**Scenario**:

I am providing a warehousing service to E-commerce stores where they can store their goods and am responsible for delivering their orders.

I have 3 departments Picking, Packaging, and dispatch.

once the order arrived the picker picks the stock and forwards it to the packaging department. the packaging department packs the order and moves it toward the dispatch department, then the dispatch department hands over the order to different courier companies for delivery.

I need to system to manage my internal affairs of the warehouse.

**Requirements** :

1:Pick order and forward the order to packaging department

2:pack order and forward the order to dispatching department

3:Dispatch order and forward to courier company

4: Receiving and recording new stock arrivals.

E-commerce warehouses play a crucial role in the smooth functioning of online retail operations. These warehouses are responsible for efficiently managing and storing inventory, ensuring timely order fulfillment, and maintaining the overall supply chain. Here are some essential functionalities of an e-commerce warehouse:

1. Inventory Management:

- Real-time tracking of inventory levels for all products.

- Receiving and recording new stock arrivals.

- Accurate and efficient stock counting through regular audits.

- Handling stock rotation (FIFO, FEFO) to prevent product expiration or obsolescence.

2. Order Fulfillment:

- Picking and packing of items to prepare orders for shipment.

- Ensuring order accuracy and completeness.

- Integrating with order management systems to process incoming orders.

- Prioritizing urgent orders and meeting specified delivery timeframes.

3. Warehousing Technology:

- Utilizing barcode scanners and RFID technology for efficient inventory tracking.

- Implementing warehouse management systems (WMS) to optimize processes.

- Using automation and robotics for repetitive tasks, such as picking and packing.

4. Storage and Organization:

- Efficient space utilization through strategic shelving and storage systems.

- Segregation of products based on size, weight, and handling requirements.

- Implementing a systematic product labeling system for easy identification.

5. Shipping and Logistics:

- Collaborating with shipping carriers and logistics providers for smooth outbound operations.

- Selecting appropriate shipping methods (e.g., ground, air, express) based on customer requirements.

- Generating shipping labels and tracking numbers for customers' visibility.

6. Returns Management:

- Processing and inspecting returned items to determine their condition and potential restocking.

- Initiating the refund or exchange process for eligible returns.

- Updating inventory levels accordingly based on the returned items' condition.

7. Security and Safety:

- Implementing strict security measures to prevent theft and unauthorized access.

- Adhering to safety regulations for handling hazardous or sensitive products.

- Maintaining a clean and organized workspace to avoid accidents and product damage.

8. Data Analytics and Reporting:

- Generating reports on inventory levels, order fulfillment rates, and warehouse efficiency.

- Analyzing data to identify trends, make informed decisions, and optimize warehouse operations.

9. Cross-Docking and Kitting:

- Facilitating the process of transferring goods directly from inbound to outbound transportation, reducing storage time.

- Creating product bundles (kits) to streamline order processing for items sold together.

10. Quality Control:

- Conducting regular quality checks to ensure products meet the required standards.

- Flagging and segregating defective or damaged items for appropriate action.

These functionalities collectively ensure the e-commerce warehouse operates at its full potential, enabling timely and accurate order fulfillment while maintaining optimal inventory levels and reducing operational costs.

Elicitation Technique:

1. Interviews

**Reason:** for gaining in-depth insights from individual stakeholders.And to understand all the aspects deeply

1. Questionare

**Reason:** gathering opinions, preferences, and general information.Collecting feedback from stakeholders on features.

1. Brainstorming

**Reason:** To present them new ideas and add in Requirements

1. Use case

**Reason:** Diagrammatically understand the interaction of system and their functionalities

In software engineering, elicitation techniques are used to gather requirements from stakeholders effectively. Different techniques are suitable for various situations and types of stakeholders. Here are some common elicitation techniques and when and where to use them:

1. \*\*Interviews:\*\*

- When to use: Interviews are best suited for gaining in-depth insights from individual stakeholders or small groups. They are ideal when you need to understand complex requirements, user needs, and preferences.

- Where to use: Conduct interviews during the early stages of the project or when you require a detailed understanding of specific aspects of the software.

2. \*\*Surveys and Questionnaires:\*\*

- When to use: Surveys are beneficial when you need to gather feedback from a large number of stakeholders simultaneously. They are useful for gathering opinions, preferences, and general information.

- Where to use: Surveys can be used at various stages of the project, such as during the initial requirements gathering phase or for collecting feedback on proposed features.

3. \*\*Workshops and Focus Groups:\*\*

- When to use: Workshops and focus groups are excellent for brainstorming sessions, fostering collaboration, and eliciting requirements from multiple stakeholders simultaneously.

- Where to use: Use workshops and focus groups during the early stages of the project to define high-level requirements and involve key stakeholders in the decision-making process.

4. \*\*Prototyping and Mockups:\*\*

- When to use: Prototypes and mockups are useful for eliciting requirements related to user interface and user experience (UI/UX). They provide stakeholders with a visual representation of the software's appearance and behavior.

- Where to use: Use prototyping and mockups when refining the user interface design or when gathering feedback on the proposed UI features.

5. \*\*Observation:\*\*

- When to use: Observation is valuable when requirements involve physical processes, user behavior, or existing systems that need improvement.

