Online Agriculture Products Store

Mr. Henry, after being successful as a businessman and has become one of the wealthiest persons in the city. Now, Mr. Henry wants to help others to fulfil their dreams. One day, Mr. Henry went to meet his childhood friends Peter, Kevin and Ben. They live in a remote village and do farming. Mr. Henry asked his friends if they are facing any difficulties in their day-to-day work. Peter told Mr. Henry that he is facing difficulties in procuring fertilizers which are very important for farm. Kevin said that he is also facing the same problem in-case of buying seeds for farming certain crops. Ben raised his concern on lack of pesticides which could help in greatly reducing pests in crops. After listening to all his friends’ problems, Mr. Henry thought that this is a crucial problem faced not only by his friends but also by so many other farmers. So, Mr. Henry decided to make an online agriculture product store to facilitate remote area farmers to buy agriculture products. Through this Online Web / mobile Application, Farmers and Companies (Fertilizers, seeds and pesticides manufacturing Companies) can communicate directly with each other. The main purpose to build this online store is to facilitate farmers to buy seeds, pesticides, and fertilizers from anywhere through internet connectivity. Since new users are involved, Application should be user friendly. This new application should be able to accept the product (fertilizers, seeds, pesticides) details from the manufacturers and should be able to display them to the Farmers. Farmers will browse through these products and select the products what they need and request to buy them and deliver them to farmers location. Mr. Henry has given this project through his Company SOONY. In SOONY Company, Mr Pandu is Financial Head and Mr Dooku is Project Coordinator. Mr. Henry , Mr Pandu , and Mr Dooku formed one Committee and gave this project to APT IT SOLUTIONS company for Budget 2 Crores INR and 18 months Duration under CSR initiative. Peter, Kevin and Ben are helping the Committee and can be considered as Stakeholders share requirements for the Project. Mr Karthik is the Delivery Head in APT IT SOLUTIONS company and he reached out to Mr Henry through his connects and Bagged this project. APT IT SOLUTIONS company have Talent pool Available for this Project. Mr Vandanam is project Manager, Ms. Juhi is Senior Java Developer, Mr Teyson, Ms Lucie, Mr Tucker, Mr Bravo are Java Developers. Network Admin is Mr Mike and DB Admin is John. Mr Jason and Ms Alekya are the Tester. And you joined this team as a BA.

**Your Team**

**Project Manager - Mr Vandanam**

**Senior Java Developer - Ms. Juhi**

**Java Developers - Mr Teyson, Ms Lucie, Mr Tucker, Mr Bravo**

**Network Admin - Mr Mike**

**DB Admin - Mr John.**

**Testers - Mr Jason and Ms Alekya**

**BA – You**

Fertilizers, seeds, pesticides details from the manufacturers and should be able to display them to the Farmers.

To gather the business requirements from the client, you went to SOONY and met Mr. Henry. When Mr. Henry was asked about the project and what are they expecting from the project, Mr. Henry stated that he is expecting to have a login for all its users (fertilizers, seeds, pesticides manufacturers and Farmers) , a product catalogue of fertilizers, seeds, pesticides, a search option to search for products, payment process, and delivery tracking.

After doing the stakeholder analysis, you have found out that Peter, Kevin, Ben are the key stakeholders and you have scheduled an appointment to meet them. After meeting with them and trying to gather the stakeholder requirements, Kevin said that, a Farmer should be able to browse through the products catalogue once they visit the website and need to have a search option so that they can search for any product they need. Peter said that, if a farmer wants to buy any product or add them to buy-later list, they need to login first using their email id and password. If it is a new user, then they can create a new account by submitting their email ID and creating a secure password. Ben added saying that, Farmers needs to have an easy-to-use payment gateway which should include cash-on-delivery (COD), Credit/Debit card and UPI options so that the user’s experience should be better. Kevin mentioned that, a user gets an email confirmation regarding their order status. A delivery tracker to track the whereabouts of their order.

**Identify Business Requirements (which includes Stakeholder Requirements)**

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

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

**Answer:**

Based on the information gathered during the meetings with Mr. Henry, Peter, Kevin, and Ben, here are the **Business Requirements (BR)** that address both the functional needs of the system and the expectations of the stakeholders:

* **Business Requirements (BR)**
* **BR001 – Product Search and Browse**

**Description**: Farmers should be able to search for available products (fertilizers, seeds, pesticides) within the application and they should also be able to browse through the entire product catalogue.

**Stakeholder Requirement**: Kevin emphasized that farmers need the ability to search for products. This aligns with the user experience of making it easy for farmers to find specific products quickly.

* **BR002 – Manufacturer Product Upload and Display**

**Description**: Manufacturers should be able to upload their product details, including descriptions, images, prices and availability and the system should display these products to farmers.

**Stakeholder Requirement**: This was directly mentioned by Mr. Henry, who stated the need for manufacturers to upload product information so that it can be displayed to the farmers. This helps ensure that farmers have access to the latest products available in the market.

* **BR003 – User Login and Registration**

**Description**: Farmers must log in to their accounts to add products to their cart or buy-later list. If the farmer is a new user, they must be able to create an account using their email address and a secure password.

**Stakeholder Requirement**: Peter emphasized the necessity for farmers to log in before making any purchase or adding products to their list. This helps in managing orders and providing a personalized experience.

* **BR004 – Easy Payment Gateway**

**Description**: The application should offer a simple, secure and user-friendly payment gateway, allowing farmers to choose from various payment methods like cash-on-delivery (COD), credit/debit cards and UPI for transactions.

**Stakeholder Requirement**: Ben expressed the need for an easy-to-use payment system to enhance the user experience, ensuring farmers can choose the payment method most convenient for them.

* **BR005 – Order Confirmation and Email Notification**

**Description**: Once a farmer places an order, they should receive an email confirmation about the order status, including the items ordered and the expected delivery date.

**Stakeholder Requirement**: Kevin mentioned that farmers should receive email confirmations to ensure they are aware of their order's status and can track it effectively.

* **BR006 – Delivery Tracking**

**Description**: The system should include a delivery tracker, allowing farmers to track the whereabouts of their orders in real-time.

**Stakeholder Requirement**: Kevin highlighted the importance of having a delivery tracking system so that farmers can know the status of their orders at any time.

* **Summary of Stakeholder Requirements**
* **Kevin**:

Farmers should be able to search for and browse products in the catalogue.

Farmers should receive email confirmation regarding their order status.

Farmers should be able to track their orders with a delivery tracker.

* **Peter**:

Farmers should be required to log in before adding products to their cart or buy-later list.

Farmers should have the option to create an account if they are new users.

* **Ben**:

The payment gateway should include COD, credit/debit card, and UPI payment options.

These **Business Requirements** and **Stakeholder Requirements** clearly outline the essential features and functionalities for the online agriculture product store that will serve both farmers and manufacturers. These requirements ensure that the application will meet user needs, support business operations, and enhance user experience.

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

**Answer**

Here are 10 potential business requirements for the online agriculture product store that Mr. Henry wants to build, along with a few assumptions:

**Assumptions:**

* The application will be accessible on both web and mobile platforms.
* The application will have an easy-to-use interface, as users may not be very familiar with technology.
* The platform will support multiple languages to cater to a wider audience, especially in remote areas.
* Payments can be done via multiple methods, including online and offline (COD).
* Delivery services will be included for product shipping and it will support tracking of orders.
* The manufacturers (companies) will be verified before they can list their products.
* The platform will also include a support/help feature for users who face technical issues.
* Inventory management will be in place to track product availability.
* The system will have security features for protecting sensitive data (user details, financial transactions etc.).

**Business Requirements:**

1. **User Registration and Authentication:**
   * The system must allow farmers and manufacturers to register and log in securely using email, mobile number or social media accounts.
   * The system should support role-based authentication (farmer, manufacturer, admin).
2. **Product Listing by Manufacturers:**
   * Manufacturers must be able to list their products (seeds, fertilizers, pesticides) on the platform by providing essential details such as product name, description, pricing, quantity and delivery details.
   * The system must allow manufacturers to upload product images to enhance visibility.
3. **Product Search and Browsing for Farmers:**
   * Farmers must be able to search for products based on categories (seeds, fertilizers, pesticides), price range and type of crop.
   * Filters should be available to narrow down product selection by manufacturer, product features, price range or ratings.
4. **Product Details and Comparison:**
   * Farmers must be able to view detailed information about the products, including description, usage instructions, ingredients and reviews from other farmers.
   * Farmers should be able to compare similar products in terms of features and pricing.
5. **Shopping Cart and Order Placement:**
   * Farmers should be able to add products to the shopping cart and proceed to checkout.
   * The application should allow farmers to review the products in their cart, adjust quantities and apply any discount coupons before finalizing the order.
6. **Order Management and Delivery:**
   * Farmers must receive an order confirmation notification, including an estimated delivery date.
   * The system must allow for tracking of order status (e.g., processing, shipped, out for delivery) and provide updates to farmers.
   * Delivery charges should be calculated and displayed at the time of checkout.
7. **Payment Integration:**
   * The system should integrate with various payment gateways to allow farmers to make payments through credit/debit cards, UPI and cash on delivery (COD).
   * The system must support secure payment processing to protect user data.
8. **Feedback and Rating System:**
   * Farmers should be able to rate and review products based on their experience.
   * The system should display average ratings for each product, allowing new customers to make informed decisions.
9. **Manufacturer and Product Verification:**
   * The platform must verify the authenticity of manufacturers before they can list their products. This could involve checking licenses, certifications and other legal documents.
   * Manufacturers should be able to update product information, but changes should be reviewed and approved by an admin before being displayed.
10. **Customer Support and Help Desk:**
    * The application must provide a customer support feature for farmers to report issues or get assistance with order tracking, product queries or payments.
    * The system should allow for both live chat and email support, ensuring that users can contact the support team easily.

