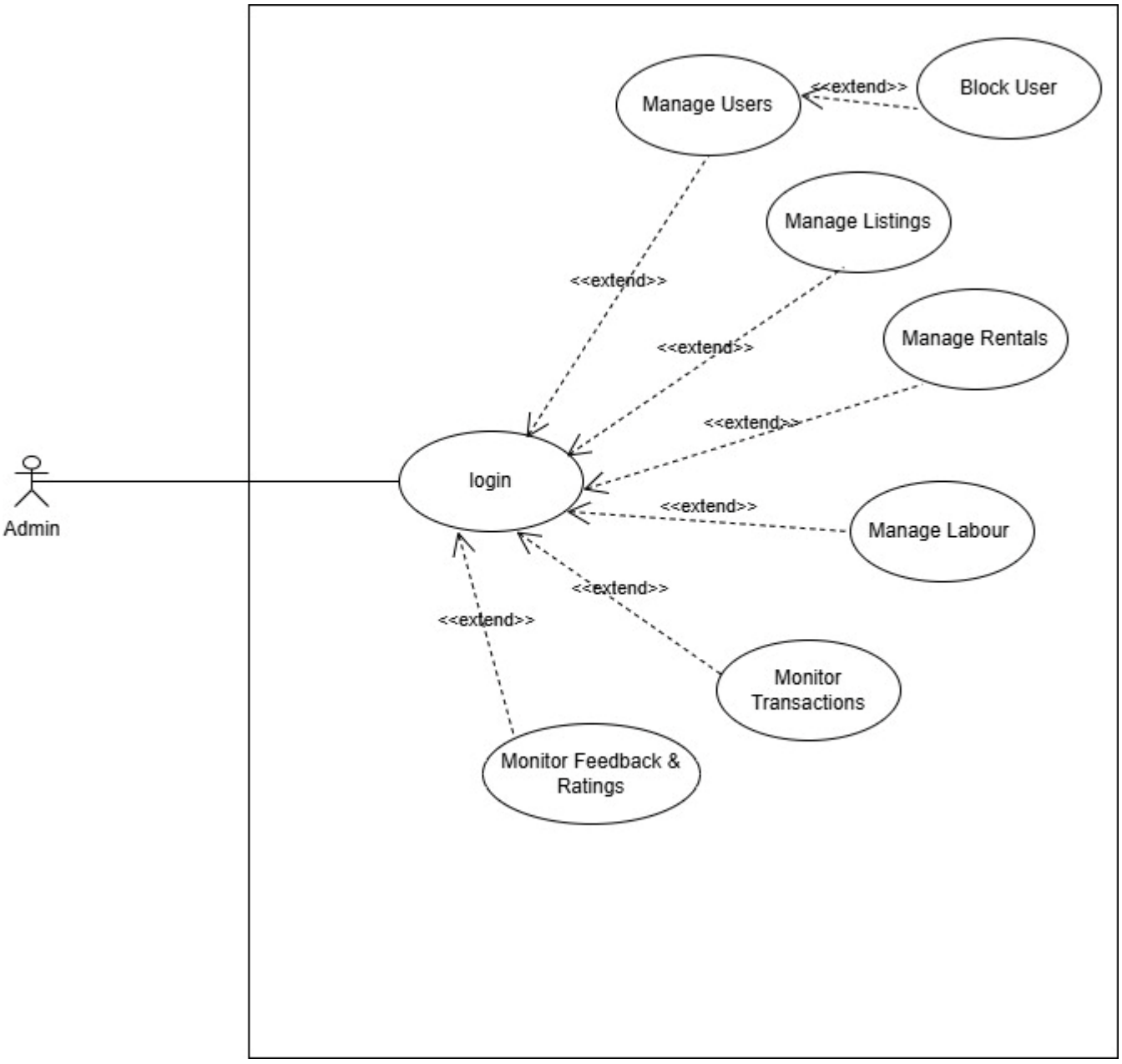
3.1.2 Buyer Usecase :



3.1.3 Labour Usecase :



3.1.4 Admin Usecase :



3.2 Fully Dressed Use Cases

3.1.1 Use Case: Authenticate:

Action	Farmer, Buyer, Labour, Admin
Preconditions	User is on the login screen
Postconditions	User is successfully authenticated and redirected to their respective dashboard.
Trigger	User opens the app or platform and attempts to log in.

Steps	**Action**	**System response**
1	User enters credentials (username/password) and clicks "Login"	System verifies credentials
2	If successful	User is redirected to their role-specific dashboard
3	If unsuccessful	System displays an error message

Extensions:

- *Forgot Password:* User can reset password via email.

- *Email Verification:* Included during password reset or registration.

3.1.2 Use Case**:** Register:

Main Success Scenario:

Action	Farmer, Buyer, Labour, Admin
Preconditions	User is not registered
Postconditions	New user account is created.
Trigger	User clicks on "Register".

Steps	**Action**	**System response**
1	User fills registration form with rolespecific info	System validates input
2	User submits form	System creates account and sends verification email.

3.1.3 Use Case: Reset Password:

Main Success Scenario:

Action	User
Preconditions	User has forgotten their password
Postconditions	Password is updated successfully
Trigger	User clicks “Forgot Password”

Steps	**Action**	**System response**
1	User enters email	System sends password reset link.
2	User resets password	System updates password and confirms success

3.1.4 Use Case: Edit/View Profile**:**

Main Success Scenario:

Action	User
Preconditions	User is logged in.

Postconditions	Profile data is updated
Trigger	User navigates to profile settings

Steps	**Action**	**System response**
1	User views or edits personal info	System displays and validates data
2	User saves changes	System updates profile info

3.1.5 Use Case: Upload / Edit Crops (Farmer Only)

Main Success Scenario:

Action	Farmer
Preconditions	Farmer is logged in
Postconditions	Crop listing is posted or updated.
Trigger	Farmer wants to sell crops

Steps	Action	System response
1	Farmer fills crop listing form	System validates and saves listing
2	Farmer edits an existing listing	System updates it accordingly

3.1.6 Use Case: Offer Services (Labour)

Main Success Scenario:

Action	Labour
Preconditions	Labour is logged in
Postconditions	Labour service listing is active
Trigger	Labour wants to offer services.

Steps	Action	System response
1	Labour enters availability, skills	System lists the services under "Available Labour"
2	Labour updates or removes info	System processes changes accordingly

3.1.7 Use Case: Buy Seeds, Crops, and Equipment

Main Success Scenario:

Action	Buyer
Preconditions	Buyer is logged in.
Postconditions	Purchase is recorded
Trigger	Buyer selects an item to purchase

Steps	**Action**	**System response**
1	Buyer selects item and confirms purchase	System processes transaction
2	Buyer gets confirmation	System stores transaction details

3.1.8 Use Case: Manage Rent Equipment

Main Success Scenario:

Action	Farmer, Admin

Preconditions	User is logged in.
Postconditions	Equipment listed for rent or approved.
Trigger	Farmer uploads or Admin reviews listing

Steps	**Action**	**System response**
1	Farmer adds rental equipment	System stores listing
2	Admin approves/rejects listing	System updates status accordingly

3.1.9 Use Case: View Transaction History

3.1.10 Main Success Scenario:

Action	All Users
Preconditions	User is logged in.
Postconditions	Transactions are shown
Trigger	User selects "Transaction History".

Steps	Action	System response
1	User views past transactions	System fetches and displays data

3.1.11 Use Case: View Weather Updates

Main Success Scenario:

Action	Farmer, Buyer
Preconditions	User is logged in.
Postconditions	Weather data is displayed
Trigger	User opens weather section

Steps	Action	System response
1	User opens weather tab	System shows location-based weather forecast

3.1.12 Use Case: Plan Farming Activities

Main Success Scenario:

Action	Farmer
Preconditions	Farmer is logged in.
Postconditions	Farming plan saved.
Trigger	Farmer navigates to planner

Steps	**Action**	**System response**
1	Farmer adds crop cycle, irrigation, fertilizer plans	System saves and optionally reminds user later

3.1.13 Use Case**:** Provide Feedback

Main Success Scenario:

1. **Actors**: **:** Buyer, Farmer, Labour
2. **Preconditions**: User has completed an interaction or purchase.
3. **Postconditions**: **:** Rating/feedback is saved.
4. **Trigger**: User chooses to rate or give feedback.