- Where to use: Use observation techniques in real-world settings, such as user environments, to understand the context and challenges users face.

6. \*\*Document Analysis:\*\*

- When to use: Document analysis is suitable for gathering existing information from various sources, such as reports, manuals, or previous project documentation.

- Where to use: Use document analysis in the early stages of the project to understand the existing system or to extract relevant data.

7. \*\*JAD (Joint Application Development) Sessions:\*\*

- When to use: JAD sessions are beneficial when you need to bring together stakeholders, subject matter experts, and developers for intensive requirements gathering and decision-making.

- Where to use: Use JAD sessions to resolve complex issues, prioritize requirements, or define the scope of the project.

The choice of elicitation technique depends on the project's characteristics, the stakeholders involved, the complexity of requirements, and the available resources. Often, a combination of different techniques is used throughout the software development life cycle to ensure comprehensive requirements gathering and a successful software product.

Questions:

Do you need a website or mobile app for the system

How are you managing the inverntory

Do you want order tracking

Which technology you want to use for order tracking

How are you sure that the product is delivered to correct destination

Do you want manual or robotic automation for picking and packaging

How do you organize stock with respect to storage requirements like weight and size

\*\*Scenario: E-commerce Website - Non-Functional Requirements Gathering\*\*

In this scenario, we are developing an e-commerce website for a fashion retail company. The website aims to provide an online platform for customers to browse and purchase clothing, accessories, and footwear. We will focus on gathering non-functional requirements for the website.

1. \*\*Performance Requirements:\*\*

- The website should load within 3 seconds on both desktop and mobile devices to ensure a smooth user experience.

- The website should be able to handle a minimum of 1000 concurrent users during peak hours without significant performance degradation.

- The server response time for any user action, such as adding items to the cart or checking out, should be less than 1 second.

2. \*\*Usability Requirements:\*\*

- The website's user interface should be intuitive and user-friendly, ensuring easy navigation and efficient product search.

- Font sizes and color contrast should adhere to accessibility standards to accommodate users with visual impairments.

- The website should have responsive design, providing a consistent experience across different devices and screen sizes.

3. \*\*Security Requirements:\*\*

- The website should implement SSL encryption to secure all data transmissions, especially during the checkout process.

- User authentication and authorization mechanisms should be in place to protect sensitive user information and ensure secure login and access to user accounts.

- Payment processing should comply with PCI DSS (Payment Card Industry Data Security Standard) requirements to ensure the safety of credit card information.

4. \*\*Reliability Requirements:\*\*

- The website should have a minimum uptime of 99.9% to minimize any service disruptions or downtime.

- Regular backups of the website's database and content should be performed to ensure data integrity and quick recovery in case of data loss.

5. \*\*Scalability Requirements:\*\*

- The website should be designed to accommodate future growth in terms of increased user traffic and product catalog size.

- The system should be scalable to handle a 50% increase in concurrent users during seasonal sales or promotions.

6. \*\*Compatibility Requirements:\*\*

- The website should be compatible with major web browsers, including Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.

- The website should support multiple operating systems, such as Windows, macOS, iOS, and Android.

7. \*\*Compliance Requirements:\*\*

- The website should adhere to relevant data protection and privacy regulations, such as GDPR (General Data Protection Regulation) if applicable.

- The website should comply with industry standards for e-commerce websites, ensuring trust and credibility for customers.

8. \*\*Maintainability Requirements:\*\*

- The website's code should be well-documented, following industry best practices and coding standards for easy maintenance and future enhancements.

- Modularity and separation of concerns should be considered during the development to facilitate code changes without affecting the entire system.

By effectively gathering and incorporating these non-functional requirements into the e-commerce website's design and development process, we can ensure that the final product meets the desired quality standards, delivers an exceptional user experience, and operates reliably and securely.

WMS contains

* Inventory Management
* Location Management
* Receiving and put away
* Picking
* Packing
* Sorting by category
* Loading
* Movement and Storage of material in warehouse
* Invoicing And dispatching
* Inventory Reconcilliation
* Manage Damage or expire products
* Manage Returns

**Picking:**

Barcode labels are applied on stock

**Receiving:**

When a new task is assign then a notification is sent to receiving team and

Pallets and put away

Once the order is packed the notification is sent to put away team

System guides the team to reserve location and pallets for order

ON packing put away team will place order and system will decide that order will be place in which bin which Stock Unit and what quantity can be placed

Location Suggestion Allocation

Reservations

SU no

Weight of the SU

If the order is placed at correct loacation notification is sent to Supervisor

And also for wrong location

Packing team will pack and forwards to dispatching team

And then forwards for delivery.

**Stakeholders:**

* Client
* Warehouse Manager
* Warehouse Supervisors
* Warehouse Workers
* Admin
* Inventory Managers
* IT Staff
* Data Entry Operators
* Order Management Team
* Quality Control Team
* Material Handlers
* Automated Systems
* Customer Service Representatives
* Reporting and Analytics Team

If a warehouse management system (WMS) is using an application for management, the actors involved would include both individuals and automated components interacting with the software. Here are the key actors in a warehouse management system that uses an application for management:

1. Warehouse Manager: Responsible for overseeing the entire warehouse operation and using the WMS application to monitor and make strategic decisions about inventory, staff allocation, and logistics.

