1. **Quarterly Audits:**
2. Quarterly audits are planned each phase by phase as:
3. Requirement Gathering phase
4. Requirement analysis phase
5. Design phase and Development phase
6. Testing phase
7. Requirement Gathering phase :

* Audit will check for if all the requirement is collected from the client as per their needs and check if completed or not .
* Checklist includes BRD template Elicitation results report Duplicate requirements report grouping of functionalities/features- Client signoff.

1. Requirement analysis Phase:

* Check if all the customer requirements are analysed and worked accordingly.
* Checklist includes UML Diagrams Business to Functional Requirements mapping Client Signoff RTM Document version control.

1. Design and Development phase:

* Audit will check if software has been designed according to client need.
* Audit will check if website has been created according to customer’s requirement.
* Checklist includes Utilization of Tools Documented evidence on client communication. Stakeholder MOM.
* Checklist includes creating detailed checklist of requirement, Creating timeline and task with list of deliverables and deadlines Meeting with Project development team.

1. Testing Phase:

* Developed application will be tested and verify before handed over to real-time user.
* If any error occurs then again handed over to development team for correction.
* Audit checklist will include Meeting with testers to check on possible outcome, Discussion with QA team on the details such as automation code, where to store the automation code and who will need access to it, who's running the tests; and writing test cases Meeting with QA team to identify where the tests will run.

1. Quarterly audit in order:

|  |  |
| --- | --- |
| Stage 1 | Requirement gathering 15 weeks (week 1 to week 15) |
| Completed | 10 weeks (week 1 to week 10) |
| Checklist |  |
|  | BRD template |
|  | Elicitation results reports |
|  | Duplicate requirements report |
|  | Grouping of features - client sign off |
|  | Email communication- To, CC, BCC |
|  |  |
| Stage 2 | Requirement Analysis Phase 13 weeks (weeks 16 to week 29) |
| Completed | 7 weeks (week 16 to week 23) |
| Checklist |  |
|  | UML diagrams |
|  | Business to functional requirements mapping |
|  | Client sign off - documents |
|  | RTM Document version control |
|  | Email communication- To , CC, BCC |
|  |  |
| Stage 3 | Design Phase - 10 weeks (week 30 to week 40) |
| Completed | 7 weeks (week 30 to week 37) |
| Checklist |  |
|  | Utilization of tools |
|  | Documented evidence on client communication |
|  | Stakeholder MOM |
|  | Email communication- To , CC, BCC |
|  |  |
| Stage 4 | Development Phase 30 weeks (week 40 to week 70) |
| Completed | 20 weeks (week 40 -to 60) |
| Checklist |  |
|  | JAD session reports |
|  | End user manual preparation diagram |
|  | BA and developer MOM |
|  | Email communication- To , CC, BCC |
|  |  |
| Stage 5 | Testing Phase 20 weeks (week 58- 78) |
| Completed | 20 weeks (week 58 - week 78) |
| Checklist |  |
|  | Test case summary |
|  | Training report to end users |
|  | Lesson learnt docs |
|  | Email communication- To , CC, BCC |

1. **Business Analyst (BA) Approach Strategy:**

The approach strategy ensures that the project meets stakeholder expectations and aligns with the defined objectives. BA strategy ensures clear communication, stakeholder engagement, and adherence to project timelines. The structured approach minimizes risks and ensures the delivery of a user-friendly platform.