Action	Buyer, Farmer, Labour
Preconditions	Rating/feedback is saved.

Postconditions	Farming plan saved.
Trigger	User chooses to rate or give feedback

Steps	**Action**	**System response**
1	User writes feedback or gives stars	System stores and updates ratings

3.1.14 Use Case: Admin – Manage Users

Main Success Scenario:

1. **Actors**: **:** Admin
2. **Preconditions**: Admin is logged in.
3. **Postconditions**: Users are added/edited/removed
4. **Trigger**: Admin selects "Manage Users".

Action	Admin
Preconditions	Admin is logged in..
Postconditions	Users are added/edited/removed
Trigger	Admin selects "Manage Users".

Steps	Action	System response
1	Admin views user list	System fetches user data
2	Admin edits/deletes accounts	System updates the database

3.1.15 Use Case: Admin – Manage Listings

Main Success Scenario:

Action	Admin
Preconditions	Admin is logged in..
Postconditions	Listings approved or rejected
Trigger	Admin opens listings dashboard

Steps	Action	System response
1	Admin views crop/equipment/labour listings	System shows status
2	Admin approves/rejects/moderates listings	System updates status

3.1.16 Use Case: Admin – Monitor Transactions:

Main Success Scenario:

Action	Admin
Preconditions	Admin is logged in..
Postconditions	Transactions reviewed
Trigger	Admin accesses transaction logs

Steps	**Action**	**System response**
1	Admin checks logs	System displays all transactional data
2	Admin flags suspicious activity	System alerts or marks for review

3.1.17 Use Case: Admin – Monitor Labour Activities:

Main Success Scenario:

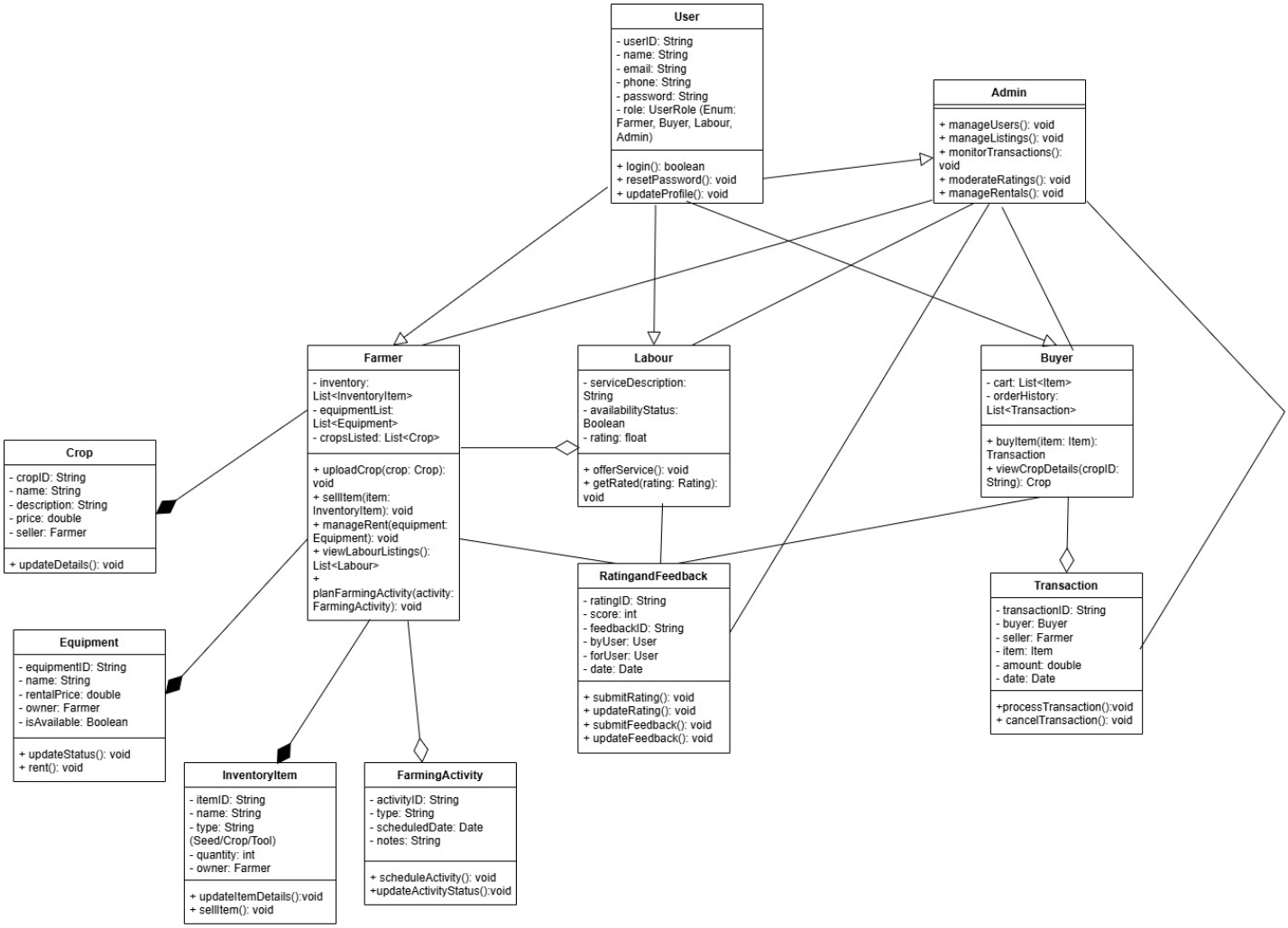
Action	Admin
Preconditions	Admin is logged in..

Postconditions	Labour data reviewed
Trigger	Admin selects “Labour Management”.

Steps	**Action**	**System response**
1	Admin views availability/performance	System displays info
2	Admin flags inappropriate listings	System removes or disables entry