2. Warehouse Supervisors: They utilize the WMS application to coordinate day-to-day activities, assign tasks to warehouse workers, and manage real-time workflows.

3. Warehouse Workers: These employees interact with the WMS application to carry out tasks such as receiving, picking, packing, and shipping goods based on the instructions provided by the system.

4. Inventory Managers: Utilize the WMS application to monitor and optimize inventory levels, track stock movement, and generate reports on inventory performance.

5. IT Staff: Responsible for the maintenance, configuration, and troubleshooting of the WMS application and its underlying infrastructure.

6. Data Entry Operators: Enter data into the WMS application, including product information, stock counts, and other relevant data to keep the system up-to-date.

7. Order Management Team: They manage and process customer orders through the WMS application, ensuring timely and accurate order fulfillment.

8. Quality Control Team: Uses the WMS application to perform quality checks on incoming and outgoing items, ensuring that products meet the required standards.

9. Material Handlers: These workers use the application to locate and move items within the warehouse, often with the aid of handheld devices or mobile terminals.

10. Automated Systems: The WMS application may interact with automated components such as conveyor belts, automated storage and retrieval systems (AS/RS), barcode scanners, and RFID readers to streamline warehouse operations.

11. Customer Service Representatives: They can access the WMS application to track orders, check inventory availability, and provide real-time updates to customers.

12. Reporting and Analytics Team: Uses the data gathered by the WMS application to generate reports, analyze warehouse performance, and identify areas for improvement.

With an application-based warehouse management system, the process becomes more efficient, accurate, and streamlined, as it allows for real-time visibility, automation, and better decision-making capabilities compared to manual systems.

Here are some use cases of the "Picker" role in a warehouse management system, presented in two verbs form:

1. \*\*Retrieve Items\*\*: The picker locates and collects items from the warehouse based on order details or internal requests.

2. \*\*Scan Barcodes\*\*: The picker uses barcode scanners to identify and verify items during the picking process.

3. \*\*Update Inventory\*\*: The picker updates the inventory system to reflect the items that have been picked and removed from the warehouse.

4. \*\*Organize Picked Items\*\*: The picker arranges and organizes the picked items in preparation for packing and shipping.

5. \*\*Collaborate with Packers\*\*: The picker communicates with packers to ensure a smooth handover of picked items for packaging.

6. \*\*Resolve Stock Discrepancies\*\*: The picker reports any stock discrepancies or damaged items found during the picking process.

7. \*\*Maintain Workspace\*\*: The picker keeps the picking area clean and organized to facilitate efficient operations.

8. \*\*Follow Safety Guidelines\*\*: The picker adheres to safety protocols while moving around the warehouse and handling items.

9. \*\*Prioritize Urgent Orders\*\*: The picker identifies and prioritizes orders with urgent delivery requirements.

10. \*\*Coordinate with Supervisors\*\*: The picker communicates with supervisors to get picking instructions and report completed tasks.

11. \*\*Utilize Warehouse Layout\*\*: The picker navigates through the warehouse, utilizing the knowledge of the warehouse layout to find items efficiently.

12. \*\*Optimize Picking Routes\*\*: The picker plans and optimizes picking routes to minimize travel time and increase productivity.

These use cases represent some of the key tasks and responsibilities of a picker in a warehouse management system. The picker plays a crucial role in ensuring timely and accurate order fulfillment, as well as maintaining efficient warehouse operations.

Here are some use cases of the "Packer" role in a warehouse management system, presented in two verbs form:

1. \*\*Pack Items\*\*: The packer carefully packages items for shipment, ensuring they are secure and protected during transit.

2. \*\*Scan and Verify\*\*: The packer uses barcode scanners to verify the contents of the package before sealing it.

3. \*\*Generate Labels\*\*: The packer generates and attaches shipping labels to packages for delivery.

4. \*\*Update Status\*\*: The packer updates the order status in the system to indicate that the items have been packed and are ready for dispatch.

5. \*\*Coordinate Dispatch\*\*: The packer communicates with the dispatcher to ensure timely pickup and delivery of packages.

6. \*\*Organize Packages\*\*: The packer organizes and groups packages for efficient loading and delivery.

7. \*\*Inspect Packaging\*\*: The packer inspects packaging materials for damage or defects, replacing them as needed.

8. \*\*Collaborate with Pickers\*\*: The packer communicates with pickers to clarify order details or confirm package contents.

9. \*\*Handle Fragile Items\*\*: The packer takes special care in handling and packing fragile or sensitive items.

10. \*\*Weigh and Measure\*\*: The packer weighs and measures packages to determine shipping costs and compliance with delivery regulations.

11. \*\*Use Packing Materials\*\*: The packer selects and uses appropriate packing materials, such as bubble wrap, packing peanuts, or cushioning, as required.

12. \*\*Follow Packing Instructions\*\*: The packer follows specific packing instructions for different types of products or customers.

These use cases illustrate some of the key tasks and responsibilities of a packer in a warehouse management system. The packer plays a crucial role in ensuring that items are properly prepared for shipment, and that packages are accurately labeled and ready for delivery.

Here are some use cases of the "Dispatcher" role in a warehouse management system, presented in two verbs form:

1. \*\*Receive Orders\*\*: The dispatcher receives incoming orders or requests for delivery from the warehouse management system.

