

CAPSTONE PART 1 PEP 2

1. 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?

Ans-

Audits in a Business Analyst (BA) role primarily focus on ensuring that the project's processes, deliverables, and documentation align with the agreed-upon standards, requirements, and timelines. For the Online Agriculture Product Store project, here's how quarterly audits (Q1, Q2, Q3, Q4) could be structured from a BA perspective:

Stage	Quarter 1 Audit Report (Requirement Gathering Phase)
Completed	weeks 18 (week 1 to week 18)
Check list	Documentation of Business Requirements (BRD)
	Elicitation results Report
	UML diagrams
	Grouping of functionalities/features-client sign off and mapping
	Client signoff-document
	RTM document version control
	Email Communication-To, CC, BCC

Stage	Quarter 2 Audit Report (Design Phase)
Completed	weeks 9 (week 28 to week 37)
Check list	Utilization of tools
	Functional design documents (FDD) created
	Documented evidence on client communication
	Stakeholder MOM
	Email Communication-To, CC, BCC

Stage	Quarter 3 Audit Report (Development Phase)
Completed	20 weeks (week 40 to 60)
Check list	JAD session report
	Code Review Documentation
	End user manual preparation document
	BA and developer MOM
	Approval from Stakeholders
	Email Communication-To, CC, BCC

Stage	Quarter 4 Audit Report (Testing Phase)
Completed	20 weeks (week 58 to 78)
Check list	Testing standards and coverage verification
	Test case summary
	Training report to end users
	Lessons learnt document
	Email Communication-To, CC, BCC

2. 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)

Ans-

A. What Elicitation Techniques to apply?

Ans-

Requirement Elicitation is a process of digging out the information from the stake holders. Requirement Elicitation serves the foundation in documenting the Requirements.

Brainstorming

Brainstorming is a group activity designed to generate a wide range of ideas and solutions through open discussions. This technique is particularly effective for generating a large number of ideas in a short time and is well-suited for collaborative projects like an Online Agriculture Product Store. During brainstorming sessions, stakeholders and team members, such as farmers' representatives and project committee members, share ideas openly without immediate judgment or criticism. This encourages creative thinking and uncovers potential opportunities or overlooked issues. The ideas generated are later analysed and refined for feasibility and alignment with project goals.

Advantages:

Encourages creativity and innovation.

Engages multiple stakeholders, enhancing collaboration.

Efficient in generating diverse ideas quickly.

Disadvantages:

Risk of dominant participants overshadowing others.

May result in too many unfeasible ideas requiring extensive filtering.

Document Analysis

Document analysis involves reviewing existing documents such as business plans, user manuals, policies, or project charters to identify relevant requirements. For this project, documents related to agricultural processes, farmer challenges, and manufacturer operations are valuable sources. This technique helps uncover implicit or unstated requirements while leveraging pre-existing information to establish a baseline for development.

Advantages:

Provides detailed and historical context.

Cost-effective as it utilizes existing information.

Minimizes reliance on stakeholder availability.

Disadvantages:

Information may be outdated or incomplete.

Requires expertise to interpret technical or complex documents correctly.

Reverse Engineering

Reverse engineering involves analysing an existing system or application to identify its components, design, and functionality to infer requirements. This is useful when a similar agricultural platform exists, providing insights into functional gaps or improvements. The team can evaluate how features align with the needs of remote farmers and adapt them accordingly.

Advantages:

Reveals functional and design specifics of existing systems.

Helps identify limitations or areas for improvement.

Useful when documentation is missing.

Disadvantages:

Time-intensive process.

May violate intellectual property if not handled ethically.

Focus Groups

Focus groups gather a selected group of stakeholders to discuss specific aspects of the project. For example, representatives of farmers, distributors, and agricultural experts can be assembled to explore challenges and expectations related to the platform. This interaction provides qualitative insights and reveals differing perspectives, leading to a comprehensive understanding of requirements.

Advantages:

Captures diverse opinions and feedback.

Promotes stakeholder engagement.

Quick turnaround for qualitative insights.

Disadvantages:

Risk of groupthink, where dominant opinions prevail.

Can be difficult to schedule with diverse participants.

Observation

Observation involves closely studying end-users or stakeholders as they perform their daily tasks to identify their needs and challenges. For this project, observing farmers' purchasing processes, including any manual or workaround methods, can highlight inefficiencies that the platform could address. This technique offers real-world insights into user behaviour and requirements.

Advantages:

Provides first-hand and authentic data.

Identifies issues that users may overlook or fail to articulate.

Enhances empathy for the end-user experience.

Disadvantages:

Time-consuming and resource-intensive.

Observer presence may influence user behaviour (Hawthorne effect).

Workshop

Workshops involve gathering stakeholders and team members in a structured environment to collaboratively define, analyse, and refine requirements. This technique fosters real-time communication and decision-making. For instance, a workshop could bring together

farmers, manufacturers, and developers to prioritize features for the web/mobile application.

Advantages:

Encourages collaboration and consensus.

Reduces communication gaps through direct interaction.

Efficient for addressing complex requirements.

Disadvantages:

Requires skilled facilitation to avoid conflicts.

Logistical challenges in coordinating all participants.

Joint Application Development (JAD)

JAD sessions are formal workshops specifically designed to gather requirements through collaboration among developers, stakeholders, and users. For the agricultural platform, a JAD session might include representatives from remote farmers and manufacturers, facilitated by the project team, to ensure all perspectives are incorporated.

Advantages:

Encourages structured collaboration.

Facilitates consensus and faster decision-making.

Reduces the risk of incomplete or ambiguous requirements.

Disadvantages:

High preparation and coordination effort.

Requires significant stakeholder availability.

Interview

Interviews involve one-on-one or group discussions with stakeholders to extract specific requirements and insights. For this project, interviewing farmers, project committee members, and manufacturers helps uncover detailed expectations and unique perspectives. Structured, semi-structured, or unstructured formats can be used depending on the goals.

Advantages:

Captures detailed and personal insights.

Flexible and adaptable to the interviewee's expertise.

Builds rapport and trust with stakeholders.

Disadvantages:

Time-consuming for both parties.

Risk of interviewer bias affecting responses.

Prototyping

Prototyping creates a preliminary version of the application or feature to visualize and validate requirements. For example, a prototype of the purchasing interface allows stakeholders to provide feedback on usability and functionality. Prototyping ensures alignment between expectations and deliverables early in the project.

Advantages:

Enhances stakeholder understanding through visualization.

Early detection of usability or design flaws.

Reduces the risk of costly changes later.

Disadvantages:

Can create unrealistic expectations if the prototype is perceived as final.
Resource-intensive for iterative refinements.

Questionnaire

Questionnaires are written sets of questions distributed to stakeholders to gather requirements efficiently. This technique is useful for reaching a wide audience, such as a survey among farmers about their preferences for purchasing fertilizers, seeds, and pesticides. Responses are analysed to identify common needs and trends.

Advantages:

Cost-effective for large groups.
Ensures anonymity, encouraging honest feedback.
Standardized format simplifies analysis.

Disadvantages:

Limited interaction for clarifying ambiguous responses.
Response rates and quality may vary.

Use Case Specifications

Use case specifications document the interactions between users and the system to achieve specific goals. For the agricultural platform, use cases could include a farmer purchasing a product, viewing order history, or receiving delivery updates. This technique ensures requirements are captured in a user-centric manner.

Advantages:

Clearly defines functional requirements.
Highlights user-system interactions and dependencies.
Useful for creating test cases and validating requirements.

Disadvantages:

Time-intensive to develop detailed use cases.
May require additional expertise for comprehensive documentation.
By applying these elicitation techniques strategically, the project team can gather robust and actionable requirements to create a user-centric and functional platform for remote farmers.

B. how to do Stakeholder Analysis RACI?

Ans-Stakeholder analysis with the RACI (Responsible, Accountable, Consulted, and Informed) model involves identifying all relevant stakeholders and mapping their roles to ensure clear responsibility and accountability throughout the project. This analysis follows these steps:

Identify Stakeholders: Start by listing all individuals, groups, or organizations affected by or influencing the project. For example, in a project involving an Online Agriculture Product Store, stakeholders might include farmers, manufacturers, the project team, and regulatory bodies.