1. **Elicitation Techniques:** Conduct interviews with the stakeholders (Mr. Henry, Mr. Pandu, Mr. Dooku, Peter, Kevin, Ben) to gather the requirements. Organize focus group sessions to understand the needs of the remote area farmers. Review existing systems and documents to gather additional information. Conduct surveys and questionnaires to gather a wider perspective.
2. **Stakeholder Analysis:** Conduct RACI (Responsible, Accountable, Consulted, Informed) or ILS (Involved, Lead, Support) analysis to determine the roles and responsibilities of each stakeholder. Identify the key stakeholders and prioritize their requirements. Establish effective communication channels with the stakeholders to keep them informed about the progress of the project.
3. **Documents:** Write a Requirements Document (RD) to outline the functional and non-functional requirements of the project. Create a Business Requirements Document (BRD) to provide a detailed description of the project's objectives, scope, and deliverables. Prepare a Project Charter to define the project's goals, deliverables, timeline, and budget. Develop a Use Case Document to describe the processes and workflows involved in the project.
4. **Sign Off:** Obtain sign-off from the stakeholders on the Requirements Document, Business Requirements Document, Project Charter, and Use Case Document. Ensure that the stakeholders understand and agree with the requirements, scope, and objectives of the project.
5. **Approvals**: Obtain the client's approval on the project deliverables, budget, timeline, and approach. Ensure that the client's expectations are aligned with the project's goals and objectives.
6. **Communication Channels:** Establish a regular communication schedule with the stakeholders to keep them informed about the project's progress. Create a communication plan to outline the channels and methods of communication. Schedule regular status meetings with the stakeholders to discuss the project's progress and address any issues or concerns.
7. **Change Requests:** Handle change requests in a structured and systematic manner. Evaluate the impact of each change request on the project's scope, timeline, and budget. Obtain approval from the stakeholders before implementing the change request.
8. **Progress Updates:** Keep the stakeholders informed about the project's progress through regular status reports and progress meetings. Highlight any risks or issues that need to be addressed. Provide regular progress updates to the stakeholders and seek their feedback.
9. **UAT Sign-off:** Conduct User Acceptance Testing (UAT) to validate the project's deliverables. Obtain sign-off from the client on the UAT results and the Project Acceptance Form. Ensure that the project meets the client's expectations and requirements.
10. **3-Tier Architecture:**

The 3-tier architecture is a software design model that divides an application into three logical layers: Presentation Layer, Application Layer, and Data Layer. Each layer is responsible for specific functionalities, ensuring separation of concerns, scalability, and maintainability.

1. **Layers in 3-Tier Architecture:**
2. Application layer.
3. Business logic layer.
4. Data layer.
5. **Application layer(Presentation layer):**

* It is the top most layer of the application and responsible for the user interface and interaction with the users.
* This tier contains components that handles user inputs, display output and perform other user related tasks.
* Implemented using web technologies such as HTML,CSS and java script.

1. **Business logic layer:**

* Processes business logic and handles communication between the Presentation and Data layers.
* This tier contains components that handle business rules, perform computations and communication with other components in the same tier and data storage tier.
* Implemented using programming language such as java, python.

1. **Data Layer:**

* Manages data storage and retrieval.
* This tier contains components that manage the applications data, file system and perform data related tasks.
* Implemented using data base such as SQL.

1. For online agriculture store 3 tier process:
2. Application Layer:

* Farmers access the platform via a mobile app to browse products.

1. Business logic Layer:

* The server processes a search query, applies filters, and retrieves product details.

1. Data Layer:

* The database fetches seed and fertilizer details for display in the app

1. **Illustrate 3-tier architecture:**



1. **BA Approach Strategy for Framing Questions**:  
     
   Before framing questions to ask stakeholders, a Business Analyst (BA) should consider several frameworks and strategies to ensure comprehensive, clear, and actionable information is gathered.
2. **5W 1H:**

This framework helps in ensuring that all necessary aspects of a question are covered. It is an Question tool where we write down or frame a question.

1. **Who:** Identifies the stakeholders involved or affected.

**Example:** Who will be using the online agriculture product store?

1. **What:** Defines what is being discussed or required.

**Example:** What functionalities are most important for the farmers?

1. **When:** Specifies the timeline or deadlines.

**Example:** When do you need the application to be ready?

1. **Where:** Determines the location or context.

**Example:** Where will the application be used (e.g., on farms, in offices)?

1. **Why:** Understands the reasons or goals behind requirements.

**Example:** Why do you need a specific feature (e.g., real-time inventory updates)?

1. **How:** Explores the methods or processes.

**Example:** How do you currently procure fertilizers, and how do you envision this changing with the online store?

1. **SMART:**

Ensures that the requirements and questions are Specific, Measurable, Achievable, Relevant, and Time-bound. It is a Validation tool.

1. **Specific:** Be clear and precise.