These requirements will provide the foundation for developing a user-friendly and efficient online agriculture product store.

**List your assumptions.**

Here are the assumptions for the project based on the context provided:

* **Multi-Platform Access:**
  + The application will be available on both web and mobile platforms to cater to a larger audience, especially in remote areas where mobile phones are more likely used since farmers may not have access to desktop computers or laptops.
* **User-Friendly Interface:**
  + The application will be designed to be simple and intuitive, as many farmers might not be tech-savvy and would require a straightforward user experience.
* **Multi-Language Support:**
  + The application will offer support for multiple languages to ensure accessibility for farmers from different regions of the country.
* **Payment Methods:**
  + The application will support both online and offline payment methods, including credit/debit cards, UPI and cash on delivery (COD), to accommodate farmers with varying access to digital payment systems.
* **Product Delivery:**
  + The platform will include delivery services to send the purchased products directly to the farmer's location, with order tracking available for farmers.
* **Manufacturer Verification:**
  + Manufacturers (fertilizer, seed and pesticide companies) must go through a verification process before they are allowed to list their products on the platform to ensure legitimacy and quality of products.
* **Inventory Management:**
  + The platform will include an inventory management system that will track the availability of products in real-time, ensuring that farmers can only order products that are in stock.
* **Customer Support:**
  + A helpdesk feature will be available to assist farmers with any technical issues or inquiries they may have regarding the application or their orders.
* **Security Features:**
  + The application will have reliable security measures in place to protect sensitive information, such as personal details, payment information and order history.
* **Order Tracking:**
  + Farmers will be able to track the status of their orders from placement to delivery, ensuring they are informed at each stage of the process.
* **Discounts and Offers:**
  + The platform may offer discounts, promotions or loyalty programs to attract more farmers and encourage repeated purchases.
* **Mobile-Friendly:**
  + The mobile version of the application will be optimized for low-end smartphones, considering that farmers in remote areas may not always have access to high-end devices.
* **Internet Connectivity:**
  + The platform will assume that users have access to the internet, though it will be optimized for lower bandwidth areas where internet speeds may be slower.
* **Product Catalogue Management:**
  + Manufacturers will be responsible for updating product details, but changes will be subject to approval by administrators to ensure product accuracy and consistency.
* **Legal Compliance:**
  + The platform will comply with local agricultural regulations, including certifications and licenses required for selling agricultural products.
* **Product Quality and Certifications:**
* **Product Quality** ensures the product performs well and is safe for use, while **Certifications** provide legal assurance that the product meets recognized standards like Key Certifications Relevant to Agricultural Products such ISO, Organic Certifications, Pest Control Certifications, Global Good Agricultural Practice Certifications (Global GAP), FSSAI, Seed Certifications from Indian Council of Agricultural Research (ICAR) etc.

These assumptions help define the scope and limitations of the project, ensuring that the final product meets the needs of the farmers and manufacturers while adhering to practical considerations.

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

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | 8 |
| BR002 | Manufacturers  upload their Products | Manufacturers should be able to upload and display their products in the application | 8 |
| BR003 | User Login and Registration | Farmers must log in to their accounts to add products to their cart or buy-later list. If the farmer is a new user, they must be able to create an account using their email address and a secure password. | 10 |
| BR004 | Easy Payment Gateway | The application should offer a simple, secure and user-friendly payment gateway, allowing farmers to choose from various payment methods like cash-on-delivery (COD), credit/debit cards and UPI for transactions. | 10 |
| BR005 | Order Confirmation and  Email Notification | Once a farmer places an order, they should receive an email confirmation about the order status, including the items ordered and the expected delivery date. | 9 |
| BR006 | Delivery Tracking | The system should include a delivery tracker, allowing farmers to track the whereabouts of their orders in real-time. | 9 |

**Once the requirements are finalized, as a business analyst, one of the major roles is to act as a liaison between the client and the project team. To gather the requirements correctly from the client side and then to deliver those requirements to the project team in a way they understand. To make the project team understand the requirements, you need to convert those requirements into UML diagrams and screen mock-ups.**

**You will draw use case diagram**

**Prepare use case specs for all use cases**

**And you will all Activity diagrams required**

**1. Identify minimum 20 functional requirements.**

**Example:**

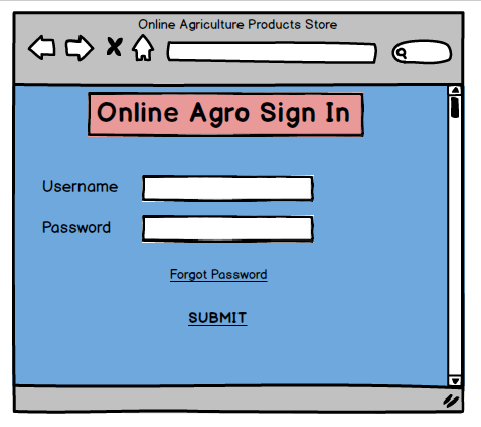
**Functional requirement: When an order is fulfilled, the local printer shall print a packing slip.**

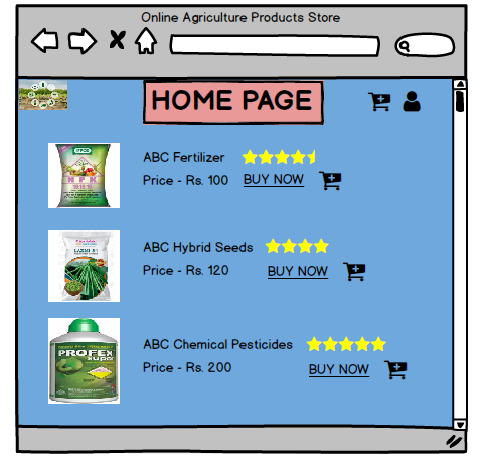
**Non-Functional Requirement: Packing slips shall be printed on both sides of 4”x 6” white paper, the standard size for packing slips used by local printers.**

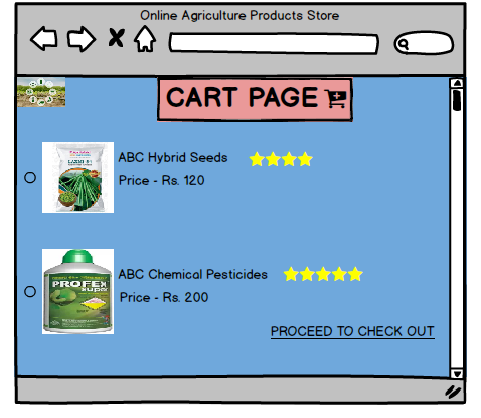
|  |  |  |  |
| --- | --- | --- | --- |
| **Req ID** | **Req Name** | **Req Description** | **Priority** |
| FR0001 | Farmer Registration | Farmers should be able to register with the application | 8 |
| FR0002 | Farmer Search for Products | Farmers should be able to search for available products in fertilizers, seeds, pesticides | 8 |
| FR0003 | No matching Product | Farmer should be notified if the searched product is not found | 8 |
| FR0004 | Product Selection | Farmer should be able to select the Product | 9 |
| FR0005 | Out of Stock | Farmer should be notified if the product is out of stock | 9 |
| FR0006 | Filter | Farmer should be able to filter and select the product as per Brand and Price | 8 |
| FR0007 | Similar Products | Farmer should be able to select similar products, if searched product is out of Stock | 8 |
| FR0008 | Bought Together | Once the product is finalized, farmer should be suggested with related products as a package with the price | 7 |
| FR0009 | Add to Cart | Farmer should be able to Add the product to the Cart for purchase | 8 |
| FR0010 | Wishlist | Farmer should have an option to Add the product into Wishlist. | 7 |
| FR0011 | Save it for Later | Farmer should be able to see the option for Save it Later, to purchase later | 8 |
| FR0012 | Delivery Address | Farmer should be able to select the Delivery address to deliver the product | 8 |
| FR0013 | Payment Options | Once the delivery address is selected, Application should show the Payment Options to the Farmer for purchasing the product. | 10 |
| FR0014 | Payment Confirmation | Farmer should receive the Payment Confirmation email and SMS | 9 |
| FR0015 | Order Confirmation | Farmer should receive the Order Confirmation email and SMS | 10 |
| FR0016 | Expected Delivery Date | Farmer should also see the Expected Delivery Date of the Product | 9 |
| FR0017 | Track Delivery | Farmer should have the option to Track the delivery on the application | 9 |
| FR0018 | Cancel / Return / Replacement | Farmer should be able to Cancel/ Return/ Replacement the product | 9 |
| FR0019 | Return Pick Up | Farmer should be given Pick Up date and time cancel / return | 9 |
| FR0020 | Return Confirmation | Farmer should receive SMS and Email confirmation for cancellation/ return. | 10 |
|  |  |  |  |
| NFR001 | Page loading time | Each Page should load within 2 seconds of time | 9 |
| NFR002 | WCAG 2.1. | The system must meet Web Content Accessibility Guidelines WCAG 2.1. | 8 |
| NFR003 | Technical Supported System | Application can be used on any OS (Android, Windows, iOS) | 9 |
| NFR004 | Time limit for OTP | OTP time limit should be given maximum of 5 minutes for Login and Registration process | 7 |
| NFR005 | Logout System | If the page is not accessed for more than 5 minutes, the page should log out automatically | 6 |
| NFR006 | Stock Availability | Stock Availability should be updated on real time basis | 8 |
| NFR007 | SMS & Mail Confirmation | Automated Email and SMS notification should be sent to users | 8 |
| NFR008 | Back Up | All data should get backup automatically | 9 |
| NFR009 | Connectivity | System should be connected with Internet | 10 |
| NFR010 | Stock Alerts | Seller should receive Stock Alert notifications when Stock is reduced, every week | 9 |
| NFR011 | Net Banking | Bank account should be active in nature for smooth payment process | 9 |
| NFR012 | Check Stock | Once the Product is sold, the stock of the product should be reduced | 7 |
| NFR013 | Email Address | Email Address should be a active to receive Email Notifications | 8 |
| NFR014 | Taxation System | All products should be included with additional Tax | 8 |
| NFR015 | Password | User should receive Password change alert every 30 days | 9 |

