# Question 1 – Audits - 5 Marks

Audit is a check on quality of the work, Here we check the process. We are making sure the project going on smooth manners.

we use data from the Gantt chart to make the audits.

Q1

Stage: Requirement gathering Phase 15 week (Week 1 to week 15)

Completed: 10 weeks (Week 1 to Week 10)

Checklist:

BRD template

Elicitation reports

Duplicate requirement report

Grouping of functionality/feature = client signed off

Email communication - To CC BCC

Q2

Stage: Requirement analysis Phase 13 week (16 Weeks to week 29)

Completed: 7 weeks (Week 16 to week 23)

Checklist:

UML diagram

Business to functional requirement mapping

Client signed off-document

RTM document version control

Email communication - To, CC, BCC

Q3

Stage : Design phase 10 weeks (week 30 to week 40)

Completed: 7 weeks (Week 30 to week 37)

Checklist:

Utilization of tools

Documented evidence on client communication

Stakeholder MOM

Email communications To, CC, BCC

Q4

Stage : Development + test phase 50 Week (Week 40 to week 78)

Completed:

Development 20 week (week 40 to 60)

Testing 20 week (Week 58 to week 78)

Checklist:

JAD session report

End user manual preparation document

BA and developer MOM

email communication To, CC, BCC

# Question 2 – BA Approach Strategy - 6 Marks

The Business Analyst (BA) Approach Strategy outlines the plan and methods a Business Analyst will use to conduct their activities effectively throughout a project. It defines how the BA will elicit, analyze, document, validate, and manage requirements, ensuring alignment with stakeholder needs and project objectives. A well-defined BA approach strategy helps ensure clarity, consistency, and successful delivery of requirements.

## What Elicitation Techniques to apply

Elicitation techniques are methods used to gather, discover, and extract information from stakeholders or other sources to understand business needs, requirements, and constraints. These techniques help ensure that business analysts and project teams have a clear and accurate understanding of what stakeholders expect from a project or product.

### Interviews

- For an online agriculture store we can conduct One-on-one or group conversations with stakeholders to gather detailed information. this elicitation technique is useful for understanding individual perspectives or domain expertise.

### workshops

- For an online agriculture store we can conduct collaborative sessions with multiple stakeholders to gather requirements or solve problems collectively.

which will encourage brainstorming and general agreement building.

### surveys

- For an online agriculture store we can conduct a survey where structured sets of questions distributed to stakeholders to collect responses. which is effective for gathering input from a large group of farmer who is facing the issue.

### observation

- For an online agriculture store we can watch stakeholders perform their daily tasks to understand current workflows and challenges.

### document analysis

- by using above elicitation technique, we can document analysis of agricultural regulations.

## How to do Stakeholder Analysis RACI/ILS

Stakeholder Analysis is important for identifying the roles, responsibilities, and influence of stakeholders in a project.

The RACI matrix clarifies the roles and responsibilities of stakeholders by categorizing them into four types:

R (Responsible): The individual(s) who perform the task or work.

A (Accountable): The individual ultimately answerable for the task's completion.

C (Consulted): The individual(s) who provide input or advice.

I (Informed): The individual(s) kept up-to-date on progress or decisions.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Mr. Karthik (Delivery Head)** | **Ms. Juhi (Senior Java Developer)** | **Mr. Henry (Sponsor)** | **Farmers (End-Users)** |
| **Define project scope** | A | C | R | I |
| **Gather requirements** | C | R | C | C |
| **Develop platform features** | I | R | C | I |
| **Perform User Acceptance Testing (UAT)** | I | C | I | R |
| **Manage financial budgeting** | C | I | R | I |
| **Network infrastructure setup** | C | I | I | I |
| **Database design and optimization** | I | R | I | I |
| **Deploy platform** | I | R | C | I |

## What Documents to Write

As a Business Analyst (BA) on the Online Agriculture Products Store project, We need to create and manage various documents such as BRD, SRD, FRD, to ensure smooth communication, proper requirement gathering, and successful delivery.

**Business Requirements Document (BRD):**

BRD defines high-level business needs and objectives.

Example: Farmers need a user-friendly platform to purchase seeds, fertilizers, etc.

**Stakeholder Requirements Document (SRD):**

SRD captures specific stakeholder needs and expectations.

**Functional Requirements Specification (FRS):**

FRS details system functionalities, such as product browsing, cart management, and payment processing.

**Non-Functional Requirements Document (NFR):**

NFR specifies system performance, scalability, security, and usability needs.

### Use Cases and User Stories:

Use Cases describe how users interact with the system (e.g., “Farmer adds a product to the cart”).

User Stories: Agile-format requirements (e.g., “As a farmer, I want to filter products by type to find what I need quickly”).

**Requirements Traceability Matrix (RTM):**

Maps requirements to design, development, and testing to ensure coverage.

**Test Cases and Scenarios:**

Write test cases for UAT and system testing to validate functionalities against requirements.

**UAT Plan and Results:**

Plan for User Acceptance Testing and document findings.

## What process to follow to Sign off on the Documents

To sign off on documents for the Online Agriculture Products Store project, we have to prepare the draft (e.g., requirements or design) and share it with stakeholders, such as Mr. Henry and the farmers, via email, requesting feedback by a specific date. Organize a face-to-face or virtual meeting to review the document collaboratively, where our manin focus is to get key concerns are addressed, especially from end-users like farmers. Incorporate feedback into a final version and email it to stakeholders, requesting explicit approval by replying to confirm. Document verbal approvals from meetings and follow up with written confirmation. Archive the signed-off documents and notify the team to move forward with implementation.

