**Capstone prep 2**

**Q1,** Quarterly audits for a project, including a Business Analyst’s (BA) contributions, are conducted to ensure compliance, quality, and alignment with the project objectives. Here’s how these audits would typically happen from a BA’s perspective:

**Objective of Quarterly Audits:**

1. **Evaluate Project Progress:**

Assess whether milestones, deliverables, and timelines are being met.

2. **Ensure Requirements Traceability:**

Verify that all requirements gathered by the BA are captured, documented, and implemented correctly.

3. **Assess Compliance:**

Ensure adherence to organizational, regulatory, and stakeholder standards.

4. **Identify Risks and Gaps:**

Review any risks, gaps, or deviations from the project scope and mitigate them

Quarterly audits for a BA in a project are essential to ensure compliance, identify risks, and track the progress of deliverables. The quarterly audits (Q1, Q2, Q3, Q4) are designed to ensure that the project stays on track and aligns with the business objectives and stakeholder expectations.

|  |  |
| --- | --- |
|   | Quarter 1 - Audit Report |
| Stage | Requirement Gathering Phase |
| Completed |  16 Weeks (Week 1 to Week 16) |
| Check list | * BRD Template
* Elicitation Result Report
* Grouping of Functionalities/Features
* Client Sign-Off
* Email Communication (To, CC, BCC)
 |

|  |  |
| --- | --- |
|   | Quarter 2 - Audit Report |
| Stage | Design Phase |
| Completed |  16 Weeks (Week 17 to Week 32) |
| Check list | * Wireframes & Mockups
* UML Diagrams
* Technical Design Document
* Client approval on design
* Design walkthrough with team
 |

|  |  |
| --- | --- |
|   | Quarter 3 - Audit Report |
| Stage | Development Phase |
| Completed |  24 Weeks (Week 33 to Week 56) |
| Check list | * Development Kickoff
* Module development status
* Code reviews
* Integration testing initiated
* Regular sprint demos
 |

|  |  |
| --- | --- |
|   | Quarter 4 - Audit Report |
| Stage | Testing & Deployment Phase |
| Completed |  16 Weeks (Week 57 to Week 72) |
| Check list | * Test cases prepared
* System testing completed
* User acceptance testing (UAT)
* Bugs fixes & retesting
* Deployment completed
 |

**Q2**, **BA Approach Strategy**

To approach this project effectively, here’s a strategy I would follow as a Business Analyst:

1. **Elicitation Techniques to Apply:**

As a Business Analyst, the first step is to gather clear and comprehensive requirements from all relevant stakeholders. I will use the following elicitation techniques:

**Interviews:** One-on-one meetings with stakeholders (Mr. Henry, Peter, Kevin, Ben, etc.) to understand their specific needs and concerns.

**Workshops:** Group discussions with internal teams (project manager, developers, testers) and external stakeholders (farmers, manufacturers) to define the features and functionalities of the online store.

**Questionnaires:** To collect data from a wider group of farmers, especially those in remote areas, on their pain points and needs.

**Document Analysis**: Review any existing documentation related to previous projects or industry standards for agriculture product stores.

1. **Stakeholder Analysis (RACI/ILS):**

I will use the RACI matrix to clarify roles and responsibilities and ILS (Influence/Interest) matrix to track stakeholders' involvement:

**RACI Matrix:**

Responsible: Business Analyst (Myself) for requirements gathering and documentation.

Accountable: Project Manager (Mr. Vandanam) for overall project delivery.

Consulted: Stakeholders (Mr. Henry, Mr. Pandu, Mr. Dooku, Peter, Kevin, Ben) for their input and feedback.

Informed: Development Team, Testing Team, and other key members to keep them updated on progress.

**ILS Matrix:**

High Influence, High Interest: Mr. Henry, Mr. Pandu, Project Manager.

High Influence, Low Interest: Senior Java Developer, Testers, DB Admin, Network Admin.

Low Influence, High Interest: Farmers (Peter, Kevin, Ben, etc.).

Low Influence, Low Interest: External vendors or third-party services.