2. \*\*Assign Vehicles\*\*: The dispatcher assigns suitable delivery vehicles to each order based on factors like package size, delivery location, and urgency.

3. \*\*Optimize Routes\*\*: The dispatcher optimizes delivery routes to minimize travel time and ensure efficient delivery schedules.

4. \*\*Coordinate Delivery\*\*: The dispatcher coordinates delivery schedules and updates with drivers and customers.

5. \*\*Track Shipments\*\*: The dispatcher tracks the status and location of shipments in real-time.

6. \*\*Communicate Updates\*\*: The dispatcher communicates delivery updates to customers or recipients.

7. \*\*Handle Exceptions\*\*: The dispatcher manages and resolves delivery exceptions or delays.

8. \*\*Prioritize Deliveries\*\*: The dispatcher prioritizes urgent deliveries and time-sensitive orders.

9. \*\*Monitor Fleet\*\*: The dispatcher monitors the availability and condition of delivery vehicles.

10. \*\*Assign Drivers\*\*: The dispatcher assigns drivers to specific delivery routes or tasks.

11. \*\*Update Delivery Status\*\*: The dispatcher updates the delivery status in the system as packages are dispatched and delivered.

12. \*\*Handle Customer Inquiries\*\*: The dispatcher handles customer inquiries related to deliveries and provides timely updates.

These use cases illustrate some of the key tasks and responsibilities of a dispatcher in a warehouse management system. The dispatcher plays a critical role in coordinating and ensuring the smooth flow of deliveries, optimizing routes, and effectively managing the fleet of delivery vehicles.

If we use TCS (Tata Consultancy Services) as a delivery module in a warehouse management system, here are some use cases represented in two verbs form:

1. \*\*Initiate Delivery\*\*: The system initiates the delivery process with TCS for selected orders.

2. \*\*Track Shipment\*\*: The system tracks the shipment status and location through TCS's tracking services.

3. \*\*Update Delivery Status\*\*: TCS updates the delivery status in the system as the shipment progresses.

4. \*\*Generate Shipping Labels\*\*: The system generates and provides shipping labels to TCS for package identification.

5. \*\*Calculate Shipping Cost\*\*: The system calculates the shipping cost for each shipment based on TCS's pricing structure.

6. \*\*Schedule Pickup\*\*: The system schedules a pickup with TCS for the packages ready for dispatch.

7. \*\*Confirm Delivery\*\*: TCS confirms the successful delivery of packages, updating the system accordingly.

8. \*\*Handle Exceptions\*\*: The system and TCS collaborate to handle any delivery exceptions, such as delays or failed deliveries.

9. \*\*Provide Delivery Updates\*\*: TCS provides real-time delivery updates to the system, ensuring customers are informed.

10. \*\*Manage Returns\*\*: The system facilitates return pickups and coordination with TCS for reverse logistics.

11. \*\*Maintain Delivery Records\*\*: The system maintains records of past deliveries and shipment history with TCS.

12. \*\*Coordinate Delivery Time\*\*: The system coordinates delivery time windows with TCS to ensure efficient deliveries.

These use cases represent the interactions and functionalities related to TCS as the delivery module in the warehouse management system. TCS, being a well-known logistics service provider, can integrate with the system to handle the actual delivery and shipment aspects, providing end-to-end delivery solutions.

1. \*\*Create storage zones:\*\* In this use case, users can use the application to create new storage zones within the warehouse management system. They can specify the zone's name, location, capacity, and any relevant attributes. Once created, the storage zone becomes available for organizing and managing inventory items efficiently.

2. \*\*Modify storage zones:\*\* Users can modify existing storage zones using the application. This includes updating zone details, such as changing the capacity, location, or attributes. Additionally, they can rename zones, merge or split them as needed, adapting the warehouse's layout to better suit the current inventory requirements.

3. \*\*Assign items to storage zones:\*\* This use case enables users to assign inventory items to specific storage zones within the warehouse. They can utilize the application to scan or input item information and associate it with the appropriate storage zone. This ensures that items are stored in their designated areas, streamlining the retrieval process and minimizing errors.

4. \*\*Track inventory movement:\*\* The application allows users to track the movement of inventory items between different storage zones. This use case helps in monitoring stock levels and identifying areas with higher or lower inventory turnover rates. Users can view historical data on item transfers and optimize the warehouse layout for improved efficiency.

5. \*\*Generate storage reports:\*\* Users can generate various storage reports using the application, providing insights into the warehouse's zone utilization and overall inventory management. Reports may include information on capacity utilization, empty or nearly full zones, and trends in stock movement over time.

6. \*\*Manage zone permissions:\*\* With the application, administrators can manage permissions for accessing and modifying storage zones. They can assign different levels of access to warehouse personnel, ensuring that only authorized personnel can make changes to zone configurations or item assignments.

7. \*\*Implement safety protocols:\*\* Users can leverage the application to implement safety protocols in storage zones. For instance, they can mark certain zones as hazardous or flammable, set restrictions on specific items' storage, or define temperature-controlled zones to maintain the quality of sensitive goods.

8. \*\*Perform zone audits:\*\* The application facilitates zone audits to ensure proper inventory placement and adherence to storage guidelines. Users can conduct regular audits to verify the accuracy of item assignments, detect misplaced or mislabeled items, and maintain warehouse organization.