## How to take Approvals from the Client

To take approvals for the Online Agriculture Products Store project, we need to ensure the document like BRD, functional requirements, or wireframes is complete and highlights sections requiring sign-off. Share the draft with the client, Mr. Henry (Sponsor), via email, requesting feedback by a specific deadline. Afterward organize a meeting (in-person or virtual) with Mr. Henry and key stakeholders like farmers to review the document collaboratively.

Address their concerns in real-time and document agreed changes. After incorporating feedback, email the updated version to Mr. Henry for formal approval. Once approval is received then we will save the email or meeting notes as formal documentation and notify the project team to proceed.

## What Communication Channels to establish and implement

Establishing and implementing communication channels is essential for the Online Agriculture Products Store project to ensure smooth coordination, effective stakeholder engagement, and the timely exchange of information.

Establishing these communication channels will facilitate smooth and effective collaboration between all parties involved in the Online Agriculture Products Store project. The channels will cater to the needs of stakeholders, project team members, and farmers, ensuring that the project runs efficiently, stays on track, and delivers a product that meets the farmers’ needs.

**1. Internal Project Communication Channels**

Internal Project Communication Channels use to facilitate seamless collaboration among project team members, ensuring alignment and smooth project execution.

we can use tool like Slack/Microsoft Teams: For real-time communication, quick discussions, and file sharing.

Email for formal communications, sharing important documents, and reporting.

Project Management Tools (JIRA/Asana/Trello) for task tracking, milestone management, and issue resolution.

the need of this channel is to keep everyone informed of progress, tasks, and deadlines.

Fosters collaboration across different teams like Development, Testing, DB admin, devops to enables easy tracking of issues, tasks, and bottlenecks.

**2. Stakeholder Communication Channels**

Stakeholder Communication Channels are user to ensure regular updates and feedback loops with key stakeholders (e.g., Mr. Henry, Mr. Pandu, Mr. Dooku) and facilitate decision-making.

we can use any email service for formal communications such as financial reports, project status updates, and decision-related documents.

Also video Conferencing (Zoom, Microsoft Teams, Google Meet) for periodic meetings with stakeholders to discuss project progress, resolve issues, and align goals.

the need of this channel is to keeps stakeholders informed about the project’s status, challenges, and achievements. Also enables smooth decision-making and feedback exchange, this will builds transparency with the project sponsor and financial head.

## How to Handle Change Requests

Change request is a formal proposal to change a product, system, or process.

Agenda behind handling the change request is to ensuring that changes do not negatively impact the scope, timeline, or budget of the Online Agriculture Products Store project.

To manage change requests effectively while minimizing risks and disruptions to the Online Agriculture Products Store project we have to establish a Clear Change Control Process where stakeholders, including farmers, project team members, and other stakeholders, can submit change requests where as project managers, business analysts, technical leads, and key stakeholders, will evaluate and approve or reject changes. To ensure that all changes are tracked and evaluated based on their impact Document the Change Request. Evaluate the Change Request to assess the potential impact of the proposed change on the project’s objectives, timeline, and budget. To ensure that the requested change aligns with business goals and stakeholders expectations we have to consult Stakeholders and Get Feedback. BA has to keep explore alternative ways of implementing the change or alternatives that may have less impact on the project. If the change is approved, document the approval, update relevant project documents, and proceed with implementation. If rejected, provide feedback to the requester.

Once the CR is approved have to execute the change request and ensure that it is incorporated into the project effectively. we have to monitor and Track the Change to ensure the change is implemented successfully and doesn’t cause unforeseen issues.

While closing the CR we have to ensure the change is fully incorporated and any necessary documentation is updated once done we need to keep all stakeholders informed of the change and its impact on the project.

## How to update the progress of the project to the Stakeholders

1. Set a Communication Plan: Define the frequency (weekly/monthly), medium (email, meetings), and format (reports/dashboards).
2. Use Dashboards: Share real-time project status via tools like Power BI, JIRA, or Trello.
3. Share Reports: Provide concise progress reports with milestones, completed tasks, risks, and budget updates.
4. Conduct Meetings: Host regular video calls or presentations for detailed discussions and feedback.
5. Focus on Key Metrics: Highlight critical KPIs (e.g., timeline adherence, deliverables).

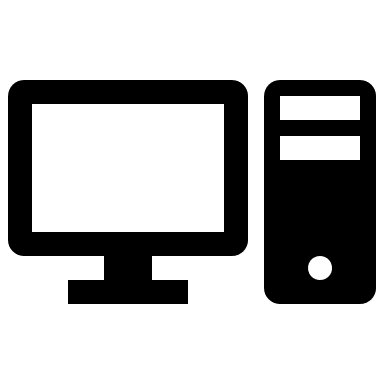
## How to take signoff on the UAT- Client Project Acceptance Form)

1. Prepare the Form: Include project details, deliverables, UAT results, and a signoff statement.
2. Share Results: Present UAT outcomes to the client, highlighting passed tests and addressing concerns.
3. Address Feedback: Resolve any remaining issues or note them for future phases.
4. Obtain Signoff: Secure client approval through a signed document (digital or physical).
5. Distribute Copies: Share the signed form with all stakeholders for record-keeping.