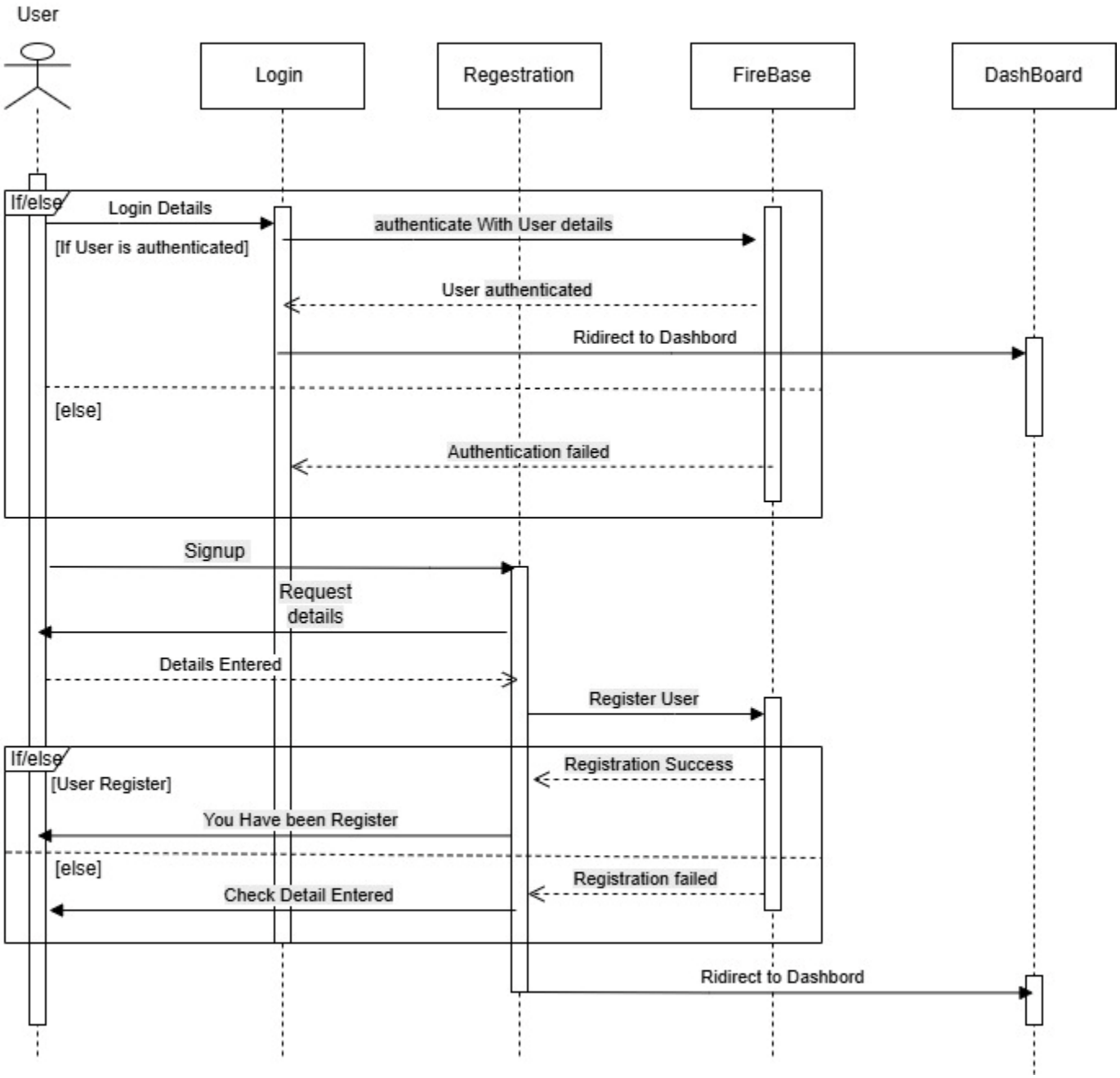
Chapter 4

Class Diagram:

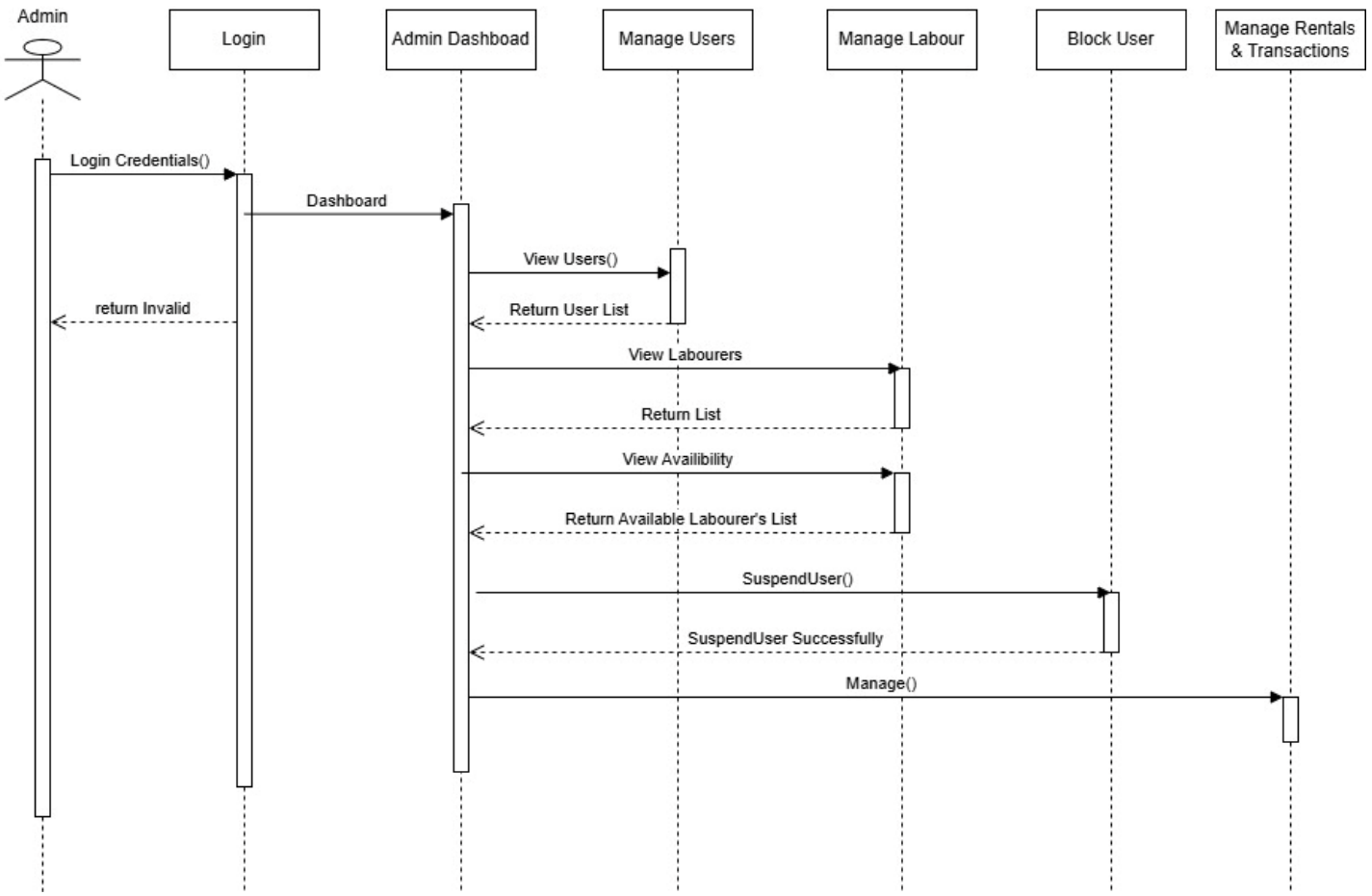


Sequence Diagram:

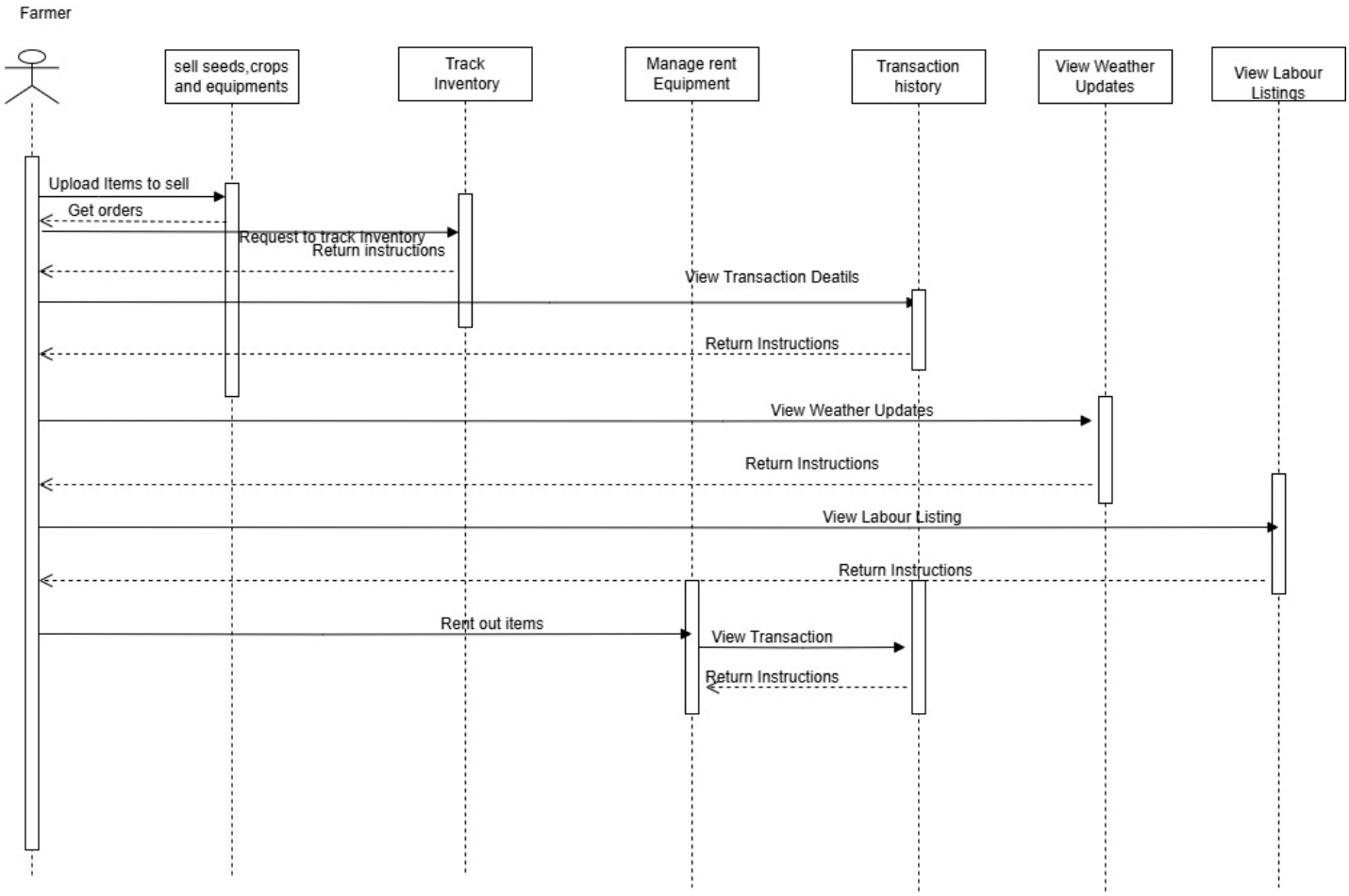
Login:



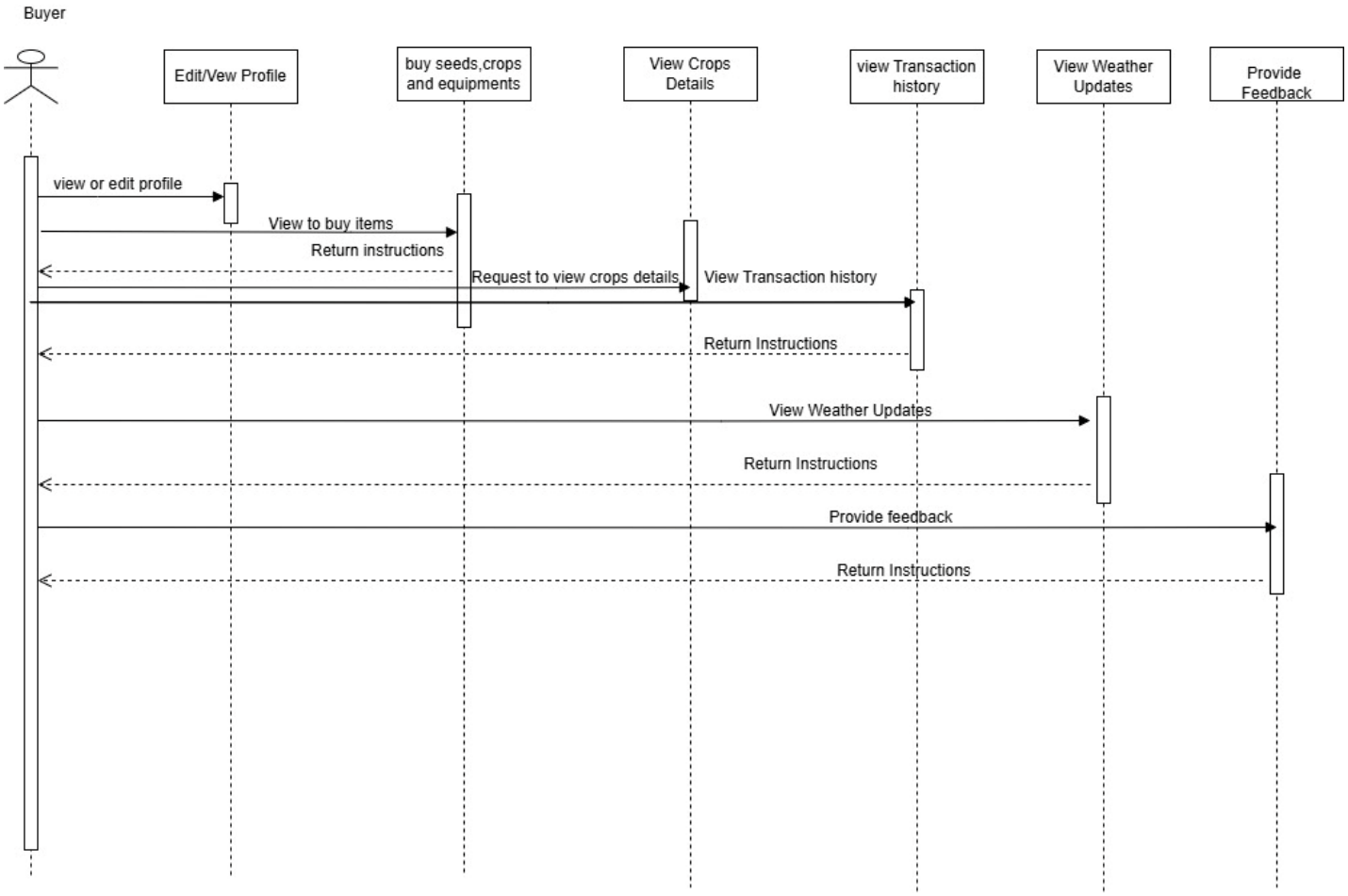
Admin:


Admin
Login
Admin Dashboad
Manage Users
Manage Labour
Block User
Manage Rentals & Transactions
Login Credentials()
Dashboard
View Users()
return Invalid
Return User List
View Labourers
Return List
View Availibility
Return Available Labourer's List
SuspendUser()
SuspendUser Successfully
Manage()

Farmer:


Farmer
sell seeds,crops and equipments
Track Inventory
Manage rent Equipment
Transaction history
View Weather Updates
View Labour Listings
Upload Items to sell
Get orders
Request to track Inventory
Return instructions
View Transaction Deatils
Return Instructions
View Weather Updates
Return Instructions
View Labour Listing
Return Instructions
Rent out items
View Transaction
Return Instructions

Buyer:


Buyer
Edit/Vew Profile
buy seeds,crops and equipments
View Crops Details
view Transaction history
View Weather Updates
Provide Feedback
view or edit profile
View to buy items
Return instructions
Request to view crops details
View Transaction history
Return Instructions
View Weather Updates
Return Instructions
Provide feedback
Return Instructions

Schema:

```
interface User {
  uid: string; // The Document ID (from Firebase Auth)
  name: string;
  email: string;
  contactInfo: string;
  cnic: string;
  role: 'Farmer' | 'Buyer' | 'Labour' | 'Admin';

  // Role-Specific Fields (Only populated based on role)
  labourInfo?: {
    serviceDescription: string;
    availabilityStatus: boolean;
    ratingAvg: number;
  };

  createdAt: Timestamp;
}


interface Crop {
  cropId: string; // Document ID
  farmerId: string; // Reference to users/{uid}
  name: string;
  description: string;
  price: number;
  imageUrl?: string; // Recommended for Firebase apps
}
```

```
interface FarmingActivity {
  activityId: string;
  farmerId: string; // Reference to users/{uid}
  type: string;
  scheduledDate: Timestamp;
  notes: string;
  status: 'pending' | 'completed';
}



interface Equipment {
  equipmentId: string;
  ownerId: string; // Reference to users/{uid} (the Farmer)
  name: string;
  rentalPrice: number;
  isAvailable: boolean;
}

interface InventoryItem {
  itemId: string;
  buyerId: string; // Reference to users/{uid}
  sellerId: string; // Reference to users/{uid} (Farmer)
  amount: number;
  quantity: number;
  purchasedAt: Timestamp;
}
```

```
interface Transaction {
  transactionId: string;
  ownerId: string; // Reference to Farmer uid
  buyerId: string; // Reference to Buyer uid
  itemName: string;
  rentalPrice: number;
  date: Timestamp;
  status: 'success' | 'failed' | 'pending';
}


interface Rating {
  ratingId: string;
  byUserId: string; // Reference to users/{uid}
  forUserId: string; // Reference to users/{uid}
  score: number; // 1-5
  comments: string;
  timestamp: Timestamp;
}

interface Feedback {
  feedback_id: string;
  by_user_id: string;
  for_user_id: string;
  message: string;
  date: timestamp;
}
```

Chapter 6

Software Testing Document

6.1 Introduction

This section presents the testing strategy, test plan, environment, and detailed test cases for the HarvestHub application. Testing ensures that each functionality—such as user registration, equipment rental, crop selling, and payment processing—works accurately and efficiently. Since HarvestHub deals with agricultural transactions, reliability and user trust are critical. The goal of this document is to verify that the application performs according to the requirements and provides a seamless experience for farmers/seller, buyers, laborers, and administrators.