Understand Their Roles: For each stakeholder, determine their role in the project. Some may be decision-makers (Accountable), while others may provide input (Consulted) or require updates (Informed).

Define RACI Components:

Responsible: Assign individuals or teams responsible for executing tasks. These are the doers who take ownership of the work.

Accountable: Designate one person accountable for the task's success, often ensuring decisions are made and objectives are met.

Consulted: Identify stakeholders whose input is crucial. These are the advisors or experts providing insights or feedback.

Informed: Specify who needs to be kept in the loop about progress or decisions but does not contribute directly.

Create a RACI Matrix: Develop a table where tasks or deliverables are listed in rows, and stakeholders are listed in columns. Assign R, A, C, or I to each cell to clarify their involvement in specific tasks.

Validate the RACI Matrix: Share the matrix with stakeholders to confirm their roles are accurately defined. This ensures alignment and prevents overlap or confusion in responsibilities.

Communicate and Use: Regularly refer to the RACI matrix during the project lifecycle to resolve role conflicts, ensure accountability, and maintain effective communication. For example, if a task is delayed, knowing who is "Responsible" and "Accountable" enables swift action.

By following this detailed approach, the RACI model ensures clarity, avoids duplication of effort, and fosters collaboration, ultimately contributing to the project's success.

C. What Documents to Write?

Ans-

a. BRD (Business Requirements Document)

The BRD defines the business needs and the solution that is being implemented to address those needs. It focuses on the business side and sets the context for the project. It includes high-level goals and objectives, and it serves as the foundation for other documents like FRD or Use Case documentation.

Key Components:

Introduction: Overview of the project, including objectives and stakeholders.

Business Objectives: Clearly defines the goals and objectives of the business.

Scope: High-level description of what is included and excluded from the project.

Stakeholders: List of individuals or groups impacted by the project.

Assumptions & Constraints: Key assumptions made and constraints on the project.

Business Requirements: The business needs that the project must fulfill.

Success Criteria: How success will be measured, including KPIs.

Timeline: High-level project timeline, milestones, and deadlines.

b. FRD (Functional Requirements Document)

The FRD describes in detail the functional aspects of the system and the features that need to be implemented. This document provides clear guidance to developers, testers, and other team members on what needs to be built.

Key Components:

Introduction: An overview of the system, its goals, and the objectives for its development.

Functional Requirements: Detailed description of all features and functionality required, often broken down into user stories or functional modules.

Use Cases: Specific scenarios explaining how the system will be used.

Non-Functional Requirements: Requirements that define system performance, security, scalability, etc.

Data Requirements: Detailed specifications on the data that will be used or generated, including data formats and database structures.

System Interface Requirements: Interaction with other systems or external interfaces.

Security Requirements: Authentication, authorization, encryption, and other security-related details.

C. Use Case Documentation

Use case documentation focuses on the interactions between the user and the system. Each use case represents a specific task or process that the system should support. It provides detailed scenarios for each business process, which guides both the development and testing phases.

Key Components:

Use Case Name: A brief, descriptive title of the use case.

Use Case ID: A unique identifier for the use case.

Actors: Users or external systems that interact with the system (e.g., Customer, Admin).

Preconditions: Conditions that must be met before the use case can begin.

Triggers: What starts the use case (e.g., User logs in).

Basic Flow: The main steps that represent the "happy path" where everything works as expected.

Alternative Flow: Steps for any deviations from the basic flow (e.g., errors, exceptions).

Postconditions: The state of the system after the use case is completed.

Business Rules: Any rules that govern the use case.

Exceptions: Scenarios where the use case doesn't proceed as expected (e.g., invalid input).

Assumptions: Any assumptions made while writing the use case.

4. Test Case Documentation

Test case documentation provides detailed instructions for testing each functionality to ensure the system meets the requirements. It is crucial for QA teams to understand how to execute tests and verify system behaviour.

Key Components:

Test Case ID: A unique identifier for each test case.

Test Case Name: A brief description of what is being tested.

Test Objective: The goal or purpose of the test case.

Preconditions: The initial conditions or setup required to execute the test.

Test Steps: A step-by-step description of the actions to be taken to execute the test.

Expected Results: The expected outcome for each step or for the entire test.

Actual Results: The actual outcome after executing the test.

Pass/Fail Criteria: Criteria to determine if the test case passes or fails.

Test Data: Any data needed to execute the test (e.g., user credentials, specific inputs).

Test Environment: Information about the environment in which the test is conducted (e.g., hardware, software version).

Priority: Importance of the test case (e.g., High, Medium, Low).

Tested By: The person responsible for executing the test.

Test Date: When the test was conducted.

Comments: Any additional remarks or observations.

D. What process to follow to sign off on the Documents?

Ans-Signing off the Software Requirements Specification (SRS) document is a critical step in ensuring all stakeholders agree on the project's requirements before development begins.

Direct Meetings:

Schedule an in-person meeting with stakeholders.

Present the SRS document, clarify sections, and address feedback.

Obtain physical or digital sign-off during the meeting.

Virtual Meetings:

Organize a virtual meeting

Share the document for review, walk the client through key areas, and resolve queries.

Request an email confirmation or digital sign-off after the session.

Email Confirmation:

Send the SRS document with a summary of critical points and a clear request for sign-off.

Specify a deadline for review and confirmation.

Follow up with a call or email if no response is received within the timeframe.

E. How to take approvals from the client?

Ans-

Clear Communication: Always ensure that the client understands the technical aspects. Use simple language or visual aids to explain complex concepts.

Documentation: Always document the approvals and decisions, whether via email, project management tools, or formal sign-offs.

Timely Updates: Keep the client informed about project progress, any potential delays, or risks to maintain trust and get the continuous feedback.

Change Control Process: Ensure that any changes requested after approvals follow a formal change control process, including assessing impact, adjusting timelines, and re-approving revised deliverables.

F. What Communication Channels to establish and implement?

Ans-

Regular meetings can be scheduled with team members to ensure consistent collaboration. These can include daily stand-ups or brief check-ins to address immediate concerns, share updates, and clarify any blockers. This fosters continuous engagement and allows quick problem resolution.

Weekly status meetings should be organized to discuss the overall project progress, review completed tasks, and set priorities for the upcoming week. This helps all team members stay aligned with project goals and timelines, fostering accountability and identifying risks early on.

Bi-weekly sprint reviews are essential for teams following agile methodologies. These meetings provide an opportunity to evaluate completed tasks within a sprint, demonstrate working features, and gather feedback from stakeholders or product owners. This helps in refining the product increment and aligning it with user expectations.

Monthly stakeholder updates are crucial for keeping external parties informed. These updates should summarize the project's progress, milestones achieved, challenges faced, and any course corrections required. Regular updates provide transparency and build trust with stakeholders, ensuring that the project remains on track and aligned with broader business objectives.

By integrating these communication channels, we can ensure that the project remains organized, on schedule, and responsive to any issues that arise.

G. How to handle change requests?

Ans-

a. Change Request Form

A Change Request Form captures the details of a proposed change. The key components of the form should include:

Change ID: Unique identifier for the change request.

Requestor: Name of the person requesting the change.

Date of Request: When the request was made.

Description of Change: A detailed explanation of the proposed change.

Reason for Change: Why the change is necessary (e.g., new requirements, project risk, technology update).

Impact on Scope, Schedule, and Budget: How the change might affect the project scope, timeline, or budget.

Priority: Urgency of the change (low, medium, high).

Requested Change Type: Categorize the type of change (e.g., scope change, schedule change).

Affected Stakeholders: People or departments impacted by the change.

Supporting Documentation: Attachments that provide further context, such as design documents or specifications.

b. Impact Analysis

Impact analysis evaluates how the change will affect the project in terms of scope, time, cost, quality, and risks. Key steps for conducting impact analysis:

Scope Impact: Determine if the change alters the original project deliverables. Consider whether new features are being added or existing features are being removed.

Schedule Impact: Analyse if the change affects the project timeline. Does it require additional time to complete tasks or create delays?

Cost Impact: Assess the financial implications, including additional resources, materials, or other costs required to implement the change.

Quality Impact: Evaluate if the change impacts the quality of the project deliverables. Could it lead to decreased quality or increase testing/quality assurance efforts?

Risk Impact: Identify new risks that might be introduced by the change or if the change mitigates any current risks.