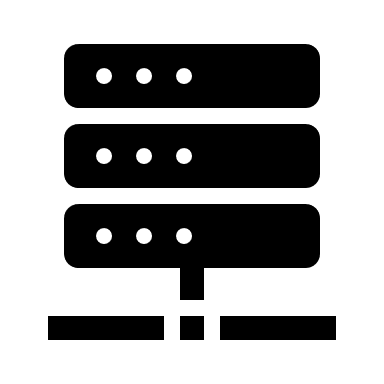
# Question 3 – 3-Tier Architecture - 5 Marks

An 3 tier architecture is where we have 3 layer at DB/backend Client layer, Busiiness layer, Database layer.

**Client Tier**

****

**Business Logic Tier**

****

**Database Tier**

For Online agriculture online store from client-side farmer want to purchase for specific brand of seed/fertilizers/pesticides that information will go to business logic layer. business logic layer will ask the same to manufacturer and is the same is available or not, depends on the availability and the manufacturer response business logic layer is respond on client the side and on the same hand the info will be store in database layer.

# Question 4 – BA Approach Strategy for Framing Questions – 10 Marks

## - The 5W1H

The 5W1H framework is a useful tool for gathering information and understanding a situation by answering questions about who, what, when, where, why, and how.

**Who**: Farmers request products; manufacturers confirm availability.

**What**: Request for specific branded agricultural products.

**When**: When the farmer places an order online.

Where: Online platform (client-side), manufacturer’s system, and database.

Why: To ensure product availability and record transactions.

How: The business logic layer communicates with the manufacturer, responds to the farmer, and updates the database.

## - SMART

The SMART technique can help in creating questions

SMART framework ensures that the project’s objectives are clear, realistic, and measurable, leading to better planning, execution, and tracking of progress. This makes the entire process more manageable, increases accountability, and get successful outcomes.

1. Specific

Why SMART: It ensures the objective is clear and focused. By specifying the goal (e.g., checking product availability, responding to farmers), everyone involved in the project understands exactly what needs to be achieved, reducing ambiguity.

2. Measurable

Why SMART: It allows tracking of progress. Measuring key metrics like response time, product availability confirmation, and successful transactions enables you to evaluate whether the system is performing as expected and identify areas for improvement.

3. Achievable

Why SMART: It ensures the goal is realistic and feasible. By making sure that the system's capabilities (e.g., business logic handling, manufacturer communication, database management) are in line with what can realistically be delivered, it prevents setting unattainable expectations.

4. Relevant

Why SMART: It ensures the goal is aligned with the broader objectives of the project, ensuring the system’s functionality supports farmers' needs and the business strategy. It helps prioritize what matters most and directs efforts toward enhancing user experience and platform efficiency.

5. Time-bound

Why SMART: It sets deadlines for completing the goal or task. By defining a clear timeframe for system responses and feedback, it ensures the project moves at an appropriate pace, and no task drags on longer than necessary, promoting efficient project delivery.

## - RACI

RACI charts help define and clarify roles and responsibilities within a team by outlining who is responsible, accountable, consulted, and informed for each task.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task/Role** | **Presentation Layer (Client-side)** | **Business Logic Layer** | **Data Layer (Database)** | **Project Manager** | **Stakeholders** |
| **Design User Interface** | Responsible |  |  | Accountable | Informed |
| **Process User Requests** | Informed | Responsible |  |  |  |
| **Validate Data** |  | Responsible |  |  |  |
| **Query Database for Data** | Informed | Responsible | Accountable |  |  |
| **Store Data in Database** |  | Responsible | Accountable |  |  |
| **Provide Response to User** | Responsible |  |  | Accountable | Informed |
| **Monitor System Performance** | Informed | Accountable |  |  | Informed |

By using the RACI model, you ensure that everyone involved in the process understands their roles and responsibilities, reducing confusion and improving project coordination.

## Use Cases

A use case is a detailed description of how a system interacts with external entities (such as users, other systems, or devices) to achieve a specific goal or functionality. It outlines the steps involved in a process, including the interactions between the actor (the entity initiating the action) and the system.

### 1. Use Case: Farmer Places a Product Request

Actor: Farmer

Description: A farmer logs into the online platform and selects a specific brand of seeds, fertilizers, or pesticides to purchase.

Preconditions: Farmer has registered and logged into the system.

Main Flow:

Farmer browses the product catalog.

Selects a product (brand of seed/fertilizer/pesticide).

Requests product availability.

System forwards the request to the business logic layer.

Postconditions: The product request is sent to the business logic layer for availability verification.

### 2. Use Case: Business Logic Layer Checks Product Availability

Actor: Business Logic Layer

Description: After receiving the request from the farmer, the business logic layer queries the manufacturer for product availability.

Preconditions: Farmer has requested a product.

Main Flow:

Business logic layer receives the farmer’s request.

Queries the manufacturer’s system to check product availability.

Receives the availability status from the manufacturer.

Postconditions: The business logic layer has the product availability information.

### 3. Use Case: Manufacturer Confirms Product Availability

Actor: Manufacturer

Description: The manufacturer verifies whether the requested product is available and sends the response back to the business logic layer.

Preconditions: The business logic layer has queried for product availability.

Main Flow:

Manufacturer receives the availability query.

Checks the stock for the requested product.

Sends the availability status (available or out of stock) back to the business logic layer.