6.2 Test Approach

Testing follows the Black Box Testing technique, emphasizing input-output validation rather than internal code inspection. Each module was tested independently, ensuring functional accuracy, usability, and data integrity. The approach validates HarvestHub's workflows for various user roles under real-world agricultural use cases, ensuring robustness and performance. **6.3 Test plan**

6.3.1 Features to be tested

- User Authentication (Registration, Login, Password Reset)
- Crop/Seed Upload and Editing
- Equipment Rental Management
- Labor Hiring and Service Listings
- Buying and Selling Operations in Marketplace
- Transaction and Payment Gateway Integration
- Profile Management
- Feedback and Reporting System

6.3.2 Features Not to be Tested

- Firebase database schema and internal queries
- Third-party APIs (e.g. Payment Gateway API)
- UI design aesthetics (colors, animations, styling)
- Backend infrastructure setup (assumed stable) **6.4 Testing Tools and Environment** **Development Tools:**
  - Frontend: Flutter (for Mobile application development)
  - Backend: Dart
  - Database: Firebase
  - IDE: Android Studio / Visual Studio Code
  - Testing Tool: Postman (for API testing)

Version Control:

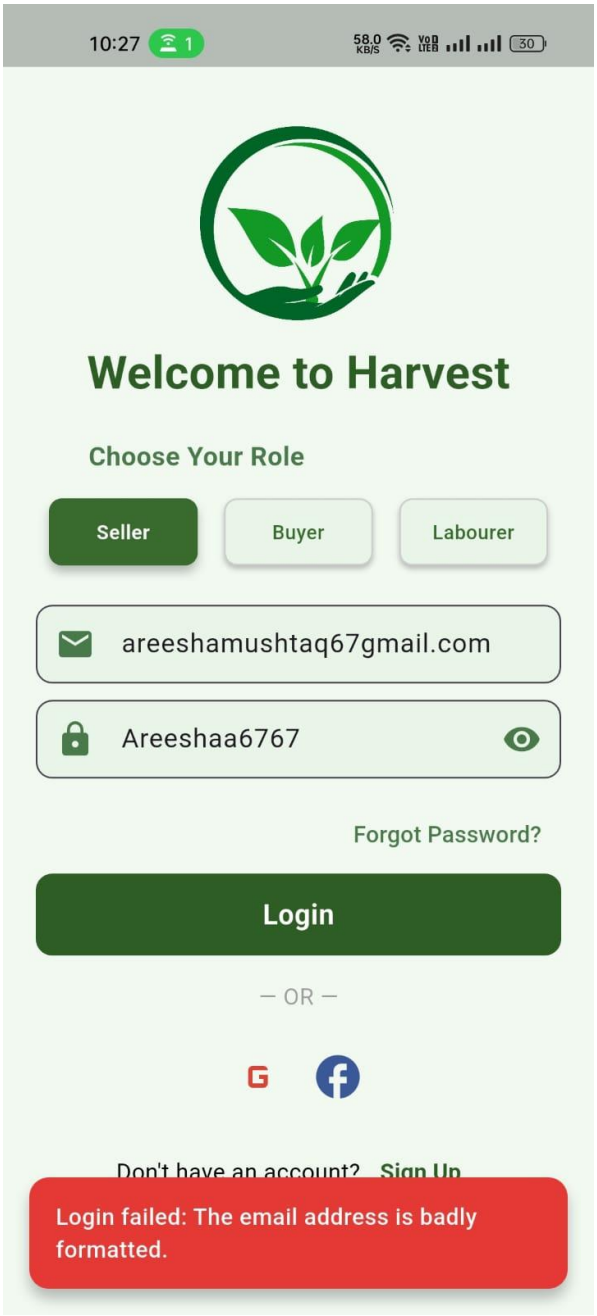
- Use Git and GitHub for version control and collaboration.

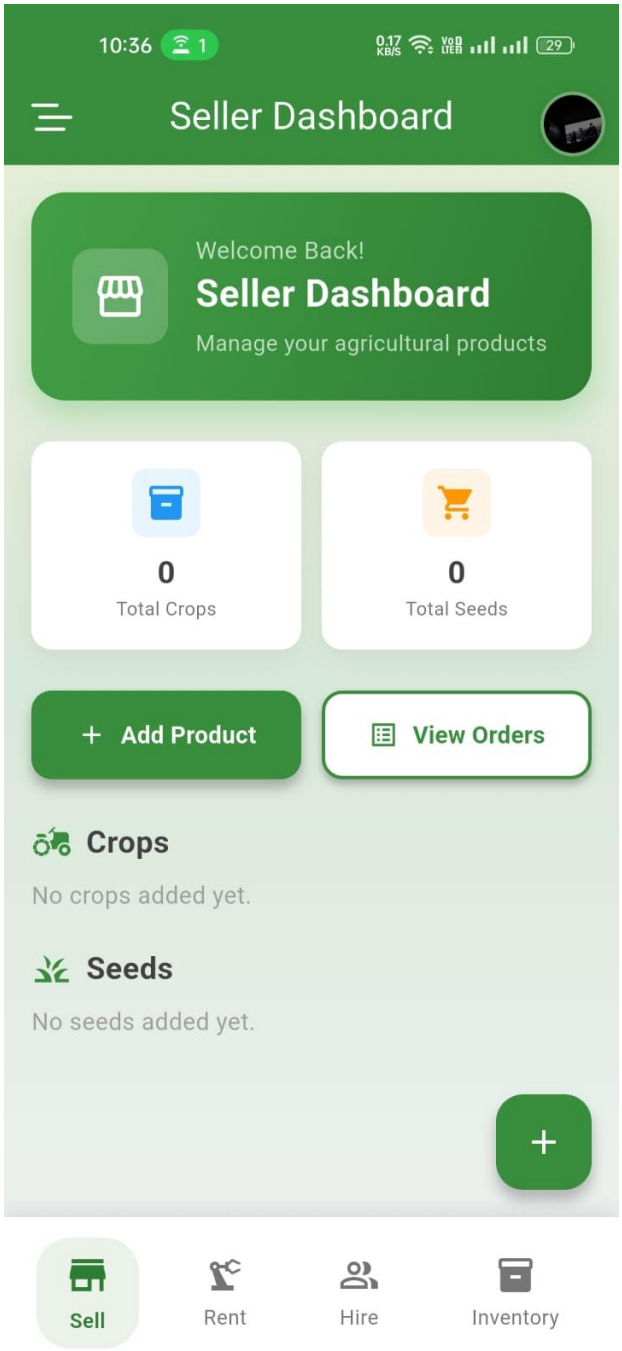
6.5 List of Test Scenarios

Test Cases include input conditions, execution steps, and expected outcomes. These help confirm whether application functions perform as intended. Below are the key test cases for HarvestHub with their respective conditions and outcomes.

6.5.1. Test case 1: Authenticate

Test Case ID	UC-1-001
Use Case Name	Authenticate User
Summary Description	Verifying user login process. Ensures that valid credentials allow login and invalid credentials trigger errors.
Pre-Condition	User is on the login screen.
Test Procedure	1. Enter valid credentials. 2. Enter invalid credentials.
Test Data	Valid credentials: role/Email /password1. Invalid credentials: role/Email /wrongpassword.
Expected Result	1. User successfully logs in and is redirected to the home screen. 2. Error message is displayed.
Actual Result	Same as expected result.
Status	Valid Input: pass Invalid Input: fail





Login Fails

Successfully Login

6.5.2 Test case 2 User Registration

Test Case ID	UC-2-001
Use Case Name	Register User
Summary Description	Validate that users can register successfully with valid information and that the system displays appropriate error messages for invalid inputs for all roles (Farmer, Buyer, Labour).
Pre-Condition	User is not already registered.
Test Procedure	1. Enter required information (Name, Email, Password, Role). 2. Click on Register. 3. Repeat with invalid or missing fields.
Test Data	**Valid Case:** Name: Laiba Email: laiba@gmail.com Password: Laiba@123 Role: farmer/buyer **Invalid Cases:** 1. Missing email 2. Invalid email format (laiba123) 3. Weak password (12345) 4. No role selected
Expected Result	**Valid Input:** Account successfully created. **Invalid Input:** Appropriate error messages displayed (e.g., “Enter valid email,” “Password too weak,” “Select role”).
Actual Result	**Valid Input:** Account successfully created. **Invalid Input:** Appropriate error messages displayed (e.g., “Enter valid email,” “Password too weak,” “Select role”).
Status	Valid Input: pass Invalid Input: fail

10:32 1 0.03 KB/S 29

Tap to add profile picture

Areesha Mushtaq

03008505236

Lahore

arishamushtaq67@gmail.com

Areesha00

Password must have 1 capital, 1 small letter, 1 ...

