

# Final Deliverable: Complete Software Documentation Package

By: Jose Flores, Andrew Puig, Cesar Collazo, Carlos Cuadra

1. Software Requirements Document (SRD):

## Table Of Contents

Cover page

Introduction

Purpose of the Software

Scope of the Software

Functional Requirements

Non-functional Requirements

Use Cases

Individual Contribution/ Task distributions

Project Schedule (using Gantt Chart)

References

## **Introduction**

The proposed software project from Group 10 is a full-stack digital inventory management system called Investorage, designed specifically for small-to-medium-sized warehouses and retail operations. The goal is to streamline how businesses manage, track, and analyze their inventory by eliminating overly complex legacy software, minimizing human errors, and offering a minimal learning curve.

## **Purpose of the Software**

Easy to understand system to allow for a user-friendly interactive menu for small and medium companies.

A well developed tracking system for the inventory, which will allow for a fast and efficient tracking of resources

Real Time insight, which will update the items current placement throughout the movement of multiple users; ensuring the last location of the item is its current placement in the facility

Algorithm technique, which will ensure we choose the most effective algorithm to search the database for minimum latency.

Provide a clean and user-friendly interface to small and medium enterprises (SMEs)

Enable fast, accurate, and real-time inventory tracking and updates

Ensure that stock movements are traceable across user roles with timestamped logs

Optimize database interactions using fast search, filtering, and grouping logic

## **Scope of the Software**

### **Included:**

- Real-time inventory visibility
- Role-based access (Admin and Staff)
- CRUD operations for inventory
- Export/Import inventory order processing
- Low-stock alerts
- Inventory report generation and filtering (7, 30, 90 days, etc.)
- Secure login and signup system
- Multi-user session handling with group ownership

### **Excluded**

- Payroll, wages, or employee schedules
- Customer management or CRM integration
- Sales or revenue management
- Barcode scanning or RFID hardware integration

## **Functional Requirements**

### **Core Functional Modules**

- The system must allow users to sign up, log in, and be assigned a role (Admin or Staff).
- Users must be able to add new inventory items with name, SKU, stock, description, and status.
- Users must be able to edit and update item quantities, stock levels, or status.
- Admins must be able to remove inventory items from the database.
- Staff and Admins must be able to search for items using keywords or filters.

### **Order Management**

- The system must allow Admins to import orders via CSV or JSON.
- The system must allow Admins to export selected inventory items and quantities.
- All imported items must appear under a generated Order ID and be editable.

## **Notification System**

- Low stock warnings should be triggered automatically when inventory drops below the LowStockWarning threshold.
- Notifications must appear visually in the dashboard.

## **Reporting**

- Admins must be able to generate reports by category or date range.
- Reports should include current inventory, changes (add, edit, remove), and import/export summaries.
- Admins must be able to export reports in PDF format.

## **Role Management**

- Admins have full access (add/edit/delete/export/report).
- Staff can view and edit inventory but cannot delete or generate exports/reports.
- Role-based access must be enforced server-side and reflected in the UI.

## **User-Friendly Interface**

- The system should have an intuitive and easy-to-navigate interface for any kind of user.

## **Audit Logs & Activity Tracking**

- Maintain logs of all user actions, stock updates, and order changes for security and compliance.
- Allow admins to review and export logs for audits.

## **Non-functional Requirements**

- Users must be authenticated with a secure login process.
- Group passwords and user credentials must be stored securely.
- SQL queries must be sanitized using prepared statements to prevent SQL injection.(Done with prepared statements like \$stmt->prepare() and bind\_param())
- Role-based access must be enforced for all features.
- The application must function across major browsers (Chrome, Firefox, Safari).
- The UI should be fully responsive
- The codebase must follow modular MVC structure to simplify updates.
- All major features must be documented in a help module.
- The system must support real-time interactions and multi-user concurrency.
- The system must maintain uptime and performance even during high traffic.
- Inventory updates and search queries should complete in under 2 seconds.
- The system should have error-handling mechanisms to recover from minor failures without user intervention.
- Implement role-based access control
- Implement multi-factor authentication for administrative access
- The system should be well-documented
- The system should have a clean, intuitive, and responsive UI for easy navigation
- Users should be able to perform common actions (e.g., adding stock, generating reports) in a few clicks.
- The system should be able to support multiple warehouses/locations without performance degradation.
- Must provide 24/7 system availability
- This will be a Web Application

## **Tech Stack:**

- Frontend: HTML, CSS, Bootstrap
- Backend: PHP
- Database: MySQL
- Hosting: HostGator Shared/Cloud Linux Server

## **Database Tables**

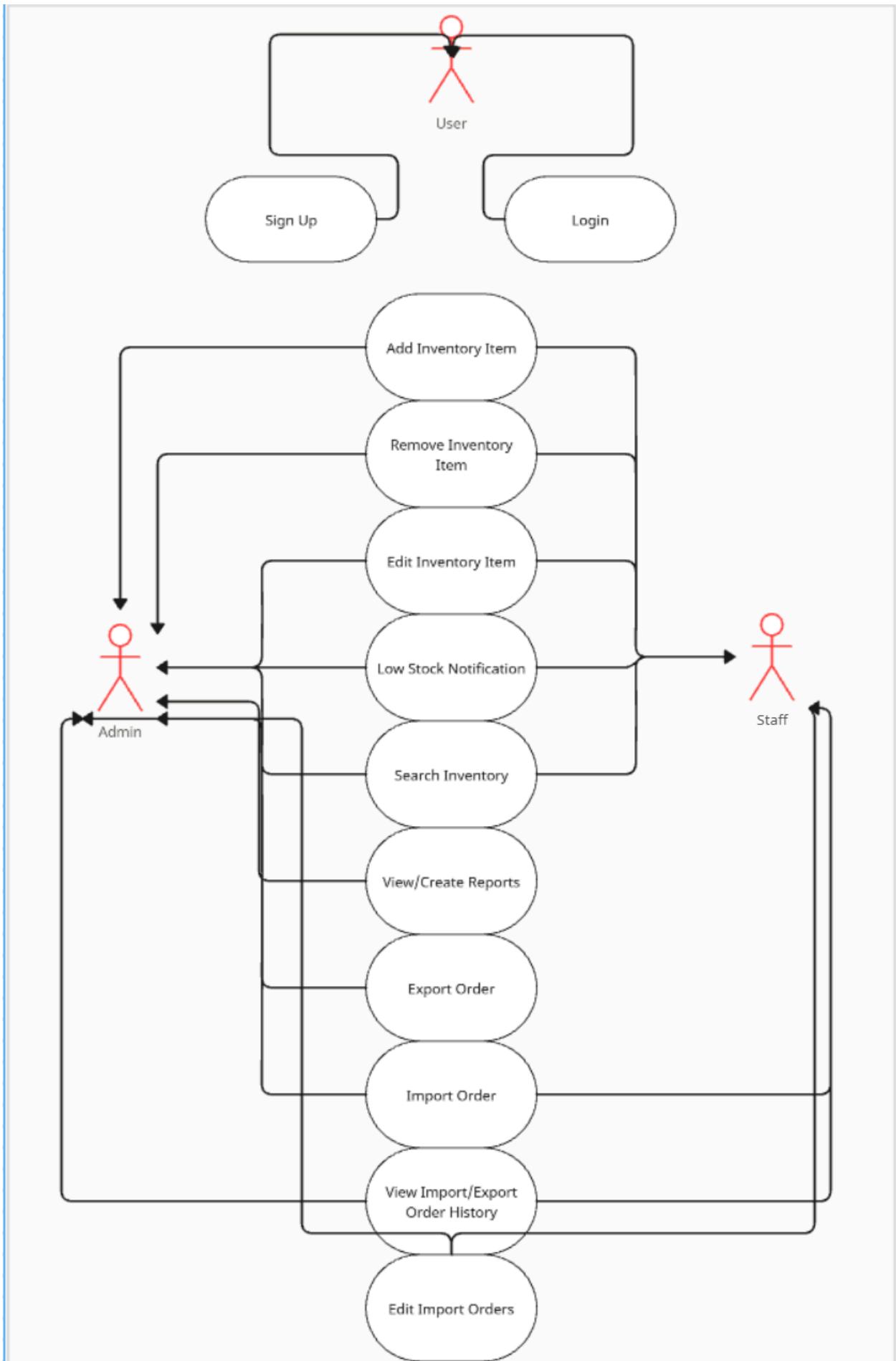
- Inventory Items:
  - **SKU** - The item's SKU Number
  - **Name** - The Name of the item