Postconditions: Availability status is returned to the business logic layer.

### 4. Use Case: System Provides Feedback to Farmer

Actor: Business Logic Layer

Description: After receiving the availability information, the system informs the farmer of the product’s availability.

Preconditions: Business logic layer has received the availability response from the manufacturer.

Main Flow:

Business logic layer processes the manufacturer's response.

If available, the system confirms availability to the farmer.

If unavailable, the system informs the farmer that the product is out of stock.

Postconditions: Farmer is informed about the product's availability.

### 5. Use Case: Store Product Request in Database

Actor: Database Layer

Description: The system stores the product request and availability information for future reference and record-keeping.

Preconditions: Product request and availability status are available.

Main Flow:

The business logic layer sends the request details to the database layer.

The database layer stores the farmer's request, product details, and availability status.

Postconditions: The product request is stored in the database.

# Use case Specs

A Use Case Specification is a detailed document that describes the functionality and behavior of a system in the form of use cases. It serves as a detailed guide to how a specific use case operates within the system, helping stakeholders, designers, and developers understand the system’s requirements and expectations.

A Use Case Specification is a formalized version of a use case that serves to describe the system’s behavior and interactions in detail, ensuring that all necessary steps and conditions are addressed.

**Use Case ID**: UC-01

**Use Case Name**: Place Order

**Actor**: Customer

**Goal**: The customer places an order for a product and completes the payment.

**Preconditions**:

Customer is logged in.

The product is in stock.

**Main Flow**:

Customer browses the product catalog.

Customer adds products to the cart.

Customer enters payment and shipping details.

System processes the payment and confirms the order.

Customer receives an order confirmation.

**Alternative Flow**:

If payment fails, the system asks the customer to retry or choose a different payment method.

**Postconditions**:

Order is confirmed, and payment is processed.

**Exceptions**:

If the product is out of stock, the system notifies the customer.

Frequency of Use: Frequently

**Assumptions**:

Customer has a valid credit card.

**Special Requirements**:

Payment should be processed within 30 seconds.

# Activity Diagrams, Models

An Activity Diagram is a type of Unified Modeling Language (UML) diagram that visually represents workflows or processes in a system. It is used to reproduce the flow of control or data and the actions that occur within a system or between system components. Activity diagrams are helpful in understanding the sequence of activities and the decisions that influence the flow of the process.

**Activity Diagram for Placing an Order (Online Agriculture Products Store)**

**Initial Node**: The process begins when the customer accesses the online store.

**Action 1**: Customer browses products.

**Action 2**: Customer adds product to cart.

**Decision Node**: Is the product available?

**Yes**: Continue to payment.

**No**: Notify customer that the product is out of stock.

**Action 3**: Customer enters payment details.

**Action 4**: System processes payment.

**Decision Node**: Was the payment successful?

**Yes**: Confirm order and notify the customer.

**No**: Ask customer to retry payment or choose an alternative payment method.

**Final Node**: Order placed and process ends.

# Page designs

Page designs or website designs, are crucial for creating an effective and visually appealing online presence that can attract and retain visitors.

When considering page designs for the Online Agriculture Products Store project, it is essential to take into account the needs and expectations of a diverse group of users, ranging from farmers in remote areas who may have limited access to technology, to manufacturers and project stakeholders who will rely on the platform's functionality for business transactions. Since the project involves a variety of stakeholders, including Mr. Henry, the main sponsor, and Mr. Pandu, the Financial Head, it is crucial that the page designs not only offer ease of use but also reflect the project's overarching goals; they must be intuitive enough for users with varying levels of digital literacy, while also accommodating features such as product search, order placement, payment integration, and delivery tracking.

# Question 5 – Elicitation Techniques - 6 Marks

Elicitation techniques are a set of methods for gathering information from people

**Brainstorming**: A collaborative technique used to generate a wide range of ideas or solutions to a problem. Participants freely suggest ideas without judgment, encouraging creativity and innovation. It's often used at the early stages of requirements gathering or problem-solving.

**Document Analysis**: This method involves reviewing existing documentation such as reports, business plans, or manuals to understand current processes, business rules, and requirements. It’s helpful for uncovering information that may not be readily accessible through other techniques.

**Reverse Engineering**: The process of analyzing an existing product, system, or process to understand its components and functionality. This can be useful when documenting legacy systems or understanding how something works before improving or modifying it.

**Focus Groups**: A facilitated discussion with a group of selected individuals (stakeholders, users, etc.) to gather insights, opinions, and ideas. It is used to understand needs, preferences, and perceptions in a more interactive setting.

**Observation**: Involves watching users or stakeholders in their natural environment to understand their tasks, challenges, and workflows. This method helps uncover implicit needs that may not be expressed verbally, providing valuable insights into user behavior.

**Workshops**: A collaborative approach where stakeholders come together to work on specific issues or requirements. Workshops are focused, structured sessions that encourage active participation and idea generation, fostering consensus on key decisions.

**Joint Application Development (JAD):** A facilitated meeting approach where key stakeholders and the project team work together to define and prioritize system requirements. JAD sessions are highly structured and aim to accelerate the decision-making process, promoting collaboration between business and IT teams.

**Interview:** One-on-one conversations with stakeholders, users, or subject matter experts to gather in-depth information about their needs, goals, and challenges. Interviews can be structured, semi-structured, or unstructured depending on the level of detail required.