Resource Impact: Consider the allocation of resources. Do you need more staff or different skill sets to implement the change?

Stakeholder Impact: Identify how stakeholders will be affected by the change and any potential communication needs.

c. Approval Process

The approval process is essential to ensure that only justified and necessary changes are implemented. A typical approval process includes:

Submit Change Request: The requestor submits the change request form for review.

Initial Review: A designated team (e.g., Project Manager, Change Control Board) reviews the request to ensure it is valid and aligns with project goals.

Impact Assessment: After reviewing the change request, an impact assessment is conducted, possibly involving various project teams (e.g., development, testing, operations).

Review and Evaluation: The Change Control Board (CCB) or project stakeholders evaluate the impact analysis and determine if the change should be accepted, rejected, or deferred.

Decision: Based on the evaluation, a decision is made to approve, reject, or request modifications to the change request. Approval may require sign-offs from the project sponsor or other key stakeholders.

Communicate Decision: Notify all stakeholders about the decision and any changes in project scope, timeline, or budget.

Update Documentation: Any approved changes should be reflected in project documentation (e.g., scope, schedule, budget).

d. Documentation

Proper documentation ensures that all changes are tracked and the project's baseline is updated. This includes:

Change Log: Maintain a log of all change requests, including the status (approved, rejected, pending) and any relevant details.

Updated Project Plan: If the change is approved, update the project scope, schedule, budget, and resources accordingly.

Revised Requirements Document: If the change affects the project's requirements, update the requirements document.

Updated Deliverables: Modify any impacted project deliverables based on the change request.

Version Control: Ensure that all changes are versioned properly and that previous versions are retained for reference.

Communication Plan: Update communication plans to reflect any changes in the project.

H. How to update the progress of the project to the stakeholders?

Ans-

a. For the weekly status report, include:

- A summary of the progress made in the last week with specific details about completed tasks, milestones, and any deviations from the initial plan.
- The current status of active tasks, including those that are on track, delayed, or at risk.
- Highlight any issues or challenges that have come up, along with actions being taken to address them.

- Provide updates on any changes in scope, schedule, or resources.
- Include a forecast of work for the upcoming week and any dependencies or support required from stakeholders.

b. For the monthly review meeting, focus on:

- A summary of the overall progress against the project timeline, emphasizing whether the project is on track or has encountered any significant delays.
- An update on key deliverables, milestones, and whether they were achieved within the agreed timeframes.
- A discussion of the project's health, such as budget status, risk management efforts, and any changes in requirements or scope.
- Insights into the performance of team members, any resource adjustments needed, and a review of dependencies or external factors impacting progress.
- A look ahead to the next month's objectives, identifying major upcoming tasks and any potential issues that need to be managed.

I. How to take signoff on the UAT-client project Acceptance Form?

Ans-

a. UAT Preparation

Objective: Ensure the system is ready for User Acceptance Testing (UAT).

Steps to follow:

Review requirements and test cases.

Prepare the environment, including data, test scripts, and required resources.

Ensure that the UAT team is ready and has the necessary understanding.

Signoff Action: Get confirmation from stakeholders (e.g., Business Analyst, UAT Manager, Project Manager) that the system is ready and UAT can proceed.

b. Conduct UAT

Objective: Test the system against user requirements to ensure it meets business needs.

Steps to follow:

Execute test cases based on the predefined scenarios.

Record issues and feedback.

Engage with users to resolve ambiguities and clarify doubts.

Signoff Action: Have users or business stakeholders sign off the UAT results, acknowledging that testing was conducted and feedback was addressed.

c. Fix Issues

Objective: Address the defects or issues found during UAT.

Steps to follow:

Identify, prioritize, and resolve issues raised during UAT.

Communicate fixes to the UAT team.

Verify fixes by re-testing and ensuring issues are resolved.

Signoff Action: Obtain confirmation from the client that all issues have been addressed and that the system is ready for final review.

d. Acceptance Form

Objective: Document the formal acceptance of the system by the client.

Steps to follow:

Prepare the UAT-Client Project Acceptance Form, summarizing the testing process, issues fixed, and the final status.

Include key points like:

UAT completion date

Issues fixed

Client's satisfaction level

Signoff Action: The client reviews and approves the document, confirming their acceptance of the system.

e. Final Review Meeting

Objective: Conduct a final meeting to discuss the UAT process, review the results, and ensure alignment on the acceptance.

Steps to follow:

Present the UAT results, any pending issues, and the fixes.

Review the Acceptance Form with the client.

Discuss any additional actions or follow-ups if required.

Signoff Action: Get verbal or written confirmation that the client is satisfied with the process.

f. Obtain Sign-off

Objective: Get formal approval that the system is accepted by the client and the project is considered complete.

Steps to follow:

Ensure the UAT-Client Project Acceptance Form is fully completed, with all sections filled out.

Request the client's sign-off on the form, acknowledging that they accept the system as delivered.

Signoff Action: Obtain signatures (either physical or digital) from authorized client representatives, indicating formal acceptance.

3.Explain and illustrate 3-tier architecture?

Ans-

A 3-tier architecture is a software architecture pattern that divides an application into three distinct layers or tiers, each with its own responsibility.

1.Application layer

Topmost layer of the architecture-also known as "Presentation Layer".

It handles Graphical user interface (GUI) components such as screens and pages, validations on pages, Organization specific business logic will be on the Application layer.

2.Business Logic Layer

Middle Layer of the architecture which acts as an intermediary between the presentation layer and the data storage layer.

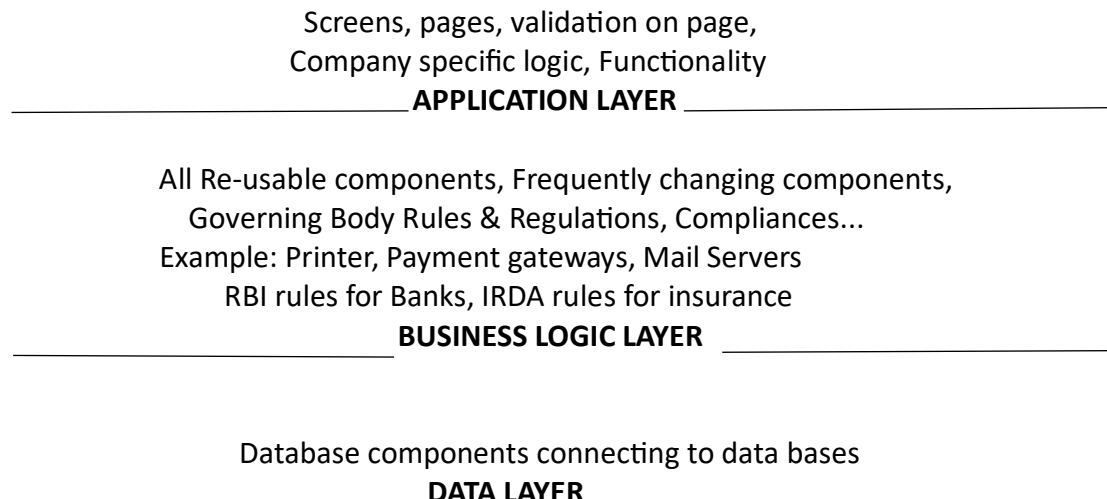
This layer contains the core logic of the application such as All reusable components (logic pertaining to industry), Frequently changing components, governing body rules and regulations, compliances should go to middle layer.

Ex: Printer, Payment Gateways, Mail server, RBI rules for banks, IRDA rules of insurance, etc

3.Database Layer

Bottom most layer of the architecture and which is responsible for storing and retrieving data.
Database components connecting to databases.
Ex: MYSQL, Oracle database.

ILLUSTRATE



In this illustration, the presentation layer interacts with the user and handles the user interface. It sends user requests and data to the business logic layer for processing. The business logic layer performs the required operations and communicates with the data storage layer to retrieve or store data. The data storage layer manages the persistence and retrieval of data from the underlying storage systems.

This three-tier architecture promotes separation of concerns, flexibility, and scalability. Each layer can be developed and maintained independently, enabling changes or updates in one layer without affecting the others. It also allows for better distribution of responsibilities and supports modular development, making the application more maintainable and extensible.

4. 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)

Ans-

A.5W 1H

If we need extract consistent requirements then probe in this direction like where, why, what, who, when, and how. 5W1H is considered as tool of a BA.

