Q1 **4 Quarterly Audits are planned Q1 , Q2, Q3, Q4 for this Project What is your knowledge on how these Audits will happen for a BA ?**

Answer:

Quarterly audits in a project involving a Business Analyst (BA) role typically focus on assessing the project's progress, compliance, and adherence to processes. Here's an overview of how these audits might happen and what a BA's role could include:

---

1. Scope of the Audits

- **Compliance Audit**: Check if processes are followed per regulatory, organizational, or project standards.
- **Deliverables Audit**: Ensure that requirements and deliverables are aligned with the business goals and meet stakeholder expectations.
- **Performance Audit**: Measure the performance of the BA in documenting, analysing, and communicating requirements effectively.

---

2. Key Steps in Each Quarterly Audit

Q1: Planning and Initial Assessment

- **BA Role**:
  - Present requirement gathering progress, including initial stakeholder engagement and documentation.
  - Ensure that the scope document is up-to-date and approved by stakeholders.
  - Provide evidence of any workshops, interviews, or surveys conducted.

Q2: Mid-Year Progress Check

- **BA Role**:
  - Provide updates on requirements traceability matrices (RTM).
  - Showcase completed and in-progress workflows, process maps, and other models.
  - Address gaps or delays and explain corrective measures.

Q3: Pre-Implementation and Validation

- **BA Role**:
  - Demonstrate readiness for testing phases (user stories, use cases, acceptance criteria).
  - Highlight changes to requirements and how they were managed (change management process).
  - Share documentation of stakeholder approvals and sign-offs.

Q4: Final Review and Lessons Learned

- **BA Role**:
  - Present final deliverables, ensuring alignment with initial objectives.
  - Provide a post-implementation review of requirements' success and stakeholder satisfaction.
  - Collaborate on documenting lessons learned for continuous improvement.

---

3. Audit Process Flow

1. **Preparation**:
   - Collect relevant documentation (BRD, FRD, RTM, user stories, process flows, meeting minutes, etc.).
   - Ensure all tools and templates used for requirements management are updated and ready.
2. **Execution**:

    - Participate in interviews or meetings with auditors.
    - Answer questions related to requirements gathering, traceability, stakeholder communication, and risk identification.
3. **Report Review**:
    - Review findings and address non-compliance issues.
    - Prepare a corrective action plan for gaps identified.
4. **Follow-Up**:
    - Track progress on identified action items and ensure closure by the next quarter.

---

4. Best Practices for BA in Audits

- Maintain thorough and organized documentation.
- Use tools like JIRA, Confluence, or Excel for RTM and project tracking.
- Communicate proactively with auditors and project managers.
- Focus on measurable results and alignment with business goals.

Example of Audit based on case study

1. Requirement Phase (Q1)
Stage: Gather and Analyse Requirements
Estimated Completion: 6 weeks
Checklist:

1. Confirm elicitation techniques used (e.g., brainstorming, interviews, workshops).
2. Ensure all stakeholders' requirements are documented and validated.
3. Verify that requirements align with the project scope (facilitating farmers and manufacturers).
4. Check the completeness and clarity of the Business Requirement Document (BRD).
5. Ensure all assumptions, constraints, and dependencies are recorded.
6. Obtain sign-offs from all stakeholders (farmers, manufacturers, and committee).
7. Validate requirements for feasibility within the given budget and timeline.

---

2. Design Phase (Q2)
Stage: System and Technical Design
Estimated Completion: 8 weeks
Checklist:

1. Confirm the design covers all functional and non-functional requirements.
2. Review the system's high-level architecture (3-tier architecture: UI, business logic, and database).
3. Check the detailed design for modules such as user registration, product management, and order processing.
4. Validate database schema with a focus on scalability and security.
5. Ensure UI/UX designs are user-friendly and suitable for farmers in remote areas.
6. Verify integration points between components (e.g., payment gateway, delivery tracking).
7. Obtain client approvals on the System Design Document (SDD).

---

3. Implementation Phase (Q3)
Stage: Code Development and Unit Testing
Estimated Completion: 12 weeks
Checklist:

1. Verify coding standards and guidelines are being followed (e.g., naming conventions,

comments).
2. Ensure individual modules (registration, product search, etc.) are being implemented according to the design.
3. Confirm code reviews are being conducted regularly by senior developers.
4. Validate unit tests for each module and confirm code coverage exceeds 90%.
5. Ensure database connections, APIs, and integrations are tested for functionality.
6. Monitor progress against the development schedule and identify delays.
7. Record and address defects found during unit testing.

4. Testing Phase (Q4)
Stage: System Integration Testing (SIT) and User Acceptance Testing (UAT)
Estimated Completion: 10 weeks
Checklist:

1. Validate that all test cases are derived from requirements and design documents.
2. Verify the SIT environment mirrors the production environment.
3. Confirm all modules (e.g., product browsing, cart management, payments) integrate correctly.
4. Ensure UAT is conducted with real end-users (farmers and manufacturers).
5. Record and prioritize defects found during testing phases.
6. Validate non-functional requirements (e.g., system performance, security, usability).
7. Obtain sign-offs from stakeholders after successful UAT.

5. Deployment and Maintenance Phase (Post-Launch)
Stage: Deployment and Support
Estimated Completion: 6 weeks (for initial deployment)
Checklist:

1. Confirm the deployment environment is prepared (servers, network, database).
2. Validate successful migration of test data to production.
3. Monitor the system for issues during initial deployment.
4. Verify that the support team is trained and ready to handle issues.
5. Collect feedback from users and prioritize enhancement requests.
6. Review the change management process for handling future updates.
7. Ensure compliance with any legal or regulatory requirements for online stores.

Q 2 BA Approach Strategy - 6 Marks
Before the Project is going to Kick Start, The Committee asked Mr Karthik to submit BA Approach Strategy
Write BA Approach strategy (As a business analyst, what are the steps that you would need to follow to complete a project – What Elicitation Techniques to apply, how to do Stakeholder Analysis
RACI/ILS, What Documents to Write, What process to follow to Sign off on the Documents, how to
take Approvals from the Client, What Communication Channels to establish n implement, how to Handle Change Requests, how to update the progress of the project to the Stakeholders, how to take signoff on the UAT- Client Project Acceptance Form)

Answer:

Defination:- A Business Analysis (BA) Approach & Strategy defines the method, techniques, and processes that a business analyst will follow to identify, analyze, document, and manage requirements for a project. It serves as a roadmap to ensure that business objectives are met efficiently.

Benefits of a BA Approach & Strategy

1. **Improves Efficiency** – Streamlines requirement gathering and analysis.
2. **Enhances Communication** – Ensures alignment among stakeholders and teams.
3. **Reduces Risks** – Identifies potential issues early in the process.
4. **Ensures Quality Deliverables** – Provides a structured approach for accurate requirements.
5. **Supports Decision-Making** – Helps stakeholders make informed business decisions.
6. **Facilitates Project Success** – Ensures business needs are met within scope, time, and budget.

A) Elicitation Technique:

- **Techniques Used**: Workshops, Interviews, Focus Groups, and Prototypes.
- **Justification**:
    - Workshops ensure collaboration among stakeholders (e.g., farmers, manufacturers, committee).
    - Interviews help gather insights from Peter, Kevin, and Ben on farmer needs.
    - Focus groups with manufacturers identify product specifics (e.g., fertilizers).
    - Prototypes visualize the application for stakeholder validation.

B) Stakeholder Analysis:

- **Stakeholders**:
    - **Primary**: Farmers, Manufacturers, Mr. Henry.
    - **Secondary**: SOONY Committee, APT IT SOLUTIONS (Developers, Testers, PM).
- Use a **RACI Chart** for responsibilities:
    - **Responsible**: Developers, PM.
    - **Accountable**: Mr. Henry, SOONY Committee.
    - **Consulted**: Peter, Kevin, Ben.
    - **Informed**: Testers, DB Admin.