**Prototype**: A preliminary version of a system or application created to visualize and test design concepts. Prototypes help gather feedback from users early in the development process, ensuring that the final product aligns with their expectations.

**Questionnaire**: A set of written questions used to gather data from a large group of people. Questionnaires are an efficient way to collect standardized information, often used for gathering quantitative data or feedback on specific issues.

**Use Case Specs**: Detailed descriptions of how users will interact with a system, often including scenarios, steps, and expected outcomes. Use case specifications help define functional requirements and ensure that the system meets the needs of its users by outlining specific actions and behaviors.

# Question 6 – This project Elicitation Techniques - 5 Marks

For the Online Agriculture Products Store project, using Interview, Focus Group, and JAD (Joint Application Development) techniques can be highly beneficial for gathering detailed insights and ensuring that all stakeholder needs are properly addressed.

**Interview**: This method is ideal for understanding the individual needs and concerns of stakeholders, such as farmers, the project manager, or business sponsors. By conducting one-on-one interviews with key stakeholders like Mr. Henry (the main sponsor) and Mr. Pandu (the financial head), you can dive deep into specific requirements, uncover expectations, and address any concerns or ideas that may not surface in a group setting. This personalized approach is also valuable for gathering information from stakeholders with different levels of knowledge or involvement in the project.

**Focus Group**: Organizing focus groups with end-users like farmers (Peter, Kevin, and Ben) will allow you to collect feedback on their pain points, preferences, and expectations for the online platform. Focus groups facilitate dynamic discussions, enabling you to understand user concerns about product accessibility, ease of use, and the online buying process. This method fosters a collaborative environment where users can share experiences and propose ideas that you might not have anticipated. It’s also useful for generating ideas on how to simplify navigation and enhance the user interface to ensure adoption by farmers with varying levels of digital literacy.

**JAD (Joint Application Development):** JAD sessions are perfect for ensuring alignment between the technical team (like Ms. Juhi, the Senior Java Developer, and Mr. Vandanam, the Project Manager) and business stakeholders. These facilitated sessions enable both sides to collaborate, define, and prioritize system requirements quickly and effectively. In this project, JAD can be used to bring together key project participants—developers, project coordinators, and business analysts—who will collaboratively define the system’s features, ensuring that technical and business needs are balanced and that the project stays on track in terms of scope, budget, and timeline.

# Question 7 – 10 Business Requirements- 10 Marks

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Requirement (BR)** | **Stakeholder Requirement** |
| **BR001** | Farmers should be able to search for available products in fertilizers, seeds, and pesticides. | Kevin: Farmers need an easy-to-use search option to find products in each category. |
| **BR002** | Manufacturers should be able to upload and display their products in the application. | Mr. Henry: Manufacturers should have the ability to upload product details (name, description, image, price, etc.). |
| **BR003** | Farmers should be able to create an account and log in using their email ID and password. | Peter: A secure login system is required for farmers to buy products or add them to a buy-later list. |
| **BR004** | The system should support multiple payment options: COD, Credit/Debit card, and UPI. | Ben: Farmers should have a variety of payment methods for a smooth transaction process. |
| **BR005** | Farmers should receive an email confirmation regarding their order status. | Kevin: After an order is placed, an email confirmation should be sent to the farmer with order details. |
| **BR006** | A delivery tracker should be provided to track the whereabouts of an order. | Kevin: A real-time delivery tracker is needed so farmers can check the status of their order. |
| **BR007** | The platform should allow users to manage their account information (update email, password). | Peter: Farmers should be able to update their personal information and account settings. |
| **BR008** | The platform should display detailed product information such as description, price, and quantity. | Mr. Henry: Each product should have clear, concise details (e.g., product name, description, price, availability). |
| **BR009** | The platform should provide a "Buy Later" list for farmers to save products for future purchase. | Peter: Farmers should be able to add products to a "buy later" list to revisit and purchase them later. |
| **BR010** | The platform should be mobile-responsive and user-friendly for farmers with varying tech skills. | Kevin: The platform should be simple to navigate, with a design that works well on mobile devices for remote farmers. |

# Question 8 –Assumptions- 5 Marks

Assumption 1: A user can login using Facebook or google account.

Assumption 2: A user has knowledge on Farming

Assumption 3: Online shopping trend is increasing. The customers prefer online shopping

Assumption 4: The manufacturers have product to display in the application. The need for the application is ever growing.

Assumption 5: The customers have online accounts for secured payment processing.