Who: Identify stakeholders, end-users, or systems involved.

Who will use this feature/system?

Who will maintain it post-implementation?

What: Understand the core requirement or problem.

What is the objective of this functionality?

What are the constraints or boundaries?

When: Define timelines and dependencies.

When should the feature be delivered?

When are critical milestones or deadlines?

Where: Determine the operational environment.

Where will the solution be used (location/platform)?

Why: Explore the rationale behind the requirement.

Why is this functionality important?

Why does this problem exist?

How: Discuss implementation, process, or flow.

How will the solution be utilized or integrated?

How will success be measured?

B. SMART

A well- formed requirement should comply with SMART.

Specific

The requirement must be clear, precise, and unambiguous. It should answer the "what, why, and how" questions.

Measurable

The requirement should have criteria to measure progress and success. This ensures that stakeholders can track whether the requirement is met.

Attainable

The requirement should be realistic and achievable within the given resources, budget, and timeframe.

Realistic

The requirement must align with the overall project goals and organizational objectives. It should have practical importance.

Time-bound (or Traceable)

The requirement should have a defined timeline for completion or milestones to ensure accountability. Alternatively, it should be traceable to its source or origin.

C. RACI

Leverage RACI to identify the right stakeholders to approach:

Responsible: Ask about the team or individual handling the requirement.

Accountable: Clarify who approves or owns the decision.

Consulted: Inquire with experts or advisors for guidance.

Informed: Determine who needs updates.

D. 3 Tier Architecture

A 3-tier architecture is a software architecture pattern that divides an application into three distinct layers or tiers, each with its own responsibility.

1.Application layer

Topmost layer of the architecture-also known as "Presentation Layer".

It handles Graphical user interface (GUI) components such as screens and pages, validations on pages, Organization specific business logic will be on the Application layer.

2. Business Logic Layer

Middle Layer of the architecture which acts as an intermediary between the presentation layer and the data storage layer.

This layer contains the core logic of the application such as All reusable components (logic pertaining to industry), Frequently changing components, governing body rules and regulations, compliances should go to middle layer.

Ex: Printer, Payment Gateways, Mail server, RBI rules for banks, IRDA rules of insurance, etc

3. Database Layer

Bottom most layer of the architecture and which is responsible for storing and retrieving data. Database components connecting to databases.

Ex: MYSQL, Oracle database.

E. Use Cases, Use case Specs, Activity Diagrams, Models, Page designs?

Use Cases represent the interactions between users (actors) and the system to achieve specific goals. For the online agriculture store, the actors would be farmers (primary users), admin staff, and manufacturers. Examples of use cases include:

- Browse Products: A farmer searches for fertilizers, seeds, or pesticides.
- Add Product to Cart: A farmer adds a product to their cart for purchase.
- Place Order: After reviewing the cart, the farmer places the order.
- Process Payment: The farmer proceeds with a secure payment.
- Track Order: The farmer tracks the status of their order.
- Manage Inventory: Manufacturers update their product stock details.
- Admin Manage Users: Admin staff manage user accounts, including registration, login, and profile updates.
- View Reports: Admin staff view sales, inventory, and customer reports.

Use Case Specifications provide detailed information on each use case, explaining the interaction flow. For example, the “Place Order” use case specification would describe the following steps:

- Pre-condition: The user must be logged in and have items in the cart.
- Main Flow: The farmer clicks “Place Order” after reviewing their cart; the system validates the items, calculates total cost, and initiates the payment process.
- Post-condition: The order is confirmed, and the farmer receives an order ID.
- Alternative Flow: If payment fails, the system prompts the user to reattempt or use an alternative payment method.
- Exceptions: The system handles cases where the cart is empty, or the payment gateway is unavailable.

Activity Diagrams represent the flow of activities and decisions within a use case or process. For instance, the “Place Order” activity diagram would show:

- Start: Farmer reviews the cart.
- Decision: Farmer decides whether to place the order.
- Action: If yes, the system verifies cart contents and calculates the total.
- Action: The farmer enters payment details.
- Decision: Payment successful?
- Yes: Proceed to order confirmation.
- No: Display error and request re-entry.
- End: Order is placed successfully or transaction fails.

Models such as sequence diagrams help design the structure of the system.

A sequence diagram for placing an order would illustrate the interaction between the Farmer, the System, and the Payment Gateway:

- The farmer initiates the process by selecting products.
- The system verifies the cart and computes the total cost.
- The farmer enters payment details.
- The system communicates with the payment gateway to process the payment.
- Based on the response from the payment gateway, the system either confirms the order or prompts for a retry.

Page Designs in UML can be represented through wireframes or mock ups, which outline the user interface of the application. These designs would include:

- Homepage: Featuring navigation to product categories (fertilizers, seeds, pesticides) and a search bar.
- Product Detail Page: Displays detailed information about a product, including its price, description, and stock status.
- Cart Page: Allows users to review selected items, modify quantities, and proceed to checkout.
- Order Confirmation Page: Displays the order summary and payment confirmation.

5. As a Business Analyst, What Elicitation Techniques you are aware of? (BDRFOWJIPQU)

Ans-

Requirement Elicitation is a process of digging out the information from the stake holders. Requirement Elicitation serves the foundation in documenting the Requirements.

Brainstorming

Brainstorming is a group activity designed to generate a wide range of ideas and solutions through open discussions. This technique is particularly effective for generating a large number of ideas in a short time and is well-suited for collaborative projects like an Online Agriculture Product Store. During brainstorming sessions, stakeholders and team members, such as farmers' representatives and project committee members, share ideas openly without immediate judgment or criticism. This encourages creative thinking and uncovers potential opportunities or overlooked issues. The ideas generated are later analysed and refined for feasibility and alignment with project goals.

Advantages:

Encourages creativity and innovation.
Engages multiple stakeholders, enhancing collaboration.
Efficient in generating diverse ideas quickly.

Disadvantages:

Risk of dominant participants overshadowing others.
May result in too many unfeasible ideas requiring extensive filtering.

Document Analysis

Document analysis involves reviewing existing documents such as business plans, user manuals, policies, or project charters to identify relevant requirements. For this project, documents related to agricultural processes, farmer challenges, and manufacturer operations are valuable sources. This technique helps uncover implicit or unstated requirements while leveraging pre-existing information to establish a baseline for development.

Advantages:

Provides detailed and historical context.
Cost-effective as it utilizes existing information.
Minimizes reliance on stakeholder availability.

Disadvantages:

Information may be outdated or incomplete.
Requires expertise to interpret technical or complex documents correctly.

Reverse Engineering

Reverse engineering involves analysing an existing system or application to identify its components, design, and functionality to infer requirements. This is useful when a similar agricultural platform exists, providing insights into functional gaps or improvements. The team can evaluate how features align with the needs of remote farmers and adapt them accordingly.

Advantages:

Reveals functional and design specifics of existing systems.
Helps identify limitations or areas for improvement.
Useful when documentation is missing.

Disadvantages:

Time-intensive process.
May violate intellectual property if not handled ethically.

Focus Groups

Focus groups gather a selected group of stakeholders to discuss specific aspects of the project. For example, representatives of farmers, distributors, and agricultural experts can be assembled to explore challenges and expectations related to the platform. This interaction provides qualitative insights and reveals differing perspectives, leading to a comprehensive understanding of requirements.

Advantages:

Captures diverse opinions and feedback.
Promotes stakeholder engagement.

Quick turnaround for qualitative insights.

Disadvantages:

Risk of groupthink, where dominant opinions prevail.

Can be difficult to schedule with diverse participants.

Observation

Observation involves closely studying end-users or stakeholders as they perform their daily tasks to identify their needs and challenges. For this project, observing farmers' purchasing processes, including any manual or workaround methods, can highlight inefficiencies that the platform could address. This technique offers real-world insights into user behaviour and requirements.

Advantages:

Provides first-hand and authentic data.

Identifies issues that users may overlook or fail to articulate.

Enhances empathy for the end-user experience.

Disadvantages:

Time-consuming and resource-intensive.

Observer presence may influence user behaviour (Hawthorne effect).