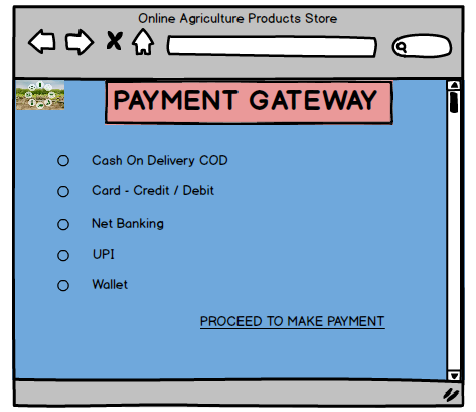
**2. Make wireframe and prototypes.**

**Balsamiq:**



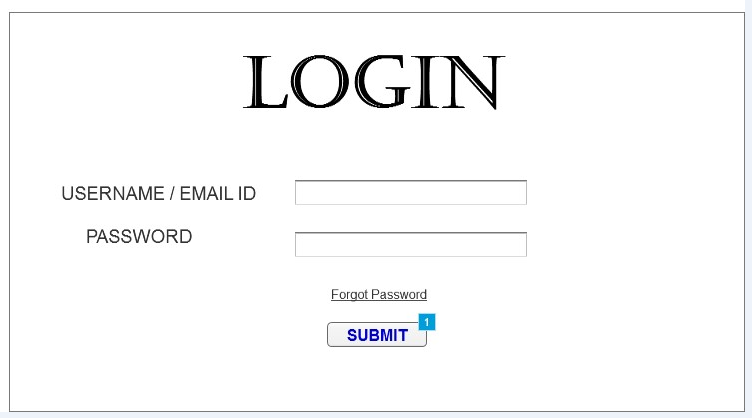


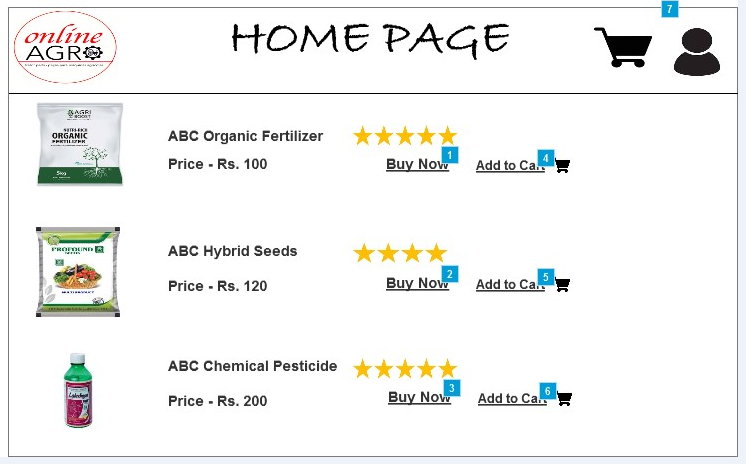


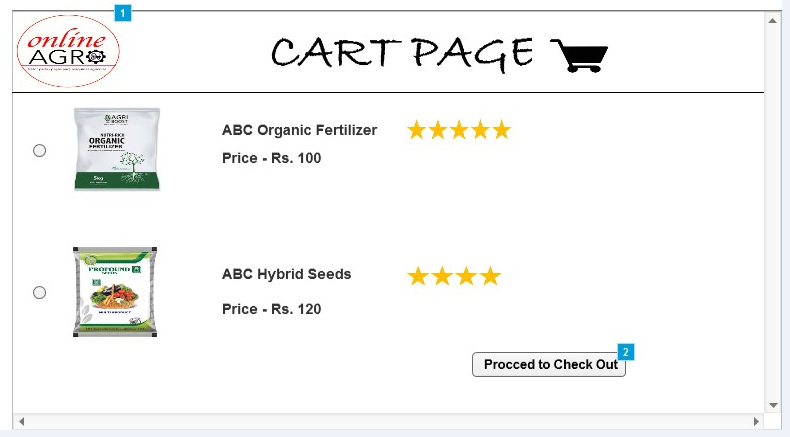


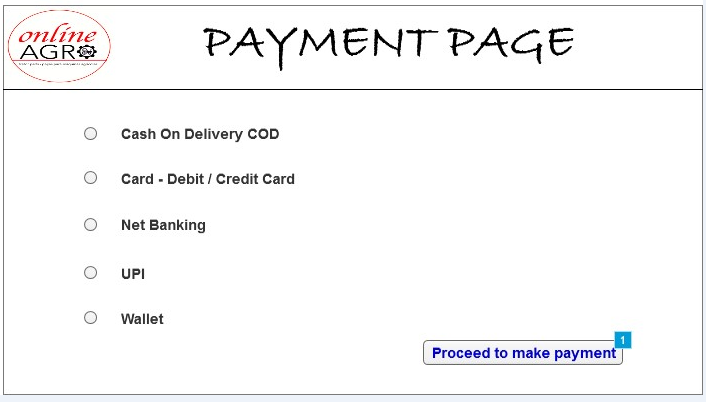


**Axure:**











**3. Make a note of the Tools, which you are using for above concepts.**

**Balsamiq**

Balsamiq is a popular wireframing tool used by designers, product managers, and developers to create low-fidelity mock-ups and prototypes for websites, apps, and software interfaces. It focuses on providing a simple, quick way to sketch ideas for user interfaces (UI) without worrying about the detailed design aspects.

**Key Features of Balsamiq:**

1. **Drag-and-Drop Interface**: Balsamiq has an intuitive drag-and-drop interface that allows you to quickly add UI elements (like buttons, text fields, menus, etc.) to your canvas.
2. **Low-Fidelity Mock-ups**: The tool is designed to encourage rapid ideation by creating rough, hand-drawn and just about enough wireframes that are intentionally basic and not polished. This helps designers and teams focus on functionality and layout before refining the visual design.
3. **Collaboration**: You can share your wireframes with colleagues or clients, allowing for easy feedback and collaboration in real-time.
4. **Pre-built Components**: Balsamiq offers a wide range of pre-built UI components, which makes it easier to build wireframes quickly without needing to create elements from scratch.
5. **Interactive Prototypes**: While Balsamiq focuses on wireframes, you can also create simple interactive prototypes by linking different screens together, which can help communicate how the user will interact with your design.
6. **Cloud and Desktop Versions**: Balsamiq is available as a desktop application (for both Windows and macOS) and also as a cloud-based version (Balsamiq Cloud) that allows you to access your designs anywhere and collaborate more easily.
7. **Export Options**: You can export your wireframes as PNGs, PDFs, or XML files, making it easy to share or use in documentation.

**Advantages:**

* **Fast Iteration**: Balsamiq allows you to quickly sketch ideas and iterate on designs, which is ideal for brainstorming or getting early feedback.
* **Focus on User Flow**: Since it's not about polished design, Balsamiq helps teams focus more on the user flow and functionality of an interface.
* **Accessible**: It's easy to learn and doesn't require advanced design skills, making it accessible to anyone involved in product development.