**Example:** What specific types of fertilizers do you use regularly?

1. **Measurable:** Ensure that the requirement can be quantified or assessed.

**Example:** How many transactions do you expect to process monthly?

1. **Achievable:** Check that the requirement is realistic.

**Example:** Can the farmers realistically use a mobile app given their current technology access?

1. **Relevant:** Make sure the requirement is pertinent to the project goals.

**Example:** Are bulk order discounts relevant to your purchasing process?

1. **Time-bound:** Define the timeframe for implementation or achievement.

**Example:** When do you need the bulk order feature implemented by?

1. **RACI:**

Identifies the roles and responsibilities in the project.

1. **Responsible:** Who will carry out the task?

**Example:** Who is responsible for maintaining the product catalog?

1. **Accountable:** Who is ultimately accountable for the task?

**Example:** Who will approve the final design of the user interface?

1. **Consulted:** Who needs to be consulted before a decision or action?

**Example:** Who should we consult to validate the pesticide procurement process?

1. **Informed:** Who needs to be informed about decisions or actions?

**Example:** Who needs to be informed about new features added to the application?

1. **3-Tier Architecture:**

Consider the structure and flow of data across the presentation, business logic, and data tiers.

1. **Presentation Tier:** How will the user interface be designed and what interactions will be required?

**Example:** What pages and features do farmers need in the mobile app?

1. **Business Logic Tier:** What business rules and processes will the application enforce?

**Example:** What are the rules for bulk purchasing and discount eligibility?

1. **Data Tier:** What data needs to be stored and retrieved, and how will this data be managed?

**Example:** What data fields are necessary for tracking pesticide inventory?

1. **Use Cases and Use Case Specifications:**

Define the functional requirements and interactions of the system.

1. **Use Cases:** Describe specific scenarios in which the system interacts with users or other systems.

**Example:** Use Case: "Purchase Fertilizer" – A farmer selects a fertilizer, adds it to the cart, and completes the purchase.

1. **Use Case Specifications:** Detail the basic flow (positive flow) and alternative flows (negative flow).

* **Basic Flow (Positive):**
* Farmer logs into the application.
* Searches for fertilizer.
* Adds fertilizer to cart.
* Proceeds to checkout.
* Confirms purchase.
* **Alternative Flow (Negative):**
* Farmer logs into the application.
* Searches for fertilizer.
* Adds fertilizer to cart.
* Checkout fails due to payment error.
* Farmer receives an error message and retries the payment or contacts support.

1. **Activity Diagrams, Models, and Page Designs:**

Visual tools and documentation to illustrate the workflow and design of the application.

1. **Activity Diagrams:** Show the flow of activities and decisions in a process.

**Example:** Diagram depicting the steps a farmer takes to search for and purchase a product.

1. **Models:** Conceptual models like ER diagrams, class diagrams to represent data and system structure.

**Example:** ER diagram showing relationships between users, products, and orders.

1. **Page Designs:** Wireframes and mockups of the user interface.

**Example:** Mockup of the product search and checkout pages.

1. **Elicitation Techniques:**

Requirement analysis is to dig the information from the stakeholders before starting the project to gather as much as information possible from the stakeholders. Requirement elicitation serves as the foundation in documenting the requirements before starting a project.

There are 11 elicitation techniques to apply accordingly: -

1. **Brainstorming:**

* This technique is used to generate new ideas and find a solution for a specific issue.
* The members included for brainstorming can be domain experts, subject matter experts.
* Multiple ideas and information give us a repository of knowledge and we can choose from different ideas.

1. **Document Analysis:**

* During this step of the requirements elicitation process, business analysts review existing documentation at hand, with the intent of identifying requirements for changes or improvements.
* Examples of document analysis sources include pre-existing project plans, system specifications, process documentation, market research dossiers, customer feedback, meeting minutes, and user manuals.
* Document analysis is performed before scheduling more in-depth requirements elicitation sessions or interviews with stakeholders.

1. **Reverse engineering:**

* In this Technique, any outdated documentation in an existing system, can be reversed to understand what the system does.
* His is an elicitation technique that can extract implemented requirements from the system.
* There are two types of reverse engineering techniques.