- **Category** - Category the Item belongs to
  - **Description** - A Brief Description of the Item
  - **SalesPrice** - The price the item is sold for
  - **Stock** - The number of items in stock
  - **Status**: InStock, Ordered, Backordered, Reserved, Dropped
  - **LowStockWarning** - The number representing the low stock warning threshold.
- 
- **OrderItems Table:**
    - **OrderID** - Associated order ID
    - **SKU** - Item SKU
    - **Quantity** - Quantity ordered
    - **UserID** - Creator of the order
- 
- **ExportOrders Table:**
    - **ExportOrderID** - Unique ID for export transaction
    - **ExportDate** - Date of export
    - **Destination** - Destination location
    - **ExportedBy** - Exporting user's name
    - **TotalItems** - Count of exported items
    - **TotalValue** - Sum of item sales prices \* quantity
    - **GroupID** - Associated warehouse group
- 
- **ExportOrderItems Table:**
    - **ExportOrderID** - Linked export ID
    - **SKU** - Item SKU
    - **Category** - Item category
    - **QuantityExported** - Exported quantity
- 
- **InventoryLog Table:**
    - **LogID** - Unique change ID
    - **SKU** - Item SKU
    - **ChangeType** - Type of change (Add, Remove, Edit, Export)
    - **OldValue** - Previous value (for updates)
    - **NewValue** - New value (for updates)

- **Timestamp** - Time and date of change
  - **GroupID** - Associated warehouse group
- 
- **Users Table:**
    - **UserID** - Unique user ID
    - **FirstName** - First name
    - **LastName** - Last name
    - **Email** - Email address
    - **Role** - Admin or Staff
    - **GroupID** - Associated warehouse group
  
  - **WarehouseGroups Table:**
    - **GroupID** - Unique group ID
    - **GroupName** - Group display name
    - **GroupPassword** - Group access password

## Use Cases

- The following diagram illustrates the various points where the specified user can interact with the system:



## Individual Contribution/ Task distributions

- Carlos Cuadra -Authentication page (full stack development)
  - Develops the Login and Sign-Up pages(including database for user information storage)
  - Implements role-based access control for Admin, Manager, and Staff users.
- Andrew- Add and remove inventory, (full stack development)
  - Develops the add, remove inventory functionalities for stock management.
  - Develops database for inventory
- Jose- Update and notifications, (full stack development)
  - Implements automated notifications for low stock, order status updates, and expiration warnings.
- Develops update functionality for inventory management.
- Cesar- Search and reporting, (full stack development)
  - Develops advanced search functionalities, allowing users to locate specific inventory items..
  - Implements the reporting system, generating inventory summaries and stock movement analytics.

## Project Schedule (using Gantt Chart)

<https://jrfz0937s-team.monday.com/boards/8459436110>

Task	February 6	February 13	February 20	February 27	March 6	March 13	March 20	March 27	March 27	April 3	April 10	April 17
Proposal Planning												
Obtain Hosting and Domain												
Log in and Sign Up												
Add and Remove Inventory Items												
Update Inventory Items and Send Notifications												
Search and Reports												
Project Demo Submission												

## References

Petersen, Kai, et al. "Guidelines for Conducting Systematic Mapping Studies in Software Engineering: An Update." *Information and Software Technology*, vol. 64, 2015, pp. 1–18, <https://doi.org/10.1016/j.infsof.2015.03.007>.

Lenberg, Per, et al. "Behavioral Software Engineering: A Definition and Systematic Literature Review." *The Journal of Systems and Software*, vol. 107, 2015, pp. 15–37, <https://doi.org/10.1016/j.jss.2015.04.084>. Lenberg, Per, et al. "Behavioral Software Engineering: A Definition and Systematic Literature Review." *The Journal of Systems and Software*, vol. 107, 2015, pp. 15–37, <https://doi.org/10.1016/j.jss.2015.04.084>.

Stol, Klaas-Jan, and Brian Fitzgerald. "The ABC of Software Engineering Research." *ACM Transactions on Software Engineering and Methodology*, vol. 27, no. 3, 2018, pp. 1–51, <https://doi.org/10.1145/3241743>.

Mahmoud, Anas, and Grant Williams. "Detecting, Classifying, and Tracing Non-Functional Software Requirements." *Requirements Engineering*, vol. 21, no. 3, 2016, pp. 357–81, <https://doi.org/10.1007/s00766-016-0252-8>.

## 2.Design Document

March 18, 2025

By: Jose Flores, Andrew Puig, Cesar Collazo, Carlos Cuadra

### Table of Contents

1. Introduction
2. High-Level Architecture
3. Detailed System Design
4. Algorithms
5. Interface Design
6. Security Considerations
7. Algorithms (Summary)
8. Future Enhancements

## **1. Introduction**

The most important part of any warehouse is the inventory section which needs the most attention, some inventory tracking software are difficult to use and navigate which is where Investorage comes in. Investorage is a web-based inventory management system designed for small to medium-sized warehouses. Investorage aims to provide a clean, intuitive, and highly efficient inventory system with features like real-time tracking, stock alerts, expiration tracking, and supplier order management.

## **2. High-level Architecture:**

The website is built on a Model-View-Controller pattern.

The Controller is based on PHP. Here we have HTML, CSS and Bootstrap saved to PHP variables that can be sent to the web browser dynamically. This function of sending the HTML and supporting code creates the View.

After elements are loaded to the web browser, a user can submit forms when displayed. On submit of these forms a post method will be activated and PHP will receive the request.

PHP will then route the request to the Model using a mysql request. The Model, which is based on MYSQL will then respond using a mysql response.

Once the PHP has received the response, it will save response data to user SESSION variables and update internal variables. With the new data, PHP logic will run and when required, will send a new View for display.

Through this MVC flow, the user can progress dynamically through the website.

## **Organization of Major Components**

### **1. Client Layer (View)**

- Web-based interface built with HTML, Bootstrap, and JavaScript.
- Delivers responsive forms and dashboards.
- Accepts user inputs and displays server-processed output.

### **2. Controller Layer**

- Built in PHP; receives and processes all client requests.
- Handles input validation, routing, session management, and role-based access control.
- Contains dedicated controller scripts for features such as login, inventory updates, order management, and reports.

### **3. Model Layer**

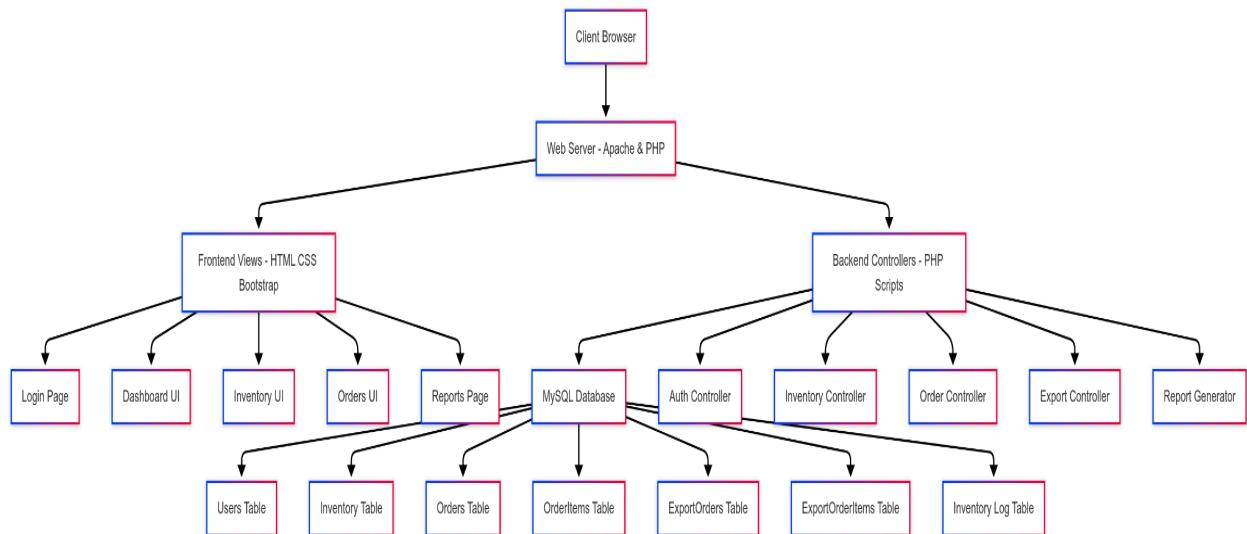
- Uses MySQL for persistent data storage.
- Each feature interacts with normalized database tables such as Inventory, Orders, ExportOrders, and InventoryLog.
- Controllers access models via secure prepared statements.