9. \*\*Optimize storage efficiency:\*\* Using the application's analytics and reporting features, users can identify areas of improvement within the warehouse's storage layout. By analyzing data on inventory movement and utilization, they can make informed decisions to optimize storage efficiency, reduce congestion, and minimize travel distances for warehouse staff.

10. \*\*Integrate with inventory systems:\*\* The application can integrate with other inventory and warehouse management systems. This allows for seamless data synchronization, ensuring that storage zone information remains up-to-date across various platforms and reducing the likelihood of data discrepancies.

1. \*\*Record damage incidents:\*\* In this use case, users can utilize the application to record instances of damaged inventory within the warehouse management system. They can input details such as item descriptions, quantities affected, and the cause of damage. By recording this information, the system keeps a comprehensive record of damaged inventory for further analysis and action.

2. \*\*Assess damage severity:\*\* Users can assess the severity of the damage to each affected item using the application. They can indicate whether an item is salvageable, requires repair, or needs to be disposed of entirely. This assessment helps prioritize actions and determine the best course of action for handling the damaged inventory.

3. \*\*Initiate damage alerts:\*\* The application can automatically generate alerts or notifications when damage incidents are recorded. This allows warehouse staff and management to promptly respond to damaged inventory and take necessary actions, such as isolating affected items, conducting inspections, or initiating repair or replacement processes.

4. \*\*Coordinate inspection process:\*\* Using the application, users can coordinate and schedule inspections of damaged inventory. They can assign tasks to designated personnel, set inspection deadlines, and track the status of inspection activities. This ensures that damaged items are thoroughly examined to determine appropriate next steps.

5. \*\*Document repair or disposal actions:\*\* When damaged inventory undergoes repair or disposal, the application allows users to document these actions. They can record repair details or specify the method of disposal (e.g., recycling, landfill). This documentation helps maintain a clear audit trail and supports decision-making regarding future handling of similar incidents.

6. \*\*Manage inventory reallocation:\*\* In cases where damaged items can still be utilized with minor adjustments, the application enables users to manage inventory reallocation. They can update item attributes, such as reduced quantities, altered specifications, or new storage locations, ensuring that repaired or partially damaged inventory is appropriately assigned.

7. \*\*Track repair progress:\*\* The application provides tools to track the progress of repairing damaged inventory. Users can monitor repair timelines, costs, and other relevant details. This allows for better resource planning and helps ensure that repaired items are made available for use as quickly as possible.

8. \*\*Analyze damage trends:\*\* By analyzing the recorded data on damage incidents, the application allows users to identify trends and patterns. This analysis can help pinpoint areas of concern, such as frequent damage occurrences, specific product vulnerabilities, or potential issues with handling procedures.

9. \*\*Generate damage reports:\*\* Users can generate comprehensive damage reports using the application. These reports may include statistics on the frequency of damage, the cost of repairs and disposals, and recommendations for process improvements. The reports assist in making informed decisions to reduce damage incidents and associated costs.

10. \*\*Implement preventive measures:\*\* Based on insights gained from damage trend analysis, users can implement preventive measures using the application. They can update warehouse procedures, invest in better packaging materials, or provide additional training to staff to minimize future damage incidents and enhance overall inventory protection.

1. \*\*Initiate return requests:\*\* In this use case, users can initiate return requests for inventory items using the application within the warehouse management system. They can input relevant details, such as the reason for the return, item descriptions, quantities, and customer information (if applicable). By initiating return requests through the application, the process becomes more streamlined and trackable.

2. \*\*Track return status:\*\* Users can track the status of return requests through the application. This includes monitoring the progress of returned items from the moment the request is initiated until the items are received and processed in the warehouse. Tracking return status helps ensure timely and accurate handling of returned inventory.

3. \*\*Authorize return approvals:\*\* The application allows authorized personnel to review and approve return requests. They can verify the reason for the return, check item conditions, and determine whether the return meets the company's policies for acceptance. Once approved, the return process moves forward accordingly.

4. \*\*Coordinate return logistics:\*\* Using the application, users can coordinate the logistics of returning invent ory items. This involves arranging for return shipping or pickup, generating return labels, and ensuring that the items are sent to the correct warehouse locations for inspection and processing.

5. \*\*Inspect returned items:\*\* Warehouse staff can use the application to conduct inspections of returned items. They can assess the condition of the returned inventory, check for damages, and verify if all necessary components or accessories are included. The inspection process helps in determining appropriate actions, such as restocking, refurbishing, or disposing of returned items.

6. \*\*Update inventory status:\*\* Once the returned items are inspected and processed, users can update the inventory status in the warehouse management system through the application. This includes adjusting stock levels, marking items as available for resale, or sending them for repair or refurbishment if needed.

7. \*\*Manage customer communications:\*\* The application facilitates communication with customers throughout the return process. Users can use the application to send automated notifications, updates on return status, or resolution details, ensuring customers are well-informed about the progress of their return requests.

8. \*\*Generate return analytics:\*\* The application can generate return analytics and reports, providing insights into the frequency and reasons for returns. This data helps identify potential issues with products, packaging, or customer service, enabling the organization to implement improvements to reduce returns in the future.