1. **Black box reverse engineering:**

The system is studied without examining its internal structure (function and composition of software).

1. **White box reverse engineering:**

The inner workings of the system are studied (analysing and understanding of software code).

1. **Focus Group:**

* By using a focus group, you can get information about a product, service from a group.
* The Focus group includes subject matter experts. The objective of this group is to discuss the top issue and provide information. A moderator manages this session.

1. **Observation:**

* An excellent elicitation technique helps understand requirements based on observations related to process flows and work environments of stakeholders.
* Observation requires a business analyst to go and look at the work – for example, observing the business processes in scope of the project.
* The elicitation technique observation is an effective means of understanding how a user does their job by assessing their work environment.

1. **Workshops**:

* Workshops comprise a group of users or stakeholders working together to identify requirements.
* A requirement workshop is a structured way to capture requirements.
* Workshops are used to scope, discover, define, and prioritize requirements for the proposed system.

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

* This technique is more process-oriented and formal as compared to other techniques.
* These are structured meetings involving end-users, PMs, SMEs.
* This is used to define, clarify, and complete requirements.

1. **Interviews:**

* An interview is a systematic approach to elicit information from a person or group of people.
* This is the most common technique used for requirement elicitation.
* Interview techniques should be used for building strong relationships between business analysts and stakeholders.
* In this technique, the interviewer directs the question to stakeholders to obtain information. One to one interview is the most commonly used technique.

1. **Prototyping:**

* Prototyping is used to identify missing or unspecified requirements.
* In this technique, frequent demos are given to the client by creating the prototypes so that client can get an idea of how the product will look like.
* Prototypes can be used to create a mock-up of sites, and describe the process using diagrams.

1. **Questionnaire & Surveys:**

* For Survey/Questionnaire, a set of questions is given to stakeholders to quantify their thoughts.
* After collecting the responses from stakeholders, data is analysed to identify the area of interest of stakeholders. Questions should be based on high priority risks.

1. **Use case specs:**

* Use cases are an effective and widely used technique for eliciting software requirements.
* The use-case approach focuses on the goals that users have with a system, rather than emphasizing system functionality.
* This technique combines text and pictures to provide a better understanding of the requirements.
* The use cases describe the ‘what’, of a system and not ‘how’. Hence, they only give a functional view of the system.
* The components of the use case design include three major things – Actor, Use cases, use case diagram.

1. **Elicitation Techniques:**

For the Online Agriculture Products Store project, multiple elicitation techniques are suitable. However, selecting the best techniques depends on the specific requirements; stakeholders involved, and project goals.

1. **Prototyping (Best for Usability Validation)**
2. **Why:**
3. Provides stakeholders (like farmers and manufacturers) with a visual representation of the application, making it easier for them to understand and provide feedback.
4. Especially helpful for non-technical stakeholders who may struggle to articulate their requirements clearly.
5. **What to Prototype:**
6. Login/Sign-up workflows.
7. Product catalog with filters and search options.
8. Payment gateway options (UI flow for COD, UPI, and card payments).
9. Delivery tracking interface.
10. **Outcome:**  
    Clear, user-approved designs that minimize ambiguity and set accurate expectations.
11. **Use Case Specifications (Best for Functional Clarity)**
12. **Why:**
13. Captures interactions between users (farmers, manufacturers) and the system in a structured format.
14. Provides a clear understanding of functional requirements and expected system behavior.
15. **Example Use Cases:**
    1. "Search for a product."
    2. "Login/Sign-up."
    3. "Place an order and make a payment."
    4. "Track delivery."
16. **Outcome:**  
    Detailed documentation that aligns developers and testers with the business requirements.
17. **Brainstorming (Best for Ideation and Innovation)**
18. **Why:**
    1. Encourages creative input from key stakeholders like Peter, Kevin, Ben, and Mr. Henry.
    2. Helps identify additional features or enhancements, such as promotional offers or farmer-specific dashboards.
19. **Topics to Brainstorm:**
    1. Features to improve the farmer's user experience.
    2. Methods to streamline product uploads for manufacturers.
    3. Effective ways to send order notifications and delivery updates.
20. **Outcome:**  
    A rich set of ideas to refine and expand the application's scope.
21. **Document Analysis (Best for Understanding Context and Constraints)**
22. **Why:**
    1. Reviews existing product catalogs, user workflows, and industry guidelines for fertilizers, seeds, and pesticides.
    2. Identifies mandatory fields for manufacturers and ensures compliance with agricultural regulations.
23. **What to Analyze:**
    1. Current formats used by manufacturers for product details.
    2. Industry norms for fertilizers and pesticide labeling.
    3. Existing farmer and manufacturer workflows.
24. **Outcome:**  
    A solid understanding of the domain to ensure the application aligns with stakeholder and regulatory requirements.
25. **Best Technique for Each Stakeholder Group**