### **4. Session and Role Management**

- After login, session variables (e.g., Role, userID, GroupID) define access scope.
- Admin users have full access, while Staff roles are restricted to editing and viewing operations.

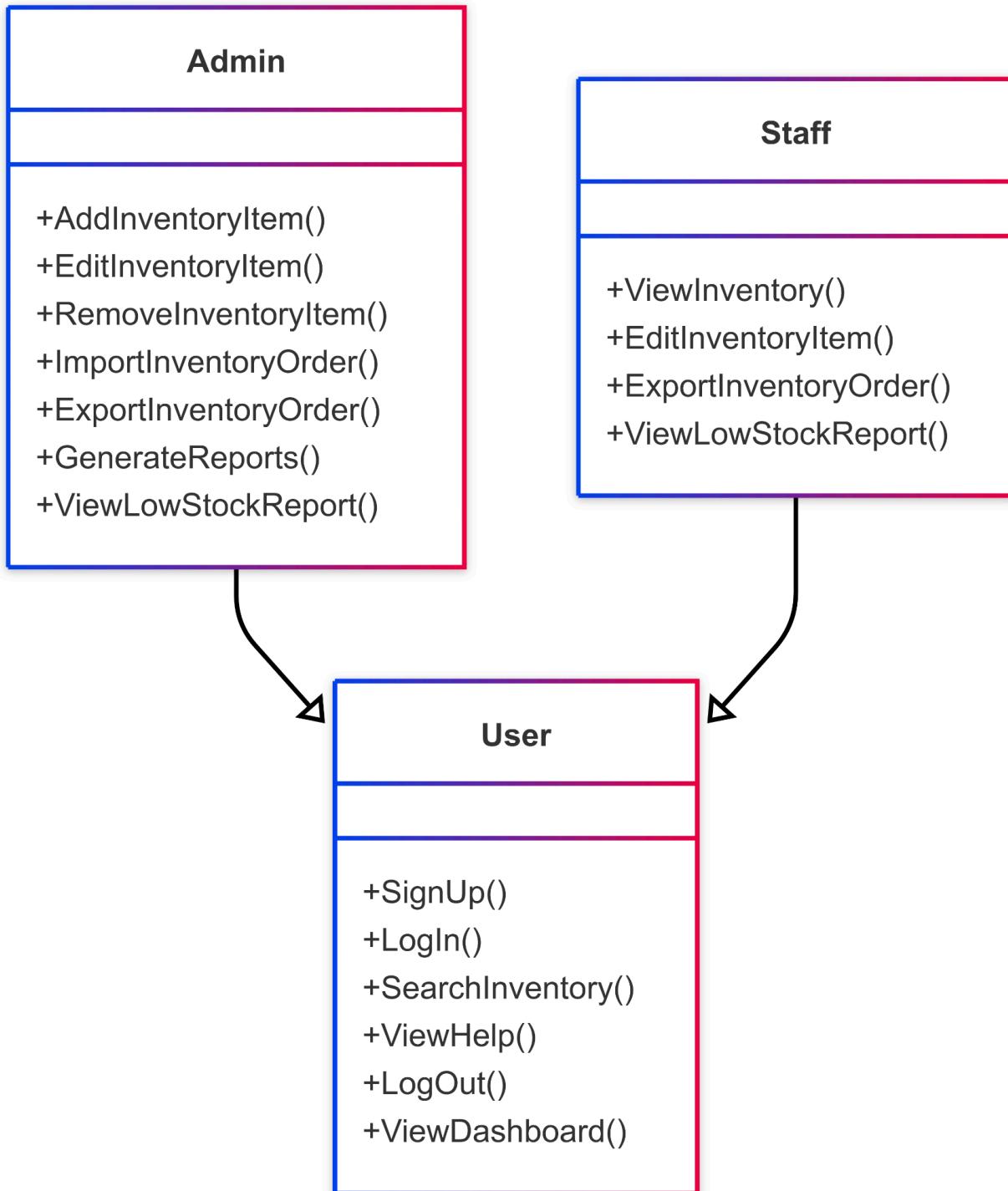
## 5. Report & Notification Integration

- Reporting modules fetch data from InventoryLog to generate filtered reports.
- Low-stock notifications are derived directly from live inventory tables based on thresholds.