**Disadvantages:**

* **Limited Design Customization**: Balsamiq’s low-fidelity approach means it's not the best tool for high-fidelity or in-detail design work or visual polish.
* **No Advanced Interactions**: While it supports basic interactivity, it doesn't offer the same depth of interactivity or animation that other tools like Figma or Adobe XD might provide.

Overall, Balsamiq is a great tool for creating quick, low-fidelity wireframes to outline the structure and functionality of an interface early in the design process.

**Axure RP 7.0**

**Axure RP 7.0** is an earlier version of Axure RP, a powerful wireframing and prototyping tool used for designing interactive, high-fidelity prototypes and UI mock-ups. Axure RP is known for its ability to create more dynamic, functional prototypes compared to simple wireframing tools.

**Key Features of** **Axure RP 7.0**:

1. **Interactive Prototyping**:
   * **Dynamic Content**: Axure allows you to create interactive prototypes with dynamic content that responds to user input. You can design interactive user flows, conditional logic, and dynamic panels.
   * **Conditional Logic and Variables**: Axure RP 7.0 introduced the ability to add conditions to interactions and store data with variables, making the prototypes more sophisticated and realistic.
2. **Advanced Interactions**:
   * **Actions and Events**: It supports a wide range of interactions, such as on-click, on-hover, and on-change, as well as more complex actions like showing/hiding elements, changing styles, and navigating between pages.
   * **Drag-and-Drop Widgets**: You can use pre-made widgets or drag-and-drop components to speed up the process of building interactive prototypes.
3. **Adaptive Views**:
   * **Responsive Design**: Axure RP 7.0 introduced adaptive views, which allow you to create responsive designs that adjust to different screen sizes, similar to how modern websites and apps adapt to various devices.
4. **Master Pages**:
   * **Reusable Components**: The concept of master pages (reusable templates) allows you to design UI elements once and reuse them across different pages. This is especially helpful for maintaining consistency across prototypes.
5. **Annotations and Documentation**:
   * **Automated Documentation**: Axure RP 7.0 can generate specifications and documentation from your prototype, helping development teams understand the design interactions without needing additional communication.
   * **Annotations**: You can add notes and comments to prototypes to clarify design intentions or provide additional instructions.
6. **Collaboration**:
   * **Team Collaboration**: Axure RP 7.0 allowed for team collaboration via shared Axure RP files. This feature helped multiple designers or stakeholders to work together on the same project.
   * **Axure Cloud**: Axure Cloud (formerly Axure Share) allowed sharing prototypes and gathering feedback from clients and stakeholders in real-time.
7. **Browser-based Preview**:
   * **Preview Prototypes**: You could view your prototypes in real-time in a web browser to simulate actual user experiences. Axure also supported generating HTML files for easy sharing.
8. **Export Options**:
   * **HTML Output**: Axure RP 7.0 allows exporting prototypes as HTML files, which can be hosted on the web for live previews or sharing with others.
   * **PDF Export**: You can export documentation and wireframes in PDF format for presentation or delivery.

**Advantages of Axure RP 7.0:**

* **Rich Interactivity**: Unlike other wireframing tools, Axure RP 7.0 allows you to build highly interactive prototypes with conditional logic, variables, and data-driven interactions, making it ideal for complex web and app prototypes.
* **Dynamic Panels and States**: The use of dynamic panels and states enables the creation of complex user interactions and transitions within prototypes, which can simulate real-world app behaviours.
* **Advanced Features for Power Users**: Axure RP 7.0 is geared towards more advanced users, such as UX designers, developers, and product managers, who require intricate prototypes with rich interactions and logic.

**Disadvantages:**

* **Learning Curve**: Axure RP has a steeper learning curve, especially for beginners or those unfamiliar with interactive design tools.
* **Complexity for Simple Projects**: For simpler projects or basic wireframing, Axure RP might be overkill. It’s often seen as more suitable for high-fidelity prototypes with significant interactivity.
* **Performance**: Larger prototypes with a lot of interactions or dynamic content could lead to performance issues or slow rendering times, especially on older hardware.

**Conclusion:**

Axure RP 7.0 was a feature-rich tool that helped UX designers and product teams build advanced, interactive prototypes with sophisticated user flows and conditional logic. Its focus on interaction and functionality made it ideal for designing prototypes that closely mimic the final product. However, it was more complex than other wireframing tools and better suited for experienced users working on complex designs or projects requiring rich interactivity.

**4**. **A business analyst’s key responsibilities are to keep track of the requirements and make sure that no requirement is missed. Mr. Henry and Peter have approached you regarding the current status of the project. How will you tackle this situation?**

**Prepare RTM**

A **Requirements Traceability Matrix (RTM)** is a tool used in project management and software development to ensure that all requirements are met throughout the project lifecycle. It is essentially a document that links (or traces) requirements to their corresponding deliverables, ensuring that each requirement is fulfilled, tested, and verified.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Req ID** | **Req Name** | **Req Description** | **Design** | **D1** | **T1** | **D2** | **T2** | **D3** | **T3** | **D4** | **T4** | **UAT** |
| FR0001 | Farmer Registration | Farmers should be able to register with the application | Y | Y | Y | Y | Y | Y | Y | Y | Y | N |
| FR0002 | Farmer Search for Products | Farmers should be able to search for available products in fertilizers, seeds, pesticides | Y | Y | Y | Y | Y | Y | Y | Y | Y | N |
| FR0003 | No matching Product | Farmer should be notified if the searched product is not found | Y | Y | Y | Y | Y | Y | Y | N | N | N |
| FR0004 | Product Selection | Farmer should be able to select the Product | Y | Y | Y | Y | Y | Y | N | N | N | N |
| FR0005 | Out of Stock | Farmer should be notified if the product is out of stock | Y | Y | Y | N | N | N | N | N | N | N |
| FR0006 | Filter | Farmer should be able to filter and select the product as per Brand and Price | Y | Y | Y | N | N | N | N | N | N | N |
| FR0007 | Similar Products | Farmer should be able to select similar products, if searched product is out of Stock | Y | Y | Y | N | N | N | N | N | N | N |
| FR0008 | Bought Together | Once the product is finalized, farmer should be suggested with related products as a package with the price | Y | Y | Y | Y | Y | N | N | N | N | N |
| FR0009 | Add to Cart | Farmer should be able to Add the product to the Cart for purchase | Y | Y | Y | Y | N | N | N | N | N | N |
| FR0010 | Wishlist | Farmer should have an option to Add the product into Wishlist. | Y | Y | Y | N | N | N | N | N | N | N |
| FR0011 | Save it for Later | Farmer should be able to see the option for Save it Later, to purchase later | Y | Y | Y | N | N | N | N | N | N | N |
| FR0012 | Delivery Address | Farmer should be able to select the Delivery address to deliver the product | Y | Y | Y | N | N | N | N | N | N | N |
| FR0013 | Payment Options | Once the delivery address is selected, Application should show the Payment Options to the Farmer for purchasing the product. | Y | Y | Y | N | N | N | N | N | N | N |
| FR0014 | Payment Confirmation | Farmer should receive the Payment Confirmation email and SMS | Y | Y | Y | Y | Y | N | N | N | N | N |
| FR0015 | Order Confirmation | Farmer should receive the Order Confirmation email and SMS | Y | Y | Y | N | N | N | N | N | N | N |
| FR0016 | Expected Delivery Date | Farmer should also see the Expected Delivery Date of the Product | Y | Y | Y | N | N | N | N | N | N | N |
| FR0017 | Track Delivery | Farmer should have the option to Track the delivery on the application | Y | Y | Y | N | N | N | N | N | N | N |
| FR0018 | Cancel / Return / Replacement | Farmer should be able to Cancel/ Return/ Replacement the product | Y | Y | Y | Y | Y | N | N | N | N | N |
| FR0019 | Return Pick Up | Farmer should be given Pick Up date and time cancel / return | Y | Y | Y | N | N | N | N | N | N | N |
| FR0020 | Return Confirmation | Farmer should receive SMS and Email confirmation for cancellation/ return. | Y | Y | Y | N | N | N | N | N | N | N |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| NFR001 | Page loading time | Each Page should load within 2 seconds of time | Y | Y | Y | Y | Y | Y | Y | Y | Y | N |
| NFR002 | Technical Supported System | Application can be used on any OS (Android, Windows, iOS) | Y | Y | Y | Y | Y | N | N | N | N | N |
| NFR003 | Time limit for OTP | OTP time limit should be given maximum of 5 minutes for Login and Registration process | Y | Y | Y | N | N | N | N | N | N | N |
| NFR004 | Logout System | If the page is not accessed for more than 5 minutes, the page should log out automatically | Y | Y | Y | N | N | N | N | N | N | N |
| NFR005 | Stock Availability | Stock Availability should be updated on real time basis | Y | Y | Y | N | N | N | N | N | N | N |
| NFR006 | SMS & Mail Confirmation | Automated Email and SMS notification should be sent to users | Y | Y | Y | Y | Y | N | N | N | N | N |
| NFR007 | Back Up | All data should get backup automatically | Y | Y | Y | Y | N | N | N | N | N | N |
| NFR008 | Connectivity | System should be connected with Internet | Y | Y | Y | Y | Y | N | N | N | N | N |
| NFR009 | Stock Alerts | Seller should receive Stock Alert notifications when Stock is reduced, every week | Y | Y | Y | N | N | N | N | N | N | N |
| NFR010 | Net Banking | Bank account should be active in nature for smooth payment process | Y | Y | Y | N | N | N | N | N | N | N |
| NFR011 | Check Stock | Once the Product is sold, the stock of the product should be reduced | Y | Y | Y | Y | Y | N | N | N | N | N |
| NFR012 | Email Address | Email Address should be a active to receive Email Notifications | Y | Y | Y | N | N | N | N | N | N | N |
| NFR013 | Taxation System | All products should be included with additional Tax | Y | Y | Y | N | N | N | N | N | N | N |
| NFR014 | Password | User should receive Password change alert every 30 days | Y | Y | Y | N | N | N | N | N | N | N |

**5. Prepare 10 Test Case Documents.**

A **Test Case Document** is a detailed document that describes the conditions, inputs, actions, and expected results for testing a specific feature or functionality of a system. It is a key part of the software testing process, used to ensure that the system behaves as expected and meets the requirements. Test cases help testers verify whether the software functions correctly, meets its specifications, and identifies potential issues.