Select Role:

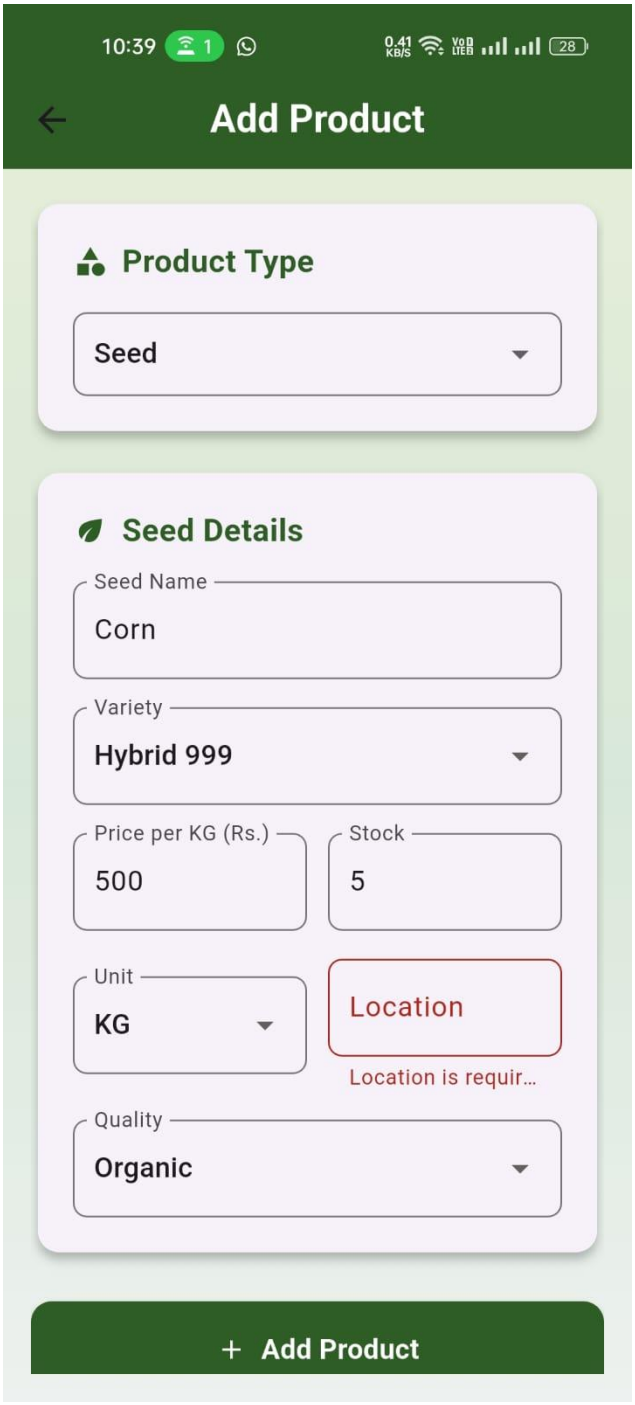
Seller | Buyer | Labourer

Sign Up

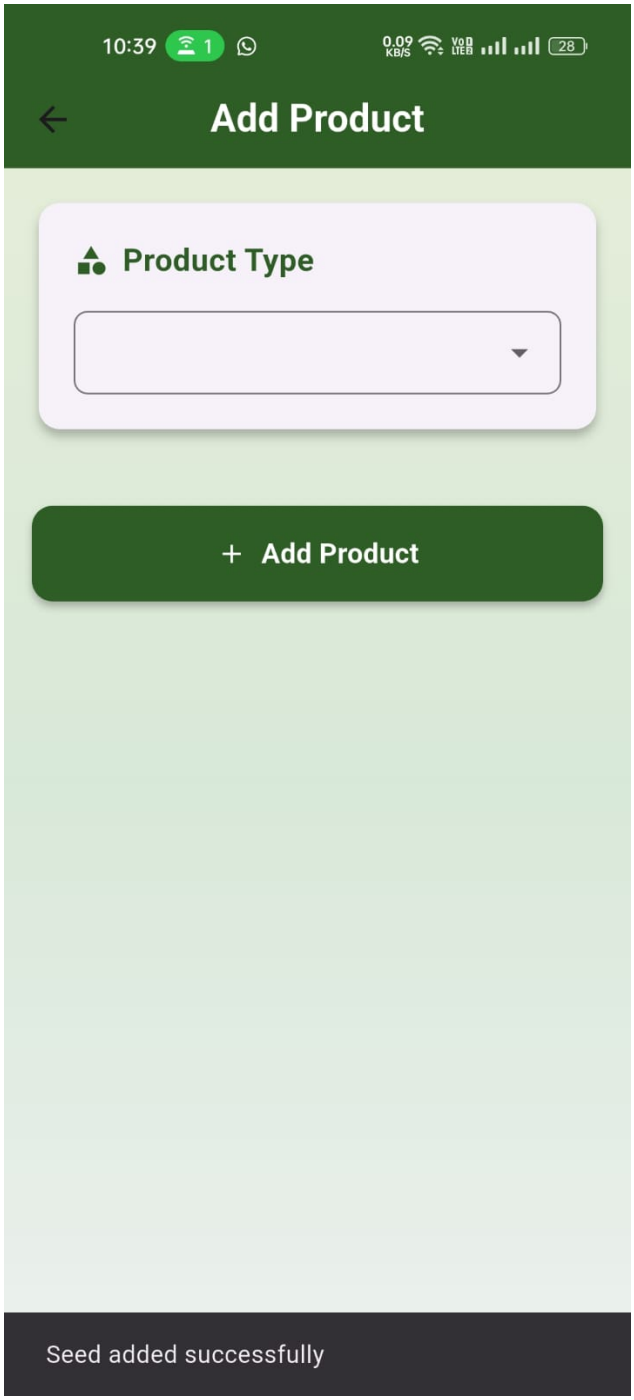
Already have an account? **Login**

6.5.3 Test case 3: Upload or Edit Crop/Seed Listing

Test Case ID	UC-3-001
Use Case Name	Upload/Edit Crop or Seed Listing
Summary Description	Verifies that the farmer can add a new crop or seed listing by selecting the correct type from the dropdown, filling in all required details, and editing the listing afterward.
Pre-Condition	Farmer is logged in and on the "Sell" screen.
Test Procedure	1. click on add product button. 2. Select Crop or Seed from the dropdown menu. 3. Fill in the form fields (Name, Price per kg, Quantity, Quality, Address). 4. Click **Submit** to add the listing. 5. Navigate to "My Listings" and edit existing data. 6. Try submitting with missing or invalid data.
Test Data	**Valid Input:** Type: Crop Name: Wheat Price per kg: 200 Quantity: 200 Unit: kg/ton Quality: A/B/organic Address: Multan, Punjab **Invalid Input:** 1. Missing "Name" 2. Missing "Price" 3. Missing "Quantity" 4. Missing "Address"
Expected Result	**Valid Input:** Crop or seed successfully listed. **Invalid Input:** Appropriate error messages displayed (e.g., "Price required," "Enter valid quantity," "Address cannot be empty").
Actual Result	**Valid Input:** Crop or seed successfully listed. **Invalid Input:** Appropriate error messages displayed (e.g., "Price required," "Enter valid quantity," "Address cannot be empty"). j
Status	Valid Input: pass Invalid Input: fail