9. \*\*Handle return refunds or replacements:\*\* If applicable, the application can manage return refunds or replacements. It allows users to process refund payments or generate replacement orders for customers who are eligible for such resolutions.

10. \*\*Integrate with customer support:\*\* The application can integrate with customer support systems, ensuring a seamless flow of information between support agents and warehouse personnel. This integration helps in addressing customer inquiries, clarifying return details, and resolving any potential discrepancies during the return process.

1. "Scan and Retrieve" Use Case:

Title: Scan and Retrieve Items in Warehouse

Description: This use case outlines the process of retrieving items from a warehouse using a mobile application equipped with barcode scanning capabilities.

Actors:

- Warehouse Staff

- Mobile Application

Preconditions:

- The mobile application is installed and functional on the staff's mobile device.

- The warehouse contains items with unique barcode labels.

Steps:

1. Warehouse staff logs into the mobile application using their credentials.

2. The staff selects the "Retrieve Items" option from the application's main menu.

3. The application activates the mobile device's camera for barcode scanning.

4. The staff positions the mobile device's camera in front of the barcode label on the item they want to retrieve.

5. The application scans the barcode and identifies the item based on the barcode data.

6. The application retrieves the item details, such as its name, location, and quantity in stock, from the database.

7. The application displays the item details on the mobile device screen.

8. If there are multiple items of the same type in different locations, the application provides a list of available locations to choose from.

9. The staff selects the appropriate location from the list (if applicable).

10. The application generates a retrieval confirmation, including the item name, quantity, and location (if applicable).

11. The staff verifies the retrieval information on the mobile device.

12. If everything is correct, the staff clicks the "Confirm" button to initiate the retrieval process.

13. The application updates the warehouse inventory in real-time, marking the item as "being retrieved."

14. The staff proceeds to the specified location and picks up the item.

15. Once the item is retrieved, the staff returns to the application and clicks the "Complete" button to finalize the retrieval process.

16. The application updates the warehouse inventory again, marking the item as "retrieved" and adjusting the stock quantity accordingly.

17. The staff can repeat the process to retrieve more items or choose to log out of the application when finished.

Postconditions:

- The warehouse inventory is updated to reflect the retrieved items.

- The staff has successfully retrieved the requested items.

2. "Search and Retrieve" Use Case:

Title: Search and Retrieve Items in Warehouse

Description: This use case outlines the process of searching for and retrieving specific items from a warehouse using a web-based application with search functionality.

Actors:

- Warehouse Staff

- Web-based Application

Preconditions:

- The web-based application is accessible and functional on the staff's computer or mobile device.

- The warehouse contains items with identifiable attributes stored in the application's database.

Steps:

1. Warehouse staff accesses the web-based application through a web browser on their computer or mobile device.

2. The staff logs into the application using their credentials.

3. The application presents a search bar where the staff can enter relevant item details, such as item name, SKU, or category.

4. The staff enters the item details they wish to retrieve into the search bar and clicks the "Search" button.

5. The application queries the database based on the search criteria and retrieves matching items.

6. The application displays a list of items that match the search criteria, along with their corresponding details (e.g., name, quantity, location).

7. The staff reviews the list and selects the desired item they want to retrieve by clicking on it.

8. The application provides additional details about the selected item, such as its current stock level and location in the warehouse.

9. The staff confirms that the selected item is the one they intend to retrieve.

10. If there are multiple items of the same type in different locations, the application provides a list of available locations to choose from.

11. The staff selects the appropriate location from the list (if applicable).

12. The application generates a retrieval confirmation, including the item name, quantity, and location (if applicable).

13. The staff verifies the retrieval information on the application interface.

14. If everything is correct, the staff clicks the "Confirm" button to initiate the retrieval process.

15. The application updates the warehouse inventory in real-time, marking the item as "being retrieved."

16. The staff proceeds to the specified location and picks up the item.

17. Once the item is retrieved, the staff returns to the application and clicks the "Complete" button to finalize the retrieval process.

18. The application updates the warehouse inventory again, marking the item as "retrieved" and adjusting the stock quantity accordingly.

19. The staff can repeat the process to retrieve more items or choose to log out of the application when finished.

Postconditions:

- The warehouse inventory is updated to reflect the retrieved items.

- The staff has successfully retrieved the requested items.

1. "Pack and Verify" Use Case:

Title: Pack Items in Packing Department of Warehouse

Description: This use case outlines the process of packing items in the packing department of a warehouse using a mobile application equipped with packing and verification functionalities.

Actors:

- Warehouse Packer

- Mobile Application

Preconditions:

- The mobile application is installed and operational on the packer's mobile device.

- The picking process has been completed, and the items to be packed are ready for packaging.

Steps:

1. The warehouse packer logs into the mobile application using their credentials.

2. The packer selects the "Pack Items" option from the application's main menu to initiate the packing process.

3. The application displays a list of items that need to be packed, along with their details, such as item name, quantity, and order number.

4. The packer verifies that the items displayed on the mobile device match the items they have in front of them for packing.

5. The packer selects the first item from the packing list and begins the packing process.

6. The application prompts the packer to select the appropriate packaging materials (e.g., box, envelope) and enters the package's tracking or shipping information if required.

7. The packer carefully places the item into the selected packaging material, ensuring it is securely packed to prevent any damage during transit.

