**Warehouse Inventory Management System Project**

Software Requirements Specification (SRS)

Version <1.0>

**Preface**

**Version History:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Author/Contributors** |
| 1.0 | 17Aug2023 | Initial SRS | Craig, Derek, Humberto, Jose A. Otero Perez |

1. **Introduction**

The Warehouse Inventory Management System (WIMS) is designed to streamline and optimize inventory management processes for general use. This document outlines the functional and non-functional requirements for the first iteration of the WIMS. The system aims to efficiently manage inventory levels, process orders, and provide tools for auditing inventory. It will be a web-based application developed using .NET Core, C#, MySQL, and Apache web server technologies.

* 1. **Purpose**

The purpose of this Software Requirements Document (SRS) is to define the scope of the Warehouse Inventory Management System and to establish a detailed set of requirements that will guide the development, implementation, and testing of the system. This document serves as a blueprint for all stakeholders involved, including developers, testers, project managers, and end-users. It ensures a common understanding of the system's features and behavior, mitigating misinterpretations and facilitating effective communication throughout the software development lifecycle.

* 1. **Scope**

The Warehouse Inventory Management System aims to address the challenges associated with efficiently managing inventory within a warehouse environment. The system will provide a user-friendly interface for warehouse personnel, administrators, and managers to effectively monitor, control, and optimize inventory operations. It will encompass a range of functionalities, including but not limited to:

User authentication and authorization to ensure secure access to the system.

Creation and management of client accounts with relevant details.

Definition and management of inventory items, including attributes such as name, description, and SKU.

Tracking of item quantities, statuses, and categories.

Management of orders with associated items and quantities.

Generation of reports for inventory, orders, and other relevant metrics.

The system's primary goal is to enhance warehouse efficiency, minimize errors, and provide real-time visibility into inventory levels and operations. It will be developed with scalability and extensibility in mind, allowing for future enhancements and integrations.

1. **Glossary**

WIMS: Warehouse Inventory Management System

Inventory: Stock of goods and materials

Order: A request for items or products

Pick: The process of selecting items from inventory for an order

Ship: The process of dispatching orders to customers

1. **User Requirements Definition**
   1. User Authentication

User should be able to provide their identity to gain access to the application

* + 1. The system shall enforce password policy and verify the identity of users attempting to log in.
    2. The system should provide password lockout and recovery mechanisms.
  1. User Authorization

The user should be able to access controls and functionality according to their assigned role and permissions.

* + 1. After authentication, the system shall determine the appropriate level of access a user has based on their role.
    2. The system shall make controls, forms, and data available according to the user’s permissions.
    3. The system should allow administrators to update permissions without modifying the code.
  1. Create and update a client.

Users should have access to a form or interface where they can create or edit client details, such as name, contact information, billing information, enable or disable the client, and other relevant data.

* + 1. The system shall enable authorized users to access a form to create new client accounts or edit existing client details.
    2. The system shall generate and assign a unique client ID to each new client.
    3. The system shall perform data validation checks to ensure that entered client information is accurate, consistent, and complete.
    4. The system should reformat client information to accommodate for uniformity of data.
    5. The system should provide user with confirmation option if a possible user duplicity is entered.
  1. Create and update an item.

Authorized users should be able to define and enter item details such as name, description, SKU (Stock Keeping Unit), category, and other relevant attributes, assign items to an existing client, and update existing item details.

* + 1. The system shall enable authorized users to access a form to create new items or edit existing items.
    2. The system shall generate and assign a unique client ID to each new item.
    3. The system shall validate that item is associated to an existing client.
    4. The system shall perform data validation checks to ensure that entered item information is accurate, consistent, and complete.
    5. The system should reformat client information to accommodate for uniformity of data.
    6. The system should prevent the user from creating the same item multiple times.
    7. The system should allow bulk item creation through data import.
  1. Submit order

Customers should be able to submit new orders through the system.

* + 1. The system shall provide an intuitive interface for customers to select items, specify quantities, enter shipping information, and confirm orders.
    2. The system shall integrate with the product catalog to display up-to-date information on product availability, prices, and descriptions.
    3. The system shall perform validation checks on order details to ensure accuracy and completeness.
    4. The system shall ensure secure transmission of customer information during the order process.
  1. Process order

Users shall be able to view and manage order details, including items, quantities, and shipping information.