# Question 9 – This project Requirements Priority - 8 Mark

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Business Requirement (BR)** | **Stakeholder Requirement** | **Priority** |
| **BR001** | Farmers should be able to search for available products in fertilizers, seeds, and pesticides. | Kevin: Farmers need an easy-to-use search option to find products in each category. | 10 |
| **BR002** | Manufacturers should be able to upload and display their products in the application. | Mr. Henry: Manufacturers should have the ability to upload product details (name, description, image, price, etc.). | 9 |
| **BR003** | Farmers should be able to create an account and log in using their email ID and password. | Peter: A secure login system is required for farmers to buy products or add them to a buy-later list. | 8 |
| **BR004** | The system should support multiple payment options: COD, Credit/Debit card, and UPI. | Ben: Farmers should have a variety of payment methods for a smooth transaction process. | 9 |
| **BR005** | Farmers should receive an email confirmation regarding their order status. | Kevin: After an order is placed, an email confirmation should be sent to the farmer with order details. | 7 |
| **BR006** | A delivery tracker should be provided to track the whereabouts of an order. | Kevin: A real-time delivery tracker is needed so farmers can check the status of their order. | 8 |
| **BR007** | The platform should allow users to manage their account information (update email, password). | Peter: Farmers should be able to update their personal information and account settings. | 6 |
| **BR008** | The platform should display detailed product information such as description, price, and quantity. | Mr. Henry: Each product should have clear, concise details (e.g., product name, description, price, availability). | 9 |
| **BR009** | The platform should provide a "Buy Later" list for farmers to save products for future purchase. | Peter: Farmers should be able to add products to a "buy later" list to revisit and purchase them later. | 6 |
| **BR010** | The platform should be mobile-responsive and user-friendly for farmers with varying tech skills. | Kevin: The platform should be simple to navigate, with a design that works well on mobile devices for remote farmers. | 8 |

# Question 10 – Use Case Diagram - 10 Marks

A Use Case Diagram visually represents the system's functionality and interactions between actors (users or external systems) and the system itself.

**Actors**:

Farmer

Admin

Manufacturer

Payment Gateway (e.g., PayPal or credit card service)

**Use Cases**:

View Products

Make Purchase (includes Checkout and Payment)

Track Orders

Register as a User (for Farmers or Manufacturers)

Manage Inventory (specific to Manufacturer)

Generate Reports (specific to Admin)

**System Boundary:**

"Online Agriculture Products Store."

**Relationships**:

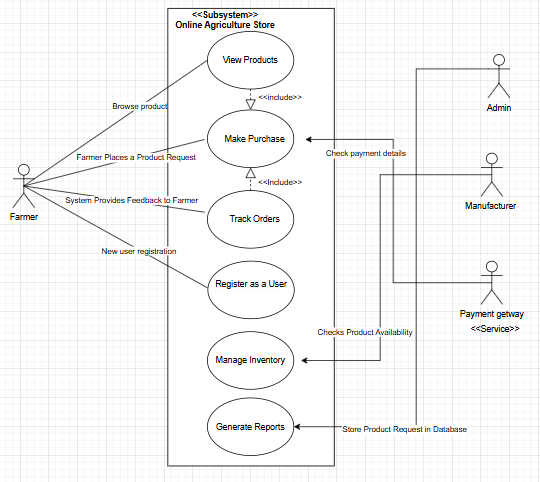
Farmers can interact with "View Products," "Make Purchase," and "Track Orders."

Admins interact with "Generate Reports."

Manufacturers interact with "Manage Inventory."

"Make Purchase" includes both "Checkout" and "Payment."

A Payment Gateway supports "Payment."



# Question 11 – (minimum 5) Use Case Specs - 15 Marks

Use Case Specifications are detailed descriptions of a use case, providing all the necessary information to understand its functionality, requirements, and interactions. Each use case specification includes structured sections to ensure clarity and completeness

### 1. Use Case: Farmer Places a Product Request

|  |  |
| --- | --- |
| **Use Case ID** | UC-02 |
| **Use Case Name** | Farmer Places a Product Request |
| **Created by** | Mr. ABC |
| **Date Created** | 24-Jan-25 |
| **Actor** | - **Primary Actor:** Farmer (Customer) |
| - **Secondary Actor:** Admin, Product Manufacturer |
| **Description** | This use case allows farmers to request specific agricultural products not currently available in the online store. The admin reviews the request and forwards it to manufacturers for potential fulfillment. |
| **Precondition** | - The farmer must have an active account and be logged into the system. |
| - The farmer must have details of the required product (e.g., name, type, and quantity). |
| **Postcondition** | - **Success:** The product request is recorded and forwarded to the admin and manufacturers. The farmer receives confirmation. |
| - **Failure:** The farmer is notified of missing or invalid details, or system errors prevent submission. |
| **Normal Flow of Events** | 1. The farmer logs into their account. |
| 2. The farmer navigates to the "Request Product" section. |
| 3. The system displays a form for product requests. |
| 4. The farmer fills in product details (name, type, quantity, optional notes). |
| 5. The farmer submits the form. |
| 6. The system validates the request (e.g., mandatory fields). |
| 7. The system records the request in the database. |
| 8. The system notifies the admin about the new request. |
| 9. The farmer receives a confirmation notification (email or SMS). |
| **Alternative Flow** | - **AF-01: Product Already Available** |
| 1. If a similar product is already in stock, the system suggests it to the farmer. |
| 2. The farmer can choose to purchase the existing product or proceed with the request. |
| **Exceptions** | - **EF-01: Missing or Invalid Details** |
| 1. The system highlights incomplete or incorrect fields and prompts for corrections. |
| - **EF-02: System Error During Submission** |
| 1. If a technical error occurs, the system displays an error message and advises the farmer to retry. |
| **Frequency of Use** | Low |