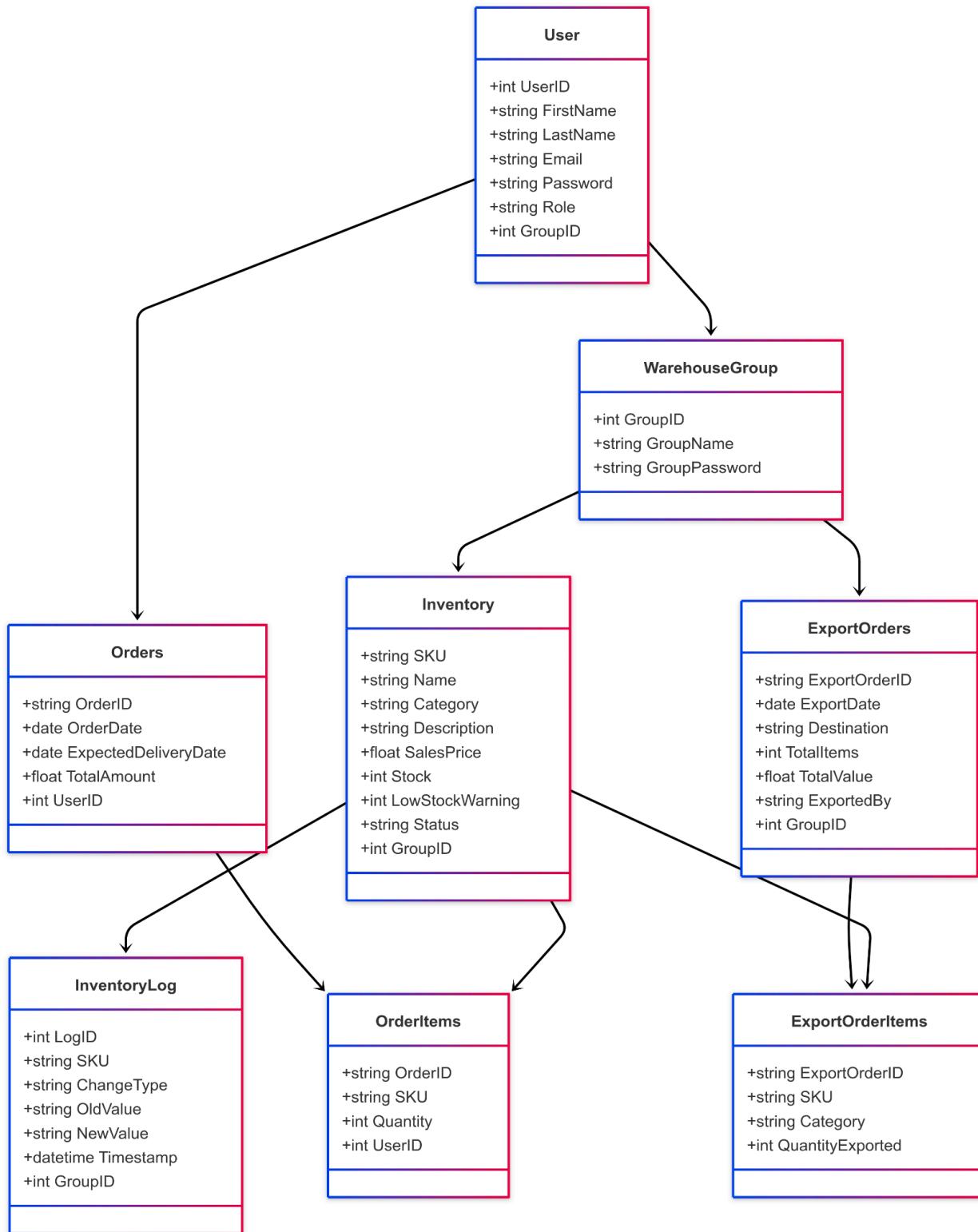


## 3.Detailed System Design

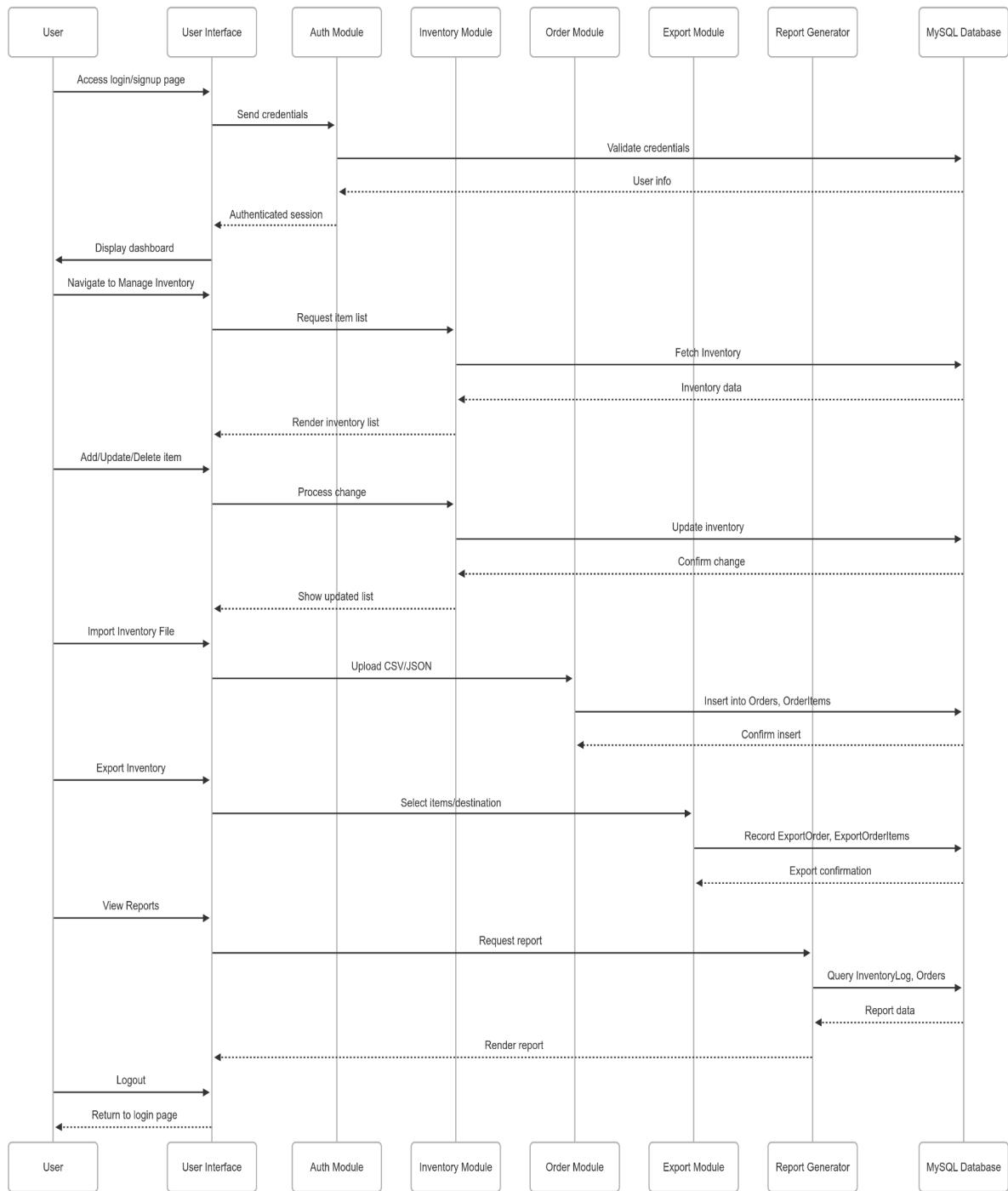
UseCase



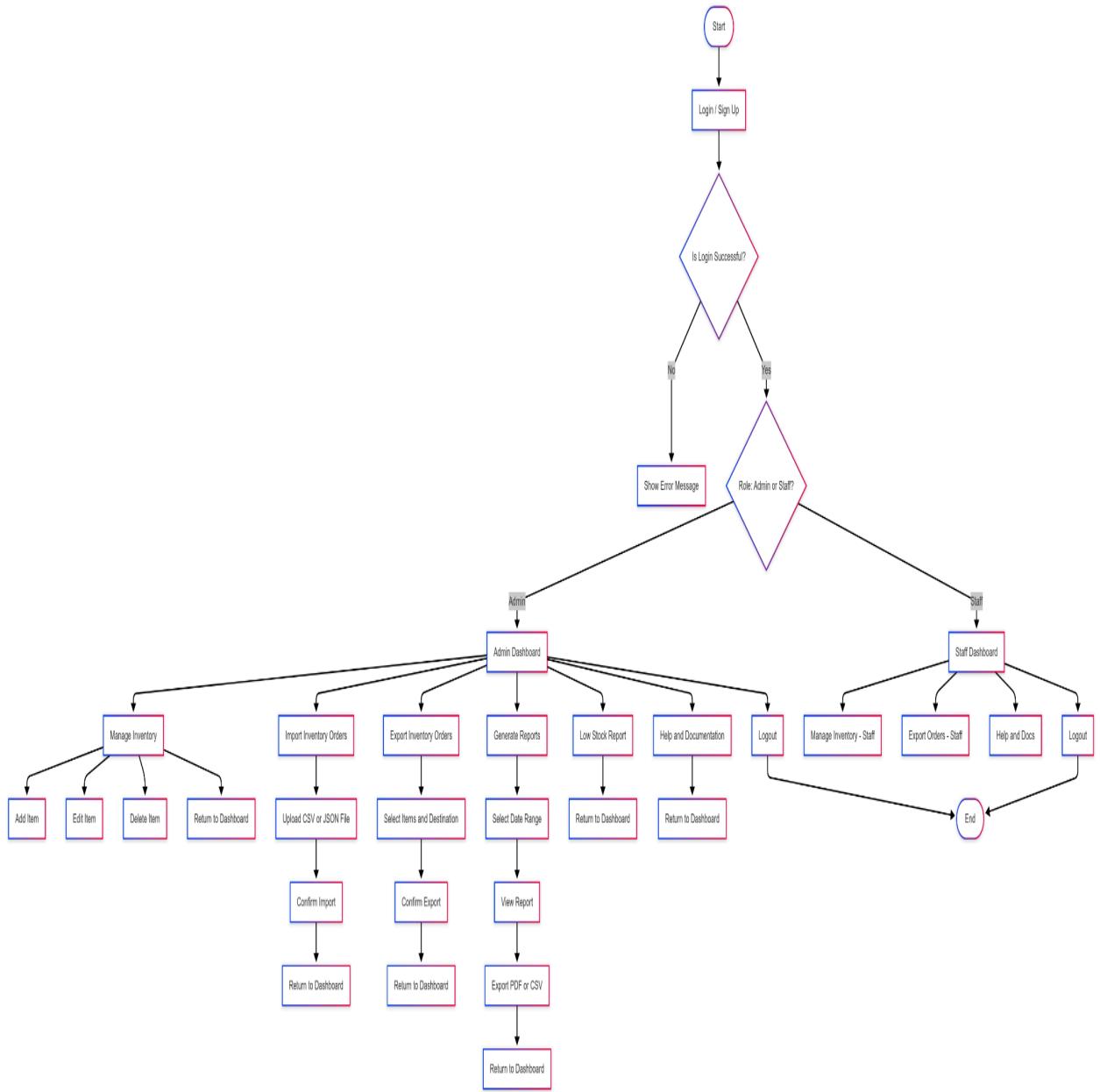
## Class Diagram



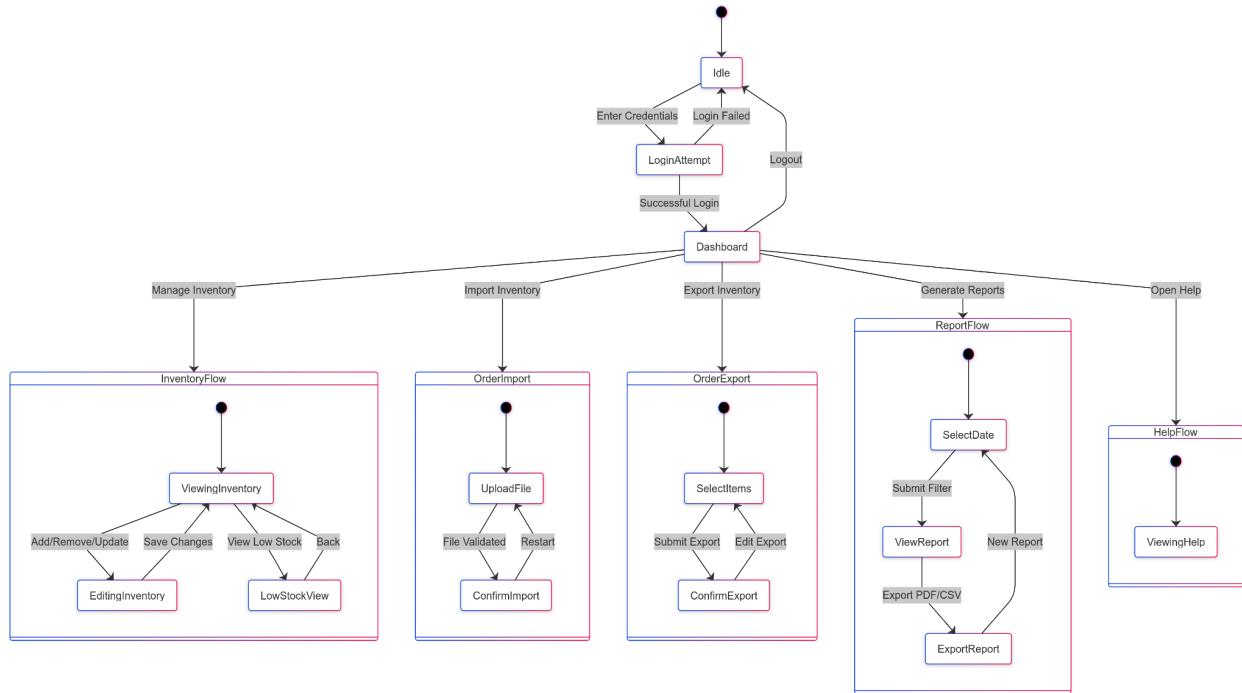
## Sequence Diagram



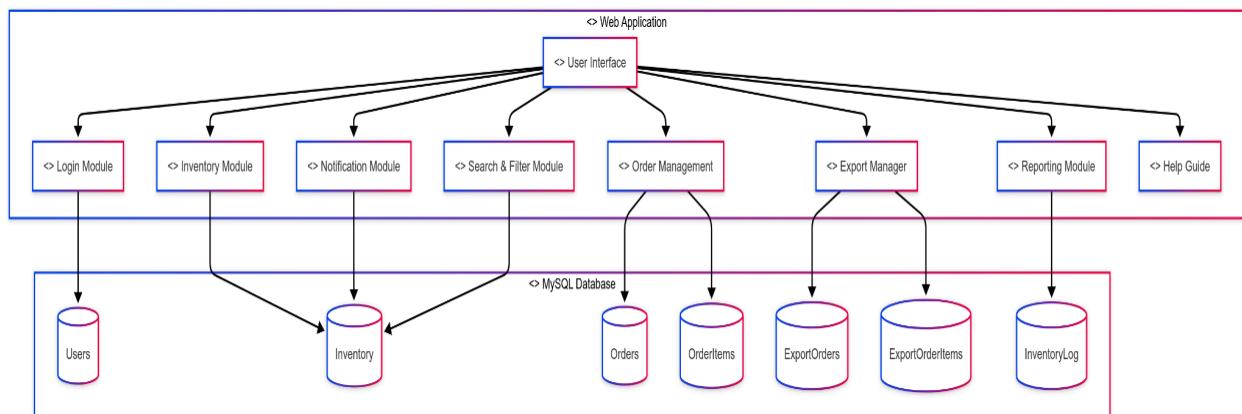
Activity Diagram



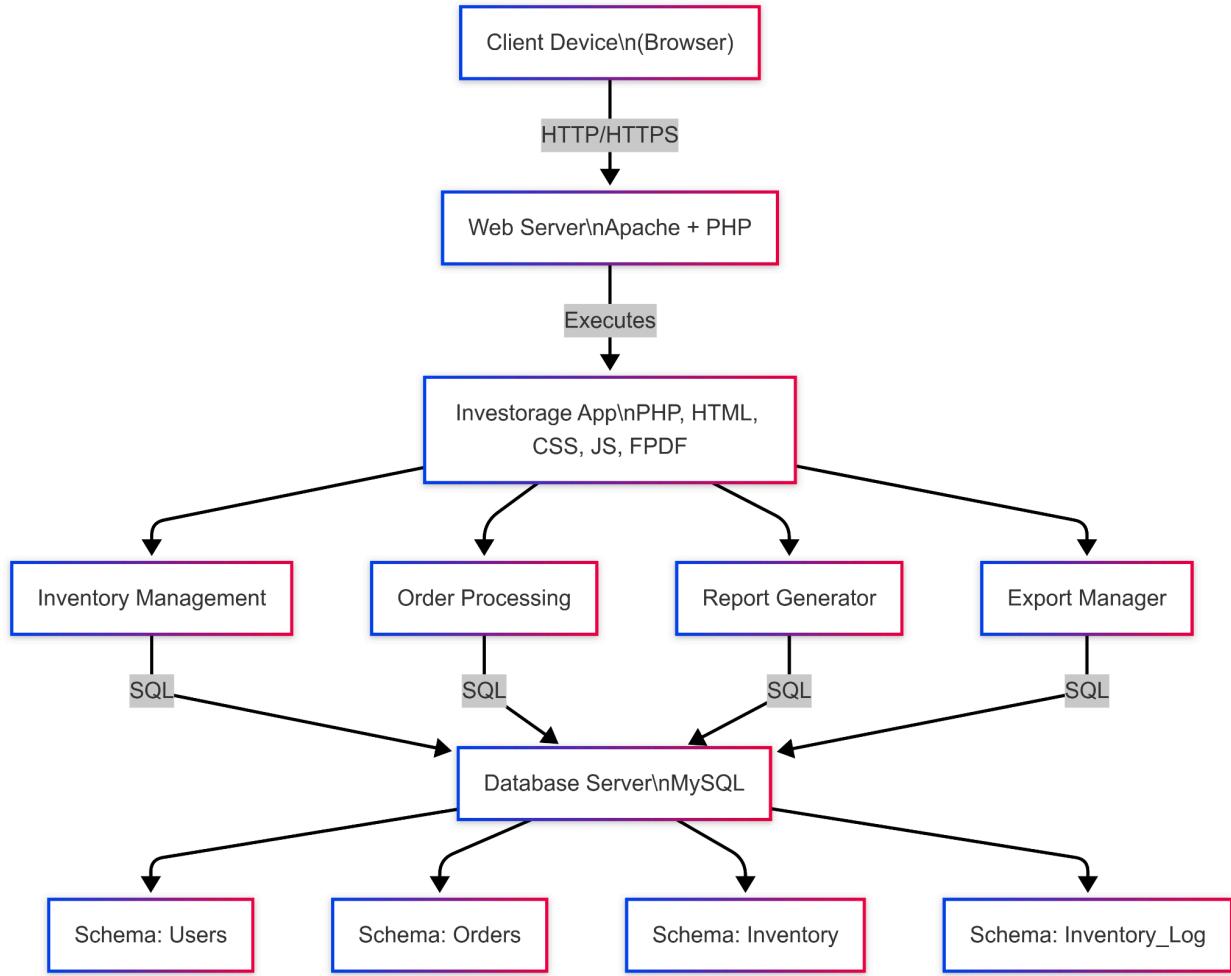
State Diagram



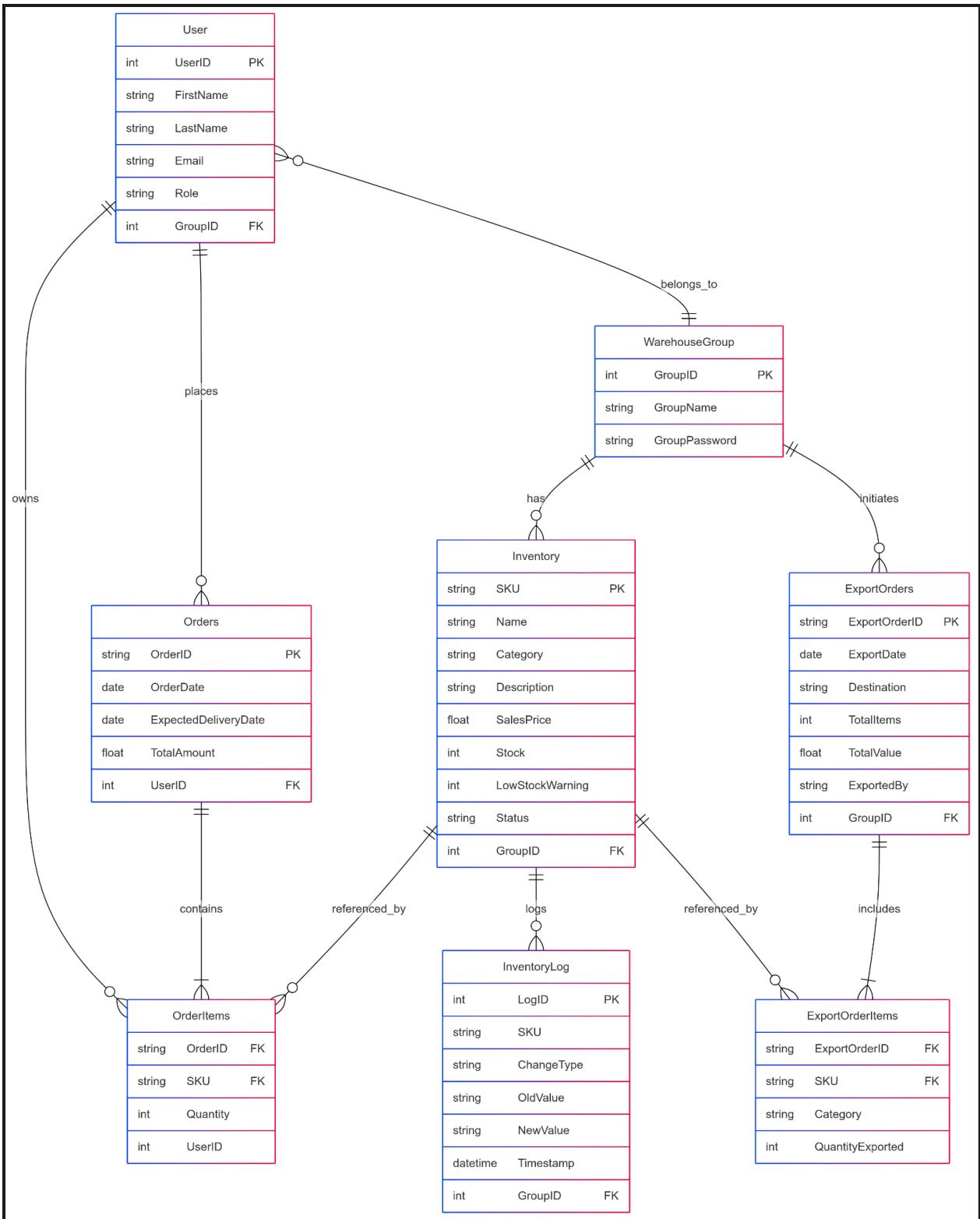