* + 1. The system shall provide authorized users with the ability to view detailed information about orders, including items, quantities, prices, and shipping information.
    2. The system shall provide authorized users with the ability to modify order details, subject to constraints such as order status and inventory availability.
    3. The system shall implement access controls to ensure that only authorized users can view or manage specific order details.
  1. Order picking

The user should access a list of order that are ready for the picking process and update each order with the picked quantity.

* + 1. The system shall allow the user to navigate a queue of orders that are ready to be picked
    2. The system shall maintain orders on the picking queue until a user selects an order for processing
    3. The system shall have a form to display the order's item name, description, location, and required quantity
    4. The system shall allow the user to input the picked quantity for each item and update the order once picking is completed with the requested quantity vs the picked quantity
    5. The system should update the order status once picking has started and completed
    6. The system should be integrated with an email service and send an email to the customer with each order status update
  1. Order shipping

The user should be able to update order shipping information and generate shipping labels

* + 1. The system shall auto populate the customer details based on the selected customer by the user
    2. The system shall have a form where the user can enter the order shipping information
    3. The system shall be able to generate a printable format of the shipping label
    4. The system should validate user manual input for preset format, such as phone number (no special characters), and valid email address (include “@” and “.com”).
    5. The system should be integrated with an email service capability to send an order update notification to the customer
  1. Inventory management

Users should be able to view the current inventory levels of all products. Users should be able to search for specific products by name, category, or item id.

* + 1. The system shall maintain accurate records of inventory items, including names, descriptions, quantities, and re-order points.
    2. The system shall allow users with appropriate permissions to add, edit, or remove inventory items.
    3. The system should generate alerts when inventory items fall below their re-order points.
    4. The system should be integrated with an email service capability to send low inventory alerts to selected users and clients
  1. Inventory audits

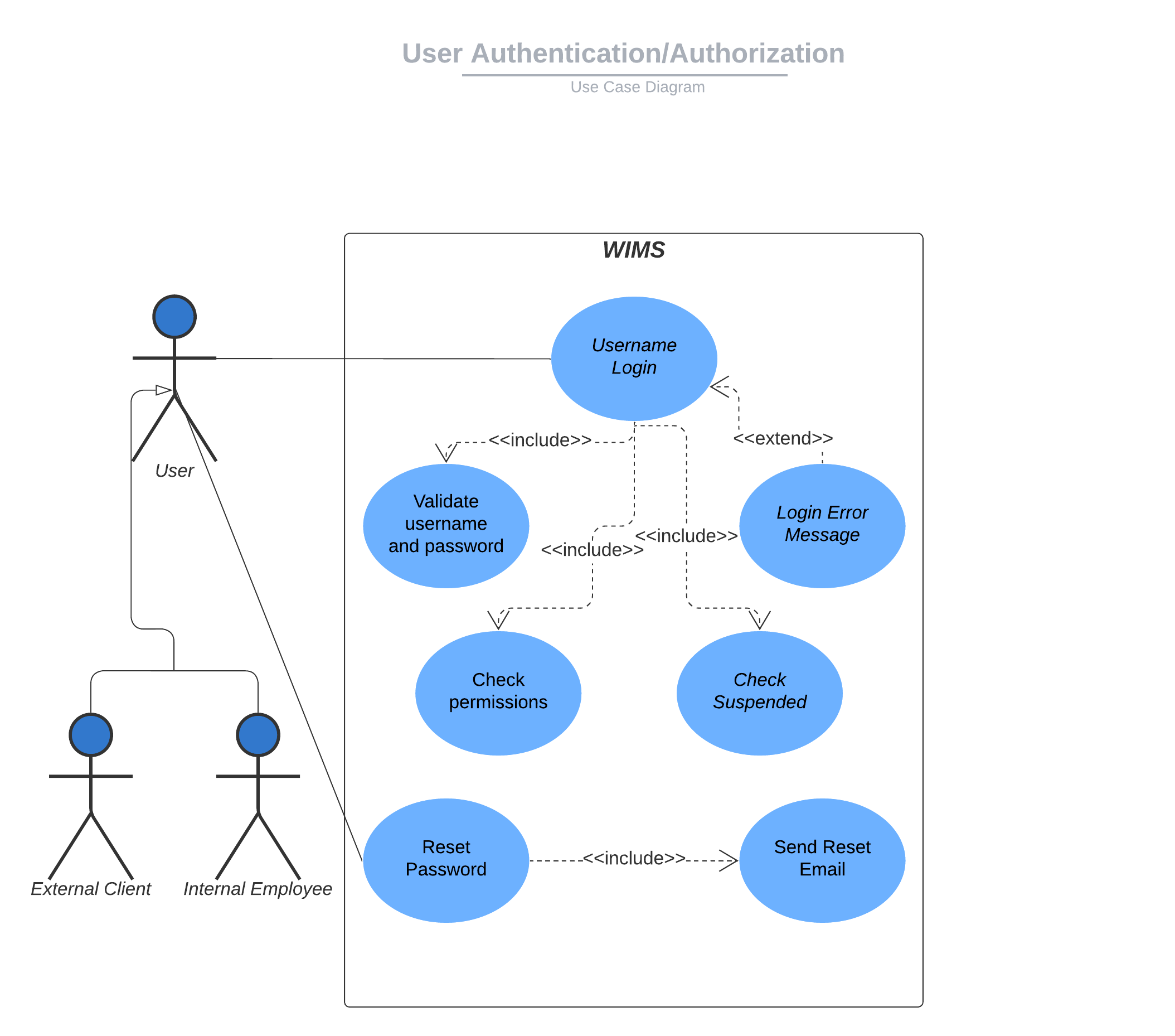
Users should be able to initiate and schedule cyclic inventory audits and generate audit reports with details of any discrepancies between recorded and counted quantities. Users should be able to reconcile physical counts with recorded inventory.