1. **Documents to Write:**

I will create and maintain various documents to ensure clarity and alignment:

**Business Requirements Document (BRD):** Outline the business goals, vision, objectives, and key functionalities.

**Functional Requirements Document (FRD):** Detail the system features, including product categories (fertilizers, seeds, pesticides), user roles, transaction flow, and delivery system.

**Use Cases/User Stories:** For user interactions with the system, ensuring the functionality meets stakeholders’ needs.

**Test Cases:** Define the criteria to validate the developed features.

1. **Process to Sign off on Documents:**

**Initial Draft Review:** Share the first draft of each document internally with the team to ensure technical feasibility and clarity.

**Stakeholder Review:** Share documents with key stakeholders for review and feedback.

**Feedback Incorporation:** Revise documents based on feedback from stakeholders.

**Formal Sign-off:** Once the documents are finalized, request formal approval via email or a signature from the stakeholders (Mr. Henry, Committee).

1. **Approvals from the Client (Mr. Henry, Stakeholders):**

**Formal Approval Process:** After each phase (e.g., requirements gathering, design), I will present the completed deliverables for approval.

**Change Request Approval:** Any change in scope, timeline, or budget will be documented as a change request, evaluated, and signed off by Mr. Henry and the committee.

1. **Communication Channels to Establish:**

**Daily Standups:** Quick updates within the team to track progress, resolve blockers, and align on tasks.

**Weekly Status Updates:** Share progress, risks, and upcoming milestones with stakeholders via email or meetings.

**Document Sharing:** Use a shared platform (e.g., Google Drive, SharePoint, Jira) to store and manage all project-related documents.

**Instant Communication:** Use WhatsApp or Slack for urgent communication and immediate clarifications.

1. **Handling Change Requests:**

**Change Request Documentation:** When new requirements or changes arise, I will document the change request, including impact analysis on scope, time, and cost.

**Impact Assessment**: Collaborate with the project manager, development team, and stakeholders to evaluate how the change will affect the project.

**Approval Process:** Present the change request to Mr. Henry and relevant stakeholders for approval, documenting their decisions.

1. **Updating the Progress of the Project to Stakeholders:**

**Weekly Reports:** Share detailed progress reports with stakeholders, highlighting completed tasks, current progress, and issues.

**Milestone Reviews:** Organize periodic milestone review meetings with stakeholders to showcase the work completed and gather feedback.

**Project Dashboard:** Implement a visual project tracking dashboard (via Jira or similar tools) that shows real-time status of tasks and deliverables.

1. **Taking Signoff on the UAT – Client Project Acceptance Form:**

**User Acceptance Testing (UAT):** After the system is developed, ensure the client (Mr. Henry, Peter, Kevin, Ben) tests the application.

**Create UAT Test Cases:** Develop test cases based on the requirements for both functional and non-functional aspects.

**Client Feedback:** Coordinate with the client and stakeholders for their feedback and to make sure the system meets their expectations.

**UAT Sign-off:** Once UAT is successfully completed and all issues are addressed, get formal sign-off on the Client Project Acceptance Form.

**Q3, -Tier Architecture**

The 3-tier architecture is a widely used software architecture that organizes applications into three layers or tiers. Each tier focuses on a specific function by making the system easier to manage and develop. The changes in one layer do not affect the others, which helps with faster updates and modifications. Each layer has a distinct role for improving scalability, maintainability, and reusability. This structure makes applications easier to build, update, and scale over time.

Each tier has its own responsibility (UI, business logic, and data management), making the system modular, easier to manage, and scalable. Modifications to one layer (like UI redesigns or database structure changes) do not require changes in the other layers.

The tiers are:

1. **Presentation Tier (User Interface)**

This is the topmost layer where users interact with the application. It is the client-facing layer. It consists of the user interface (UI) and handles input/output, such as web browsers or mobile apps.

* This is the top layer that interacts with the user.
* Its main job is to collect input from users and display the relevant data to them.
* It sends user requests to the Logic Tier and displays the results back to the user.
* Example: A web page or a mobile app where users browse agricultural products like fertilizers, seeds, or pesticides.
1. **Application Tier (Logic)**

The middle layer that processes the business logic, validates inputs, and manages the application’s core functionality. It acts as a bridge between the user interface and the database.