Workshop

Workshops involve gathering stakeholders and team members in a structured environment to collaboratively define, analyse, and refine requirements. This technique fosters real-time communication and decision-making. For instance, a workshop could bring together farmers, manufacturers, and developers to prioritize features for the web/mobile application.

Advantages:

Encourages collaboration and consensus.

Reduces communication gaps through direct interaction.

Efficient for addressing complex requirements.

Disadvantages:

Requires skilled facilitation to avoid conflicts.

Logistical challenges in coordinating all participants.

Joint Application Development (JAD)

JAD sessions are formal workshops specifically designed to gather requirements through collaboration among developers, stakeholders, and users. For the agricultural platform, a JAD session might include representatives from remote farmers and manufacturers, facilitated by the project team, to ensure all perspectives are incorporated.

Advantages:

Encourages structured collaboration.

Facilitates consensus and faster decision-making.

Reduces the risk of incomplete or ambiguous requirements.

Disadvantages:

High preparation and coordination effort.

Requires significant stakeholder availability.

Interview

Interviews involve one-on-one or group discussions with stakeholders to extract specific requirements and insights. For this project, interviewing farmers, project committee members, and manufacturers helps uncover detailed expectations and unique perspectives. Structured, semi-structured, or unstructured formats can be used depending on the goals.

Advantages:

- Captures detailed and personal insights.
- Flexible and adaptable to the interviewee's expertise.
- Builds rapport and trust with stakeholders.

Disadvantages:

- Time-consuming for both parties.
- Risk of interviewer bias affecting responses.

Prototyping

Prototyping creates a preliminary version of the application or feature to visualize and validate requirements. For example, a prototype of the purchasing interface allows stakeholders to provide feedback on usability and functionality. Prototyping ensures alignment between expectations and deliverables early in the project.

Advantages:

- Enhances stakeholder understanding through visualization.
- Early detection of usability or design flaws.
- Reduces the risk of costly changes later.

Disadvantages:

- Can create unrealistic expectations if the prototype is perceived as final.
- Resource-intensive for iterative refinements.

Questionnaire

Questionnaires are written sets of questions distributed to stakeholders to gather requirements efficiently. This technique is useful for reaching a wide audience, such as a survey among farmers about their preferences for purchasing fertilizers, seeds, and pesticides. Responses are analysed to identify common needs and trends.

Advantages:

- Cost-effective for large groups.
- Ensures anonymity, encouraging honest feedback.
- Standardized format simplifies analysis.

Disadvantages:

- Limited interaction for clarifying ambiguous responses.
- Response rates and quality may vary.

Use Case Specifications

Use case specifications document the interactions between users and the system to achieve specific goals. For the agricultural platform, use cases could include a farmer purchasing a product, viewing order history, or receiving delivery updates. This technique ensures requirements are captured in a user-centric manner.

Advantages:

- Clearly defines functional requirements.
- Highlights user-system interactions and dependencies.

Useful for creating test cases and validating requirements.

Disadvantages:

Time-intensive to develop detailed use cases.

May require additional expertise for comprehensive documentation.

By applying these elicitation techniques strategically, the project team can gather robust and actionable requirements to create a user-centric and functional platform for remote farmers.

6. Which Elicitation Techniques can be used in this Project and Justify your selection of Elicitation Techniques?

Ans-

Prototyping

Purpose: To create visual representations of the web/mobile application to gather feedback.

Justification:

- Helps visualize the user interface for farmers and manufacturers, ensuring the application is intuitive and user-friendly, which is critical for new users.
- Enables early validation of design and functionality, reducing misunderstandings about requirements.
- Provides stakeholders (e.g., farmers like Peter, Kevin, and Ben) a concrete way to express feedback on usability and functionality.
- Facilitates alignment with business objectives by allowing Mr. Henry's committee to ensure the project meets their vision.
- Enhances collaboration between the development team and stakeholders by bridging technical and non-technical perspectives.

Use Case Specifications

Purpose: To document detailed interactions between users and the application.

Justification:

- Clearly defines how farmers and manufacturers will interact with the application, covering key functionalities like product browsing, selection, and order placement.
- Captures scenarios like registration, order tracking, and delivery management, ensuring comprehensive coverage of requirements.
- Provides a structured format for developers and testers to understand the expected application behaviour.
- Serves as a reference for identifying edge cases and non-functional requirements such as performance and scalability.
- Supports effective communication among stakeholders, ensuring all interactions are accounted for and agreed upon.

Document Analysis

Purpose: To study existing documents and data to extract relevant information.

Justification:

- Allows analysis of existing agricultural product distribution processes to identify gaps and pain points.
- Helps derive insights from similar CSR initiatives or projects that could inform design and implementation.

- Assists in understanding regulatory or compliance requirements for selling fertilizers, seeds, and pesticides.
- Facilitates the extraction of requirements from legacy systems or previous applications used by manufacturers.
- Provides a foundation for creating accurate requirement specifications and process flows.

Brainstorming

Purpose: To generate a wide range of ideas and solutions collaboratively.

Justification:

- Encourages active participation from all stakeholders, including farmers, developers, and committee members, fostering a collaborative environment.
- Helps identify innovative features that could enhance the application's value, such as personalized recommendations or localized language support.
- Aids in prioritizing features based on the needs and preferences of end-users.
- Allows the team to anticipate and address potential challenges, such as connectivity issues in remote areas.
- Supports rapid consensus-building on critical aspects like delivery logistics and payment methods.
- These techniques collectively ensure a thorough understanding of user needs, technical feasibility, and project goals, thereby contributing to the successful delivery of the Online Agriculture Products Store.

7. Make suitable Assumptions and identify at least 10 Business Requirements.

Ans-

Business Requirements are higher-level statements of the goals, objectives, or needs of the enterprise. They describe the reasons why a project has been initiated, the objectives that the project will achieve, and the metrics that will be used to measure its success. Business requirements describe needs of the organization as a whole, and not groups or stakeholders within it. They are developed and defined through enterprise analysis.

The Business Requirements according to the case study-

BR001- Users should be able to search fertilizers, seeds and pesticides.

BR002- Users should be able to browse through the product catalog.

BR003- Users need to create login id and password.

BR004- If the User is the new user then he should create the login id and password first.

BR005- Manufacturers should be able to upload and display their products in the application.

BR006- Farmers should be required to create an account or log in to add products to the cart or the buy-later list.

BR007- The application should provide multiple payment options, including Cash-on-Delivery (COD), Credit/Debit cards, and UPI.

BR008- A delivery tracker should be available to track the current status and location of the order.

BR009- The system should send SMS/email notifications for account creation.

BR010- An order confirmation email should be sent to farmers upon successful placement of an order.

8. List your assumptions?

Ans-

1. The online agriculture product store will primarily cater to farmers and companies involved in the manufacturing of fertilizers, seeds, and pesticides.
2. The store will operate as a web and mobile application to provide accessibility to users.
3. The project will be developed by APT IT SOLUTIONS company, which has the necessary talent pool.
4. The project duration is 18 months, and it is being carried out as part of a Corporate Social Responsibility (CSR) initiative.
5. Mr. Karthik is the Delivery Head overseeing the project, and Mr. Vandanam is the assigned Project Manager.
6. Peter, Kevin, and Ben are considered key stakeholders in the project as they shared their requirements and are part of the committee helping Mr. Henry.
7. The store will require a user login system for manufacturers and farmers to access different functionalities.
8. A product catalog will be available, featuring detailed information about fertilizers, seeds, and pesticides, including pricing and manufacturer details.
9. Users will have the ability to search for specific products within the catalog.
10. Farmers will need to create an account using their email ID and password to make purchases or add products to a buy-later list. New users can create a new account by providing their email ID and creating a secure password.