### 2. Use Case: Business Logic Layer Checks Product Availability

|  |  |
| --- | --- |
| **Use Case ID** | UC-02 |
| **Use Case Name** | Business Logic Layer Checks Product Availability |
| **Created by** | Mrs. Qwerty |
| **Date Created** | 24-Jan-25 |
| **Actor** | - **Primary Actor:** Business Logic Layer |
| - **Secondary Actor:** Inventory Database |
| **Description** | This use case ensures that the Business Logic Layer checks the inventory database for product availability when a farmer adds products to their cart or places a request. |
| **Precondition** | - The inventory database is connected and operational. |
| - The product details (ID, quantity) are provided by the farmer’s action (e.g., adding to cart or placing a request). |
| **Postcondition** | - **Success:** The system confirms product availability and allows the farmer to proceed. |
| - **Failure:** The system notifies the farmer about insufficient stock or unavailability. |
| **Normal Flow of Events** | 1. The farmer initiates an action requiring a product check (e.g., adds to cart, places request). |
| 2. The Business Logic Layer receives the product details (ID, quantity). |
| 3. The Business Logic Layer queries the inventory database for product availability. |
| 4. The inventory database returns the stock status (available or not available). |
| 5. The Business Logic Layer validates the quantity requested against the available stock. |
| 6. The system informs the farmer of the result: |
| - If available, the system allows the farmer to proceed. |
| - If not available, the system notifies the farmer. |
| **Alternative Flow** | - **AF-01: Partial Availability** |
| 1. If the requested quantity exceeds available stock, the system suggests the available quantity to the farmer. |
| 2. The farmer can adjust their request accordingly. |
| **Exceptions** | - **EF-01: Database Connectivity Issue** |
| 1. If the inventory database is unreachable, the system displays an error message and advises the farmer to retry later. |
| - **EF-02: Invalid Product ID** |
| 1. If the product ID is invalid or not found, the system returns an error and prompts the farmer to correct the input. |
| **Frequency of Use** | High |

### 3. Use Case: Manufacturer Confirms Product Availability

|  |  |
| --- | --- |
| **Use Case ID** | UC-04 |
| **Use Case Name** | Manufacturer Confirms Product Availability |
| **Created by** | Miss. PQRS |
| **Date Created** | 24-Jan-25 |
| **Actor** | - **Primary Actor:** Manufacturer |
| - **Secondary Actor:** Admin |
| **Description** | This use case allows the manufacturer to confirm the availability of requested products and update the system with the stock status. |
| **Precondition** | - The manufacturer must be notified of a product request by the admin. |
| - The manufacturer has access to their account on the platform to update the product status. |
| **Postcondition** | - **Success:** The manufacturer confirms product availability, and the system updates the stock status. |
| - **Failure:** The manufacturer fails to confirm availability or provide necessary details, and the admin is notified. |
| **Normal Flow of Events** | 1. The admin forwards the farmer's product request to the manufacturer. |
| 2. The manufacturer logs into their account. |
| 3. The manufacturer navigates to the "Pending Requests" section. |
| 4. The manufacturer reviews the request details (product type, quantity, and specifications). |
| 5. The manufacturer checks their inventory for availability. |
| 6. The manufacturer updates the system with the availability status: |
| - Marks the product as "Available" and provides the available quantity. |
| - If unavailable, marks the product as "Out of Stock." |
| 7. The system notifies the admin of the manufacturer's response. |
| 8. The farmer receives an update on the product request status. |
| **Alternative Flow** | - **AF-01: Partial Availability** |
| 1. If the requested quantity exceeds the available stock, the manufacturer updates the system with the available quantity and marks the rest as pending. |
| 2. The system notifies the admin and farmer of the partial availability. |
| **Exceptions** | - **EF-01: Manufacturer Fails to Respond** |
| 1. If the manufacturer does not confirm availability within a specified time, the system sends a reminder to the manufacturer and notifies the admin. |
| - **EF-02: Technical Error** |
| 1. If the manufacturer encounters a technical issue while updating the system, an error message is displayed, and they are advised to retry. |
| **Frequency of Use** | Medium |

### 4. Use Case: System Provides Feedback to Farmer

|  |  |
| --- | --- |
| **Use Case ID** | UC-05 |
| **Use Case Name** | System Provides Feedback to Farmer |
| **Created by** | Mr. Dummy |
| **Date Created** | 24-Jan-25 |
| **Actor** | - **Primary Actor:** Farmer |
| - **Secondary Actor:** System |
| **Description** | This use case allows the system to notify farmers of the status of their product requests or actions, including confirmation of availability, unavailability, or partial fulfillment. |
| **Precondition** | - The farmer has submitted a product request. |
| - The system has received input from the admin or manufacturer regarding the status of the product. |
| **Postcondition** | - **Success:** The system provides accurate feedback to the farmer based on the latest product status. |
| - **Failure:** The farmer is informed of any errors or delays in the process. |
| **Normal Flow of Events** | 1. The system receives updates on the product request status from the admin or manufacturer. |
| 2. The system determines the appropriate feedback based on the update (e.g., available, unavailable, or partially available). |
| 3. The system compiles the feedback, including relevant details: |
| - Product name |
| - Quantity available (if applicable) |
| - Expected delivery timelines (if applicable) |
| 4. The system sends a notification to the farmer via email, SMS, or platform notification. |
| 5. The system marks the feedback as sent in the database for record-keeping. |
| **Alternative Flow** | - **AF-01: Delayed Manufacturer Response** |
| 1. If the manufacturer has not responded within the expected timeframe, the system notifies the farmer of the delay and provides an estimated update time. |
| **Exceptions** | - **EF-01: Notification Failure** |
| 1. If the notification fails to send due to technical issues (e.g., network failure), the system retries and logs the error. |
| 2. If the error persists, the system notifies the admin to resolve the issue. |
| **Frequency of Use** | High |