**Test Case 1: User Registration**

**Test Case ID**: TC001  
**Test Case Name**: User Registration (Farmer/Manufacturer)  
**Test Case Description**: Test if the registration feature works correctly for farmers and manufacturers.

**Preconditions**:

* User is on the registration page.
* User must have valid information to register.

**Test Steps**:

1. Navigate to the registration page of the application.
2. Enter valid details (Name, Email, Phone, Address, User Type [Farmer/Manufacturer]).
3. Click on the “Register” button.
4. Check if a confirmation email or OTP is sent.
5. Enter the OTP received and submit.
6. Verify that the user is successfully registered and logged into the application.

**Expected Result**:

* The user should receive a confirmation message/OTP and successfully register on the platform.

**Actual Result**:

* To be filled after execution.

**Status**:

* Pass/Fail

**Test Case 2: Product Upload (Manufacturer)**

**Test Case ID**: TC002  
**Test Case Name**: Product Upload by Manufacturer  
**Test Case Description**: Test if the manufacturer can upload product details correctly.

**Preconditions**:

* Manufacturer is logged in.
* Manufacturer has valid product information (name, price, description, stock, etc.).

**Test Steps**:

1. Login as Manufacturer.
2. Navigate to the “Product Upload” section.
3. Enter product details (Name, Description, Category [Fertilizer/Seeds/Pesticides], Price, Stock Availability).
4. Upload product image (optional).
5. Click “Submit”.
6. Verify that the product is displayed in the product catalogue.

**Expected Result**:

* The product should be uploaded successfully and be visible in the product catalogue.

**Actual Result**:

* To be filled after execution.

**Status**:

* Pass/Fail

**Test Case 3: Product Search (Farmer)**

**Test Case ID**: TC003  
**Test Case Name**: Product Search by Farmer  
**Test Case Description**: Test if the farmer can search for products (seeds, fertilizers, pesticides).

**Preconditions**:

* Farmer is logged in.
* Products are available in the catalogue.

**Test Steps**:

1. Login as Farmer.
2. Go to the search bar.
3. Enter the product name (e.g., “Fertilizer”).
4. Press the search button.
5. Verify if relevant products appear in the search results.

**Expected Result**:

* The search results should display products related to the entered query.

**Actual Result**:

* To be filled after execution.

**Status**:

* Pass/Fail

**Test Case 4: Add Product to Cart (Farmer)**

**Test Case ID**: TC004  
**Test Case Name**: Add Product to Cart  
**Test Case Description**: Test if the farmer can add a product to the cart for purchasing.

**Preconditions**:

* Farmer is logged in.
* At least one product is available for purchase.

**Test Steps**:

1. Login as Farmer.
2. Browse the products catalogue.
3. Select a product to purchase.
4. Click the “Add to Cart” button.
5. Verify if the product is added to the cart successfully.

**Expected Result**:

* The product should be added to the cart with correct details (name, price, quantity).

**Actual Result**:

* To be filled after execution.

**Status**:

* Pass/Fail

**Test Case 5: Product Purchase Request (Farmer)**

**Test Case ID**: TC005  
**Test Case Name**: Product Purchase Request by Farmer  
**Test Case Description**: Test if the farmer can successfully place an order for products.

**Preconditions**:

* Farmer is logged in.
* Product is added to the cart.

**Test Steps**:

1. Login as Farmer.
2. Go to the cart page.
3. Review the cart items.
4. Click on “Proceed to Checkout”.
5. Fill in the delivery details (address, phone number).
6. Click on “Place Order”.
7. Verify if an order confirmation is received (email/notification).

**Expected Result**:

* The farmer should receive an order confirmation, and the order should be placed successfully.

**Actual Result**:

* To be filled after execution.

**Status**:

* Pass/Fail

**Test Case 6: View Product Details (Farmer)**

**Test Case ID**: TC006  
**Test Case Name**: View Product Details  
**Test Case Description**: Test if the farmer can view product details such as description, price, and availability.

**Preconditions**:

* Farmer is logged in.
* Products are available in the catalogue.

**Test Steps**:

1. Login as Farmer.
2. Browse the products catalogue.
3. Click on any product to view detailed information.
4. Verify if all relevant details (price, description, availability, and product image) are displayed.

**Expected Result**:

* All the product details should be displayed correctly.

**Actual Result**:

* To be filled after execution.

**Status**:

* Pass/Fail

**Test Case 7: Admin Login and Dashboard Access**

**Test Case ID**: TC007  
**Test Case Name**: Admin Login and Dashboard Access  
**Test Case Description**: Test if an admin (System) can successfully log in and access the dashboard.

**Preconditions**:

* Admin has valid login credentials.

**Test Steps**:

1. Login as Admin.
2. Verify access to the admin dashboard.
3. Ensure that the admin can access various sections like product management, user management, and order management.

**Expected Result**:

* Admin should successfully log in and have access to the dashboard and all administrative functions.

**Actual Result**:

* To be filled after execution.

**Status**:

* Pass/Fail

**Test Case 8: Update Product Information (Manufacturer)**

**Test Case ID**: TC008  
**Test Case Name**: Update Product Information by Manufacturer  
**Test Case Description**: Test if the manufacturer can successfully update the product details after it’s uploaded.

**Preconditions**:

* Manufacturer is logged in.
* Manufacturer has at least one product uploaded.

**Test Steps**:

1. Login as Manufacturer.
2. Go to the “Manage Products” section.
3. Select a product to edit.
4. Update product details (price, stock, description).
5. Save changes.
6. Verify if the product details are updated in the catalogue.

**Expected Result**:

* The product information should be updated correctly.

**Actual Result**:

* To be filled after execution.

**Status**:

* Pass/Fail

**Test Case 9: Order Status Update (Admin/Manufacturer)**

**Test Case ID**: TC009  
**Test Case Name**: Update Order Status (Admin/Manufacturer)  
**Test Case Description**: Test if the admin or manufacturer can update the order status (e.g., Shipped, Delivered).

**Preconditions**:

* Order has been placed by a farmer.
* Admin/Manufacturer has valid credentials to update the order.

**Test Steps**:

1. Login as Admin or Manufacturer.
2. Navigate to the “Orders” section.
3. Select an order to update.
4. Change the order status (e.g., from Pending to Shipped).
5. Save changes.
6. Verify if the order status is updated correctly.

**Expected Result**:

* The order status should be updated correctly, and the farmer should receive a notification.

**Actual Result**:

* To be filled after execution.

**Status**:

* Pass/Fail

**Test Case 10: User Logout**

**Test Case ID**: TC010  
**Test Case Name**: User Logout  
**Test Case Description**: Test if the user can log out successfully from the application.

**Preconditions**:

* User is logged into the application.

**Test Steps**:

1. Login to the application.
2. Click on the logout button.
3. Verify that the user is logged out and redirected to the login page.

**Expected Result**:

* The user should be logged out and redirected to the login page.

**Actual Result**:

* To be filled after execution.

**Status**:

* Pass/Fail

**6. After the requirements are thoroughly explained to the entire project team by business analyst, the Database architects have decided to do the database design and also to represent the in-flow and out-flow of data.**

**Draw database schema and ER diagram**

**DB Schema**



**ER Diagram**



1. **What is a data flow diagram? Draw a data flow diagram to represent the in-flow and out flow of data when a Farmer is placing an order for the product.**

A Data Flow Diagram (DFD) is a graphical representation that shows the flow of data within a system. It illustrates how inputs are processed and how the data moves through various components or processes in the system. DFDs are typically used in systems analysis to help visualize the system's processes, data storage, and interactions between external entities.