Missing location

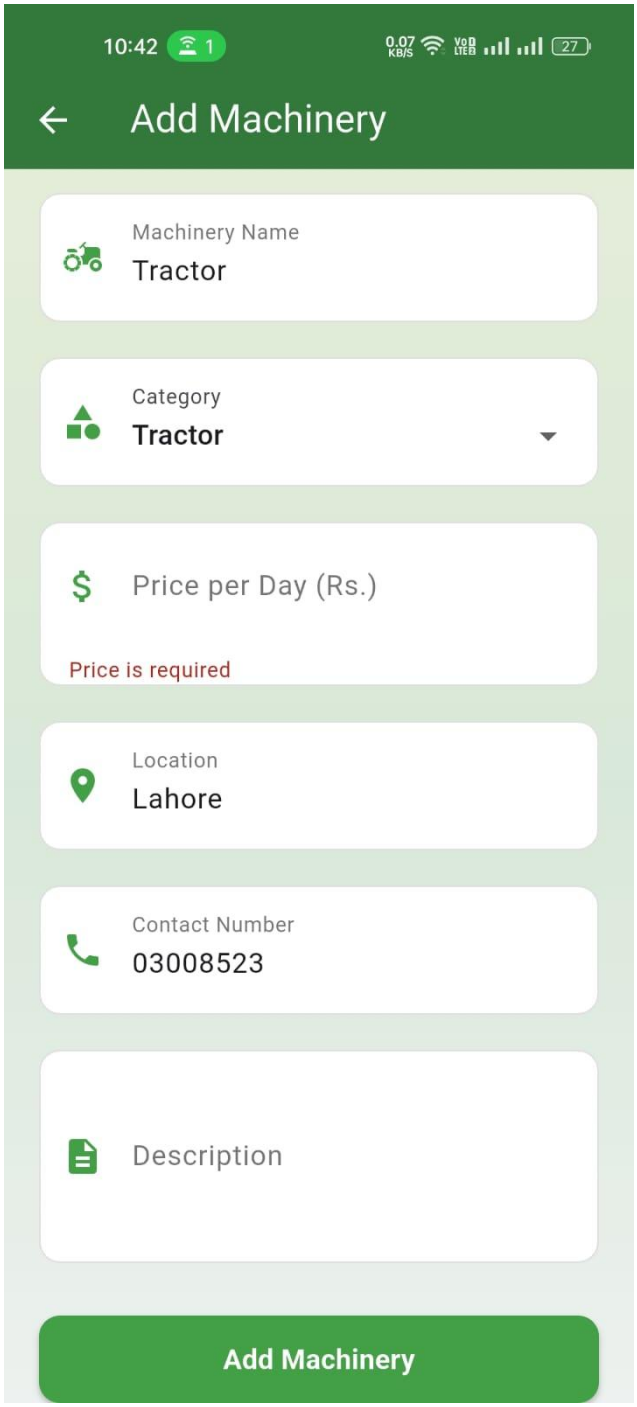


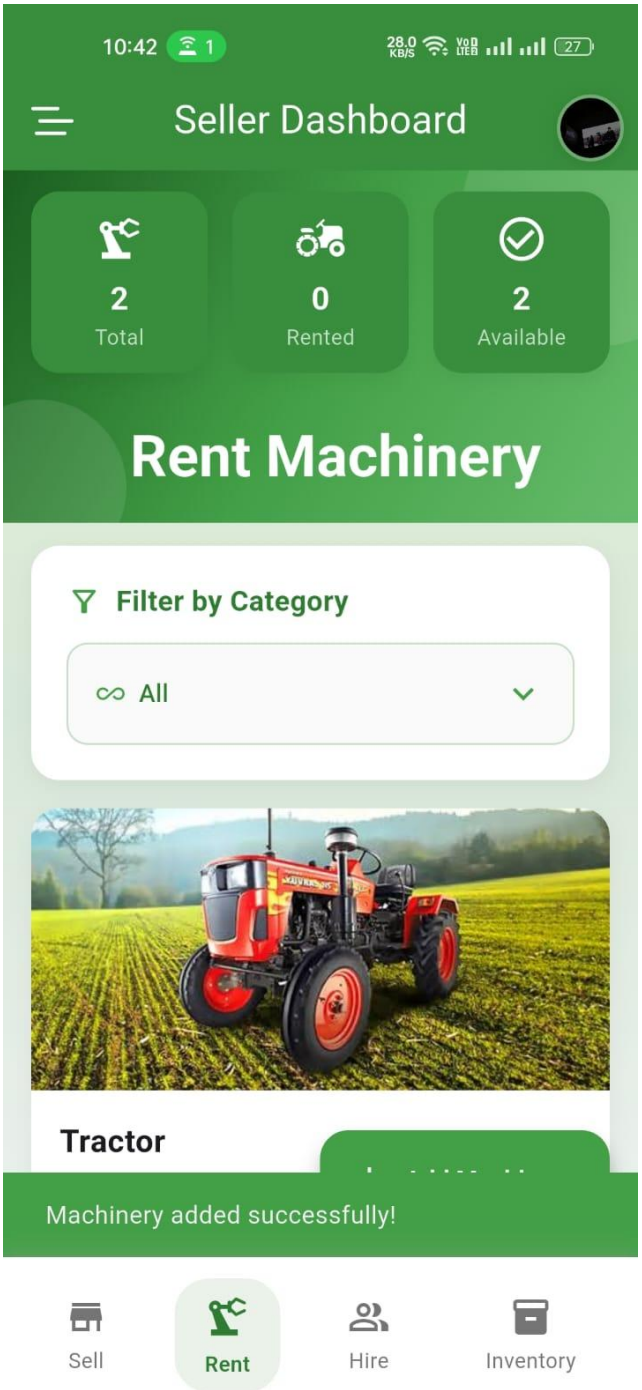
Seed Added successfully

6.5.4 Test case 4: Rent Agricultural Equipment

Test Case ID	UC-4-001
Use Case Name	Rent Agricultural Equipment
Summary Description	Validates that a seller/farmer can successfully add machinery for rent by filling all required details and that the system properly handles invalid or missing input cases.
Pre-Condition	Farmer is logged in and on the “Rent Machinery” tab.
Test Procedure	1. In Rent Machinery tab click on **Add Machinery**. 3. Upload machinery image. 4. Select machinery category from dropdown (e.g., Tractor, Harvester, Plough). 5. Enter required details: Name, Price per day, Location, and Contact Number. 6. Click **Add Machinery** to save listing. 7. Verify listing appears under “Rent Machinery” tab. 8. Try submitting with missing or invalid fields.
Test Data	**Valid Input:** Machinery Image: tractor.jpg Name: Mahindra 575 Di Category: Tractor Price/Day: 5000 Location: Lahore Contact: 0302-XXXXXXXX **Invalid** **Input:** 1. Missing image 2. Category not selected 3. Missing price 4. Empty contact field or invalid number
Expected Result	**Valid Input:** Machinery successfully added and displayed in the Rent Machinery tab with correct details. **Invalid Input:** Appropriate error messages shown (e.g., “All fields required,” “Enter valid contact number,” “Select machinery category”).

Actual Result	Same as expected result.
Status	Valid Input: pass Invalid Input: fail





6.5.5 Test case 5. Hire Labour

Test Case ID	UC-5-001
Use Case Name	Hire Labour
Summary Description	Verifies that farmers can view all available laborers, send hire requests, and that once a laborer is booked, they are no longer visible to other farmers.
Pre-Condition	1. Farmer is logged in. 2. At least one laborer is registered and available with valid details (name, skills, timing, location).
Test Procedure	1. Open **Hire Labour** section on the Seller Dashboard. 2. View list of all available labourers. 3. Select one or more labourers from the list. 4. Click **Hire/Book** to confirm hiring. 6. Attempt to hire a labourer who is already booked. 7. Try hiring without selecting any labourer.
Test Data	**Valid Input (from farmer side):** Select Labourer: Ahmad Select Labourer: Ali **Invalid Input:** 2. Attempting to hire a labourer already booked. 3. Network disconnected during hiring process.
Expected Result	**Valid Input:** Selected labourers are successfully booked. Notifications sent to each hired labourer. They disappear from the “Available Labour” list for all other farmers. **Invalid Input:** This labourer is already booked or Network connection required.
Actual Result	**Valid Input:** Selected labourers are successfully booked. Notifications sent to each hired labourer. They disappear from the “Available Labour” list for all other farmers. **Invalid Input:** This labourer is already booked or Network connection required.
Status	Valid Input: pass Invalid Input: fail