C) Documents to Write:

1. **Business Requirements Document (BRD) -** A **Business Requirements Document (BRD)** is a formal document that outlines the high-level business needs, goals, and objectives of a project. It defines what the business wants to achieve without specifying how it will be implemented.
2. **Functional Requirements Specification (FRS)-** A **Functional Requirements Specification (FRS)** describes the functional capabilities and behaviors of a system or application. It defines how the system should perform specific tasks based on business needs**.**
3. **System Requirement Specification (SRS)- A System Requirement Specification (SRS)** is a detailed document that defines **functional and non-functional** requirements for a software system. It acts as a contract between the business and the technical team
4. **Stakeholder Register- A Stakeholder** Register is a document that lists all individuals, groups, or organizations **involved in or affected** by a project. It includes their **roles, interests, influence, and engagement levels**.
5. **Change Request Log - A Change Request Log** is a document used to track all **requested changes** to a project, system, or process. It records details about change requests, their

status, and their impact on scope, time, and cost.

D) Process for Sign-Off: - The Sign-Off Process refers to the formal approval of a document, project phase, or deliverable by relevant stakeholders. It ensures that all requirements have been met and that the work is ready for the next stage or final delivery

- Draft documents → Conduct review meetings → Address feedback → Stakeholder approval → Sign-off.

E) Client Approvals: - Client approval is the formal acceptance given by a client for a **project, deliverable, document, or change request** after reviewing and ensuring that it meets their requirements and expectations. It serves as a confirmation that the work has been completed as per the agreed scope and specification

- Use structured review meetings with Mr. Henry, documented minutes of meetings (MoMs), and email confirmations.

F) Communication Channels:

- Tools: Email, Slack, Jira, MS Teams.
- Frequency: Weekly updates, bi-weekly sprint reviews.

G) Change Request Handling: - Change Request Handling is the formal process of managing and responding to requests for changes in a project, product, or system. These requests could involve adjustments to the scope, schedule, budget, or requirements during the project's lifecycle. The process ensures that any changes are evaluated, approved, and documented properly to minimize risks and ensure the project remains aligned with its objectives**.**

- **Impact Analysis**: Assess scope, timeline, cost changes.
- **Feasibility Study**: Evaluate technical and resource feasibility.
- **Effort Estimation**: Calculate developer and testing hours for change.

H) Progress Updates:

- Weekly dashboards, sprint reviews, and monthly progress reports to stakeholders.

I) **UAT Sign-Off: - UAT Sign-Off (User Acceptance Testing Sign-Off)** refers to the formal approval or acceptance given by the client, business stakeholders, or end-users after successfully completing **User Acceptance Testing (UAT).** It signifies that the system or product has been thoroughly tested by the intended users and meets the defined business requirements, and is ready for deployment or go-live

- Deliver UAT results → Incorporate feedback → Obtain client acceptance form.

Q3 Explain and illustrate 3-tier architecture?

Answer: -

The 3-tier architecture is a client-server architecture pattern that organizes applications into three distinct layers, each with a specific function. It enhances scalability, maintainability, and flexibility.
Layers of 3-Tier Architecture

1. **Presentation Layer (Client Tier)**
   - The topmost layer that interacts with users.
   - Responsible for displaying UI (User Interface) and handling user interactions.
   - Can be a web browser, mobile app, or desktop application.
   - Technologies: HTML, CSS, JavaScript, React, Angular, Vue.js.
2. **Application Layer (Business Logic Tier or Middle Tier)**
   - Handles business logic and processes user requests.
   - Communicates between the Presentation Layer and Data Layer.
   - Processes inputs, applies business rules, and makes decisions.

    - Technologies: Java, .NET, Python, Node.js, PHP, Spring Boot, Django.
3. **Data Layer (Database Tier or Backend Tier)**
    - Stores and retrieves data efficiently.
    - Ensures data integrity, security, and availability.
    - Can use SQL or NoSQL databases.
    - Technologies: MySQL, PostgreSQL, MongoDB, Oracle, Firebase.

Advantages of 3-Tier Architecture

- Scalability – Each layer can be scaled independently.
- Security – Business logic and data are separated from the user interface.
- Maintainability – Changes in one layer don't affect the others.
- Flexibility – Supports different technologies for each layer

1. **Presentation Layer**: User interface for farmers and manufacturers (web/mobile app).
2. **Business Logic Layer**: Application logic for transactions, search, and product management.
3. **Data Layer**: Database for storing product and user information.

Q4 Business Analyst should keep What points in his/her mind before he frames a Question to ask to the Stakeholder (5W 1H – SMART – RACI – 3 Tier Architecture – Use Cases, Use case Specs, Activity Diagrams, Models, Page designs)

Answer

5W1H Framework:

- **What**: What products do farmers need?
- **Why**: Why is online purchasing important?
- **When**: When will products be delivered?
- **Where**: Where are manufacturers located?
- **Who**: Who are the stakeholders?
- **How**: How will the system handle transactions?

SMART Technique:

- **Specific**: Enable farmers to buy products online.
- **Measurable**: Reduce procurement time by 30%.
- **Achievable**: Develop within 18 months.
- **Relevant**: Address remote farmers' needs.
- **Time-bound**: Deploy within budget and schedule.

RACI Chart:

Defines roles and responsibilities for stakeholders and team members

1. Use Cases (Functional Requirements Gathering)

Use cases define how users interact with the system.

General Use Case Questions

- What are the core functionalities of the online agriculture store? (E.g., product browsing, ordering, payment, tracking)
- Who are the key users of the system? (Customers, Admin, Vendors, Delivery Personnel, etc.)
- What types of agricultural products will be sold? (Seeds, fertilizers, equipment, organic

products, etc.)

- What are the payment methods supported? (UPI, Credit/Debit Card, Net Banking, COD)
- How will the delivery process work? (Own logistics, third-party shipping)
- What types of user accounts need to be created? (Customer, Admin, Vendor)
- Will the system have a recommendation engine for personalized suggestions?

Use Case Example: "Placing an Order"

- How does a customer select and add products to the cart?
- Will users be able to apply discounts, promo codes, or loyalty points?
- What happens if an item goes out of stock after adding to the cart?
- How will order tracking work?

---

2. **Use Case Specifications (Detailed Flow & Requirements)**

Use Case Specifications include details of how each use case operates.

User Authentication & Registration

- What details are required for user registration? (Name, Email, Mobile, Address, etc.)
- Will there be social media or Google login integration?
- What security measures are in place for authentication? (OTP, Two-factor authentication)
- How will password recovery and reset work?

Product Management (Admin Panel & Vendor Side)

- How will new products be added, updated, and removed?
- Will there be an approval process for vendors adding products?
- What categories and filters will be available for product browsing?

Cart & Checkout Process

- Can users modify the cart before checkout?
- What details will be captured at checkout? (Billing address, Shipping method, Payment option)
- How will stock be validated before checkout completion?
- How will taxes and shipping charges be calculated?

Order Tracking & Delivery

- How will order status updates be handled? (Email, SMS, Push notifications)
- Will customers have an option to cancel or modify their orders after placement?
- How will return and refund policies work?

---

3. **Activity Diagrams (Process Workflows & Scenarios)**

Activity diagrams help in visualizing the workflows and decision points.

Questions to Frame Activity Diagrams

- What happens when a user logs in successfully or fails authentication?
- What is the process flow for order placement and payment?
- How does inventory update when a product is purchased?
- What are the possible refund scenarios and steps?
- How will customer support or complaint handling workflow be structured?

---

4. **Models (ERD, Class Diagrams, Process Models, Data Flow Diagrams)**

Models help define the structure and data flow of the system.

Entity-Relationship Diagram (ERD)

- What are the key entities in the system? (Users, Orders, Products, Payments, Vendors)
- How are products linked to categories and subcategories?
- What relationships exist between users and orders?

Data Flow Diagram (DFD)

- How does data flow from the frontend to the backend and the database?
- What are the interactions between external services (payment gateway, shipping API)?

Class Diagrams (For Object-Oriented Design)

- What are the main classes, attributes, and methods in the system?
- How will user roles (Customer, Admin, Vendor) be structured?

---

5. **Page Designs (UI/UX Wireframes & Prototypes)**

The goal is to define how each page will look and function.