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

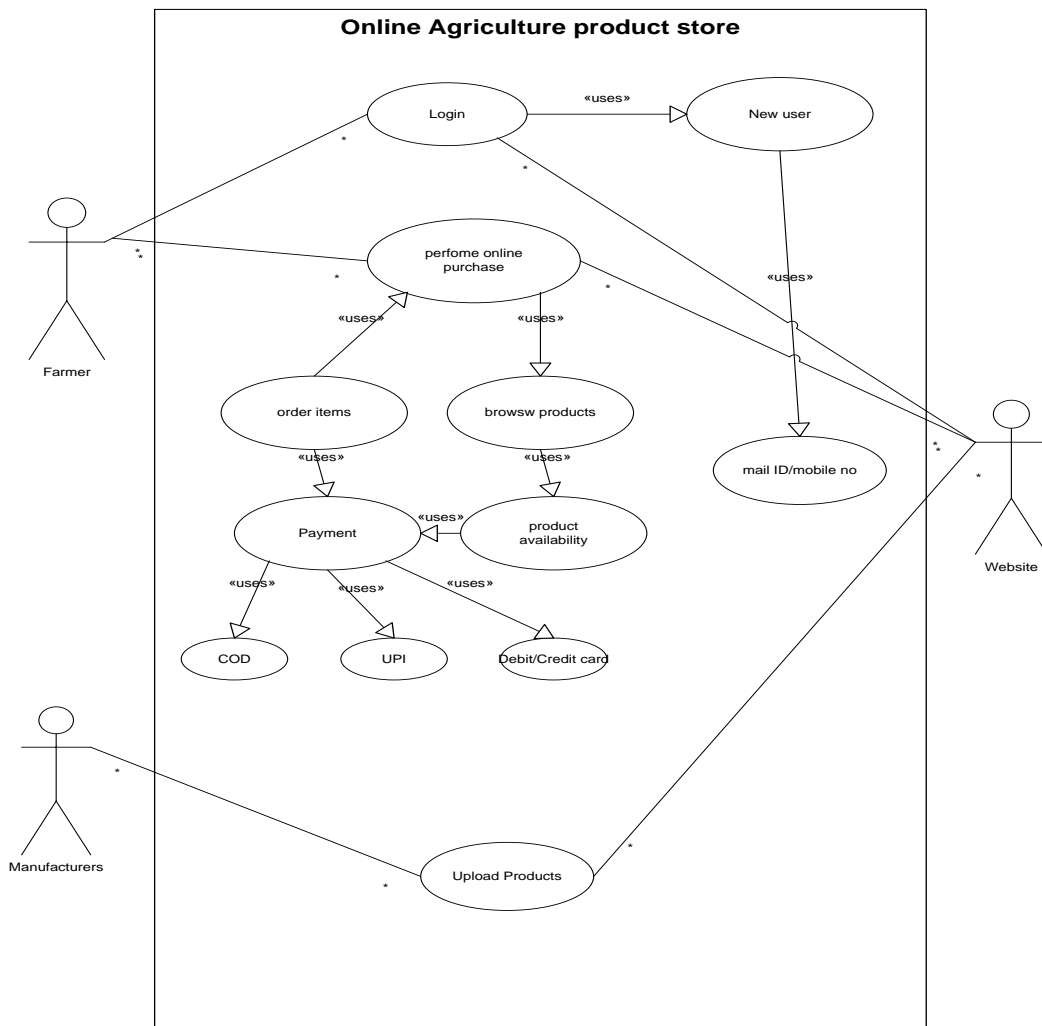
Ans-

Req ID	Req Name	Req Description	Priority
BR001	User search for product	Users should be able to search fertilizers, seeds and pesticides.	3
BR002	User browse through product	Users should be able to browse through the product catalog.	2
BR003	User create login	Users need to create login id and password.	1
BR004	New user create login	If the User is the new user then he should create the login id and password first.	1
BR005	Manufacturer upload their product	Manufacturers should be able to upload and display their products in the application.	8
BR006	Add to cart	Farmers should be required to create an account or log in to add products to the cart or the buy-later list.	5
BR007	Payment Gateway	The application should provide multiple payment options, including Cash-on-Delivery (COD), Credit/Debit cards, and UPI.	7

BR008	Product delivery tracking	A delivery tracker should be available to track the current status and location of the order.	9
BR009	E-mail conformation	The system should send SMS/email notifications for account creation.	4
BR010	E-mail conformation	An order confirmation email should be sent to farmers upon successful placement of an order.	4

10. Draw use case diagram

Ans-A use case diagram is a visual representation of the interactions between users (actors) and a system.



11. Prepare use case specs for all use cases?

Ans-

A Use Case Specification Document which provides a detailed description of a use case, outlining how users (actors) will interact with the system to achieve a specific goal.

Use case: User login

Use case ID	BR003		
Use case Name	Users create login		
Created by	Charani	Last update	25/12/2024
Date created	17/12/2024	Last revision	22/12/24
Actor	Farmers, Manufacturer		
Description	Users need to create login id and password.		
Pre-condition	None		
Post-condition	User is authenticated and logged in		
Normal Flow	Step 1: The user navigates to the login page of the application. Step 2: The system displays the login form with fields for email ID and password. Step 3: The user enters a valid email ID and password and clicks the "Login" button. Step 4: The system verifies the email ID and password against stored credentials. Step 5: If the credentials match, the system logs the user in and redirects them to the home/dashboard page. Step 6: A successful login confirmation message is displayed.		
Alternative Flow	Step 1: The user enters an incorrect email ID or password Step 2: The system verifies the credentials and identifies a mismatch. Step 3: The system displays an error message: "Invalid email ID or password. Please try again." Step 4: The user clicks on the "Forgot Password" link on the login page. The system redirects the user to the password recovery page. The user enters their registered email ID and submits the form. The system sends a password reset link or OTP to the user's email. The user follows the instructions in the email to reset their password and logs in successfully.		
Expectations	Secure login for users (Farmers and Manufacturers). Ability to reset forgotten passwords.		
Frequency of use	High; accessed by every user each time they log in or register.		
Assumptions	User has an email address. Internet connectivity is available. Security measures like CAPTCHA and strong password requirements are in place.		

Use case: Search Products

Use case ID	BR001		
Use case name	Search Products		
Created by	Charani	Last updated by	Dec 13 th 2024
Date created	Dec 11 th 2024	Last revision date	Dec 13 th 2024
Actor	Farmer		
Description	Users should be able to search fertilizers, seeds and pesticides.		
Pre – condition	Farmer is logged in. The Product catalog is displayed		
Post – condition	Search results matching the entered query are displayed		
Normal flow	Step 1 –The farmer visits the website and accesses the homepage. Step 2 –The farmer browses the product catalog, which includes fertilizers, seeds, and pesticides. Step 3 – The farmer uses the search bar, located prominently on the webpage, to enter keywords or product names. Step 4 – The system processes the search query and displays relevant results, including product images, descriptions, and prices. Step 5 – The farmer clicks on a product for detailed information, such as specifications, usage instructions, and availability.		
Alternate flow	Step 1 - If the farmer enters incorrect or partial keywords, the system provides suggestions or displays products with similar names or categories. Step 2 – If no matching results are found, a "No Results Found" message appears with recommendations to refine the search. Step 3 – In case of internet connectivity issues during the search, the system alerts the user to check their connection and retry.		
Expectations	The search should support filters like price range, product category, and manufacturer. The search results should be accurate and relevant, enhancing user satisfaction.		
Frequency of use	High		
Assumptions	Farmers have access to devices with stable internet connectivity for seamless search experiences		

Use case: Add to Cart

Use case ID	BR007		
Use case name	Add to cart option		
Created by	Charani	Last updated by	Jan 7 th 2024
Date created	Jan 1 st 2024	Last revision date	Jan 8 th 2024
Actor	Farmer		
Description	Farmers should be required to create an account or log in to add products to the cart or the buy-later list		
Pre – condition	Farmer is logged in. The products catalog is displayed.		
Post – condition	Selected products are added to the buy-later list		
Normal flow	Step 1 - The user visits the website and browses through the product catalog.		

	<p>Step 2 - The user logs into their account using their email ID and password.</p> <p>Step 3 - The user selects a product they are interested in but does not wish to purchase immediately.</p> <p>Step 4 - The user clicks the "Add to Buy-Later List" option associated with the product.</p> <p>Step 5 - The system confirms the addition of the product to the Buy-Later List.</p> <p>Step 6 - The user can view the Buy-Later List at any time by navigating to their account dashboard.</p>
Alternate flow	<p>Step 1 - If the user is not logged in and tries to add a product to the Buy-Later List, the system prompts the user to log in or create an account.</p> <p>Step 2 - If the user enters incorrect login details, the system displays an error message and allows them to retry.</p> <p>Step 3 - If the user does not have an account, they can register by providing their email ID and creating a secure password.</p>
Expectations	The system should notify users if an item in their Buy-Later List goes out of stock or has a price change.
Frequency of use	Low
Assumptions	Users are familiar with the concept of a Buy-Later List and understand its purpose.