| **Stakeholder Group** | **Best Elicitation Technique** | **Reason** |
| --- | --- | --- |
| Farmers (Peter, Kevin) | **Prototyping, Brainstorming** | Easy visualization and ideation for usability improvements. |
| Manufacturers (Ben, etc.) | **Document Analysis, Prototyping** | Ensures clarity in product uploads and compliance with guidelines. |
| Mr. Henry (Client) | **Use Case Specs, Prototyping** | Provides structured scenarios and visual feedback for approval. |

1. **Recommendation:**
2. Start with Prototyping: Create mock-ups of critical features and gather feedback from farmers and manufacturers.
3. Develop Use Case Specs: Write detailed use cases for login, product search, payment, and order tracking.
4. Conduct a Brainstorming Workshop: Explore additional features and workflows with stakeholders.
5. Perform Document Analysis: Review existing product documentation and industry standards.
6. **10 Business Requirements:**
7. BR001- The platform should have a product catalogue that includes all fertilizers ,seeds and pesticides from different manufacturer and vendors.
8. BR002- The platform should allow farmers to search for products by name, category and brand.
9. BR003- The platform should have a login feature for all users including farmers, manufacturers and vendors.
10. BR004- the platform should allow new users to create an account by submitting their E- mail id and creating a secure PW.
11. BR005- The platform should have a user friendly interfaces and easy navigation for a better user experience.
12. BR006- The platform should have a payment gateway that includes COD, credit/debit cards and UPI options.
13. BR007- The platform should send E-mail confirmation regarding order status to users.
14. BR008- The platform should have a delivery tracker to track the whereabouts of the order.
15. BR009- The platform should have scalable to accommodate future growth and expansion.
16. BR010- The platform should have a secure infrastructure to protect user data and prevent data breaches.
17. **Assumptions:**
18. The project for an e-commerce platform for fertilizers, seeds and pesticides targeted towards farmers.
19. The platform will have a product catalogue and will allow users to search feertilizers,seeds and pesticides.
20. This platform will have login feature for farmers, Manufacturers and vendors and allow new users to create account by submitting their E-mail id and creating a secure password.
21. The platform will have a payment gateway that includes COD, credit/debit card and UPI options.
22. The platform will send E-mail confirmations regarding order status and delivery tracker to track product.
23. The platform will have user-friendly interfaces and easy navigation for a better user experience.
24. **This project Requirements Priority:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement** | **Description** | **Priority** | **Justification** |
| **BR001** | Product catalogue including fertilizers, seeds, and pesticides from various manufacturers/vendors. | **10** | Core functionality; without this, the platform cannot fulfil its purpose. |
| **BR002** | Search feature by name, category, and brand. | **9** | Enhances usability and ensures users can find products quickly, crucial for user experience. |
| **BR003** | Login feature for all users, including farmers, manufacturers, and vendors. | **9** | Ensures user-specific access and management, foundational for platform security and operations. |
| **BR004** | Account creation for new users by submitting email ID and creating a secure password. | **8** | Essential for on boarding new users but can follow after initial core functionalities are built. |
| **BR005** | User-friendly interface and easy navigation. | **10** | Critical for adoption, especially given farmers may have limited technical skills. |
| **BR006** | Payment gateway with COD, credit/debit cards, and UPI options. | **8** | Important for completing transactions; COD can serve as a fall back if online payment is delayed. |
| **BR007** | Email confirmation regarding order status. | **7** | Nice-to-have; adds convenience but can be implemented after core order functionalities. |
| **BR008** | Delivery tracker to track the whereabouts of the order. | **6** | Enhances user experience but is less critical during the initial implementation phase. |
| **BR009** | Scalable platform to accommodate future growth. | **9** | Vital for long-term sustainability and should be considered during the architectural design phase. |
| **BR010** | Secure infrastructure to protect user data and prevent data breaches. | **10** | Critical to build trust and comply with data protection laws. |