Homepage & Product Listing Page

- What key elements should be displayed on the homepage? (Top categories, featured products, offers)
- How should the search and filter options be structured?

Product Detail Page

- What details should be included? (Description, Images, Pricing, Reviews, Availability)
- Should there be a "Compare Products" option?

Cart & Checkout Page

- What steps should be included in the checkout process?
- How will users enter their shipping address and choose a delivery method?

Admin Dashboard

- What key metrics should be displayed on the dashboard? (Orders, Revenue, Active Users)
- How will product management and stock updates be handled?

Q 5 **As a Business Analyst, What Elicitation Techniques you are aware of**

Definition of Elicitation Technique

Elicitation Techniques are structured methods used by **business analysts, project managers, and stakeholders** to gather, extract, and clarify requirements for a system, product, or process. These techniques help in identifying **business needs, user expectations, and constraints** to ensure the successful development of a project

Benefits of Elicitation Techniques

1. **Accurate Requirements Gathering** – Ensures business and user needs are well-defined.
2. **Improved Stakeholder Communication** – Enhances clarity and understanding.
3. **Reduces Project Risks** – Identifies gaps, assumptions, and constraints early.
4. **Enhances Decision-Making** – Provides data-driven insights for better solutions.
5. **Saves Time & Cost** – Prevents rework by capturing requirements correctly.
6. **Encourages Collaboration** – Engages different stakeholders for a comprehensive view

1. Document Analysis

Definition:

Reviewing existing documentation such as business reports, policies, manuals, and competitor websites to gather insights into system requirements.

Application in Online Agriculture Store:

- Analyzing existing agriculture e-commerce platforms (e.g., Amazon Agri, BigHaat) for features and best practices.
- Reviewing agricultural product catalogs to understand product categorization.
- Examining legal requirements for selling fertilizers and seeds online.

2. Reverse Engineering

Definition:

Analyzing an existing system or competitor’s platform to understand its functionality and workflows.

Application in Online Agriculture Store:

- Studying competitor websites to identify key features like crop-based product recommendations.
- Understanding how existing agricultural e-commerce systems handle logistics and inventory.
- Identifying the user journey from browsing products to completing a purchase.

3. Focus Groups

Definition:

A discussion involving selected stakeholders (farmers, suppliers, customers) to gather feedback on product requirements and expectations.

Application in Online Agriculture Store:

- Conducting a session with farmers to understand their online shopping behavior.
- Discussing with vendors about their challenges in selling products online.
- Gathering insights on preferred payment methods and delivery expectations.

4. Observations

Definition:

Watching how users (farmers, suppliers, customers) interact with an existing system or perform tasks in real-world scenarios.

Application in Online Agriculture Store:

- Observing how farmers currently buy agricultural products (physical stores vs. online platforms).
- Analyzing the process of product selection and purchasing behavior.
- Noting challenges farmers face in using digital platforms.

5. Workshops

Definition:

Interactive sessions with stakeholders to collaboratively define requirements, discuss problems, and brainstorm solutions.

Application in Online Agriculture Store:

- Conducting workshops with agricultural product suppliers to define product categories.
- Discussing challenges faced in supply chain and logistics.
- Gathering feedback from UX/UI designers and developers on system functionality.

6. JAD (Joint Application Development) Interview

Definition:

A structured session involving key stakeholders (business owners, customers, developers) to define requirements collaboratively.

Application in Online Agriculture Store:

- Bringing together farmers, logistics providers, and IT developers to finalize system workflows.
- Conducting structured discussions on order processing and delivery tracking.
- Identifying customization requirements like seasonal product promotions.

7. Prototyping
Definition:
Developing mockups, wireframes, or a working model of the system to visualize and validate requirements.
Application in Online Agriculture Store:

- Creating wireframes of product catalog and checkout pages for user feedback.
- Building a clickable prototype to test user navigation.
- Developing an MVP (Minimum Viable Product) with core functionalities for testing.

8. Survey / Questionnaire
Definition:
A set of structured questions to collect quantitative or qualitative data from stakeholders.
Application in Online Agriculture Store:

- Conducting surveys with farmers to understand their preferences for online shopping.
- Collecting data on payment methods, preferred delivery time, and product expectations.
- Identifying the percentage of users willing to switch to an online agricultural store.

9. Brainstorming
Definition:
A creative idea-generation technique where multiple stakeholders contribute innovative ideas and solutions.
Application in Online Agriculture Store:

- Brainstorming with business owners to define unique features like "Crop-based product recommendations."
- Discussing marketing strategies for increasing customer engagement.
- Generating ideas for loyalty programs or referral discounts.

10. Use Case Specifications
Definition:
Detailed descriptions of how users will interact with the system, including steps, actors, and expected outcomes.
Application in Online Agriculture Store:

- Defining use cases like **"Placing an Order," "Managing Inventory,"** and **"Processing Payments."**
- Outlining alternate flows for order cancellations or refunds.
- Documenting system responses for failed payments or stock unavailability.

Q6 **Which Elicitation Techniques can be used in this Project and Justify your selection of Elicitation Techniques**?

- **Techniques Used**: Workshops, Interviews, Focus Groups, Document analysis and Prototypes.
- **Justification**:
  - Workshops ensure collaboration among stakeholders (e.g., farmers, manufacturers, committee).
  - Interviews help gather insights from Peter, Kevin, and Ben on farmer needs.
  - Focus groups with manufacturers identify product specifics (e.g., fertilizers).
  - Prototypes visualize the application for stakeholder validation.

- Document Analysis: Analyse SOONY's prior projects

Q7 **Make suitable Assumptions and identify at least 10 Business Requirements**

Definition of Business Requirement

A **Business Requirement** is a high-level **need, goal, or objective** that a business must fulfill to provide value to its stakeholders. It defines **what** a business aims to achieve rather than detailing **how** it will be implemented. Business requirements typically focus on **business goals, processes, rules, and constraints** that drive the need for a system, product, or service.

Key Characteristics of Business Requirements

- **Strategic in Nature** – Aligns with business goals and objectives.
- **High-Level Definition** – Describes the **what** rather than the **how** of a solution.
- **Stakeholder-Driven** – Focuses on user needs and business benefits.
- **Independent of Technology** – Does not specify technical implementation details.

Business Requirements (Including Stakeholder Requirements)

1. **User Login and Registration** – Farmers and manufacturers should be able to register and log in using email and password.
2. **Product Catalog** – The system should display details of fertilizers, seeds, and pesticides provided by manufacturers.
3. **Product Search** – Farmers should be able to search for products using keywords.
4. **Add to Cart & Buy-Later List** – Farmers should be able to add products to the cart or save them for later purchases.
5. **Secure Payment Options** – The system should support payments via COD, Credit/Debit Card, and UPI.
6. **Order Confirmation** – Users should receive an email confirmation regarding their order status.
7. **Order Tracking** – Farmers should be able to track the delivery status of their orders.
8. **Admin Panel** – Admin should have access to manage inventory, orders, and user accounts.
9. **Multi-Language Support** – The system should support multiple languages to accommodate regional farmers.
10. **User-Friendly Interface** – The platform should be easy to navigate for farmers with limited technical knowledge.

Q8 **List your assumptions**

Definition of Assumption

An **Assumption** is a statement or condition that is **believed to be true** without concrete proof, which serves as a basis for planning, decision-making, or execution in a project, business, or analysis. Assumptions help define the **context, constraints, and expectations** within which a system, project, or process operates.

Key Characteristics of Assumptions

- **Unverified but Expected to Hold True** – Not proven but considered valid for planning.
- **Affects Decision-Making** – Impacts strategies, solutions, and risk assessments.
- **Can Change Over Time** – May need validation or revision as more information becomes available.

1. Internet connectivity is available in remote areas.
2. Farmers and manufacturers will use smartphones.
3. Delivery service providers are available.
4. The project stays within budget and timeline

Q9 **Give Priority 1 to 10 numbers (1 being low priority – 10 being high priority) to these Requirements**
after discussions with the stakeholders

Business Requirement Priority

Business Requirement Priority refers to the ranking or categorization of business requirements based on their importance, urgency, and impact on the project or business objectives. Prioritizing requirements helps teams focus on the most critical needs first, ensuring efficient resource allocation and project success.