* + 1. The system shall allow authorized users to schedule or initiate inventory audits for a selected client and its inventory based on percentage, item type or item value.
    2. The system shall generate an operational pause for the selected client until the inventory audit is completed.
    3. The system shall provide users with the list of the selected item with the item name, description and location.
    4. The system shall allow the user to input the amount of inventory found for the selected item.
    5. The system shall generate a report reflecting the amount and rate of discrepancy of inventory for each item.
    6. The system shall allow authorized users to adjust reconsolidate levels of inventory.
    7. The system should allow users to use barcode scanning to conduct inventory audits.

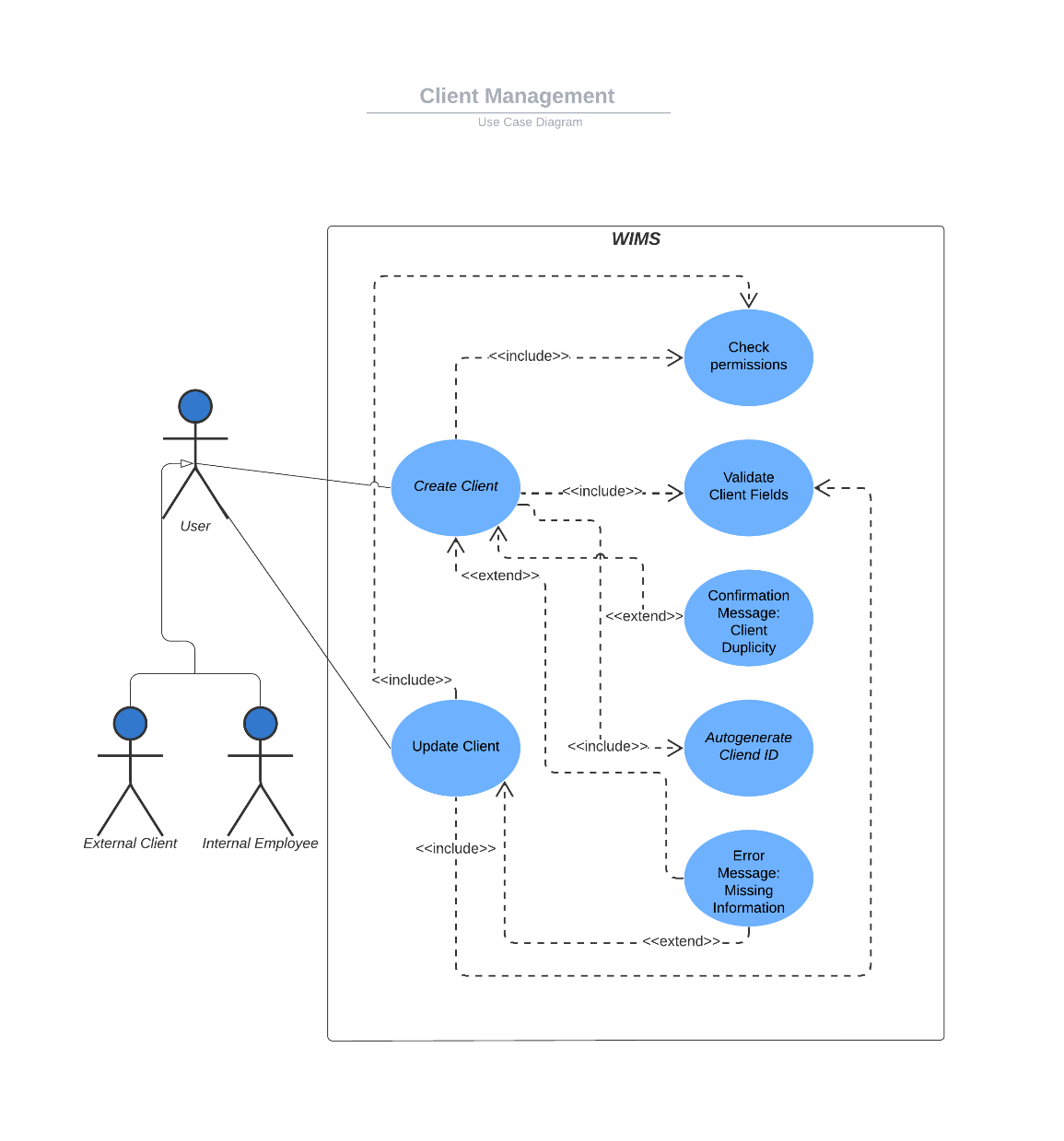
1. **System Architecture**

The WIMS will follow a three-tier architecture, consisting of a presentation layer, business logic layer, and data access layer. The presentation layer will be a web-based user interface developed using HTML5, CSS3, and JavaScript. The business logic layer will be implemented using .NET Core and C#, handling core functionalities such as order processing and inventory management. The data access layer will utilize a relational database to store inventory and order information.