1. **Use Case Diagram:**



1. **Use Case Specs:**

|  |  |
| --- | --- |
| Use Case ID | UC001 |
| Use Case Name | Buying a product |
| Actors | Customer, Seller |
| Description | This use case describes how users can make purchase via App |
| Pre - Condition | User should have been registered into the application |
| Post - Condition | Successfully able to login the Account |
| Basic Flow | Step 1: User create and account and login.  Step 2: User search for a product from the search bar.  Step 3: same product and related product option from different manufacture will be appeared on the screen.  Step 4: User select one product, selects the size and quantity of the product and click on "buy now option".  Step 5: System will take to another page, where total price calculation will be displayed along with the products added to cart.  Step 6: User click on "Place order button".  Step 7: User need to to choose the mode of the payment.  Step 8: User need to enter the banking details and make payment.  Step 9: User will receive order confirmation on email along with the tracking id. Step 10: Basic flow end here. |
| Alternate Flows | Step 1: User is not able to login and redirected to forgot "Username/Password" page.  Step 2: If you user is not able to get the right information, he can request for a call from customer care.  Step 3: once he get connected with the customer care he will explain the issue to the customer care representative,  Step 4: Customer care will send a link to reset password to his email account.  Step 5: User will go to that link and system will take to new page, where user will be able to change new password  Step 6: User will be put a new password.  Step 7: System will ask to reconfirm the password.  Step 8: User will be able to login the account now. |
| Exceptions | If internet connectivity lost while doing this use case, system displays " check with your internet connectivity " |
| Frequency of use | High |
| Assumptions | It is assumed that the customer is registered.  It is assumed that the customer has the computer knowledge.  It is assumed that the customer has a suitable device to use the APP. |

|  |  |
| --- | --- |
| Use Case ID | UC002 |
| Use Case Name | Exchange of product |
| Actors | Customer, Seller |
| Description | This use case describes how users can exchange a purchased product. |
| Pre - Condition | User should have purchased a product before in order to make a exchange. |
| Post - Condition | Successfully able to exchange the product |
| Basic Flow | Step 1: User login to account via credentials.  Step 2: User click on "Account" .  Step 3: System takes to different page with other details.  Step 4: User select option "Exchange" among those options.  Step 5: System will take to another page, where recently ordered products will be displayed on the screen.  Step 6: User has to choose the product which he wants to exchange.  Step 7: User will get another option where he will be asked- "different size in same product" or "want to buy another product"  Step 8: User need to choose one of the options and take action according to chosen option. Step 9: Once the product is chosen, user will have to click on button "Exchange".  Step 10: User will get the confirmation on email. |
| Alternate Flows | Step 1: User couldn't find the size which he wanted.  Step 2: User call customer care agent to get a solution  Step 3: Agent suggested to wait for the size to be restocked and gave a tentative date or go for similar products.  Step 4: Agent shares the link of similar products to the registered email of the customer.  Step 5: User choose the product  Step 6: User will be put a new password.  Step 7: System will ask to reconfirm the password. Step 8: User will be able to login the account now. |
| Exceptions | If internet connectivity lost while doing this use case, system displays " check with your internet connectivity " |
| Frequency of use | High |
| Assumptions | It is assumed that the customer is registered.  It is assumed that the customer has the computer knowledge.  It is assumed that the customer has a suitable device to use the APP. |