8. The application allows the packer to take a photo of the packed item (optional) to document the packaging process and provide evidence of the item's condition before shipping.

9. The packer repeats steps 5 to 8 for each item in the packing list, following the same packaging procedures for each item.

10. Once all items have been packed, the application prompts the packer to verify the contents of each package against the packing list to ensure accuracy.

11. The packer reviews the list and checks the packed items against the order details displayed on the mobile device.

12. If there are any discrepancies or missing items, the packer can adjust the packing list accordingly and update the application with the correct information.

13. Once the packing list is verified and accurate, the packer clicks the "Complete" button to finalize the packing process.

14. The application updates the order status and the warehouse inventory in real-time, marking the items as "packed" and adjusting the stock quantity accordingly.

15. If the packing process requires further steps, such as labeling or sealing the packages, the application may provide additional instructions or prompts to guide the packer through those tasks.

16. After the packing process is complete, the packer can either proceed to the shipping department for further processing or log out of the application.

Postconditions:

- The items are successfully packed and ready for further processing.

- The warehouse inventory is updated to reflect the packed items.

2. "Multi-Item Packing" Use Case:

Title: Pack Items in Packing Department of Warehouse (Multi-Item Packing)

Description: This use case outlines the process of packing multiple items together in the packing department of a warehouse using a mobile application with multi-item packing functionality.

Actors:

- Warehouse Packer

- Mobile Application

Preconditions:

- The mobile application is installed and functioning on the packer's mobile device.

- The picking process has been completed, and the items to be packed are ready for packaging.

Steps:

1. The warehouse packer logs into the mobile application using their credentials.

2. The packer selects the "Multi-Item Packing" option from the application's main menu to initiate the multi-item packing process.

3. The application displays a list of orders that require multi-item packing, along with the respective items and quantities to be packed together.

4. The packer selects the first order from the packing list and begins the multi-item packing process.

5. The application prompts the packer to select the appropriate packaging materials (e.g., box, container) suitable for packing all the items together.

6. The packer carefully places all the items from the selected order into the chosen packaging material, arranging them securely to prevent any damage during transit.

7. The application allows the packer to take a photo of the packed items (optional) to document the packing process and provide evidence of the items' condition before shipping.

8. The packer repeats steps 4 to 7 for each order requiring multi-item packing, following the same packaging procedures for each order.

9. Once all orders with multi-item packing requirements are packed, the application prompts the packer to verify the contents of each package against the packing list to ensure accuracy.

10. The packer reviews the list and checks the packed items against the order details displayed on the mobile device.

11. If there are any discrepancies or missing items, the packer can adjust the packing list accordingly and update the application with the correct information.

12. Once the packing list is verified and accurate, the packer clicks the "Complete" button to finalize the multi-item packing process.

13. The application updates the order status and the warehouse inventory in real-time, marking the items as "packed" and adjusting the stock quantity accordingly.

14. If the packing process requires further steps, such as labeling or sealing the packages, the application may provide additional instructions or prompts to guide the packer through those tasks.

15. After the multi-item packing process is complete, the packer can either proceed to the shipping department for further processing or log out of the application.

Postconditions:

- The multi-item orders are successfully packed and ready for further processing.

- The warehouse inventory is updated to reflect the packed items.

1. "Generate and Print Labels" Use Case:

Title: Generate Labels in Packing Department of Warehouse

Description: This use case outlines the process of generating and printing labels for packed items in the packing department of a warehouse using a dedicated label generation application.

Actors:

- Warehouse Packer

- Label Generation Application

- Label Printer

Preconditions:

- The label generation application is installed and operational on the packer's computer or mobile device.

- The packing process has been completed, and items are ready for labeling.

Steps:

1. The warehouse packer logs into the label generation application using their credentials.

2. The packer selects the "Generate Labels" option from the application's main menu to initiate the label generation process.

3. The application displays a list of packed items that require labels, along with their respective details, such as item name, quantity, and order number.

4. The packer selects the first item from the list to generate a label.

5. The application prompts the packer to choose the label format and template based on the packaging requirements and shipping standards.

6. The packer enters any necessary information, such as shipping address, tracking number, or special handling instructions, to be included on the label.

7. The packer reviews the label preview on the application to ensure all information is accurate and correctly formatted.

8. Once satisfied, the packer clicks the "Generate Label" button to create the label.

9. The label generation application communicates with the label printer to send the label data for printing.

10. The label printer prints the generated label according to the chosen format and template.

11. The packer attaches the printed label securely to the corresponding package, ensuring it is prominently displayed and visible for shipping or handling.

12. The packer repeats steps 4 to 11 for each item that requires labeling.

13. After all labels have been generated and attached to the packages, the packer reviews the packing list and labels to verify accuracy and completeness.

14. If any discrepancies or errors are identified, the packer can make corrections in the label generation application and reprint the labels as needed.

15. Once the labeling process is complete and accurate, the packer can proceed to the shipping department for further processing or log out of the application.

Postconditions:

- The packed items are labeled with accurate and properly formatted labels.

- The packages are ready for shipping or handling with the generated labels.

2. "Auto-Generate Labels" Use Case:

Title: Auto-Generate Labels in Packing Department of Warehouse

Description: This use case outlines the process of automatically generating and printing labels for packed items in the packing department of a warehouse using an application with auto-generate label functionality.