Use case: Payment Gateway

Use case ID	BR009		
Use case name	Payment Gateway		
Created by	Charani	Last updated by	Jan 12 th 2024
Date created	Jan 10 th 2024	Last revision date	Jan 13 th 2024
Actor	Farmer		
Description	The application should provide multiple payment options, including Cash-on-Delivery (COD), Credit/Debit cards, and UPI.		
Pre – condition	Farmer has selected products for purchase.		
Post – condition	Payment is processed and confirmed.		
Normal flow	<p>Step 1 - The user selects the desired products and proceeds to the checkout page.</p> <p>Step 2 - The payment gateway displays available payment options: Credit/Debit card, UPI, and Cash on Delivery (COD).</p> <p>Step 3 - The user selects a payment method and provides necessary details (e.g., card number or UPI ID).</p> <p>Step 4 - If using COD, no additional details are required, and the order is confirmed.</p> <p>Step 5 - Upon successful payment, the user receives a confirmation message on the website.</p> <p>Step 6- An email confirmation is sent to the user, detailing the order summary and payment status.</p>		

	Step 7 - The order is forwarded for processing and shipping.
Alternate flow	<p>Step 1 - If the payment fails due to incorrect details, insufficient funds, or technical issues, the user is notified immediately.</p> <p>Step 2 - The user can retry with the same method or choose an alternate payment method.</p> <p>Step 3 - In case of persistent failure, the system provides a contact support option to assist the user.</p> <p>Step 4 - If COD is selected, and the user wants to modify the payment method later, they can do so before order processing begins.</p>
Expectations	<p>The payment gateway should be secure and comply with industry standards</p> <p>Transaction success or failure notifications must be instantaneous and clear.</p> <p>Email confirmations must include all order details, payment status, and estimated delivery timelines.</p>
Frequency of use	High
Assumptions	<p>All users have access to the internet and a valid payment method (card, UPI, or COD).</p> <p>Users understand basic online payment processes.</p>

Use case: Product delivery tracking

Use case ID	BR010		
Use case name	Product delivery tracking		
Created by	Charani	Last updated by	Jan 17 th 2024
Date created	Jan 15 th 2024	Last revision date	Jan 18 th 2024
Actor	Farmer		
Description	A delivery tracker should be available to track the current status and location of the or order.		
Pre – condition	Farmer has made a purchase and received order confirmation.		
Post – condition	Delivery status is displayed		
Normal flow	<p>Step 1 - The user logs into the platform using their registered email ID and password.</p> <p>Step 2 - The user navigates to the "Order History" or "Delivery Tracking" section.</p> <p>Step 3 - The platform displays a list of recent orders with details such as order ID, product name, and status.</p> <p>Step 4 - The user selects an order to view its delivery tracking details.</p> <p>Step 5 - The system fetches the delivery status in real-time and displays details such as dispatch date, current location, estimated delivery date, and delivery progress.</p> <p>Step 6- The user reviews the tracking information and logs out or navigates to other sections of the platform.</p>		

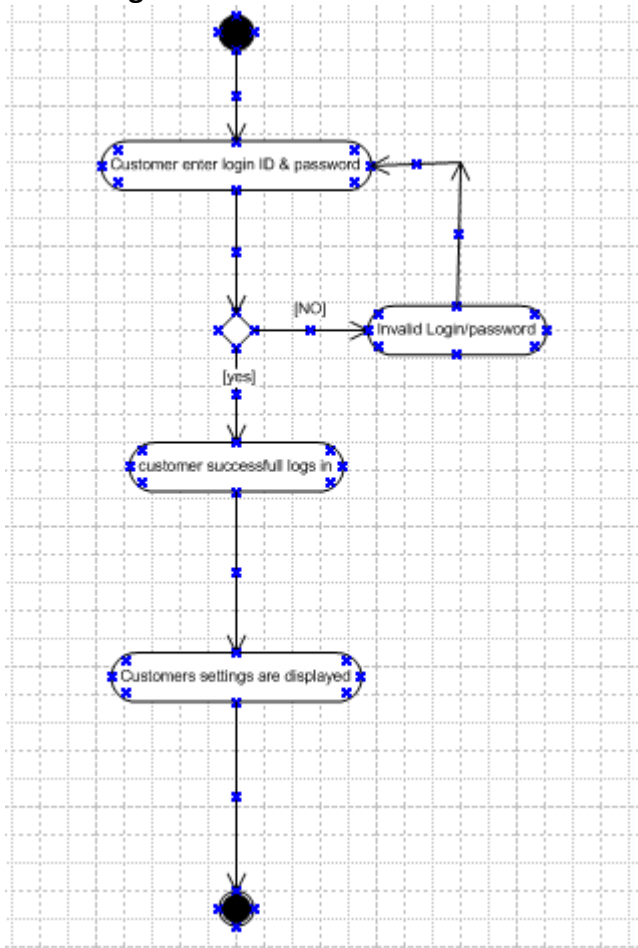
Alternate flow	<p>Step 1 - If a delivery is delayed, the system sends an updated notification with the revised estimated delivery time.</p> <p>Step 2 - If the order is canceled by the user or due to unforeseen circumstances, the system processes a refund and updates the user via email.</p> <p>Step 3 - If the real-time delivery tracking system is temporarily unavailable, the platform displays the last known status of the order and an estimated delivery date.</p>
Expectations	<p>Users receive email notifications with updates on critical delivery milestones, such as dispatch, transit, and delivery completion.</p> <p>Delivery tracking information includes comprehensive details like package current location, expected delivery time, and delivery agent contact (if applicable).</p>
Frequency of use	High
Assumptions	The delivery tracking system is robust and functions seamlessly across different devices, including desktops and smartphones.