|  |  |
| --- | --- |
| Use Case ID | UC003 |
| Use Case Name | Return of product |
| Actors | Customer, Seller |
| Description | This use case describes how users can return a purchased product. |
| Pre - Condition | User should have purchased a product before in order to make a return. |
| Post - Condition | Successfully able to exchange the product |
| Basic Flow | Step 1: User login to account via credentials.  Step 2: User click on "Account"  Step 3: System takes to different page with other details.  Step 4: User select option "Return" among those options.  Step 5: System will take to another page, where recently ordered products will be displayed on the screen.  Step 6: User has to choose the product which he wants to return. Step 7: User will get another option where he will be asked to provide the bank account number for amount of the returned product to be credited.  Step 8: User need to enter the account number and submit.  Step 9: User will get the confirmation on email. |
| Alternate Flows | Step 1: User didn't get the amount in his account within the TAT.  Step 2: User call customer care agent to ask payment status.  Step 3: Payment was stuck due to a technical glitch.  Step 4: User was shared complaint form to be filled.  Step 5: Once form submitted, user received another TAT on the email of amount to be credited.  Step 6: User get the payment id in registered email |
| Exceptions | User put the incorrect bank account. |
| Frequency of use | High |
| Assumptions | It is assumed that the customer has a valid bank account number.  It is assumed that the customer has good internet connectivity.  It is assumed that the customer has computer knowledge. |

|  |  |
| --- | --- |
| Use Case ID | UC004 |
| Use Case Name | Update the delivery address |
| Actors | Customer, Seller |
| Description | This use case describes how users can update address. |
| Pre - Condition | User should have a valid deliverable postal address. |
| Post - Condition | Successfully able to update address. |
| Basic Flow | Step 1: User login to account via credentials.  Step 2: User click on "Account".  Step 3: System takes to different page with other details.  Step 4: User select option "Update" among those options.  Step 5: System will take to another page, where mandatory fields like; Apt number, landmark, pin code, city name will be displayed and has to be field.  Step 6: User need to click on "submit" button.  Step 7: User can use the updated address for products delivery. |
| Alternate Flows | Step 1: User is not able to update the address.  Step 2: User will refresh the page.  Step 3: User gets error again while submitting details.  Step 4: User use live chat box  Step 5: User is asked to not leave blank any star marked field.  Step 6: after updating all mandatory field, address was successfully submitted. |
| Exceptions | User put the incorrect address details like; pin exceeds the maximum number of digits |
| Frequency of use | High |
| Assumptions | It is assumed that the customer has a valid postal address It is assumed that the customer has good internet connectivity.  It is assumed that the customer has computer knowledge.  It is assumed, customer understands, what details has to be put in every field. |

|  |  |
| --- | --- |
| Use Case ID | UC005 |
| Use Case Name | Update the new contact number |
| Actors | Customer, Seller |
| Description | This use case describes how users can update/ change new phone number |
| Pre - Condition | User should have a new contact number. |
| Post - Condition | Successfully able to change contact number. |
| Basic Flow | Step 1: User login to account via credentials. Step 2: User click on "Account" . Step 3: System takes to different page with other details. Step 4: User select option "Manage your Account" among those options. Step 5: System will take to another page, where personal details will be displayed. 6: User has to click Mobile number Step 7: User will get a red popup button "CHANGE". Step 8: OTP will be sent to existing updated number Step 9: once number is verified with the OTP user put. User can update new contact number. Step 10: New contact number is successfully updated. |
| Alternate Flows | Step 1: User didn't get the OTP in registered existing number. Step 2: User restarts the phone. Step 3: User raised a ticket with the customer care Step 4: User was shared issue ticket number in the registered email. Step 5: Issue got fixed with the help of support team Step 6: contact number is successfully changed. |
| Exceptions | User put the incorrect phone number. |
| Frequency of use | Low |
| Assumptions | It is assumed that the customer has a valid phone number. It is assumed that the customer has good phone network to receive OTP. It is assumed that the customer has checked the message inbox for OTP. |

1. **Activity Diagram:**

* Activity diagram are one of the 5 diagram in the UML for modelling the dynamic aspects of the system. An activity is essentially a flowchart showing the flow of control from activity to activity.
* In this online case study, we use certain elements for the use case model.
* Control flow an arrow showing the direction of the workflow.
* Activity indicates a step in the process it is a unit of work done by the system or a consistent achieved.
* Connector is a trigger attached to control flow when the guard condition is true.