Common Priority Levels

1. High (Must-Have) – Essential for business operations; failure to implement these will cause project failure.
2. Medium (Should-Have) – Important but not critical; can be implemented in later phases.
3. Low (Nice-to-Have) – Additional features that enhance usability but are not mandatory

Req ID	Req Name	Req Description	Priority
BR001	**User Login and Registration**	Farmers and manufacturers should be able to register and log in using email and password.	1
BR002	**Product CatLog**	The system should display details of fertilizers, seeds, and pesticides provided by manufacturers	3
BR003	**Product Search**	Farmers should be able to search for products using keywords	7
BR004	**Add to Cart & Buy-Later List**	Farmers should be able to add products to the cart or save them for later purchases.	8
BR005	**Secure Payment Options**	The system should support payments via COD, Credit/Debit Card, and UPI.	2
BR006	**Order Confirmation**	Users should receive an email confirmation regarding their order status.	4
BR007	**Order Tracking**	Farmers should be able to track the delivery status of their orders	5
BR008	**Admin Panel**	Admin should have access to manage inventory, orders, and user accounts	6
BR009	**Multi-Language Support**	The system should support multiple languages to accommodate regional farmers.	9
BR0010	**User-Friendly Interface**	The platform should be easy to navigate for farmers with limited technical knowledge.	10

Q10 USE CASE Diagram

A **use case** is a description of how users interact with a system to achieve specific goals. It represents the functional requirements of a system and describes the steps, interactions, and conditions involved in accomplishing a task.

1. What Are Use Cases?

- **Definition:** A use case defines a specific interaction between a user (actor) and a system to accomplish a goal.
- **Purpose:** To capture the functional requirements of a system and provide a clear understanding of what the system needs to do.

2. How to Derive a Use Case Diagram?

A **use case diagram** visually represents the relationships between actors (users) and use cases in a system. Here's how to derive it:

Steps:

1. **Identify Actors:**
   - Determine who or what interacts with the system (e.g., users, external systems, or devices).
   - Examples: Customers, Admins, Payment Gateway, etc.
2. **Identify Use Cases:**
   - List all the goals or tasks that actors want to accomplish using the system.
   - Examples: "Login", "Place Order", "Generate Report".
3. **Define System Boundaries:**
   - Clearly specify what is inside and outside the system.
4. **Determine Relationships:**
   - **Association:** Links between actors and use cases (e.g., a customer places an order).
   - **Include:** Represents a common functionality used by multiple use cases (e.g., "Login" is included in "Place Order").
   - **Extend:** Represents optional or conditional behaviour (e.g., "Apply Discount" extends "Checkout").
   - **Generalization:** Represents inheritance between actors or use cases (e.g., "Admin" is a specialized "User").
5. **Draw the Diagram:**
   - Use standard UML notation:
     - Actors are stick figures.
     - Use cases are ovals.
     - The system boundary is a rectangle around use cases.

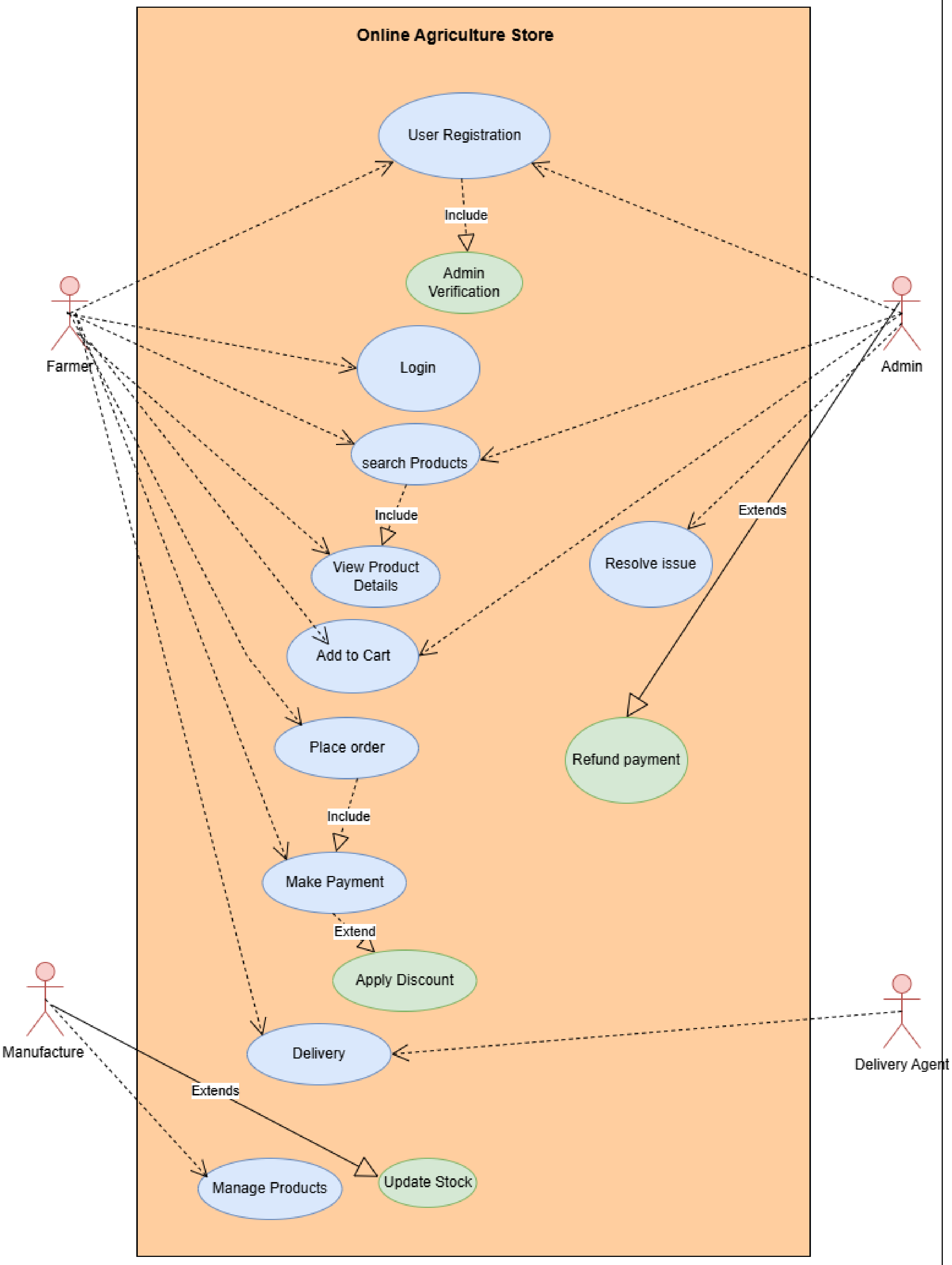
Actors

1. **Farmer (Primary Actor)**:
   - Searches for products, views product details, places orders, and tracks deliveries.
   - Registers and logs into the system.
2. **Manufacturer (Primary Actor)**:
   - Manages product listings, updates stock, and tracks orders.
3. **Admin (Secondary Actor)**:
   - Oversees the platform, manages users and products, and resolves issues.
4. **Delivery Agent (Secondary Actor)**:
   - Picks up and delivers orders.