In a DFD:

* **Entities** represent the external actors or systems that interact with the system.
* **Processes** represent the activities or operations that transform the data.
* **Data Flows** are the pathways through which data moves from one entity to another or between processes.
* **Data Stores** represent the places where data is stored.





1. **Due to change in the Government Taxation structure, we should change the Tax structure How do you handle change requests in a project?**

Handling change requests, such as the one regarding the **change in government taxation structure**, is a critical part of project management and ensures that the project remains aligned with the evolving business and regulatory needs. Here's a structured approach for handling change requests in a project:

**1**. **Acknowledge the Change Request**

* **Immediate Acknowledgment**: When a change request is received, it is crucial to acknowledge it promptly to the requester. In this case, it would be the stakeholders or the team responsible for adhering to the government taxation structure.
* **Documentation**: Ensure the change request is well documented, including details like:
  + **Nature of the change** (e.g., changes in tax structure due to government policy)
  + **Reason for the change** (e.g., change in government taxation policy)
  + **Date of the request**

**2.** **Assess the Impact of the Change**

* **Analyse the Impact**: The next step is to assess the impact of the change on the project in several areas:
  + **Scope**: Does the change affect the current scope of the project? In this case, you may need to modify how taxes are calculated or handled in the system.
  + **Cost**: What will the change cost? You’ll need to evaluate whether additional resources are needed to implement the changes.
  + **Timeline**: How will this change affect the project timeline? Adjustments may be needed to accommodate new functionality, testing, and deployment.
  + **Quality**: Will this change affect the quality of the solution? Ensure that the change does not compromise the quality standards of the application.
* **Regulatory Compliance**: This particular change is related to government taxation, so ensuring that the system is fully compliant with legal requirements is crucial.

**3.** **Evaluate Alternatives and Solutions**

* **Consult with Experts**: Engage relevant stakeholders such as legal advisors, finance teams, and tax consultants to understand the specifics of the new taxation structure.
* **Technical Feasibility**: Assess whether the current system can handle the new tax rules and calculations or if significant changes are needed in the code, database, or user interface.
* **Cost-Benefit Analysis**: Evaluate the cost of implementing the change versus the benefits it will bring to compliance and business operations.

**4.** **Create a Change Request Form**

* **Document the Details**: Create a detailed change request document/form that includes:
  + A description of the change (e.g., government tax structure change).
  + The expected impact on scope, cost, timeline, and quality.
  + A technical solution (if applicable).
  + Approval requirements (e.g., from stakeholders, legal team, etc.).
* This document serves as a formal record and ensures that all parties are aligned on the change.

**5.** **Review and Approval Process**

* **Internal Review**: The change request needs to be reviewed by the relevant stakeholders, including:
  + **Project Manager**: To assess the feasibility, timeline, and resources.
  + **Finance and Legal Teams**: To ensure compliance with tax regulations and assess financial implications.
  + **Project Sponsor or Committee**: To ensure that the change aligns with the overall business objectives and the budget.
* **Approval or Rejection**: After reviewing, the project manager and stakeholders should approve or reject the change request. If approved, the change request is added to the project plan.

**6.** **Update the Project Plan and Documents**

* **Scope Update**: If the change request is approved, update the project scope to reflect the new requirements. For example, the system will now need to incorporate the new tax rates and logic for tax calculations.
* **Schedule Update**: Adjust the project timeline to account for the time required to implement the change, which may include additional testing, development, and deployment time.
* **Budget Update**: If there are any additional costs associated with the change (e.g., new development resources or software tools), update the project budget.
* **Risk Management Update**: Review any risks associated with the change and update the project’s risk management plan.

**7**. **Implement the Change**

* **Development and Testing**: Begin working on implementing the change in the system. This may involve:
  + Modifying the tax calculation logic in the system.
  + Updating the database schema if new tax fields are needed.
  + Updating user interfaces for tax-related features.
* **Quality Assurance (QA)**: Thoroughly test the changes to ensure that they work as expected and that no other parts of the system are negatively impacted.
* **Stakeholder Communication**: Keep stakeholders updated on the progress of the change implementation.

**8.** **Deploy the Change**

* **Deployment**: Once the change has been developed and tested, it is time to deploy the update to production. Ensure that:
  + All tax-related features are properly implemented.
  + Documentation is updated to reflect the changes.
  + Any necessary training is provided to users, especially those who are involved in tax-related workflows.

**9. Post-Implementation Review**

* **Monitor the System**: After deployment, continuously monitor the system to ensure that it is operating as expected and the new tax structure is being applied correctly.
* **Feedback Collection**: Collect feedback from end-users to identify any potential issues with the implementation.
* **Lessons Learned**: Document any lessons learned from handling the change request for future reference.

**10.** **Communication with Stakeholders**

* **Keep Stakeholders Informed**: Throughout the process, ensure clear and consistent communication with all stakeholders to manage expectations and provide updates on the progress of the change request.

**Synopsis of Key Steps in Handling Change Requests:**

1. **Acknowledge** the change request.
2. **Assess the impact** of the change on scope, cost, timeline, and quality.
3. **Evaluate alternatives** and technical feasibility.
4. **Create a change request document** outlining details and impact.
5. **Review and approve** the change request with stakeholders.
6. **Update project plans, scope, budget**, and risk management strategies.
7. **Implement the change**, including development, testing, and deployment.
8. **Deploy** the change to production.
9. **Monitor** the system and collect feedback.
10. **Communicate** regularly with stakeholders.

Handling changes effectively ensures that the project remains flexible and adaptable to new requirements, while still maintaining its overall goals and objectives.

**9. As the project is in process, Ben and Kevin have contacted you. The reason is to inform you that they want the Farmers to sell their crop yields through this application i.e., Farmers should be able to add their crop yields or products and display to general public and should be able to sell them. They also want to introduce Auction system for their Crop yields. As a BA, what will be your response?**

**Is this a change request or an enhancement???**

As a **Business Analyst (BA)**, my primary responsibility is to understand and evaluate the needs of stakeholders and assess how those needs can be incorporated into the existing system. In this case, Ben and Kevin are asking for new features to be added to the system, which involve **Farmers being able to sell their crop yields** and introducing an **Auction system**.

**Response to Ben and Kevin's Request:**

**1. Clarify the Requirements:**

* **Understanding the Requirements**: The first thing I would do is to clarify the details of the request to ensure I fully understand what they need.
  + **Farmers Selling Crop Yields**: Are you asking for a feature that will allow Farmers to list their crops or products for sale? If so, what type of information do you want to capture? (e.g., crop type, quantity, price, description, images)
  + **Auction System**: What exactly is meant by the auction system? Will it allow farmers to set a base price and bid on their products? Should it work in real-time, and what auction rules should we implement?

**2. Understand the Impact:**

* These new features would significantly expand the scope of the application beyond just buying agricultural products (seeds, fertilizers, pesticides) and would introduce new features that would involve additional processes, workflows, and data structures (e.g., farmer profiles, auction bidding system, payment for crops).
* I would assess how these new features align with the existing objectives of the application and whether they will require changes to the backend, UI/UX design, or any integration with external systems (e.g., payment gateways, shipping logistics, etc.).

**3. Analyse the Impact on the Current Project Plan:**

* **Timeline**: Adding the ability for Farmers to sell crop yields and introducing an auction system will likely require additional development time, testing, and deployment. This would need to be assessed and accounted for in the project timeline.
* **Budget**: These new features might require additional resources, such as new development work, more testing, and potentially additional infrastructure to support the changes. The budget could need to be revised.
* **Scope**: These features could be considered as **out-of-scope** if they were not part of the original project scope, and we would need to evaluate whether they align with the project goals. If they are outside the original scope, they may require a formal change request process.

**Is This a Change Request or an Enhancement?**

* **Change Request**: A change request typically involves modifications to the project that were not initially planned or anticipated, often due to a change in external circumstances, such as new regulations or unforeseen issues. In this case, adding the ability for Farmers to sell crops and introducing an auction system represents a significant shift in functionality that was likely **not part of the original scope** as defined by Mr. Henry and the initial project team.
* **Enhancement**: An enhancement typically refers to improving or adding to existing features or functionalities. If the system had already planned to allow Farmers to list products but didn’t plan on an auction system, adding the auction could be considered an **enhancement** to the selling feature.

Since this request is introducing **new capabilities** (selling crops, auction system) that weren’t originally planned for, **this is more likely out of scope when compared to the originally planned requirements.** So I would say it could be considered as an **Enhancement.**

**Steps to Process This Request:**

1. **Document the Request**: I will document the request as a **formal change request**. This will include:
   * Description of the new feature (Farmer crop selling and auction system).
   * Impact on the existing system.
   * Proposed timelines and resources required.
   * Budget adjustments, if any.
2. **Analyse the Impact**:
   * **Technical Impact**: Assess the technical changes needed in the application (backend, frontend, database, etc.).
   * **Business Impact**: Evaluate how this will affect the end-users (Farmers, general public, etc.).
   * **Regulatory Impact**: Determine if there are any regulatory or compliance issues with enabling direct sales or auctions for Farmers.
3. **Review with Stakeholders**:
   * Since this is a change request, it needs to be reviewed and approved by key stakeholders, including Mr. Henry, the project sponsor, and the development team, to assess feasibility, impact on timeline, and costs.
4. **Create a Formal Change Request**:
   * Prepare a formal change request document/form for approval, which will outline:
     + The objectives of the new functionality.
     + The resources needed.
     + Revised budget and timeline estimates.
     + Any risks involved.
5. **Approval Process**:
   * The committee (Mr. Henry, Mr. Pandu, and Mr. Dooku) should evaluate whether to approve or reject the request based on the added value, feasibility, and impact on the project.
6. **Update Project Plans**:
   * If the change request is approved, I will update the project’s scope, timeline, budget, and risk management plan accordingly.
7. **Communicate the Change**:
   * Keep all relevant stakeholders informed throughout the process.