* It acts as an intermediary between the user interface (Presentation Tier) and the data (Data Tier).
* The Logic Tier validates the inputs, processes requests, and sends necessary data to the Data Tier, then returns the processed results to the Presentation Tier.
* This layer contains the core functionality, such as handling user authentication, managing business rules, and interacting with external services.
* It also allows for flexibility in modifying business processes or adding new features without changing the data or presentation layer.
* Example: The logic for validating stock availability, calculating discounts, handling payment processing, and managing delivery requests is all handled by the Logic Tier. If the farmer requests to buy a product, the Logic Tier checks whether the product is in stock and calculates the total price (with any applicable discounts or taxes).
1. **Data Tier (Database)**

The layer responsible for storing and managing data in a database. It retrieves, stores, and updates data as requested by the Logic Tier.

* The Data Tier is responsible for storing and managing the application’s data.
* This layer is responsible for maintaining persistent data such as user profiles, order history, product details, pricing, and stock levels. It ensures that data is securely stored and efficiently retrieved.
* The Data Tier ensures data integrity and consistency while interacting with the Logic Tier to provide the correct data when requested.
* It provides centralized data storage, which simplifies data management and ensures consistency across the application.
* Example: A database that stores details about products (seeds, fertilizers, pesticides), user profiles, orders, delivery information, and pricing. The data layer responds to requests from the Logic Tier, such as retrieving a list of available products or updating an order’s status.

**Q4,** **BA Approach Strategy for Framing Questions**

When a Business Analyst (BA) frames questions to ask stakeholders, it's important to ensure that the questions are clear, purposeful, and aligned with the project’s objectives. It helps in understanding the context and gathering complete information by addressing Who is involved, What needs to be done, When it happens, Where it occurs, Why it’s important, and How it will be implemented.

1. **5W 1H (Who, What, When, Where, Why, How)**

This will help the BA to ask comprehensive and open-ended questions that cover all aspects of a scenario.

* Who: Who are the key stakeholders involved? Who will use the system? Who will be responsible for a task?
* What: What are the requirements or challenges? What functionalities should the system have?
* When: When should certain features or tasks be completed? What is the timeline?
* Where: Where will the application be used (web, mobile, location-specific requirements)?
* Why: Why is this feature needed? Why are certain processes in place?
* How: How will the system be used? How will data flow through the system?

Example:

Who are the users who will access the product catalog?

1. **SMART (Specific, Measurable, Achievable, Relevant, Time-bound)**

SMART is a framework for setting clear, actionable, and realistic goals. It ensures that objectives are Specific, Measurable, Achievable, Relevant, and have a Time-bound deadline to make them more focused and easier to track.

* **Specific:**

The goal should be clear and unambiguous, addressing exactly what needs to be achieved. For example, "Increase product sales" is too vague, but "Increase product sales by 10% in the next quarter" is specific.

Is the question clear and detailed enough to avoid ambiguity?

* **Measurable:**

You should be able to quantify progress or completion. Measurability ensures that you can track and evaluate success, such as "Increase website traffic by 20% in the next six months."

Can you measure the success or outcome of the question’s answer?

* **Achievable:**

The goal must be realistic given the available resources and constraints. It should be challenging, but not so difficult that it’s unattainable. For example, setting a goal to increase sales by 500% might not be achievable in a short timeframe.

Is the question asking for something realistic given the resources?

* **Relevant:**

 The goal should align with broader business objectives and be meaningful. It should matter to both the project and the organization.

Does the question directly relate to the project’s goals or stakeholders' needs?

* **Time-bound:**

Every goal needs a deadline. Setting a timeframe helps create urgency and keeps the project on track. For example, “Launch new mobile app feature by the end of Q2” is time-bound.

Example:

What specific product categories should be available on the agriculture store by the end of Q2?

What is the target delivery time for each order after placing it?

1. **RACI (Responsible, Accountable, Consulted, Informed)**

RACI is a matrix used to define roles and responsibilities in a project. It clarifies who is Responsible for performing tasks, Accountable for decision-making, consulted for input, and Informed about progress or decisions.