## Component Diagram



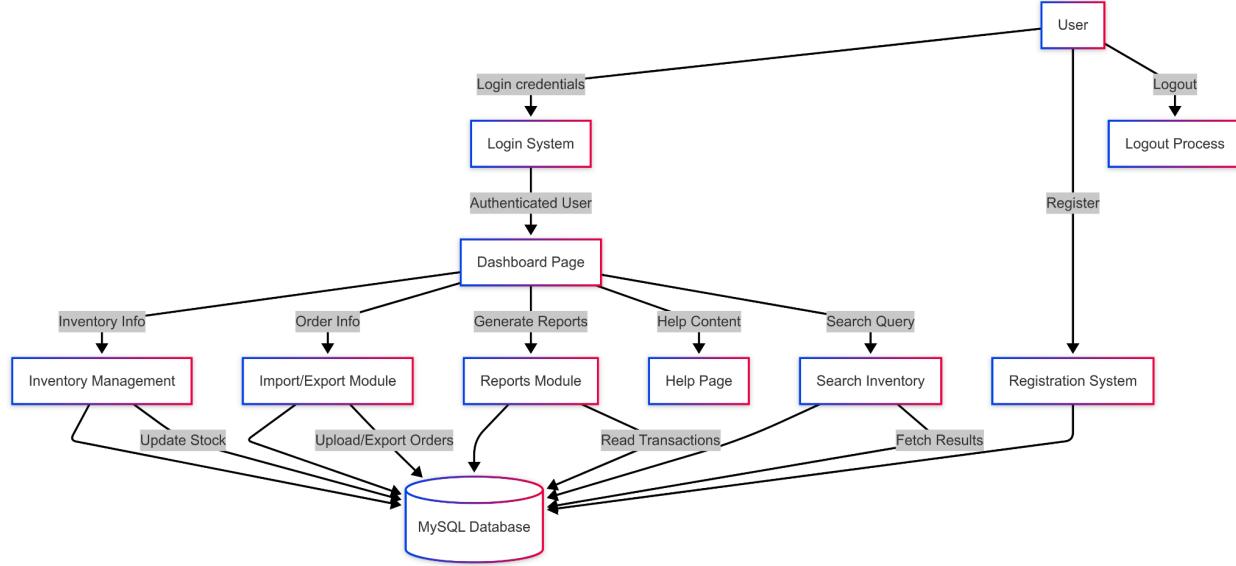
## Deployment Diagram



Entity-Relationship Diagram (ERD)



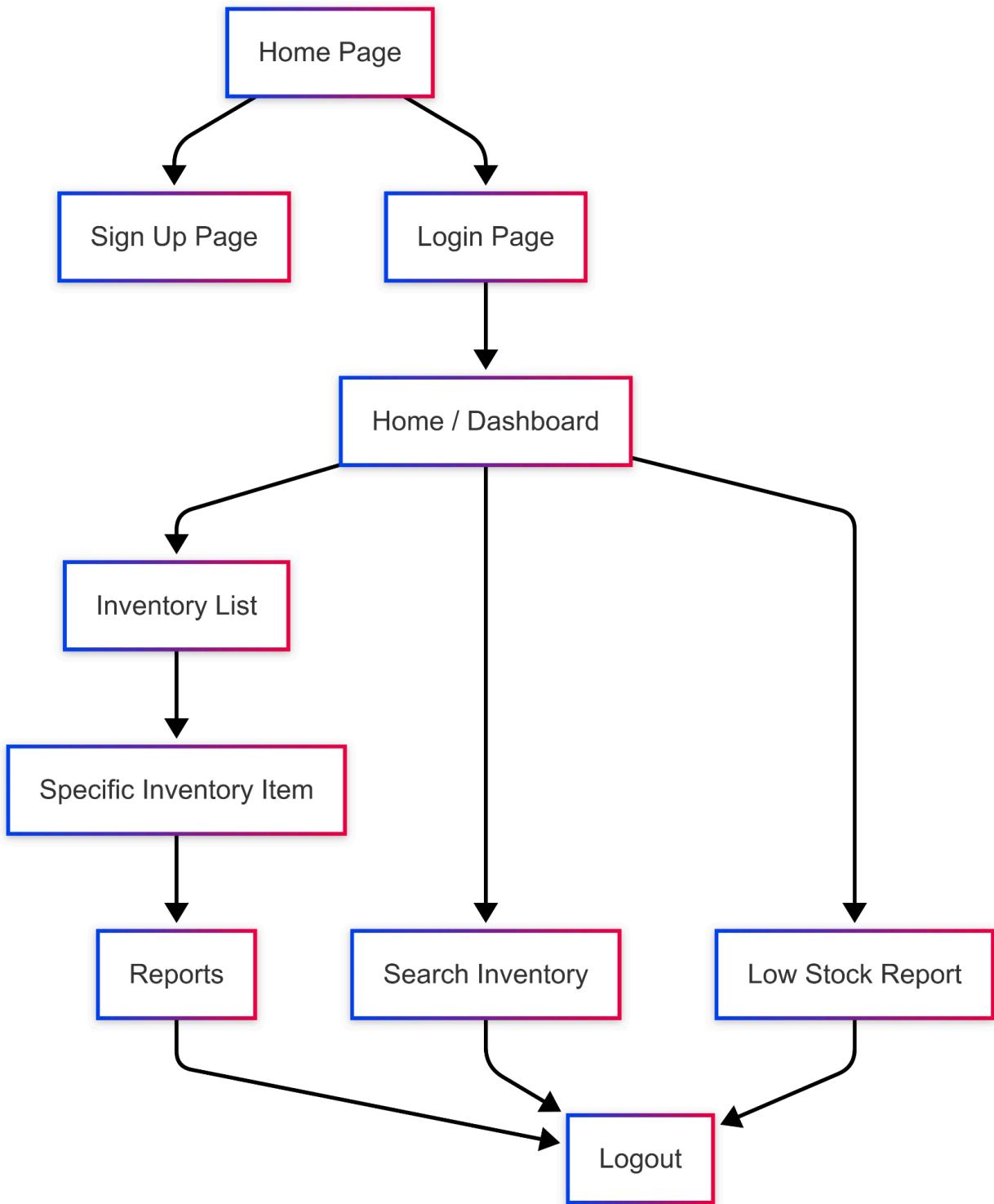
Data Flow Diagram



## 4.Algorithms:

Event driven low stock warning can be incorporated using heap sort by implementing a min-heap data structure that will allow for the stocks with the lowest level of product to be prioritized. This will allow for the system to identify and alert about the items with the lowest amount of stock. When a stock update is triggered the product quantity can be inserted into a min-heap where the products with the lowest number will stay at the top. The heap will use heap sort logic to add and remove items in logarithmic time to preserve the heap property.