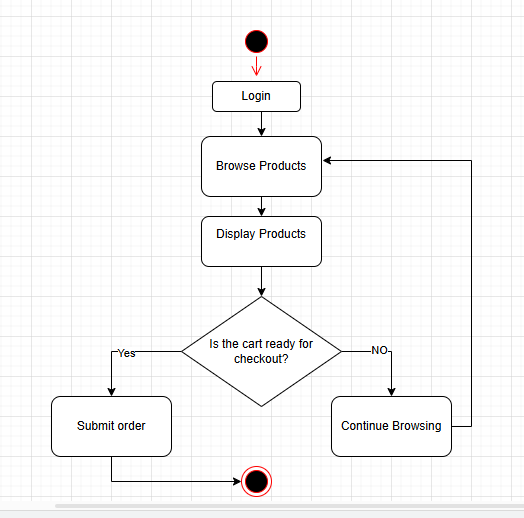
### 5. Use Case: Store Product Request in Database

|  |  |
| --- | --- |
| **Use Case ID** | UC-06 |
| **Use Case Name** | Store Product Request in Database |
| **Created by** | [Your Name] |
| **Date Created** | 24-Jan-25 |
| **Actor** | - **Primary Actor:** System (Database) |
| - **Secondary Actor:** Admin, Farmer |
| **Description** | This use case describes the process of storing a farmer's product request in the database upon submission. The system ensures that all necessary details are saved and can be retrieved for further processing. |
| **Precondition** | - The farmer has successfully submitted a product request. |
| - The system is connected to the database and operational. |
| **Postcondition** | - **Success:** The product request is successfully stored in the database with relevant details (product name, quantity, farmer ID, etc.). |
| - **Failure:** The system logs an error and notifies the admin if the request could not be stored. |
| **Normal Flow of Events** | 1. The farmer submits a product request through the platform. |
| 2. The system captures the request details (product name, type, quantity, etc.). |
| 3. The system validates the data (e.g., required fields, correct format). |
| 4. The system stores the request in the product request table of the database. |
| 5. The system confirms successful storage and proceeds with further processing (e.g., notifying the admin). |
| **Alternative Flow** | - **AF-01: Invalid Request Data** |
| 1. If the system detects invalid or incomplete data, it prompts the farmer to correct the request before submission. |
| **Exceptions** | - **EF-01: Database Connection Failure** |
| 1. If the database connection fails, the system displays an error message and asks the farmer to retry. |
| - **EF-02: Data Integrity Error** |
| 1. If an error occurs while saving the request (e.g., data conflict), the system logs the error and notifies the admin. |
| **Frequency of Use** | High |

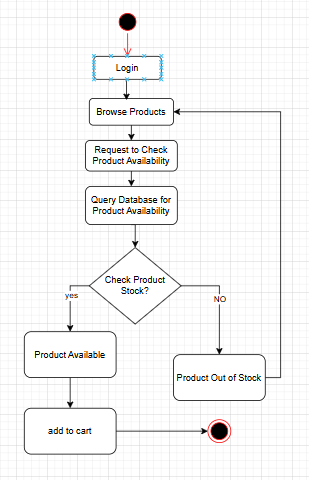
# Question 12 – (minimum 5) Activity Diagrams - 15 Marks

An Activity Diagram represents the workflow or business process of a system. Activity Diagram will illustrate the step-by-step process, from a farmer searching for products to completing the purchase and tracking the order.

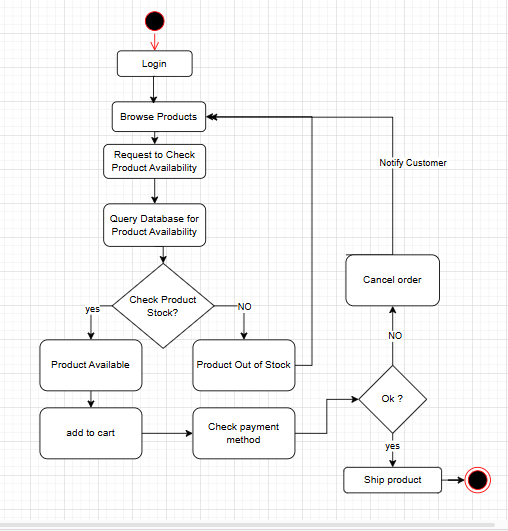
### Activity Diagram 1: For Farmer Places a Product Request



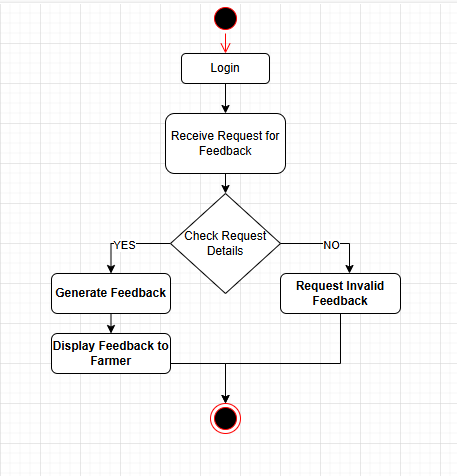
### Activity Diagram 2: Business Logic Layer Checks Product Availability



### Activity Diagram 3: Manufacturer Confirms Product Availability



### Activity Diagram 4: System Provides Feedback to Farmer



### Activity Diagram 5: Store Product Request in Database