* **Responsible:** Who will do the task or work?

The person or people who perform the work to achieve a task or deliverable. They are responsible for executing the activity and ensuring the task is completed.

Example: A developer responsible for coding a new feature.

* **Accountable:** Who is ultimately responsible for the success of the task?

The person who is ultimately accountable for the success or failure of the task. There should only be one accountable person per task. They ensure that the work is done correctly and meets standards.

Example: The project manager who oversees the overall success of the feature rollout.

* **Consulted:** Who needs to be consulted for input or feedback during decision-making?

These are the people who need to be consulted for input or advice. Their feedback or expertise is important during the process, but they are not directly involved in completing the work.

Example: A subject matter expert who provides guidance on the feature’s functionality.

* **Informed:** Who needs to be informed of progress or decisions?

These are the people who need to be kept informed about the progress or outcomes of the task, but they do not participate in the work itself.

Example: Senior stakeholders who are kept informed about the project’s status and updates.

1. **3-Tier Architecture**

3-Tier Architecture divides an application into three layers.

* **Presentation Layer (Client Tier):** The user-facing layer (UI/UX) where the user interacts with the system (e.g., web or mobile app). It handles input and output but does not process or store data.

Example: The app where farmers browse products and make purchases.

* **Logic Layer (Application Tier):** This middle layer processes business logic. It acts as an intermediary between the user and the database, validating input, processing data, and executing business rules.

Example: Logic that checks if products are in stock or calculates pricing and discounts.

* **Data Layer (Database Tier):** The backend layer responsible for storing, retrieving, and managing data. It ensures that data is safely stored in databases and accessed by the logic layer when required.

Example: A database storing product details, order history, user information, etc.

1. **Use Cases** Use cases describe how a user interacts with a system to achieve a specific goal. They outline the functional requirements of a system from the user's perspective, providing a clear understanding of how the system should behave in various scenarios. The BA needs to understand user roles, actions, and outcomes.

**Example:** “A farmer searches for seeds on the online store and adds them to the cart."

Use cases typically include the actor (user), the system actions, and the goal or outcome (e.g., successfully purchasing seeds).

1. **Use Case Specifications**

Use case specifications provide a more detailed and structured description of a use case. It includes step-by-step actions, alternate scenarios, preconditions, postconditions, and exceptions, ensuring that the system's behavior is well-defined.

**Example:** For the "Farmer makes a purchase" use case, the specification would include:

Preconditions: The user is logged in.

Main Flow: User selects products, adds them to the cart, and proceeds to checkout.

Alternate Flow: If the product is out of stock, show a notification.

Postconditions: Order is confirmed, and payment is processed.

1. **Activity Diagrams**

Activity diagrams represent the flow of activities or actions in a process. It helps to understand the processes and workflows in a visual way. The BA needs to ask questions related to activities, transitions, and conditions within the process. Ask about steps involved, decision points, and flow of activities.

**Example:** An activity diagram could show the process of a farmer searching for products, adding items to the cart, and proceeding to checkout. It also includes decision points, like whether the product is available in stock.

1. **Models**

Models are abstract representations of real-world processes or systems. They can include data models (structure of the database), process models (workflow of tasks), or object models (system components).

**Example:** A data model representing how different entities like "Farmers," "Orders," and "Products" are related in the database.

1. **Page Designs**

Page designs (also called wireframes or UI mockups) are visual representations of a website or application's layout. They show the placement of key elements like buttons, forms, and navigation, helping stakeholders visualize the user interface.

Example: A wireframe for the agriculture store showing the layout for the homepage, including product categories, search bar, and promotional banners.

**Q5, Elicitation Techniques**

Elicitation Techniques refer to the methods used by Business Analysts (BAs) to gather information, requirements, and insights from stakeholders, users, and other relevant sources to better understand the needs of the project. The goal of elicitation is to uncover the right information for creating effective solutions.

**BDRFOWJIPQU**

1. **Brainstorming**

Brainstorming is a creative group technique where participants contribute ideas freely and without judgment to solve a problem or generate solutions. It is particularly useful in the early stages of the project, when you need to come up with creative ideas for solutions or approaches to meeting business needs.