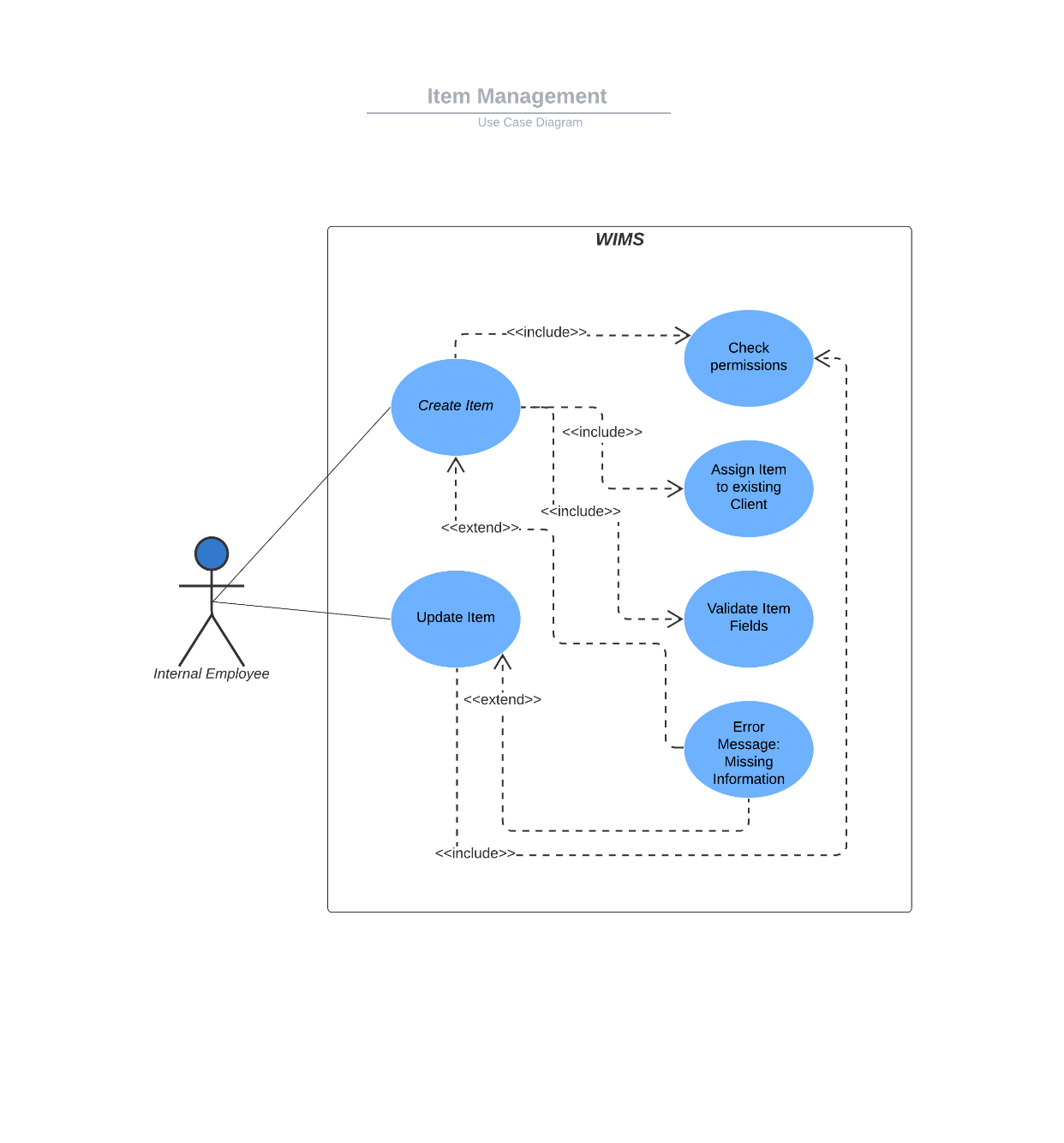
1. **System Requirements**
   1. **Functional**
      1. User Authentication and Authorization
         1. Users must be able to log in using unique credentials.
         2. Different user roles (e.g., admin, warehouse staff, and customers) should have varying levels of access and permissions.
      2. Inventory Management
         1. The system should allow users to view the current inventory levels of all products.
         2. Users should be able to search for specific products by name, category, or SKU.
      3. Order Tracking
         1. Users should be able to track the status of their orders, from placement to shipment.
         2. The system should provide real-time updates on order processing stages.
      4. Order History
         1. Users should have access to their order history, including details of past orders, quantities, and dates.
         2. Historical order data should be searchable and filterable.
      5. Reorder Point Management
         1. The system should allow administrators to set reorder points for each product.
         2. Automatic alerts should be generated when inventory falls below reorder points.
      6. Cyclic Inventory Audits
         1. Administrators should be able to initiate and schedule cyclic inventory audits.
         2. Audit reports should detail any discrepancies between recorded and counted quantities.
      7. Reporting and Analytics
         1. The system should provide various reports, including inventory levels, order statistics, and audit results.
         2. Reports should be exportable in common formats (e.g., PDF, CSV).
   2. **Nonfunctional**
      1. Performance
         1. The system shall be capable of handling concurrent user interactions without significant performance degradation.
         2. Response times for critical functions (order processing, inventory updates) shall not exceed 3 seconds
      2. Usability
         1. The user interface shall be intuitive and user-friendly, requiring minimal training for end users.
         2. Error messages shall be clear and informative, aiding users in understanding and resolving issues.
      3. Security
         1. User authentication and authorization mechanisms shall be implemented to ensure data security.
         2. Personally identifiable information (PII) and payment information shall not be stored within the system.
      4. Scalability
         1. The system architecture shall be designed to accommodate potential growth in the number of users and inventory items.
      5. Reliability
         1. The system shall maintain data integrity and consistency during power outages or unexpected shutdowns.
      6. Compatibility
         1. The system shall be compatible with popular web browsers, including Chrome, Firefox, and Edge.
      7. Integration
         1. The system should integrate with existing warehouse management tools, if any, to avoid duplication of data.
2. **System Models**
   1. Use Case Models
      1. User Authentication/Authorization