**Conclusion:**

* **Response to Ben and Kevin**: I would inform them that their request to enable Farmers to sell their crop yields and introduce an auction system is significant and requires thorough evaluation. It needs to be processed as an **Enhancement**, and I would initiate the necessary steps to assess the impact and get approval from the stakeholders before proceeding.

**10. Come up with estimations – How many Manhours required?**

Estimating **man-hours** for a project is a critical task that helps determine the resources, timeline, and costs involved. Since the change request involves enabling Farmers to sell their crop yields and introducing an auction system, the estimates must cover various aspects such as requirements gathering, development, testing, and deployment.

Here’s how we can break down the estimation process:

**1. Requirements Gathering & Analysis**

* **Task**: Understanding the exact requirements from stakeholders (Ben, Kevin, etc.) and clarifying any ambiguities. This involves gathering all the functional and non-functional requirements for the new crop selling feature and auction system.
* **Manhours Estimate**:
  + **Business Analyst (BA)**: 20–30 hours
  + **Stakeholder Meetings, Documentation, Analysis**:
  + **Total**: 20–30 hours

**2.** **System Design & Architecture**

* **Task**: Designing how the new features (crop selling and auction system) will be integrated into the current application. This includes database changes (e.g., adding product/crop listings, auction bid tracking), system architecture, and UI/UX design.
* **Manhours Estimate**:
  + **Project Manager**: 10–15 hours (to coordinate with stakeholders and approve designs)
  + **Solution Architect/Lead Developer**: 30–40 hours (for designing backend and auction system logic)
  + **UI/UX Designer**: 25–30 hours (for designing user interfaces for crop selling, auction features, etc.)
  + **Total**: 65–85 hours

**3.** **Development Phase**

* **Task**: Implementing the functionality for:
  + **Crop Selling**: Allowing farmers to list and manage their crop yields (adding details such as price, quantity, description, images).
  + **Auction System**: Developing the logic for creating auctions, bidding, setting auction start/end times, and notifying users.
  + Changes to **payment processing**, since now there will be a mechanism for the payment of crops.
* **Manhours Estimate**:
  + **Senior Java Developer (Ms. Juhi)**: 40–50 hours (for high-level coding, backend logic)
  + **Java Developers (Mr. Teyson, Ms. Lucie, Mr. Tucker, Mr. Bravo)**: 150–200 hours (for coding auction features, integration with payment systems, managing crop listings, etc.)
  + **Database Administrator (John)**: 30–40 hours (for modifying database schema, adding tables for crops, auction bids, etc.)
  + **Total**: 220–290 hours

**4.** **Testing Phase**

* **Task**: Testing the newly implemented features to ensure that the crop selling and auction systems work as expected. This includes functional testing, integration testing, and user acceptance testing (UAT).
* **Manhours Estimate**:
  + **Testers (Mr. Jason, Ms. Alekya)**: 60–80 hours (for creating test cases, performing testing, and ensuring system stability)
  + **Total**: 60–80 hours

**5.** **Deployment & Release**

* **Task**: Deploying the new functionality to the production environment, ensuring all configurations are set up correctly, and making the new features available to end-users.
* **Manhours Estimate**:
  + **Network Admin (Mr. Mike)**: 10–15 hours (for configuring servers, setting up networking)
  + **DevOps Engineer (if applicable)**: 10–20 hours (for deployment pipelines and continuous integration setup)
  + **Total**: 20–35 hours

**6.** **Post-Deployment Support**

* **Task**: Monitoring the system post-launch to fix any issues that may arise and providing support for end-users.
* **Manhours Estimate**:
  + **Project Manager**: 10–15 hours (to monitor and gather feedback from stakeholders)
  + **Developers**: 30–40 hours (for bug fixes and system tweaks based on user feedback)
  + **Total**: 40–55 hours

**Total Manhours Estimate**

1. **Requirements Gathering & Analysis**: 20–30 hours
2. **System Design & Architecture**: 65–85 hours
3. **Development**: 220–290 hours
4. **Testing**: 60–80 hours
5. **Deployment**: 20–35 hours
6. **Post-Deployment Support**: 40–55 hours

**Total Estimated Manhours** = **425–575 hours**

**How to Break It Down into Teams**

If we consider the **team size** and the available resources, this is how the man-hours might be distributed:

* **Business Analyst (BA)**: ~20–30 hours (requirements gathering)
* **Solution Architect/Lead Developer**: ~30–40 hours (system design)
* **Senior Java Developer (Ms. Juhi)**: ~40–50 hours (development)
* **Java Developers (Mr. Teyson, Ms. Lucie, Mr. Tucker, Mr. Bravo)**: ~150–200 hours (development)
* **Database Administrator (John)**: ~30–40 hours (database changes)
* **Testers (Mr. Jason, Ms. Alekya)**: ~60–80 hours (testing)
* **Network Admin (Mr. Mike)**: ~10–15 hours (deployment)
* **Project Manager**: ~10–15 hours (deployment, post-deployment support)
* **DevOps Engineer (if applicable)**: ~10–20 hours (deployment)

**Conclusion**

The total estimated man-hours for implementing the **Farmer crop selling feature** and **auction system** would be in the range of **425 to 575 hours**, depending on the complexity and any unforeseen challenges encountered during the implementation.

These estimations will need to be refined further based on more detailed analysis, but this gives a rough idea of the resources required to deliver the new features.

1. **Project has finally completed all the stages i.e., design, development, testing etc. Now, it is the role of a business analyst to contact the client for testing of the final product and have to successfully complete it. How are you going to handle this situation? And once it is done, what will be the process to close the project?**

**Explain UAT Acceptance process.**

As a **Business Analyst (BA)**, once the project has completed all the stages like **design**, **development**, and **testing**, my role shifts towards ensuring that the client is satisfied with the final product. The most important step now is to facilitate **User Acceptance Testing (UAT)**, which is crucial for verifying that the system meets the client's requirements and is ready for deployment. Here's how I would handle this situation:

**1. Preparing for UAT (User Acceptance Testing)**

**a. Coordinate with Stakeholders**

* **Identify the Key Stakeholders**: This typically includes the client (in this case, Mr. Henry and possibly his team, such as Ben and Kevin), the end-users (Farmers), and other relevant parties.
* **Define UAT Goals**: Clearly outline the goals of the UAT. This involves ensuring that the system functions according to the business requirements, as well as validating whether the system is intuitive and user-friendly.
* **Set Expectations**: I will make sure the client and stakeholders are clear on the expectations from the UAT process, such as:
  + What specific scenarios and features need to be tested?
  + What kind of feedback is expected (e.g., critical bugs, usability issues)?
  + The timelines for completing the tests and reporting feedback.

**b. Create UAT Plan and Test Cases**

* **UAT Test Plan**: Prepare a comprehensive **UAT test plan** that includes:
  + **Objectives** of UAT.
  + **Test scenarios** and use cases covering all critical functionalities (e.g., crop listing, auction bidding, tax calculations).
  + **Success Criteria**: Clear guidelines on what constitutes a pass or fail for each test case.
  + **Test Environment**: Ensure that the UAT will be done in an environment similar to production, with real data or mock data.
  + **Roles and Responsibilities**: Assign roles, so the stakeholders know who is responsible for what during UAT (e.g., who will log defects, who will approve tests, etc.).
* **Test Cases**: Collaborate with the testers (QA team) to design specific UAT test cases that represent how end-users will interact with the system.

**c. Ensure a Smooth Transition for UAT**

* **Training and Documentation**: Provide users with necessary training or instructions on how to conduct the tests. This ensures that they understand the features they are testing and what the expected behaviour should be.
* **UAT Environment Setup**: Coordinate with the technical team (Developers, Database Admins) to ensure the UAT environment is set up with the correct configuration, data, and access permissions.
* **Test Data Preparation**: Make sure the test data is realistic and representative of real-world usage, especially for features like crop listings, auctions, and payments.