Use Cases

1. **User Registration**:
   - Farmers and manufacturers register on the platform.
   - Admin verifies user details if needed.
2. **Login**:
   - Users authenticate themselves using their credentials.
3. **Search Products**:
   - Farmers search for seeds, fertilizers, pesticides, or other items.
4. **View Product Details**:
   - Farmers view detailed information about a selected product.
5. **Add to Cart**:
   - Farmers add desired products to their shopping cart.
6. **Place Order**:
   - Farmers confirm the items in their cart and place an order.
7. **Make Payment**:
   - Farmers choose a payment method (e.g., online banking, cash on delivery).
8. **Delivery**:
   - Delivery agents handle logistics for delivering the order to the farmer.
9. **Manage Products**:
   - Manufacturers add, update, and manage product listings.
10. **Resolve Issues**:

- Admin handles disputes, resolves technical issues, and ensures smooth operations


Online Agriculture Store
User Registration
Include
Admin Verification
Login
search Products
Include
View Product Details
Add to Cart
Place order
Include
Make Payment
Extend
Apply Discount
Delivery
Manage Products
Update Stock
Extends
Resolve issue
Extends
Refund payment
Farmer
Admin
Manufacture
Delivery Agent

-

Q **11 Use Case Specification**

Definition of Use Case Specification

A **Use Case Specification** is a detailed document that describes how a system interacts with users (actors) to achieve a specific goal. It outlines the steps, conditions, and requirements for a particular **use case** in software or system development. The specification typically includes **actors, preconditions, flow of events, postconditions, and exceptions.**

Benefits of Use Case Specification

1. **Clear Requirements** – Defines system functionalities and interactions in a structured manner.
2. **Improves Communication** – Bridges the gap between stakeholders, developers, and testers.
3. **Enhances System Understanding** – Provides a clear picture of system behaviour from a user's perspective.
4. **Aids in Testing** – Helps in creating test cases and validating system functionality.
5. **Identifies Edge Cases** – Documents alternate and exceptional flows, reducing ambiguity.
6. **Facilitates Documentation** – Acts as a reference for future system modifications and updates.
7. **Supports Agile Development** – Can be used in iterative development to refine system behaviour.

1. Understand the components:

- **Use Case Name**: Identify the use case that you're mapping to an activity diagram.
- **Description**: Summarize the overall goal of the use case.
- **Primary Actor**: The main person/system interacting with the system to achieve the use case goal.
- **Secondary Actor**: Any other participants, such as external systems or users who support the use case.
- **Basic Flow**: The standard flow of activities that occur when everything goes as expected.
- **Alternate Flow**: Variations of the basic flow where certain conditions or events cause deviations from the standard path.
- **Exceptional Flow**: Activities that handle error or unexpected situations.
- **Pre-condition**: What must be true before the use case can begin.
- **Post-condition**: What must be true after the use case completes successfully.
- **Assumptions**: Conditions or facts assumed to be true for the activity to occur.
- **Constraints**: Limitations or rules that apply.
- **Dependencies**: Other use cases, systems, or services the use case depends on.
- **Inputs and Outputs**: The data or objects that are input into and output from the system during the use case.
- **Business Rules**: Specific business constraints or rules that govern the activity.
- **Miscellaneous Information**: Any additional notes or considerations

1. User Registration

Use Case Name	**User Registration**
Description	Allows users to register to the platform by providing necessary details.
Primary Actor	Farmer
Secondary Actor	Admin
Basic Flow	User provides registration details → System validates details → System creates account.
Alternate Flow	If the user details are incomplete, prompt the user to fill in missing details.
Exceptional Flow	If the username already exists, display an error message.
Pre-condition	User must have an email address or mobile number for registration.
Post-condition	User account is successfully created, and the user receives a confirmation.
Assumptions	Users will have internet access to register.
Constraints	Registration should not take more than 2 minutes.
Dependencies	Database must be operational to store the user data.
Inputs and Outputs	**Inputs**: User details; **Output**s: Confirmation message and user account.
Business Rules	Users must be at least 18 years old to register.
Miscellaneous Information	Support for multilingual registration is planned for future releases.

2 Search Product

Use Case Name	**Search Product**
Description	Enables users to search for available products (seeds, fertilizers, pesticides).
Primary Actor	Farmer
Secondary Actor	Admin
Basic Flow	User enters search criteria → System displays matching products.
Alternate Flow	If no products match the criteria, suggest related products or categories.
Exceptional Flow	If the system is offline, display a maintenance notification.
Pre-condition	User must be logged in to search for products.
Post-condition	Relevant product results are displayed to the user.
Assumptions	Users are familiar with the product categories.
Constraints	The search function must respond within 3 seconds.
Dependencies	Product database should be up-to-date.
Inputs and Outputs	**Inputs**: Search criteria; **Outputs**: List of matching products.
Business Rules	Only available products will be shown to users.
Miscellaneous Information	Future iterations may include voice search functionality.

3 Add Product to Cart

Use Case Name	**Add Product to Cart**
Description	Allows users to add selected products to their shopping cart.
Primary Actor	Farmer
Secondary Actor	Admin
Basic Flow	User selects a product → System adds it to the cart → System confirms addition.
Alternate Flow	If the product is out of stock, notify the user and suggest alternatives.
Exceptional Flow	If the system crashes, display an error and retain cart data for recovery.
Pre-condition	User must have an active account and be logged in.
Post-condition	Selected product is added to the user’s cart.
Assumptions	Users are aware of the product they want to add.
Constraints	Cart updates must reflect in real-time.
Dependencies	Inventory system must synchronize stock levels with the cart system.
Inputs and Outputs	**Inputs**: Product selection; **Outputs**: Confirmation message and updated cart.
Business Rules	Users cannot add more than 50 items to the cart at a time.
Miscellaneous Information	Wishlist feature integration is under development.

4 Making Payment

Use Case Name	**Making Payment**
Description	Facilitates users in paying for the products in their cart.
Primary Actor	Farmer
Secondary Actor	Payment Gateway
Basic Flow	User selects payment method → Enters payment details → System processes payment → Displays receipt.
Alternate Flow	If payment fails, suggest alternate methods or retry.
Exceptional Flow	If the payment gateway is down, notify the user and log the error.
Pre-condition	User must have products in their cart and sufficient funds.
Post-condition	Payment is successfully processed, and the order is confirmed.
Assumptions	Users have access to secure payment methods.
Constraints	Payment must be completed within the session timeout period.
Dependencies	Payment gateway and bank systems must be operational.
Inputs and Outputs	**Inputs**: Payment details; **Outputs**: Payment confirmation and receipt.
Business Rules	Transactions are encrypted for security.
Miscellaneous Information	Refund processing feature will be enhanced in future updates.

5 Delivery

Use Case Name	**Delivery**
Description	Ensures ordered products are delivered to the user’s specified location.
Primary Actor	Farmer
Secondary Actor	Delivery Partner
Basic Flow	Order is confirmed → Products are packed → Dispatched → Delivered to user.
Alternate Flow	If delivery is delayed, notify the user with an estimated delivery time.
Exceptional Flow	If the address is invalid, contact the user for clarification.
Pre-condition	User must provide a valid delivery address and contact information.
Post-condition	Products are delivered successfully to the user’s location.
Assumptions	Users are aware of the delivery process and timelines.
Constraints	Delivery time should not exceed 7 days from the order date.
Dependencies	Delivery partner must update order tracking status.
Inputs and Outputs	**Inputs**: Order details; **Outputs**: Delivered products and delivery confirmation.
Business Rules	Only available products will be shown to users.
Miscellaneous Information	Users can track their orders in real-time.

6 Manage Products

Use Case Name	**Manage Products**
Description	Ensure all inventory supply chain manage through dark house
Primary Actor	Manufacture
Secondary Actor	Admin
Basic Flow	Inventory management system helps track 3 things 1.Manage stock level 2.Sales 3.Customer orders
Alternate Flow	Use inventory tracking and control to reduce the risk of loss or theft
Exceptional Flow	Reduce the amount of inventory that's not selling well to increase profitability
Pre-condition	Shows how many days of inventory are left for a product, so sellers can restock before running out
Post-condition	Tracks how well popular products are kept in stock
Assumptions	Uses historical data and tools like the Inventory Performance Index (IPI) to predict future demand
Constraints	Choose the right inventory management system
Dependencies	Maintain four weeks of inventory cover
Inputs and Outputs	**Inputs**: Using a shipping and logistics program **Outputs**: to automate inventory management aspects such as demand planning, maintaining healthy inventory levels, and avoiding aging inventory or spoilage
Business Rules	they gain access to machine learning-based inventory management system.
Miscellaneous Information	Third-party providers can also manage inventory at scale, helping you save on costs such as order fulfilment while offering fast shipping

Q12 Activity Diagram

Definition of an Activity Diagram
An **Activity Diagram** is a type of behavioural diagram in **Unified Modelling Language (UML)** that visually represents workflows, processes, and system activities. It illustrates the flow of control or data from one activity to another and is commonly used to model business processes and system behaviours.