12. Activity diagrams

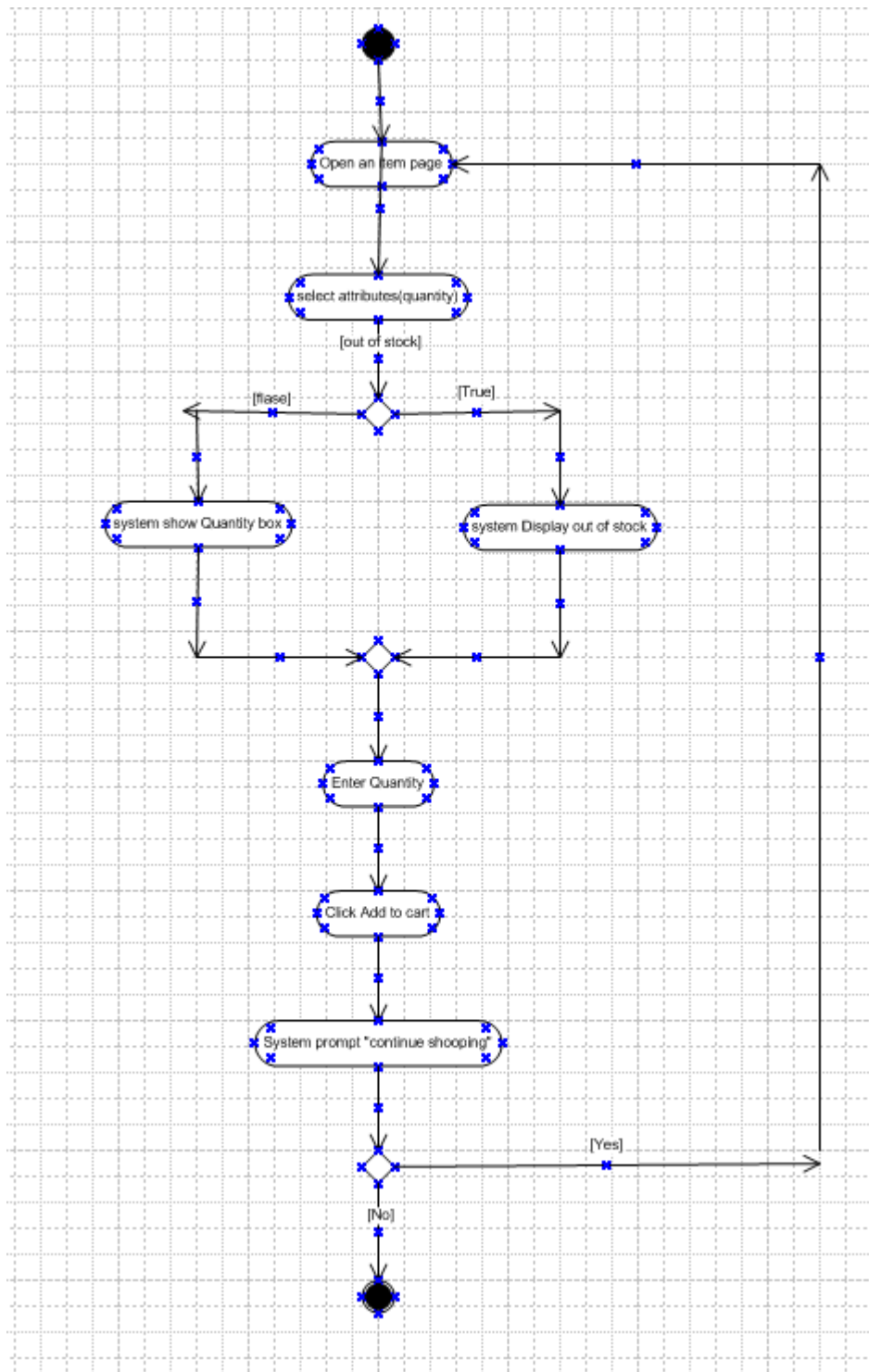
Ans-

An activity diagram is a type of diagram in the unified modelling Language (UML) that visually represents the flow of activities within a system.

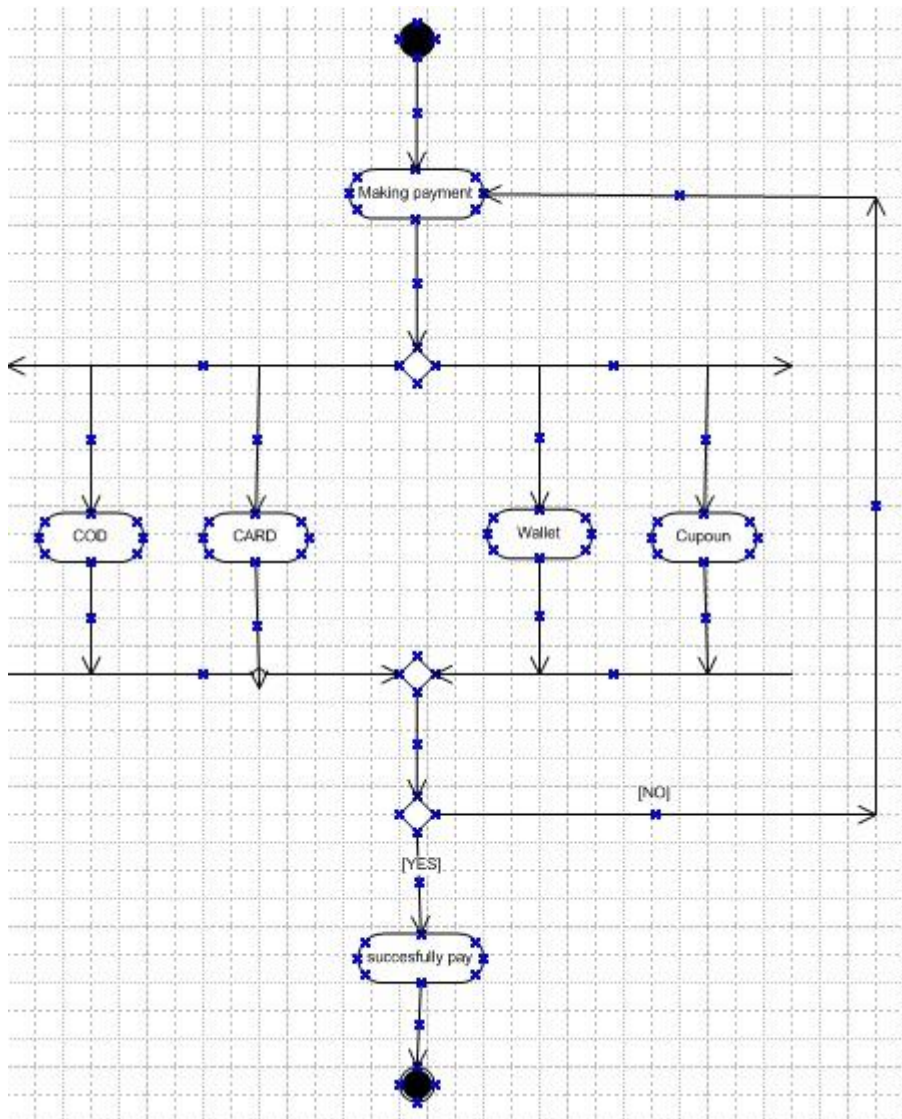
1.User Login



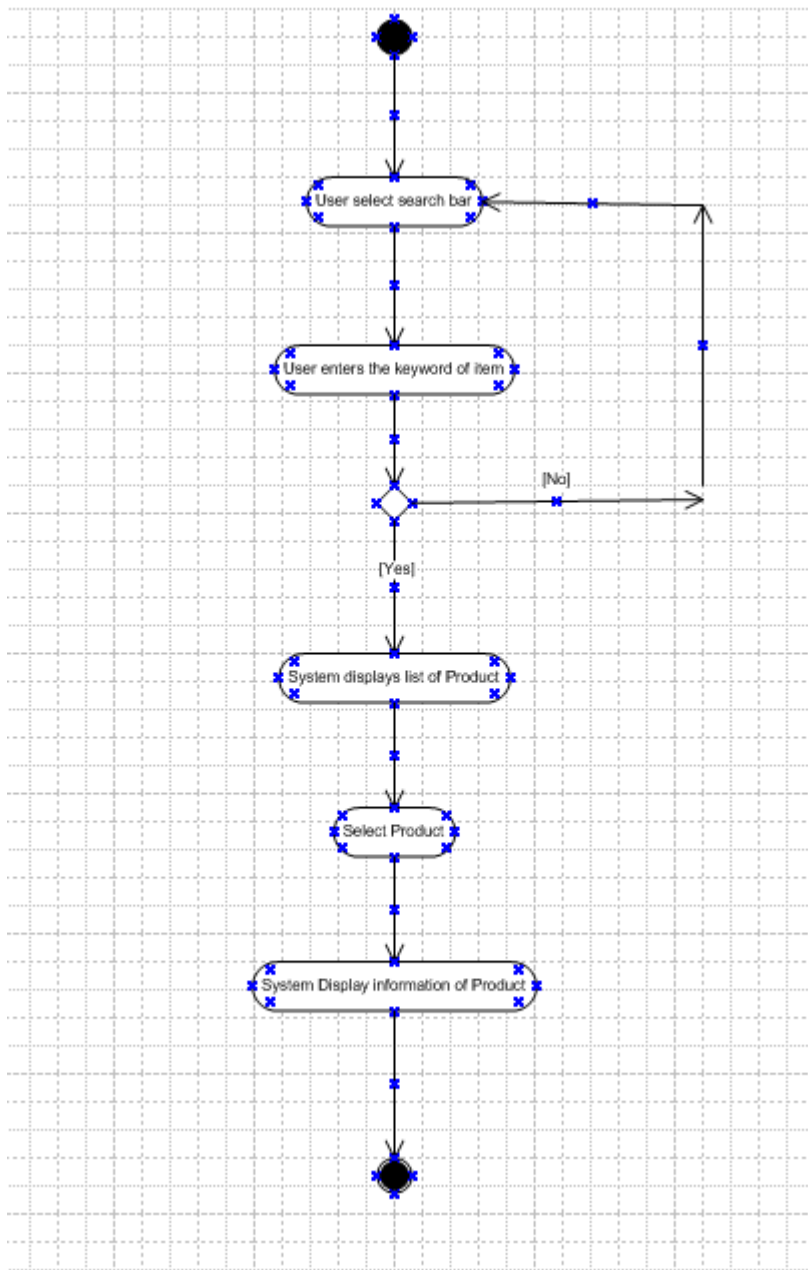
2.Add product to cart



3. Making a payment



4. Search Products



5.Delivery