Actors:

- Warehouse Packer

- Label Generation Application

- Label Printer

Preconditions:

- The label generation application is installed and operational on the packer's computer or mobile device.

- The packing process has been completed, and items are ready for labeling.

Steps:

1. The warehouse packer logs into the label generation application using their credentials.

2. The packer selects the "Auto-Generate Labels" option from the application's main menu to initiate the automated label generation process.

3. The application retrieves the list of packed items that require labels, along with their corresponding details, such as item name, quantity, and order number.

4. The packer reviews the list to ensure it accurately reflects the items that need labels.

5. The packer confirms the list and provides any additional information required for labeling, such as shipping addresses or special handling instructions.

6. The application processes the information and automatically generates labels for each packed item based on predefined label templates and standards.

7. The label generation application communicates with the label printer to send the label data for printing.

8. The label printer prints the generated labels according to the predefined formats and templates.

9. The labels are automatically dispensed or collected for the packer's retrieval.

10. The packer attaches the printed labels securely to the corresponding packages, ensuring they are prominently displayed and visible for shipping or handling.

11. The packer reviews the packing list and labels to verify accuracy and completeness.

12. If any discrepancies or errors are identified, the packer can address them through the application and initiate a reprint of the affected labels.

13. Once the labeling process is complete and accurate, the packer can proceed to the shipping department for further processing or log out of the application.

Postconditions:

- The packed items are labeled with automatically generated, accurately formatted labels.

- The packages are ready for shipping or handling with the auto-generated labels.

1. "Receive and Verify Order" Use Case:

Title: Receive Order in Dispatching Department of Warehouse

Description: This use case outlines the process of receiving and verifying an incoming order in the dispatching department of a warehouse using a mobile application.

Actors:

- Warehouse Dispatcher

- Mobile Application

Preconditions:

- The mobile application is installed and operational on the dispatcher's mobile device.

- The order has been packed, labeled, and is ready for dispatch.

Steps:

1. The warehouse dispatcher logs into the mobile application using their credentials.

2. The dispatcher selects the "Receive Order" option from the application's main menu to initiate the order receiving process.

3. The application displays a list of pending orders that are ready for dispatch, along with their order numbers and relevant details.

4. The dispatcher selects the first order from the list to receive and verify.

5. The application presents the details of the selected order, including the customer's information, shipping address, items in the order, and their quantities.

6. The dispatcher compares the displayed information with the physical order and checks for any discrepancies or irregularities.

7. If the order details are accurate, the dispatcher confirms the receipt by clicking the "Receive" button.

8. The application updates the order status to "Received" and marks it as ready for final processing.

9. The dispatcher repeats steps 4 to 8 for each pending order in the list.

10. If any discrepancies or issues are identified, the dispatcher can make necessary adjustments in the application or contact relevant personnel to address the concerns.

11. Once all pending orders have been received and verified, the dispatcher can review the complete list of received orders on the mobile device screen.

12. The dispatcher can generate and print a receipt or confirmation document for each received order, if required.

13. The application may provide options for additional actions, such as generating shipping labels, assigning delivery routes, or initiating the loading of the received orders onto delivery vehicles.

14. After completing the necessary post-receipt tasks, the dispatcher can choose to proceed with further order processing or log out of the application.

Postconditions:

- The pending orders are received, verified, and ready for final processing and dispatch.

- The order status is updated to reflect the successful receipt.

2. "Batch Order Receiving" Use Case:

Title: Batch Order Receiving in Dispatching Department of Warehouse

Description: This use case outlines the process of batch receiving multiple orders simultaneously in the dispatching department of a warehouse using a mobile application.

Actors:

- Warehouse Dispatcher

- Mobile Application

Preconditions:

- The mobile application is installed and operational on the dispatcher's mobile device.

- Multiple orders have been packed, labeled, and are ready for dispatch.

Steps:

1. The warehouse dispatcher logs into the mobile application using their credentials.

2. The dispatcher selects the "Batch Order Receiving" option from the application's main menu to initiate the batch order receiving process.

3. The application displays a list of pending orders that are ready for dispatch, along with their order numbers and relevant details.

4. The dispatcher selects multiple orders from the list that are part of the same batch for receiving.

5. The application presents a summary of the selected orders, including the total number of orders, items, and quantities in the batch.

6. The dispatcher reviews the summary to ensure the correct orders are included in the batch.

7. If the batch summary is accurate, the dispatcher confirms the receipt of the batch by clicking the "Receive Batch" button.

8. The application updates the status of the selected orders to "Received" and marks them as ready for final processing.

9. The dispatcher can view a detailed breakdown of the received orders within the batch and check for any discrepancies or issues.

10. The dispatcher can generate and print a receipt or confirmation document for the entire received batch, if required.

11. The application may provide options for additional actions, such as generating shipping labels, assigning delivery routes, or initiating the loading of the received orders onto delivery vehicles for the batch.

12. After completing the necessary post-batch receipt tasks, the dispatcher can choose to proceed with further order processing or log out of the application.

Postconditions:

- The selected orders within the batch are received, verified, and ready for final processing and dispatch.

- The order statuses are updated to reflect the successful batch receipt.