## 5.Interface Design:



**Homepage:** About US, Our Goal, What we offer, get started button

**Sign Up / Login:** Credential-based access, with group name and role entry

**Dashboard:** Clean cards and navbar with low stock alerts

**Inventory Screens:** Inline form submissions, editable records

**Import/Export:** File upload or item selector + export location

**Reports:** Date filtering, summary tables, and PDF export

**Search:** Real-time item filtering by name, category, or status

**Help Page:** User documentation and tutorials

## 6. Security Considerations

Using Role-Based Access Control (RBAC), access will be restricted according to user roles. There will be two levels of access defined by the system:

- Admin - Full system access, including managing users and accessing reports
- Staff - Can perform inventory operations but can't create accounts or access reports

Implementing RBAC is essential to ensure that users interact with the system strictly within the bounds of their responsibilities. By defining clear access levels, the system prevents unauthorized actions such as unintentional deletions or unauthorized configuration changes.

**Session Management:** All critical pages check for active session and valid role

**SQL Injection Prevention:** All queries use prepared statements

**Access Control:** Staff role restricted from generating reports, deleting, or exporting

**Input Validation:** On all forms using htmlspecialchars(), trim(), intval(), etc.

**File Uploads:** Restricted to CSV/JSON with validation in processOrderImport.php

## 7. Algorithms

### Low Stock Detection Algorithm

- Logic checks if an item's Stock <= LowStockWarning
- Used in:
  - Low Stock Report
  - Dashboard bell notification
- Executes SQL filtering and displays a badge or warning
- **Efficiency:** O(n) query scan per inventory list

### Role-Based Access Control

- Decision tree logic that:
  - Checks \$\_SESSION["Role"]
  - Displays or blocks content accordingly
- Uses conditional guards in controller files

### File Parsing & Import Logic (CSV/JSON)

- CSV/JSON parsing loop reads uploaded data
- Deduplicates items before inserting (SELECT before INSERT)
- Batches item creation under one OrderID
- **Algorithm type:** Line-by-line file parser with conditional branching

## 8. Future Enhancements

- Password encryption with password\_hash() and password\_verify()
- Support for XML import
- Admin user management panel
- Audit log tracking per user ID and action type
- Dashboard metrics and data visualization
- Email notifications for low stock and order confirmations

## 3. Implementation (Coding)

The development of the **Investorage Inventory Management System** reflects the structured and systematic application of the design principles outlined in the Software Requirements Document (SRD) and Design Document (DD). This web-based application is built using a **Model-View-Controller (MVC)** architecture, ensuring scalability, maintainability, and a clear separation of concerns.

### Technology Stack

- **Frontend:** HTML5, CSS3 (with Bootstrap 5), JavaScript
- **Backend:** PHP 8+
- **Database:** MySQL
- **Hosting Environment:** HostGator (cPanel + Apache)
- **Version Control:** Git & GitHub

### Code Structure

The system is organized into modular files and directories, grouped according to functionality:

- **Authentication:** logIn.php, signUp.php, logOut.php, with session handling and role validation
- **Inventory Management:** addInventory.php, changeInventory.php, removeInventory.php, all using form validation and SQL prepared statements
- **Order Management:** orderImport.php, exportWarehouseManagement.php, supporting both CSV and JSON parsing
- **Reporting:** inventoryReport.php, generateInventoryReportPDF.php, complete with time filters and PDF export
- **User Interface Components:** Consolidated through indexElements.php for reusable navbars, footers, and headers

- **Security Modules:** Session verification, SQL injection prevention via prepared statements, and restricted access based on user role

## Features Implemented

- **User Registration & Group Creation:** Admin users can create warehouse groups; staff can join using credentials
- **Inventory CRUD:** Users can add, edit, delete, and search items; all actions are logged for audit purposes
- **Import & Export System:** Items can be imported via CSV or JSON; orders can be exported as PDF, CSV, or viewed on web
- **Role-Based Access Control (RBAC):** Admins have full permissions; staff users are restricted from deleting, exporting, or accessing reports
- **Report Generation:** Users can generate inventory reports filtered by predefined date ranges (e.g., 7 days, 30 days, 1 year)
- **Low Stock Alerts:** System notifies users via a dynamic notification badge and dedicated report page
- **Error Handling:** All forms include client- and server-side validation with contextual error messages

## Code Quality & Best Practices

- All database interactions use **prepared statements** to safeguard against SQL injection
- Role logic is handled via **secure session variables**, preventing unauthorized access to critical features
- Modular functions and consistent naming conventions improve **readability and maintainability**
- **Responsive UI** ensures usability across devices

## **Repository Access**

The full codebase, along with supporting documentation and development history, is publicly available on GitHub:

 **GitHub Repository:**  
<https://github.com/AndrewPuig77/Investorage>

This implementation successfully translates all design intentions into a fully functional system and demonstrates careful planning, security awareness, and user-focused development.

# **4. Test Cases for Investorage Inventory Management System**

## **Table of Contents**

- 1. Introduction**
  - 2. Test Case Format**
  - 3. Functional Test Cases**
  - 4. Negative Test Cases**
  - 5. Security Test Cases**
  - 6. Performance/Stress Test Cases**
  - 7. Role-Based Access Test Cases**
  - 8. Summary Table**
  - 9. Notes**
-

# 1. Introduction

This document outlines a comprehensive suite of test cases for the Investorage Inventory Management System done on multiple accounts. The purpose of this document is to validate the system's functionality, performance, security, and role-based access control. The test cases include both positive and negative scenarios to ensure robust and reliable performance.

## 2. Test Case Format

Each test case is structured with the following attributes:

- Test Case ID
- Description
- Input
- Expected Output
- Actual Output (if tested manually)
- Status (Pass/Fail)

## 3. Functional Test Cases

Test Case ID	Description	Input	Expected Output	Status
TC100	Create Warehouse Group	Admin enters group name and password	New warehouse group created	Pass
TC101	Sign up as Admin	Valid name, email, password, group name	Admin account created, redirected to dashboard	Pass

TC102	Sign up as Staff	Valid name, email, password, existing group info	Staff account created, redirected to dashboard	Pass
TC103	Join Existing Warehouse	Staff inputs existing warehouse name and password	User is added to the correct warehouse group	Pass
TC104	Log in with correct credentials	Registered email/password	User logged in, redirected	Pass
TC105	Add Inventory	Valid SKU, name, category, low stock warning, price, and status	Item added to inventory table	Pass
TC106	Edit Inventory	Change price, stock, status, SKU, and low stock warning	Changes reflected in inventory	Pass
TC107	Remove Inventory	Select a specific amount of items then click delete selected items	Item removed from database	Pass
TC108	Remove Inventory	Click Delete all Items	Item removed from database	Pass
TC109	Import Order (CSV)	Valid CSV with 5 rows	5 items added, order created	Pass
TC110	Import Order (JSON)	Valid JSON with 5 rows	5 items added, order created in history ready to confirm	Pass
TC111	Export Order (PDF)	Select order and export	Downloadable PDF generated	Pass

TC112	Export Order (CSV)	Select order and export	Downloadable CSV generated	Pass
TC113	View Order Web Preview	Select order and view in browser	Web preview of order is shown	Pass
TC114	Report Generation (Export PDF)	Click export report as PDF	Downloadable report PDF generated	Pass
TC115	Report Generation (Date Range)	Select time frame (7, 30, 90, 180 days, last year, all time)	Report displays data based on selected range	Pass
TC116	Search Inventory	Search by name, category, and status	Matching results shown correctly	Pass