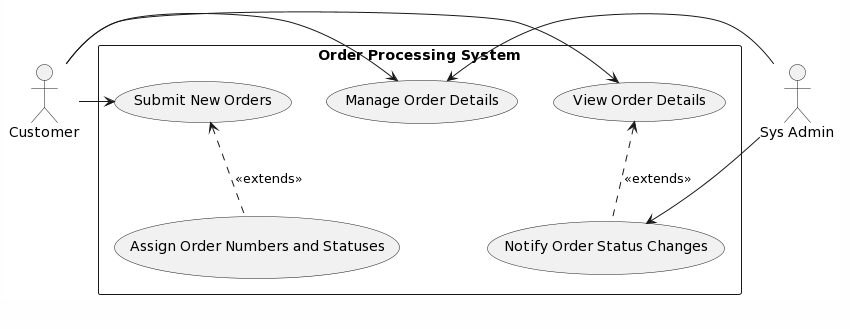
* + 1. Create and update client



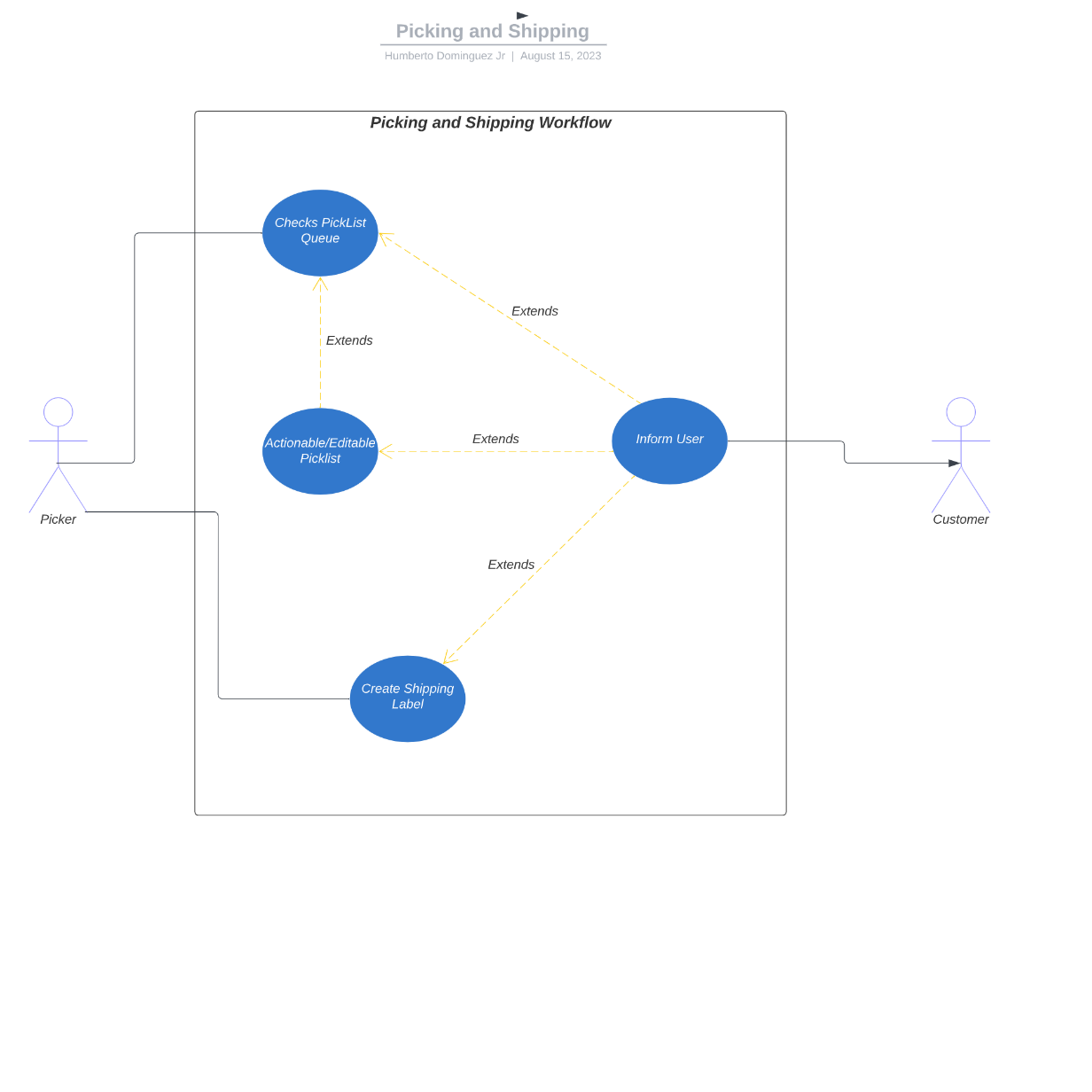
* + 1. Create and update item



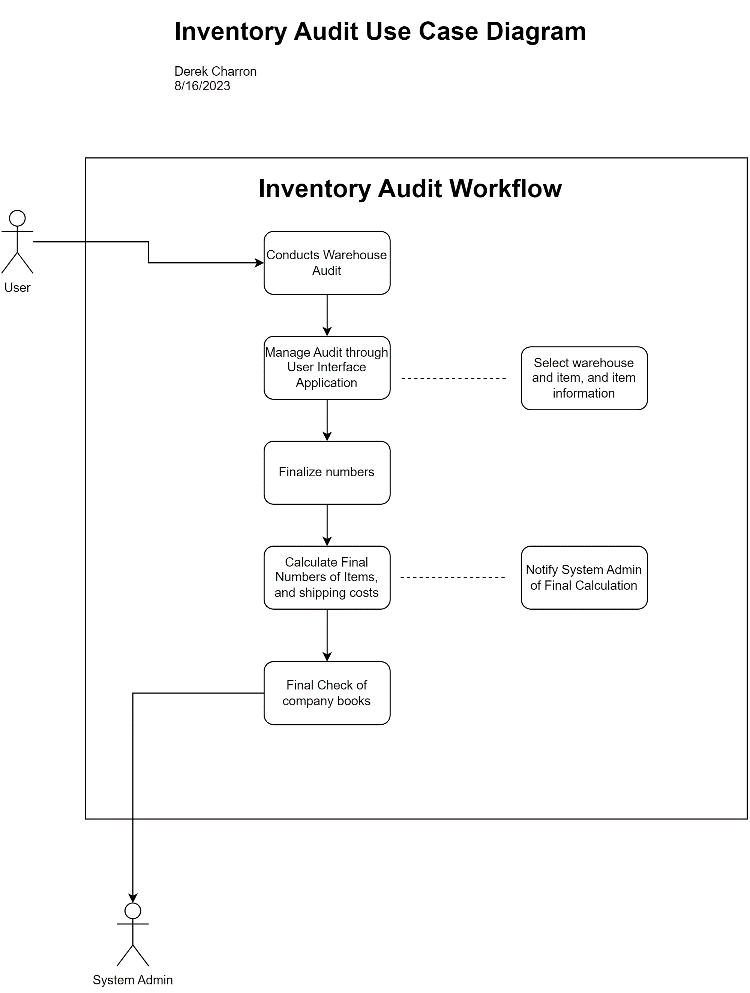
* + 1. Order management

****

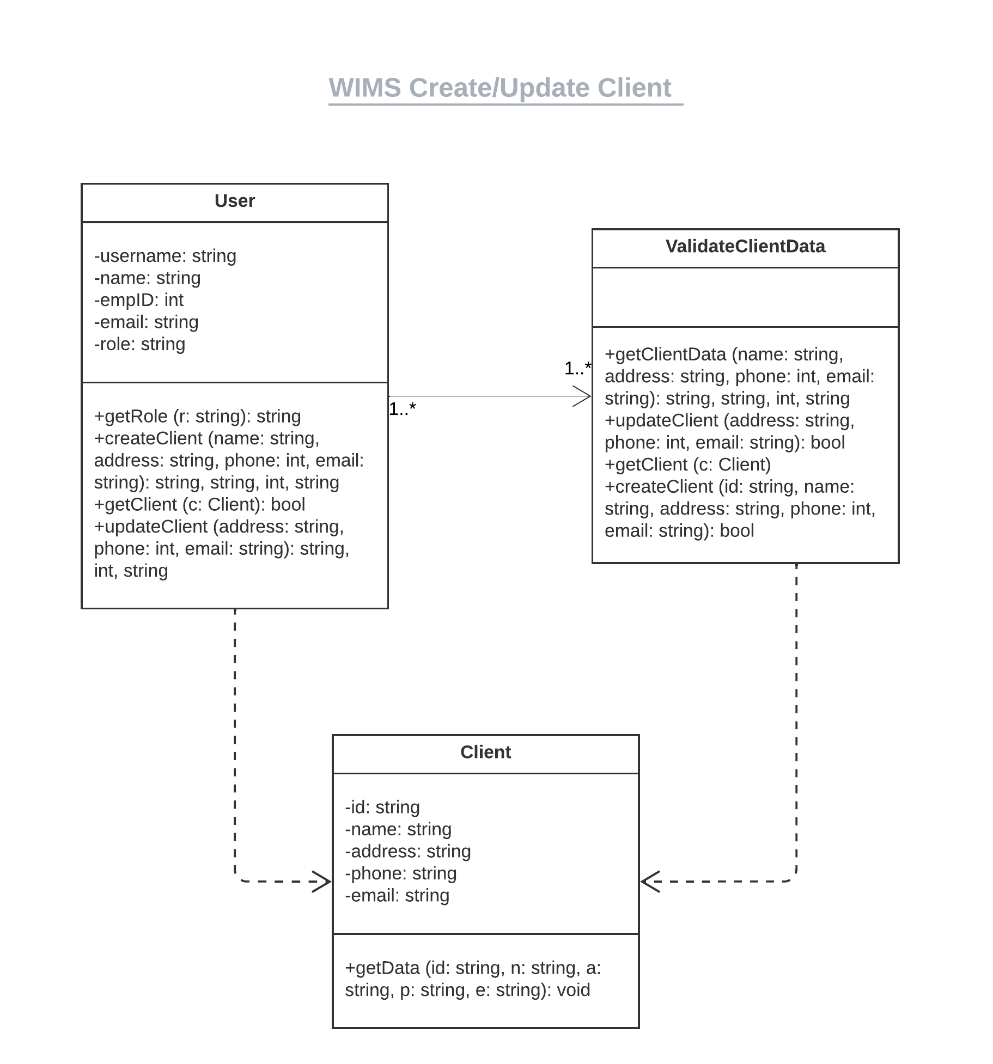
* + 1. Order picking and shipping

****

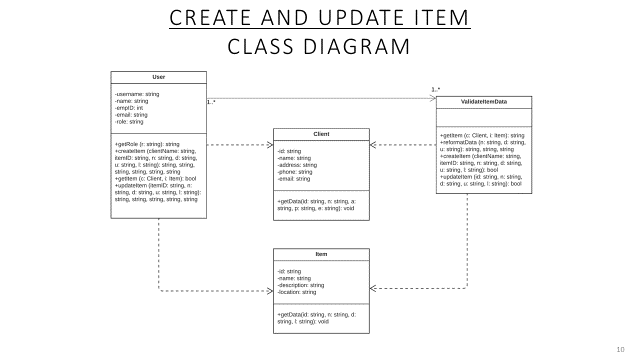
* + 1. Inventory audit

****

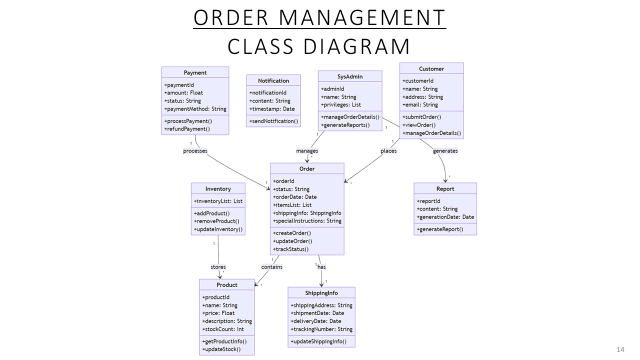