1. **Document Analysis**

Document analysis involves reviewing existing documentation such as reports, manuals, guidelines, or system records to gather relevant information. This technique is useful for gathering background information when analyzing existing processes or systems that need improvement.

1. **Requirements workshops**

Requirements workshops are structured, collaborative sessions where stakeholders and project team members gather to define, clarify, and prioritize requirements for a system or project. This typically used when a group of stakeholders needs to discuss and agree on the requirements for the project in a short time.

1. **Focus Groups:**

A focus group is a small group of people who are selected to provide their opinions, thoughts, or feedback on a specific topic or product. Focus groups are used when you want to gather qualitative insights, understand user attitudes, and explore detailed opinions from a group of people who represent the target audience.

1. **Observation:**

Observation involves watching users or stakeholders in their natural environment to understand their behavior, tasks, and challenges. This technique is useful when we want to understand user behavior and needs in a more natural, unspoken way.

1. **Workshops**

Workshops are structured, focused sessions that are designed to bring stakeholders together to solve problems, define requirements, or make decisions. Workshops are used for gathering detailed requirements, making decisions, and addressing specific issues where multiple stakeholders are involved.

1. **Joint Application Development (JAD)**

JAD is an intensive, collaborative approach where both business users and IT professionals work together in structured sessions to define the system's requirements. JAD is typically used for more complex systems where we need input from both technical and business perspectives and want to make decisions quickly.

1. **Interviews**

Interviews involve one-on-one or group discussions with stakeholders to gather detailed insights into their needs, expectations, and concerns. Interviews are particularly useful when you need to collect personalized or detailed feedback from stakeholders, especially if they are experts or have deep knowledge of the subject matter.

1. **Prototyping**

Prototyping involves creating a basic working model (prototype) of the system or product that users can interact with and provide feedback on. Prototyping is used when the requirements are not fully clear, and we need to explore options with users, refine features, and adjust the design based on feedback.

1. **Questionnaires**

Questionnaires are a set of written questions designed to gather specific information from a large group of people. Questionnaires are used when we need to gather responses from a large group of stakeholders or users in a systematic way, especially for quantitative data collection.

1. **Use Case and Scenarios**

Use cases and scenarios describe how users will interact with the system to achieve specific goals or tasks. Use cases are typically used to capture functional requirements and clarify how the system will behave when interacting with users.

**Q6,** **This project Elicitation Techniques**

I would recommend **‘Prototyping’** as the elicitation technique for this project.

* Since the project involves creating an online platform where farmers can browse products, search for them, and interact with a catalog of fertilizers, seeds, and pesticides, prototyping is an excellent technique to visualize the user interface and functionality.
* A prototype will help stakeholders see and interact with a basic version of the website or mobile application early in the development process. This allows them to provide quick feedback on how the features (such as search options, login process, product display) work in practice.
* It helps validate requirements and reduce misunderstandings early on.
* For example, farmers can test the product search feature, the payment gateway, and the delivery tracking system, giving clear feedback for refinement.

**Q7, Business Requirements (BR) and Stakeholder Requirements for the project:**

Based on the context provided, the Business Requirements (BR) and Stakeholder Requirements for the project are as below:

**BR001 – Farmers should be able to search for available products in fertilizers, seeds, pesticides**

Business Requirements (BR) :This requirement specifies that the system should allow farmers to browse and search for different products like fertilizers, seeds, and pesticides within the application.

Stakeholder Requirements:Kevin (Farmer): Kevin emphasized that farmers need an easy way to browse through products, and a search option should be available to find specific items they are looking for.

Mr. Henry: Mr. Henry expects that farmers should have an intuitive way to find the products they need, including a search feature to enhance the user experience.

Expected Outcome: Farmers can search for and quickly locate fertilizers, seeds, and pesticides within the online platform.

**BR002 – Manufacturers should be able to upload and display their products in the application**

Business Requirements (BR): This requirement ensures that manufacturers (fertilizers, seeds, pesticides) can add and manage their products in the online store, allowing them to display product details such as name, description, price, and stock availability.

