DIRECT MARKET ACCESS FOR FARMER'S

CSI-1007 - SOFTWARE ENGINEERING AND PRINCIPLES LABORATORY

PREPARED BY

1. **S KISHORE (23MIC0100)**

2. **THANASHREE T(23MIC0056)**

DIRECT MARKET ACCESS FOR FARMER'S

Description:

This project aims to create a digital platform that enables farmers to sell their products directly to consumers, retailers, or wholesalers. It eliminates the need for middlemen, ensuring better profits for farmers and fresher produce for buyers. The system will feature an e-commerce website where farmers can list their products, and buyers can purchase them online.

Scope of the Project:

In-Scope:

- Online marketplace for farmers and buyers.
- Product listing and order management.
- Payment integration for secure transactions.
- Farmer and buyer account management.
- Basic inventory tracking for farmers.

Out-of-Scope:

- Logistics and delivery management.
- Advanced AI-based price prediction.

Impact of the Project:

- **For Farmers:** Directly connect with buyers, leading to higher profits.
- **For Buyers:** Access to fresh farm produce at competitive prices.
- **For Economy:** Reducing dependency on intermediaries and improving the agricultural supply chain.

Work Breakdown Structure

Process-Based Work Breakdown Structure:

Initiation:

Requirement gathering

Feasibility study

Stakeholder identification

Project charter creation

Planning:

Develop project plan and timeline

Allocate resources

Risk assessment

Approval of project scope

Design:

UI/UX design for the app

Database schema design

System architecture planning

Development:

Frontend development

Backend API development

Database setup and integration

Payment gateway integration

Logistics module implementation

Testing :

Unit testing

Integration testing

User acceptance testing

Performance testing

Deployment:

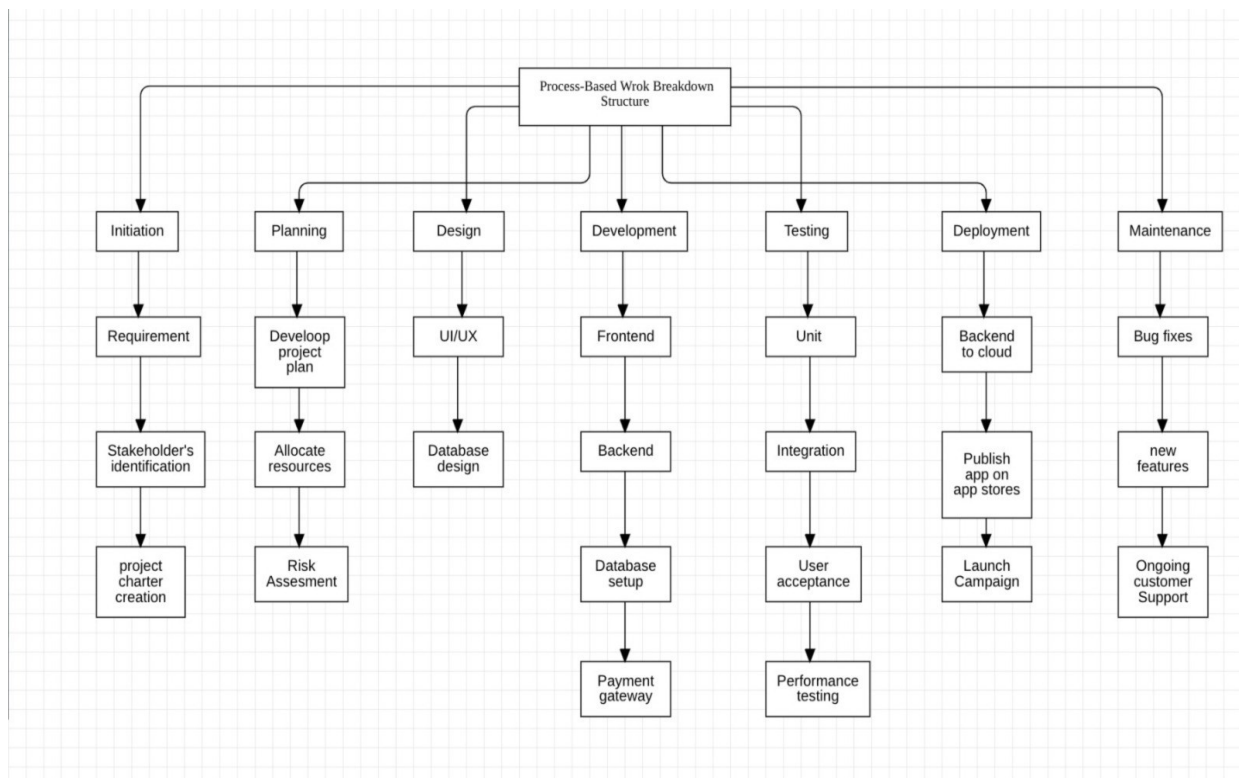Deploy backend to cloud

Publish app on app stores

Launch campaign and training

Maintenance:

Bug fixes and updates

New feature development

Ongoing customer support

Product-Based Work Breakdown Structure

Mobile Application:

Farmer registration module

Product listing feature

Buyer dashboard

Order management system

Multi-language support

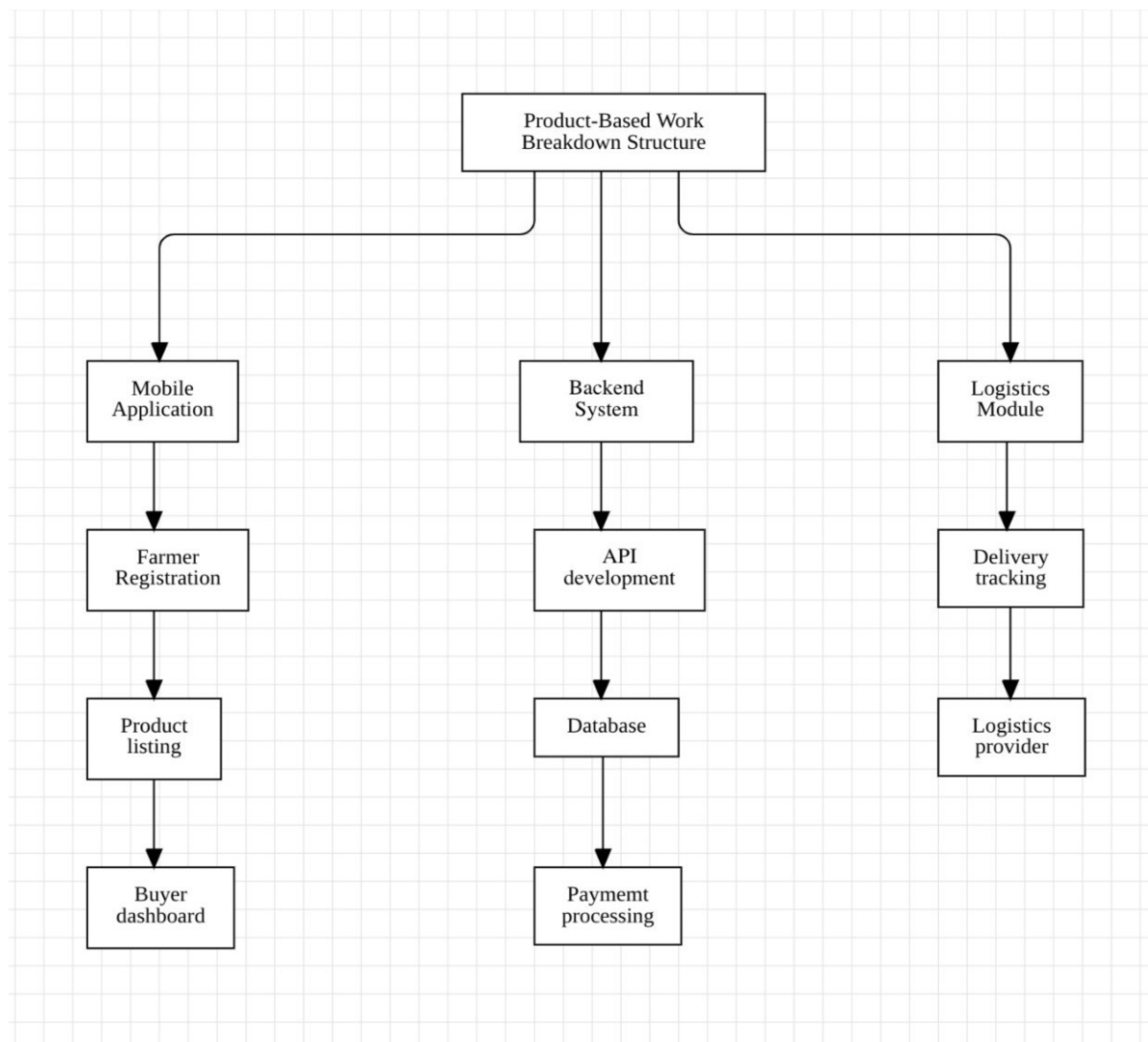**Backend System:**

API development

Database integration

Payment processing system

Market analytics module

Logistics Module:

Delivery tracking feature

Logistics provider API integration

SRS Document

Non-Functional Requirements

Performance Requirements

System should handle at least 10,000 concurrent users.
Response time should be under 2 seconds.

Security Requirements

User data encryption
Secure authentication mechanisms (e.g., OTP, 2FA)

Usability Requirements

Intuitive UI for farmers and buyers.
Mobile-friendly design.

Scalability

The system should be able to support growing numbers of users and transactions.

Compliance

Adherence to government regulations on direct farmer sales.
GDPR compliance for user data protection.

Functional Requirements:

User Authentication: Farmers and buyers must be able to register, log in, and manage their accounts.

Product Listing: Farmers should be able to add, update, and delete product listings.

Search and Filtering: Buyers should be able to search for products based on category, price, and location.

Order Management: Buyers should be able to place orders, and farmers should bable to accept or reject them.

Payment Processing: Secure online transactions should be integrated into the platform.

Messaging System: Farmers and buyers should be able to communicate through a chat system.

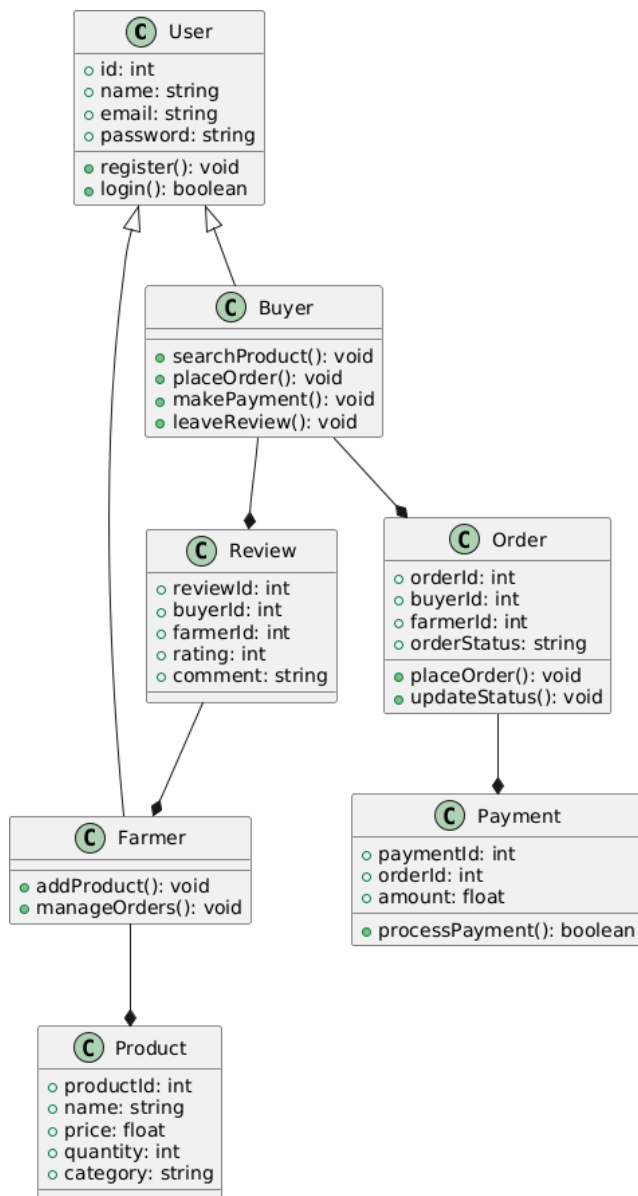**Reviews and Ratings:** Buyers should be able to rate and review products and sellers.