**2. Conducting UAT**

**a. UAT Kick-off**

* **Kick-off Meeting**: I would schedule a kick-off meeting with the client and end-users to explain the UAT process, review the test cases, and set expectations on timelines and feedback.
* **Test Execution**: Users begin testing the product by executing the defined test cases. I will facilitate communication during the testing phase to ensure that any issues are logged and escalated appropriately.
* **Issue Tracking**: Ensure there is a clear system in place for stakeholders to log any bugs or issues they encounter during testing. This could be done using a tool like **JIRA**, **Trello**, or any issue tracking system.
* **Client Feedback**: Collect detailed feedback on how the system performs in real-world conditions. I will ensure that feedback is structured, actionable, and addresses key issues such as functionality, usability, and performance.

**b. Review Feedback and Perform Fixes**

* **Monitor Progress**: As UAT is conducted, I will ensure that any issues identified are logged, categorized, and prioritized for resolution.
* **Defect Resolution**: Work with the development team to fix any high-priority issues identified during UAT. Re-testing may be necessary to ensure that fixes are effective.
* **Revalidation**: Once issues are addressed, re-validate the system with the client or the same UAT testers to ensure that changes have been applied successfully.

**c. Final Approval**

* Once all the issues have been resolved and the system meets the business requirements, I will work with the stakeholders (Mr. Henry, Ben, Kevin, etc.) to get **formal approval** for the product.
* This could be in the form of a **UAT sign-off document**, where the client formally accepts that the system is ready for deployment and meets their requirements.

**3. Closing the Project (Post-UAT)**

Once UAT is complete and the client has signed off on the system, the project enters the **closure phase**. Here's how to handle it:

**a. Final Documentation**

* **User Manuals**: Ensure that user manuals and any documentation are up to date with all features and functionalities, especially those related to the new crop selling and auction systems.
* **Technical Documentation**: Ensure that all technical documentation is complete, including code, database schemas, API documentation (if applicable), and deployment procedures.
* **Training Materials**: If applicable, ensure that all necessary training materials are delivered to the end-users to ensure they can use the system effectively.

**b. Final Review with Stakeholders**

* **Review**: Conduct a final review meeting with the project stakeholders to ensure that the project has met all business objectives and requirements. This meeting should discuss:
  + What went well in the project.
  + What could have been improved.
  + Any open issues or follow-up items.
* **Lessons Learned**: Document any lessons learned during the project that can help improve future projects.

**c. Handover**

* **Handover to Maintenance and Support**: Once the project is signed off, it is handed over to the maintenance and support team. This team will be responsible for resolving any post-deployment issues and performing regular system updates.
* **Deployment to Production**: Coordinate the deployment of the final system to the production environment, ensuring minimal downtime and smooth transition.

**d. Final Project Sign-Off**

* Once the system is deployed and all deliverables are provided, I will ensure that the final project sign-off is obtained from the client (Mr. Henry or the project sponsor).
* This ensures that the client acknowledges the successful completion of the project, and that all deliverables are in place.

**4.** **UAT Acceptance Process**

The **User Acceptance Testing (UAT) Acceptance Process** involves the following steps:

1. **Planning UAT**: As discussed, we define test cases, test objectives, and success criteria.
2. **Test Execution**: The client or end-users perform the testing according to the plan, simulating real-world scenarios and providing feedback.
3. **Issue Tracking**: All defects or issues are tracked and prioritized for resolution.
4. **Issue Resolution**: Development team fixes high-priority issues. Retesting may occur after fixes are applied.
5. **Approval**: Once all issues are resolved, the system is presented for final approval, and the client signs off on the UAT.
6. **UAT Sign-Off**: This formal document confirms that the product meets the business requirements and is ready for production.
7. **Deployment**: After UAT sign-off, the system is deployed into the production environment, and the project is officially closed.

**Conclusion**

As a BA, my role during the UAT and project closure phase involves ensuring smooth communication, timely feedback, and resolving any issues promptly. Once UAT is successful, the project closure involves finalizing documentation, handing over the system to the support team, and obtaining the final project sign-off. This ensures that the system is ready for production and the project is formally concluded.

1. **Explain Project Closure Document.**

A **Project Closure Document** is a formal document that signifies the official completion of a project. It is created at the end of a project, following the successful delivery and acceptance of the project’s final deliverables. This document ensures that all project goals and objectives have been met and outlines any remaining issues or concerns. It is often required for record-keeping and for closing out the project’s financial and administrative processes.

The **Project Closure Document** serves as a key reference for future projects and provides stakeholders with assurance that the project was completed in a structured, organized manner.

**Components of a Project Closure Document**

Here’s a breakdown of the key components typically found in a **Project Closure Document**:

**1. Project Overview**

* **Project Name**: The title of the project.
* **Project ID/Code**: A unique identifier for the project.
* **Project Sponsor/Client**: The individual or organization that commissioned the project (e.g., Mr. Henry, as the project sponsor).
* **Project Manager**: The individual who managed the project throughout its lifecycle.
* **Start and End Dates**: The official dates the project began and ended.
* **Project Objective**: A brief summary of the project’s objectives, goals, and deliverables. For example, in this case, developing an online agriculture product store for farmers.

**2. Project Deliverables**

* **Final Deliverables**: List all the key deliverables produced during the project (e.g., the fully developed online store application for farmers to buy and sell crops, including the auction feature).
* **Completion Status**: The status of the deliverables, whether they were completed, accepted, or if any issues remain unresolved.
* **Quality Standards**: Ensure that all deliverables meet the agreed-upon quality standards, such as performance, functionality, and security requirements.

**3.** **Project Performance and Results**

* **Project Objectives Met**: A summary of whether the project met the original objectives, such as fulfilling the needs of the farmers and manufacturers, implementing the online store and auction system, and achieving the desired user experience.
* **Scope Achievement**: An evaluation of whether the project scope was adhered to, or if there were any changes made (like the crop-selling and auction features).
* **Timeline Adherence**: Did the project meet the initial timelines, or were there delays? Any major deviations from the schedule should be highlighted, along with the reasons for them.
* **Budget Adherence**: A comparison of the original project budget with the final expenditure. If there were any cost overruns, this should be documented and explained.

**4. Issue and Risk Resolution**

* **Issues Faced During the Project**: A summary of any significant issues encountered during the project (e.g., technical challenges, delays due to resource unavailability, or new requirements like crop selling and auctions). This could include a brief description of how these issues were resolved.
* **Risk Management**: A review of the risks identified at the beginning of the project, along with how those risks were mitigated or managed throughout the project lifecycle.

**5.** **Client/Stakeholder Feedback**

* **Feedback from the Client**: Summarize feedback from the client (e.g., Mr. Henry, Ben, Kevin) and other stakeholders. Was the client satisfied with the project outcome? Were their expectations met?
* **Feedback from End Users**: If available, include feedback from end users, such as farmers, on how the system performs in real-world conditions (e.g., usability, performance, and user experience).

**6.** **Lessons Learned**

* **Challenges**: A description of the challenges faced during the project, including anything that was difficult to achieve or unexpected issues.
* **Best Practices**: Highlight any best practices or effective strategies used during the project, such as successful teamwork, communication methods, or tools that worked well.
* **Areas for Improvement**: Reflect on areas that could be improved for future projects. This could include better risk management, clearer requirements gathering, or enhanced testing procedures.

**7. Handover Process**

* **Handover of Deliverables**: Confirmation that all deliverables (final product, documentation, user manuals, etc.) have been successfully handed over to the client or relevant stakeholders.
* **Training**: If applicable, note that any training required for end-users (farmers, administrative staff, etc.) has been completed.
* **Maintenance and Support**: Provide information about the maintenance and support process post-project. Who will be responsible for managing the system once it’s live, and what processes are in place for issue resolution and updates?

**8.** **Formal Sign-Off**

* **Client Approval/Sign-Off**: The final section of the closure document typically includes the formal **sign-off** from the client or project sponsor, indicating that they approve the deliverables and consider the project complete.
* **Signatures**: The signatures of the client, project manager, and any other key stakeholders.
* **Date of Sign-Off**: The official date when the project was concluded.

**9.** **Post-Project Review and Follow-Up**

* **Post-Project Evaluation**: A short assessment of how the project can be improved in the future. This can include a review of internal processes, client relationships, and overall performance.
* **Support/Change Requests**: If any post-deployment issues or changes arise (such as bugs or additional features), outline the process for requesting these changes or providing support.
* **Transition to Ongoing Operations**: The transition of the system to production and ongoing support or maintenance teams.

**Purpose of the Project Closure Document**

* **Formal Completion**: It signifies the official end of the project, confirming that all work has been completed as per the scope, budget, and timeline.
* **Accountability and Transparency**: It ensures accountability by documenting the outcomes, challenges, and resolutions of the project. It provides transparency for stakeholders about how the project was handled.
* **Knowledge Transfer**: By documenting lessons learned, best practices, and areas for improvement, the closure document aids in knowledge transfer for future projects.
* **Record of Success**: It serves as an official record that can be referenced later if there are any questions or follow-up actions required.

**Conclusion**

The **Project Closure Document** is an essential part of project management as it ensures that the project has been completed to the satisfaction of all stakeholders. It formally closes the project by documenting its performance, the issues faced, and the client’s approval. This helps in providing a clear conclusion to the project, while also laying the groundwork for future improvements and maintenance.