Stakeholder Requirements: Peter, Kevin, Ben (Farmers): These stakeholders, while focusing on the farmer's needs, also indirectly rely on manufacturers being able to list their products effectively so they can access a full range of available products.

Mr. Henry: Mr. Henry expects the application to facilitate seamless interaction between manufacturers and farmers. Manufacturers should be able to upload and update their products easily for visibility and access by farmers.

Manufacturers (External Stakeholders): They need a simple interface to upload their product details, including fertilizers, seeds, and pesticides. The system should support easy management of product listings.

Expected Outcome: Manufacturers can add, update, and display their products in the online store, making them accessible for farmers to browse, purchase, and track.

**Q7**, **10 Business Requirements:**

1. **BR001 – Farmers should be able to search for available products (fertilizers, seeds, pesticides)**

Farmers should have an easy-to-use search feature to find products quickly based on various criteria such as product type, price, brand, etc.

1. **BR002 – Manufacturers should be able to upload and display their products in the application**

Manufacturers must have the ability to add, update, and manage their product listings, ensuring that product details are accurate and visible to farmers.

1. **BR003 – Farmers should be able to register and create an account using their email and secure password**

Farmers must have a secure registration process that enables them to create an account using their email and password, ensuring the safety of their personal data.

1. **BR004 – Farmers should be able to log in to their account to browse and purchase products**

After registration, farmers should be able to log in using their credentials to access the platform and make purchases.

1. **BR005 – Farmers should be able to add products to their cart and proceed with the checkout process**

Farmers need a simple process to select and purchase products, adding them to a cart and proceeding with payment.

1. **BR006 – The platform should provide a secure payment gateway with multiple payment options (COD, credit/debit card, UPI)**

A payment system must be in place that allows farmers to choose their preferred method of payment securely.

1. **BR007 – Farmers should receive email notifications confirming their order and its status**

After placing an order, farmers should receive an email confirmation with the order details and expected delivery information.

1. **BR008 – A delivery tracking feature should be available for farmers to monitor the status of their orders**

Farmers need the ability to track their orders in real-time to check the progress of their deliveries.

1. **BR009 – Manufacturers should be able to manage and update product inventory (e.g., product availability, stock levels)**

Manufacturers need to maintain control over their product listings, updating inventory levels to reflect availability in real-time.

1. **BR010 – Admin users should be able to manage and approve product listings uploaded by manufacturers**

The platform should have an admin interface where admins can review and approve or reject product listings submitted by manufacturers to ensure quality control and compliance with platform standards.

**Q8,** **Assumptions**

Assumptions for the Project:

1. Farmers and manufacturers have access to the internet, either through mobile or broadband, to use the online platform.
2. The platform will be mobile-responsive and user-friendly to accommodate users in remote areas who may not be very tech-savvy.
3. The payment gateway will be secure, supporting multiple payment options like cash-on-delivery (COD), credit/debit cards, and UPI.
4. Delivery tracking and shipping logistics will be handled by a third-party courier service that partners with the platform.
5. Manufacturers will provide accurate and up-to-date product details such as pricing, availability, and descriptions.
6. The login and registration process will be secure, with strong password protection and email verification to prevent unauthorized access.
7. The platform will support different types of users: Farmers, Manufacturers, and Admins (for managing the system).
8. The platform will comply with data protection laws, ensuring that user information is stored securely and not shared without consent.
9. The platform will provide multi-language support for farmers who speak different regional languages.
10. The platform will be designed to scale as the user base grows, especially as more manufacturers and farmers join.