* 1. Class Models
     1. Create and update client

****

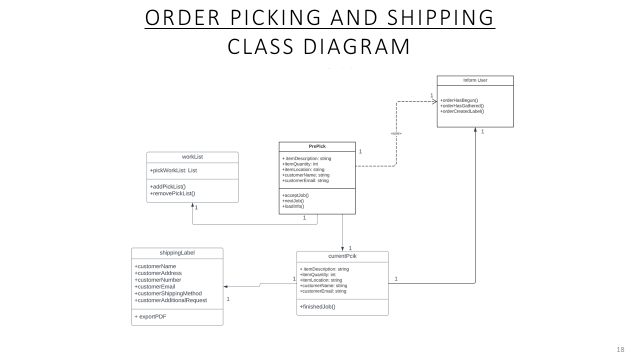
* + 1. Create and update item

****

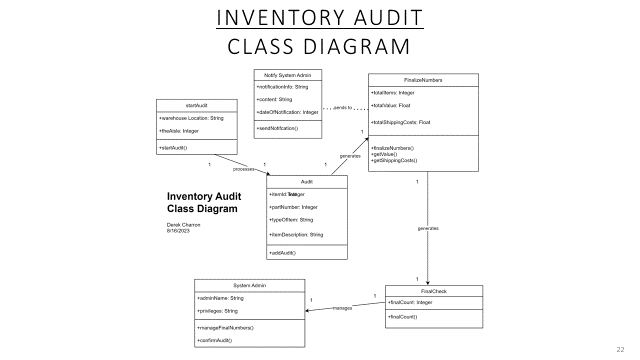
* + 1. Order management

****

* + 1. Order picking and shipping

****

* + 1. Inventory audit



1. **System Evolution [Pending]**

The WIMS is designed with flexibility in mind to accommodate future changes and enhancements. Potential hardware evolution, user feedback, and business needs will influence future system updates.

1. **Appendices [Pending]**
2. Hardware Requirements
   1. Minimum and optimal hardware configurations for server and client devices.
3. Database Requirements
   1. Logical organization of data tables, relationships, and indexes.
4. **Index [Pending]**
5. Index