Notifications: Users should receive real-time notifications about orders, paymentsand messages.
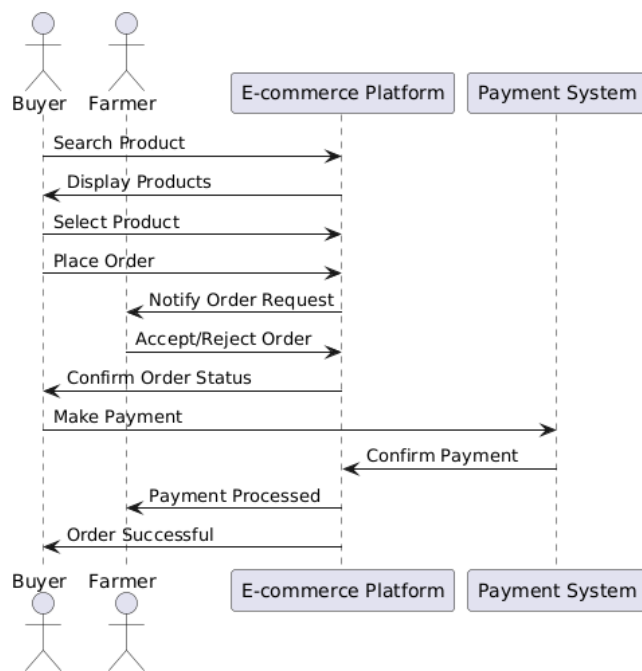
UML – Diagrams:

USE-CASE DIAGRAM:

Direct Market Access for Farmers

CLASS DIAGRAM :

User
- o id: int
- o name: string
- o email: string
- o password: string
- • register(): void
- • login(): boolean

Buyer
- • searchProduct(): void
- • placeOrder(): void
- • makePayment(): void
- • leaveReview(): void

Review
- o reviewId: int
- o buyerId: int
- o farmerId: int
- o rating: int
- o comment: string

Order
- o orderId: int
- o buyerId: int
- o farmerId: int
- o orderStatus: string
- • placeOrder(): void
- • updateStatus(): void

Farmer
- • addProduct(): void
- • manageOrders(): void

Payment
- o paymentId: int
- o orderId: int
- o amount: float
- • processPayment(): boolean

Product
- o productId: int
- o name: string
- o price: float
- o quantity: int
- o category: string

SEQUENCE DIAGRAM:

Buyer   Farmer   E-commerce Platform   Payment System

Search Product
Display Products
Select Product
Place Order
Notify Order Request
Accept/Reject Order
Confirm Order Status
Make Payment
Confirm Payment
Payment Processed
Order Successful

Buyer   Farmer   E-commerce Platform   Payment System

Testing

Scenario 1: User Registration and Login

Objective: Verify that users can successfully register and log in to the system.

Test Steps:

1. Open the application and navigate to the registration page.
2. Enter valid details (name, email, password) and submit the form.
3. Verify if an account is created and confirmation email is sent.
4. Log in using registered credentials.
5. Check if the system allows access to the user's dashboard.

Expected Result: Users should be able to register and log in without errors.

Scenario 2: Placing an Order and Processing Payment

Objective: Ensure buyers can place an order and complete a payment transaction.
Test Steps:

1. Log in as a buyer and search for a product.
2. Select a product and add it to the cart.
3. Proceed to checkout and confirm order details.
4. Choose a payment method and complete the transaction.
5. Verify if payment is processed successfully and order confirmation is received.

Expected Result: Order should be placed, payment should be processed, and a confirmation message should be displayed.

UI SCREENSCHOT OF THE LOGIN :