6.5.6 Test case 6: Buy Crops/Seeds

Test Case ID	UC-6-001
Use Case Name	Buy Crops/Seeds
Summary Description	Verifies that a buyer can view all products (crops and seeds) uploaded by farmers, purchase selected items, complete payment, and trigger notifications for both buyer and seller. Also checks system behavior for invalid or failed transactions.
Pre-Condition	1. Buyer is logged in and on the **Buyer Dashboard**. 2. At least one seller/farmer has listed crops or seeds for sale. 3. Internet and payment gateway are active.
Test Procedure	1. View the list of available crops and seeds. 2. Select a product to purchase. 3. Click **Buy Now** or **Add to Cart**. 4. Confirm the order and proceed to payment. 5. Enter valid payment details and complete the transaction. 6. Observe order confirmation and verify that notifications are sent to both buyer and respective seller. 7. Repeat process with invalid or failed payment data to test error handling.
Test Data	**Valid Input:** Product: Wheat Seeds Name: Ali Email: ali@gmail.com Phone no:0302-XXXXXXXXX Quantity: 50 kg Address: Lahore, Punjab Payment: Valid card (e.g., 4111-XXXX-XXXX-1234) **Invalid Input:** 1. No product selected. 2. Invalid payment details. 4. Network disconnected during payment.

Expected Result	**Valid Input:** Order successfully placed and payment confirmed. Notification sent to both buyer and the seller who listed the product. Product stock updates automatically in seller's inventory. **Invalid Input:** Appropriate error displayed (e.g., "Please select a product," "Payment failed," "Connection error"). No order created until payment succeeds.
Actual Result	Same as ER.
Status	Pass

6.5.7 Test case 7: Rent Machinery (Booking by User)

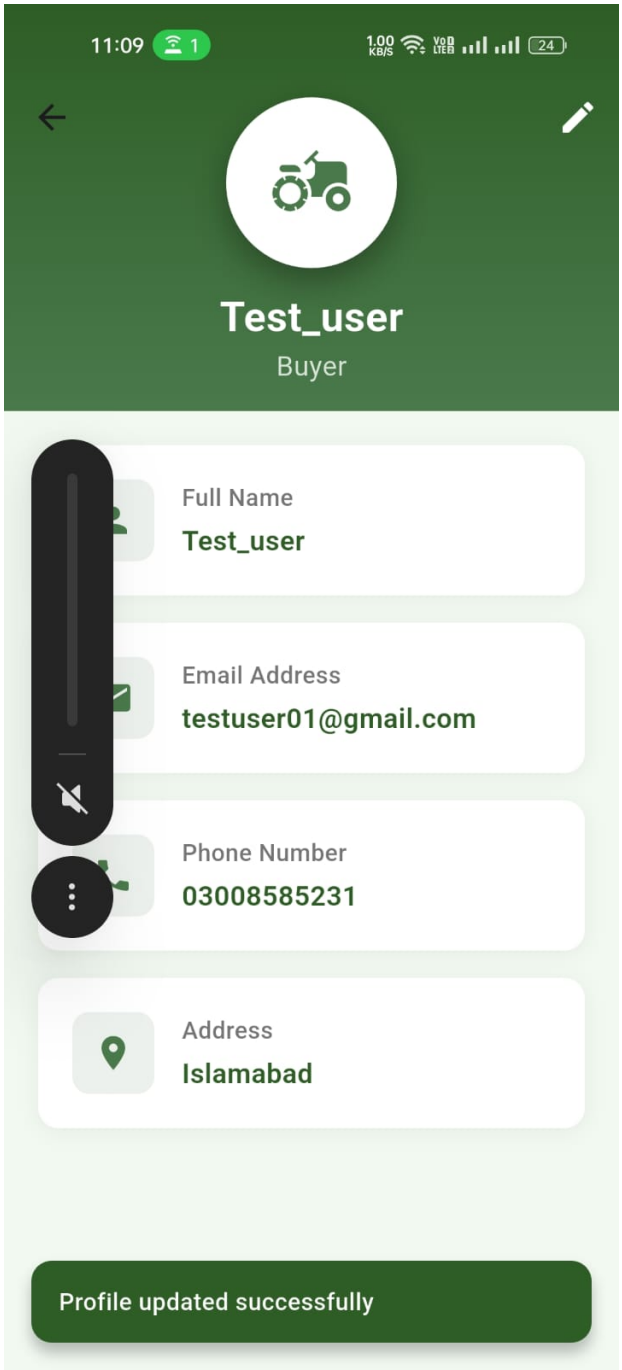
Test Case ID	UC-7-001
Use Case Name	Rent Machinery (Booking)
Summary Description	Validates that a buyer can rent available machinery listed by seller/farmer. Ensures booking confirmation, payment processing, and notification delivery work correctly, and that rented machinery becomes unavailable for others during the booking period.
Pre-Condition	1. buyer is logged in. 2. At least one seller/farmer has added machinery for rent. 3. Internet connection and payment service are active.
Test Procedure	1. View list of available machinery with details (name, category, price/day, location, contact). 2. Select machinery to rent. 3. Enter rental duration (start date, end date). 4. Choose payment method (e.g., Credit/Debit Card).
	5. Confirm booking. 6. Verify booking confirmation notification. 7. Try booking with invalid or missing data.

Test Data	**Valid Input:** Machinery: BM25 Category: Tractor Rental Duration: 3 Days (1–3 Nov 2025) Renter Name: Laiba Ahamd Email: laiba@gmail.com Address: Lahore, Punjab Payment Method: Credit/Debit Card (Successful Transaction) **Invalid Input:** 1. No machinery selected. 2. Missing rental dates. 3. Invalid email format. 4. Payment failed or cancelled.
Expected Result	**Valid Input:** Booking successful, payment confirmed, and notifications sent to both renter and machine owner. **Invalid Input:** System displays appropriate error messages (e.g., “Select machinery,” “Enter valid dates” “Payment failed,” “Fill all inputs”).
Actual Result	To be filled after testing.
Status	Valid Input: pass Invalid Input: fail

6.5.8 Test case 8: Update User Profile

Test Case ID	UC-8-001
Use Case Name	Update User Profile
Summary Description	Verifying the user profile update functionality. Ensures that the user can successfully update personal information.
Pre-Condition	User is logged in.
Test Procedure	1. Navigate to "Profile" section. 2. Edit personal details (e.g., contact info, address). 3. Save changes.
Test Data	New contact: 0302-XXXXXXXX, New address: Lahore, Punjab.

Expected Result	The system successfully updates the profile and confirms changes.
Actual Result	To be filled after testing.
Status	To be filled after testing.



6.5.9 Test case 9:
Offer Labour Services

Test Case ID	UC-9-001
Use Case Name	Offer Labour Services
Summary Description	Verifies that a labourer can successfully register and publish their service details (skills, rate, timings, and location) so that they appear in the "Available Labour" list for farmers. Also ensures the system validates missing or incorrect inputs.
Pre-Condition	1. Labourer is login. 2. Internet connection is active.
Test Procedure	1. Navigate to the **Offer Services** section. 2. Click on **Add Service**. 3. Fill out the required fields (Full Name, Skill Type, Daily Rate, Availability Timings, Location, Contact Info). 4. Optionally upload a profile image. 5. Submit the form. 6. Try submitting with missing or invalid data to check form validation.
Test Data	**Valid Input:** Name: Ali Skill: Tractor Driver Daily Rate: 1500 Timings: 8 AM – 5 PM Location: Lahore Contact: 0301-XXXXXXX Profile Image: ali.jpg **Invalid Input:** 1. Missing skill type 2. Missing rate 3. Invalid contact number 4. Missing availability timings 5. Empty location field

Expected Result	**Valid Input:** Labourer's details successfully saved and displayed in the "Available Labour" list for farmers. Confirmation message displayed (e.g., "Your service has been added successfully"). **Invalid Input:** Appropriate error messages shown (e.g., "All fields are required," "Enter a valid contact number," "Please specify your timing").
Actual Result	To be filled after testing.
Status	To be filled after testing.