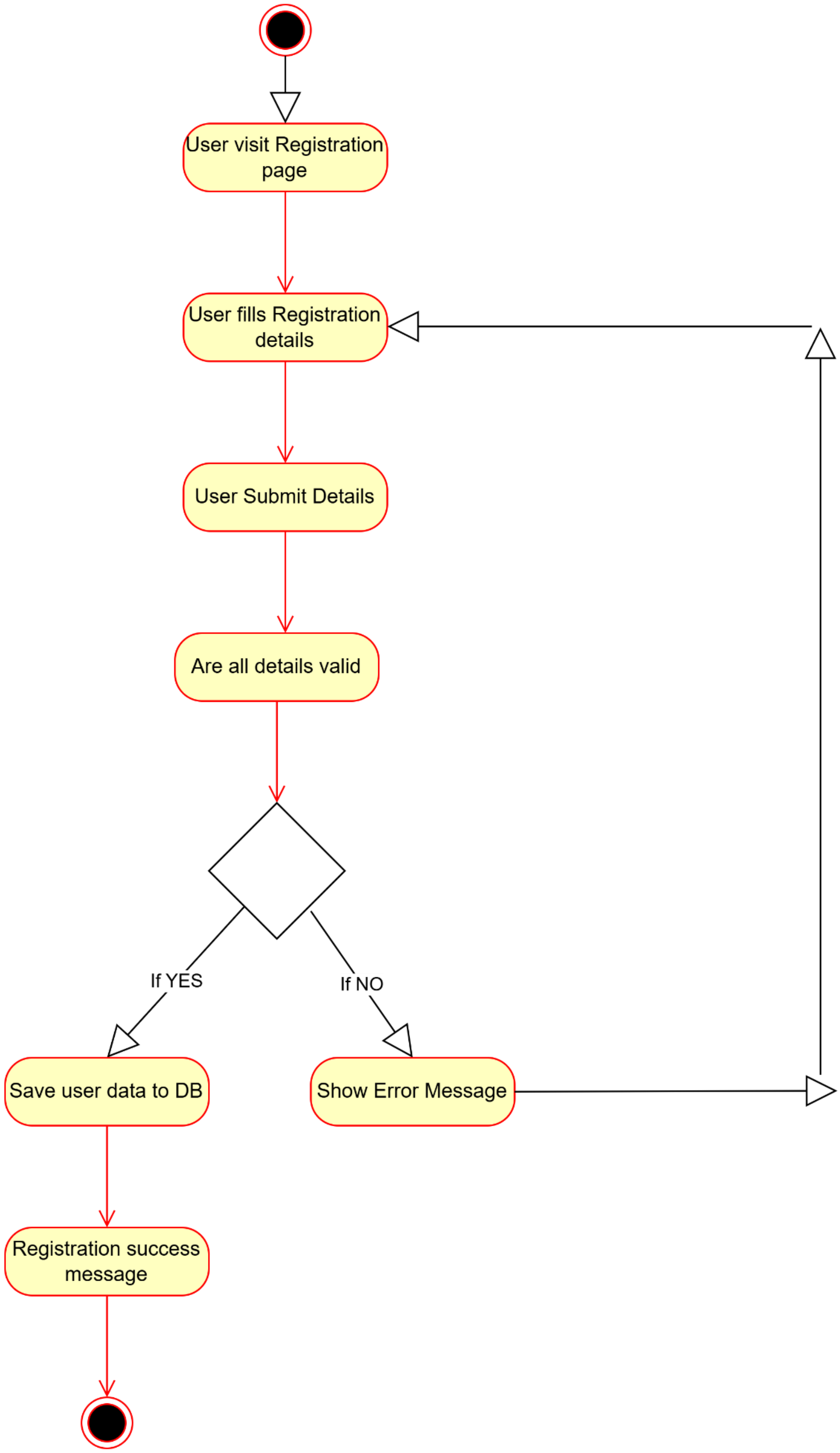
Benefits of Activity Diagrams

1. **Visual Representation** – Provides a clear and structured way to model business and system workflows.
2. **Better Understanding** – Helps stakeholders, developers, and analysts understand process flows.
3. **Identifies Bottlenecks** – Helps in analysing and optimizing workflows by identifying inefficiencies.
4. **Facilitates Communication** – Bridges the gap between technical and non-technical team members.
5. **Enhances Documentation** – Serves as a reference for system development and maintenance.
6. **Supports Parallel Processing** – Shows concurrent workflows, making it useful for modelling real-world scenarios.
7. **Simplifies Complex Processes** – Breaks down intricate workflows into manageable steps

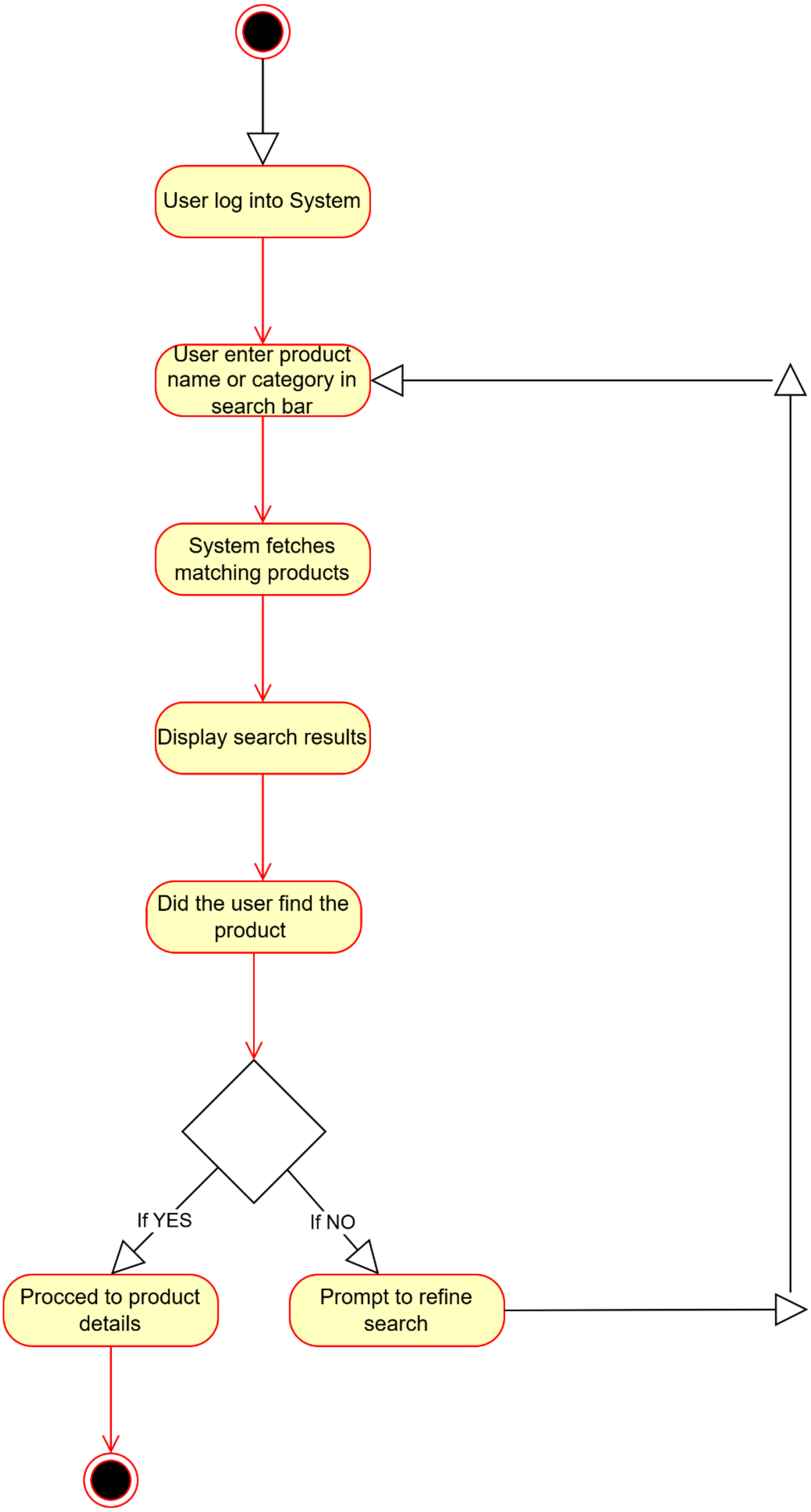
Key Symbols in the Activity Diagram:

- **Initial Node (Black Circle)**: Marks the start of the process.
- **Action/Activity (Rounded Rectangle)**: Represents a step or activity in the process.
- **Decision Node (Diamond)**: A decision point where the flow splits based on conditions.
- **Merge Node (Diamond)**: Where multiple flows join back together.
- **Fork Node (Thick Horizontal or Vertical Line)**: Splits the flow into parallel paths.
- **Join Node (Thick Horizontal or Vertical Line)**: Combines parallel flows into one path.
- **Final Node (Circle with Border)**: Marks the end of the process.
- **Swim lanes**: Horizontal or vertical divisions used to show which actor or system is responsible for which activities
-

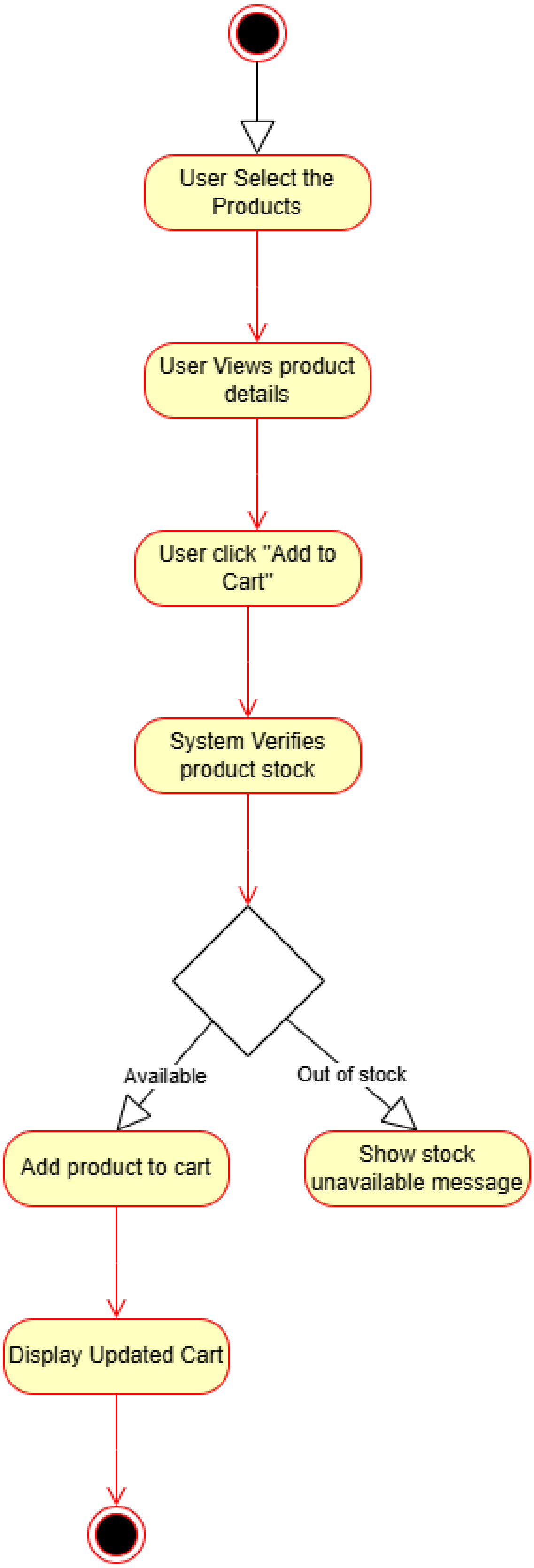
1 User Registration



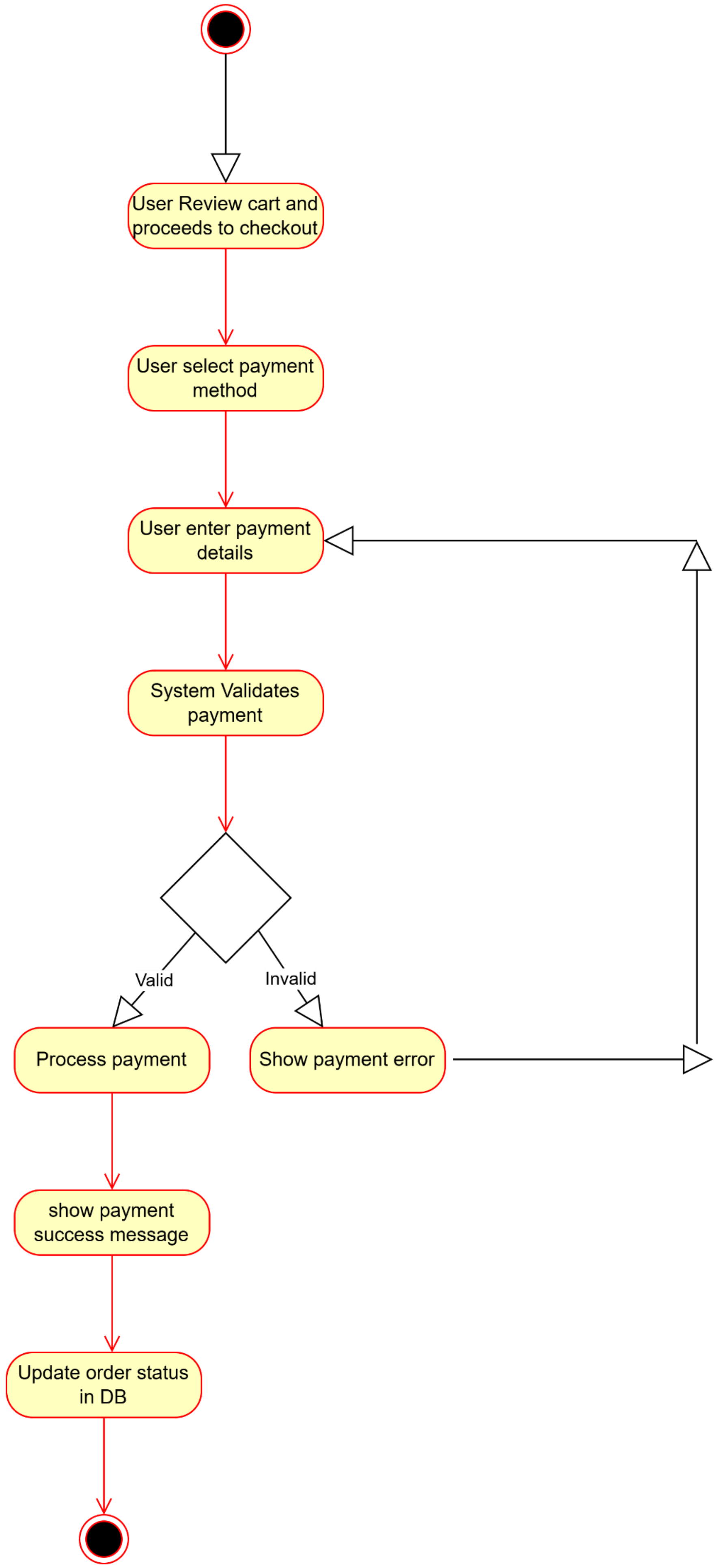
2 Search Product


User log into System
User enter product name or category in search bar
System fetches matching products
Display search results
Did the user find the product
If YES
If NO
Procced to product details
Prompt to refine search