**Q9**, **This project Requirements Priority –**

|  |  |  |  |
| --- | --- | --- | --- |
| **Req ID** | **Req Name** | **Req Description** | **Priority** |
| BR001 | Farmer Search for Products | Farmers should be able to search for available products in fertilizers, seeds, pesticides  | 9 |
| BR002 | Manufacturers upload their Products | Manufacturers should be able to upload and display their products in the application  | 10 |
| BR003 | Farmers Registration Process | Farmers should be able to register and create an account using their email and secure password. | 9 |
| BR004 | Farmers Login Process | Farmers should be able to log in to their account to browse and purchase products. | 9 |
| BR005 | Add Products to Cart & Checkout | Farmers should be able to add products to their cart and proceed with the checkout process. | 8 |
| BR006 | Secure Payment Gateway | The platform should provide a secure payment gateway with multiple payment options (COD, credit/debit card, UPI). | 10 |
| BR007 | Email Notifications | Farmers should receive email notifications confirming their order and its status. | 7 |
| BR008 | Delivery Tracking | A delivery tracking feature should be available for farmers to monitor the status of their orders. | 8 |
| BR009 | Manufacturers Product Inventory | Manufacturers should be able to manage and update product inventory (e.g., product availability, stock levels). | 8 |
| BR010 | Admin Management for Product Listings | Admin users should be able to manage and approve product listings uploaded by manufacturers. | 6 |

**Q10,**

1. **Use Case Diagram**

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

****

**Q11, Use Case Specs**

A use case specification document is 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 ID | UC-001 |
| Use Case Name | Add products to cart |
| Created by |  Ajas | Last updated by | 30-Jan-25 |
| Date created | Dec-23 | Last revision date | 26-Jan-25 |
| Actors | Farmer |
| Description | The farmer selects products and adds them to their shopping cart to proceed with the purchase. |
| Precondition: | Farmer must be logged into the system.The product must be available in stock. |
| Post condition | The selected product is added to the cart.The cart reflects the updated list of products. |
| Normal Flow of Events / Basic Flow: | The farmer browses the product catalog.The farmer selects a product to add to the cart.The system confirms the addition of the product.The cart is updated with the new item. |
| Alternative Flow: | If the product is out of stock, the system displays a message: "Product unavailable." |
| Exceptions: | If the system encounters a technical issue, the cart is not updated, and an error message is displayed. |
| Frequency of Use: | High, occurs every time a farmer adds a product to their cart. |
| Assumptions: | The system has an up-to-date product inventory.The farmer has a valid login session. |

|  |  |
| --- | --- |
| Use Case ID | UC-002 |
| Use Case Name | Complete payment |
| Created by |  Ajas | Last updated by | 30-Jan-25 |
| Date created | Dec-23 | Last revision date | 26-Jan-25 |
| Actors | Farmer |
| Description | The farmer completes the payment for the selected products in the cart. |
| Precondition: | The farmer has selected the products and is at the checkout page. |
| Post condition | Payment is processed, and the order is ready for shipping.The farmer receives a payment confirmation notification. |
| Normal Flow of Events / Basic Flow: | The farmer selects a payment method (credit/debit card, online wallet, etc.).The farmer enters the required payment details.The system processes the payment and verifies the transaction.The system confirms payment and proceeds to the delivery process. |
| Alternative Flow: | If the payment fails due to insufficient funds or incorrect details, the system prompts the farmer to retry. |
| Exceptions: | If there is a failure in the payment gateway, the system asks the farmer to try again later. |
| Frequency of Use: | High, occurs every time an order is placed. |
| Assumptions: | The payment gateway is functioning correctly.The farmer has a valid payment method. |

**Q12, Activity Diagrams –**

**Activity Diagrams –**

**Add Products to Cart**

Browse Product

View Product Details

Specify Quantity

List the items in cart for checkout

Go to payment page

Go to registration page

Start

Logged In

Yes

**Payment**

View cart contents

Proceed to checkout

Go to login page

Review order summary

Select payment method

Provide payment details

Place order

Yes

If payment is successful

Yes

No

Is logged in

**Delivery Process**

Receive order request

Validate order details

Prepare products for shipment

Generate shipping label

Handover package to courier

Track shipment

Notify customer about shipment

Start

Check product availability

**Order cancellation**

View order details

Notify customer about cancellation policy & why product cannot be cancelled

Initiate cancellation request

Process refund if applicable

Notify customer about cancellation

End

No

Check if product can be cancelled

Search Flow

|  |
| --- |
| Farmers successfully logs in &Lands on home screen |

|  |
| --- |
| Farmer Searches for a Product |

|  |
| --- |
| Product not Availableappears |

 Product doesn’t exist

|  |
| --- |
| Product List Appears |

 Product Exists