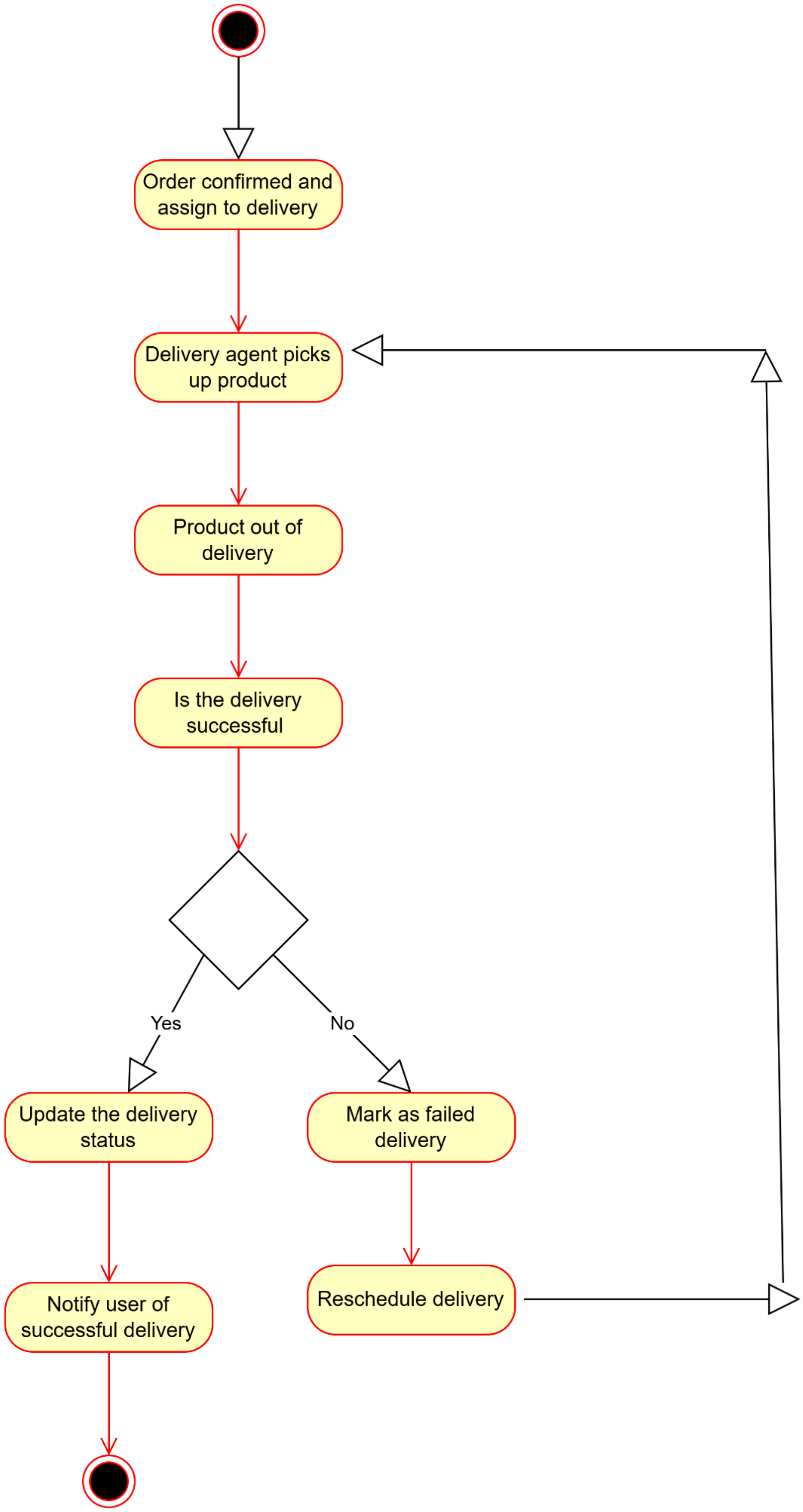
3 Add Product to Cart



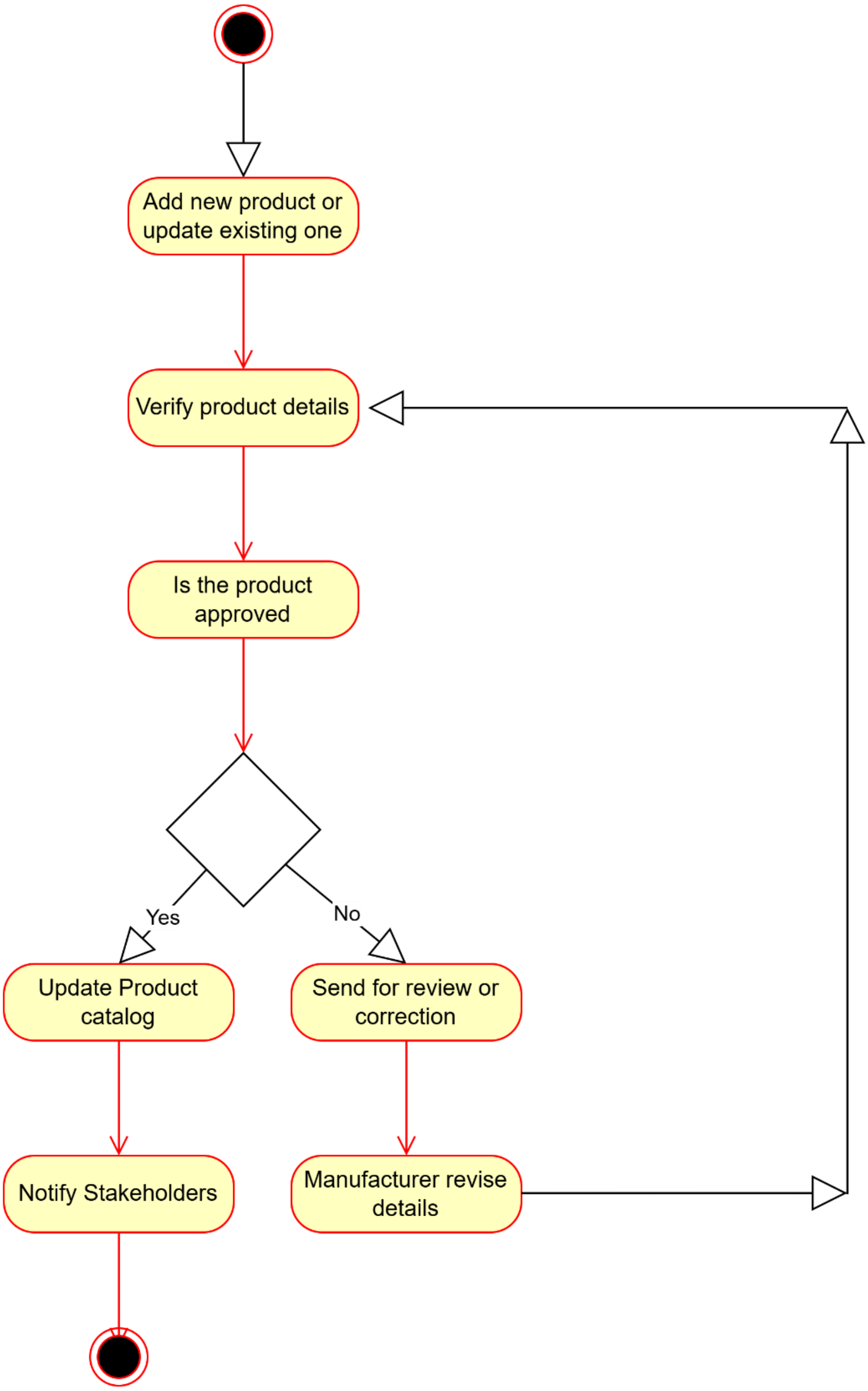
4 Making Payment


User Review cart and proceeds to checkout
User select payment method
User enter payment details
System Validates payment
Valid
Invalid
Process payment
Show payment error
show payment success message
Update order status in DB

5 Delivery


Order confirmed and assign to delivery
Delivery agent picks up product
Product out of delivery
Is the delivery successful
Yes
No
Update the delivery status
Mark as failed delivery
Notify user of successful delivery
Reschedule delivery

6 Manage Products



Add new product or update existing one
Verify product details
Is the product approved
Yes
No
Update Product catalog
Send for review or correction
Notify Stakeholders
Manufacturer revise details