## 4. Negative Test Cases

Test Case ID	Description	Input	Expected Output	Status
TC201	Submit empty sign-up form	No fields filled	Error messages on required fields	Pass
TC202	Passwords do not match	Mismatched passwords	Error: "Passwords do not match"	Pass
TC203	Upload unsupported file format	Upload .exe	Error: "Unsupported file type"	Pass
TC204	Import order with missing SKU	CSV with blank SKU	Error or skipped row	Pass
TC205	Add Inventory without stock	Empty stock field	Error: "Stock is required"	Pass

TC206	Submit blank fields across multiple forms	Leave each field empty one-by-one in Add, Edit, Signup	Proper field-specific validation errors shown	Pass
TC207	Login with invalid credentials	Nonexistent email/password	Error message: "Invalid credentials"	Pass
TC208	Login with empty fields	Empty email or password	Field-specific error: "Email is required" / "Password is required"	Pass
TC209	Add item with invalid sales price	Non-numeric price (e.g., "five dollars")	Error: "Sales Price must be numeric"	Pass
TC210	Add item with invalid low stock warning	Negative or non-numeric low stock threshold	Error: "Low Stock must be a number"	Pass
TC211	Add item with duplicate SKU	Existing SKU value	Error or overwrite protection message	Pass
TC212	Import JSON with missing required fields	JSON missing "Name" or "SalesPrice"	Skip row or validation error	Pass
TC213	Export non-existent order	Manually changed URL with invalid exportOrderID	Error: "Order not found"	Pass

## 5. Security Test Cases

Test Case ID	Description	Input	Expected Output	Status

TC301	Staff tries to delete inventory	Logged in as Staff	Error: Insufficient Permissions	Pass
TC302	Direct access to report page as Staff	/inventoryReport.php	Redirect + alert	Pass
TC303	SQL Injection attempt in login	' OR '1'='1	Login fails, no data exposed	Pass
TC304	Access export PDF without login	Open export PDF link directly after logout	Redirect to login page	Pass
TC305	Modify orderId in URL to access other orders	Change exportOrderId/orderId in URL manually	Error: Unauthorized or Order not found	Pass
TC306	Submit SQL commands in input fields	DROP TABLE Inventory; -- in SKU/Name fields	Input sanitized, command not executed	Pass
TC307	Bruteforce login attempt	Rapid failed login attempts	Temporary lockout or rate limiting	Pass

## 6. Performance/Stress Test Cases

Test Case ID	Description	Input	Expected Output	Status
TC401	Import large CSV	100+ inventory rows	Import completes without crash	Pass
TC402	Export all items	Export PDF for 100 items	Clean PDF generated	Pass
TC403	Rapid search operations	20 searches in 30 seconds	No server lag or delay	Pass

TC404	Simultaneous logins	10 users log in at the same time	All users successfully logged in	Pass
TC405	Bulk export of multiple orders	Export 10+ orders in rapid succession	All PDFs/CSVs generated without error	Pass
TC406	Search with long strings	200+ character search input	No crash, system handles input properly	Pass
TC407	Load dashboard with large data	500+ inventory items, 100+ logs	Dashboard loads in <5 seconds	Pass
TC408	Continuous form submissions	Add 10+ items rapidly back-to-back	No crashes, all items added	Pass
TC409	Multi-tab report/load test	Open report/help pages in new tabs	Pages load correctly, no freeze	Pass

## 5. User's Guide: Investorage Inventory Management System

### Table of Contents

1. Introduction
2. System Requirements
  
3. Getting Started
  1. Using the System
    - o Signing Up
    - o Logging In
    - o Navigating the Dashboard

- Adding Inventory
  - Editing Inventory
  - Removing Inventory
  - Importing Orders
  - Exporting Orders
  - Viewing Reports
  - Searching Inventory
  - Managing Low Stock Alerts
2. Troubleshooting
  3. FAQs
  4. Contact and Support
- 

## 1. Introduction

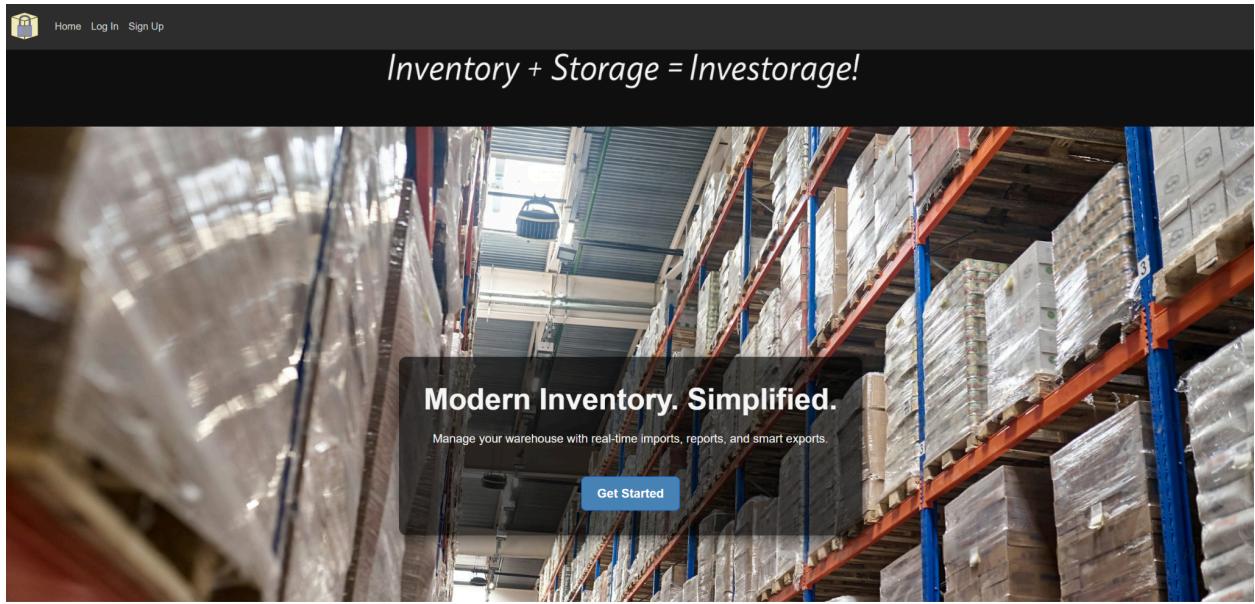
This user guide provides a comprehensive walkthrough for installing, using, and troubleshooting the Investorage Inventory Management System. Designed for warehouse teams, this system enables Admins and Staff to manage inventory, process orders, and generate reports in a secure and user-friendly environment.

## 2. System Requirements

- **Web Browser:** Chrome, Firefox, Safari, Edge (latest versions)
- **Internet Connection:** Required for all functionalities
- A device of capable of using a web browser

## 3. Getting Started

Visit the homepage (e.g., <https://investorage.site/index.php>) to begin. New users should register before logging in.



## 4. Using the System

### Signing Up

- Click "Sign Up" or get started
- Choose **Admin** or **Staff**
- Admin: Create a Warehouse Group and password
- Staff: Join an existing group using the Group Name and Password

A screenshot of the sign-up form interface. The form has a dark background with white input fields. At the top, there is a dropdown menu labeled 'Select Role' with options 'Select Role', 'Admin', and 'Staff'. The 'Admin' option is currently selected. Below the dropdown are several input fields: 'First Name', 'Last Name', 'Password', 'Confirm Password', 'Warehouse Group Name (optional)', and 'Group Password (for create/join)'. At the bottom of the form is a large blue 'Join Investorage' button.

### Logging In

- Enter registered email and password
- You'll be redirected to the appropriate dashboard based on your role

Login

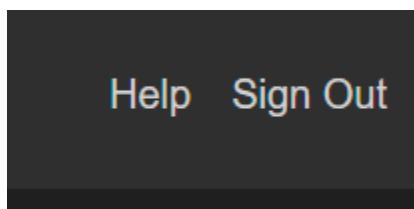
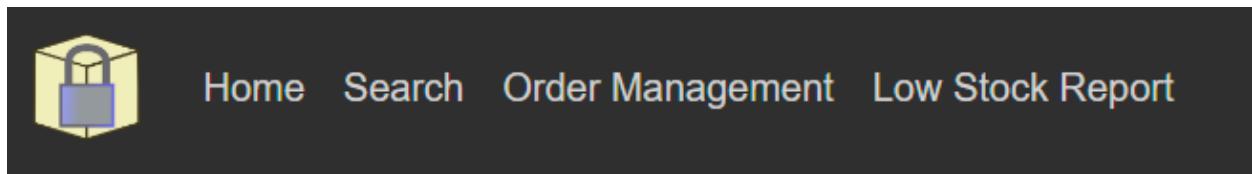
Email

Password

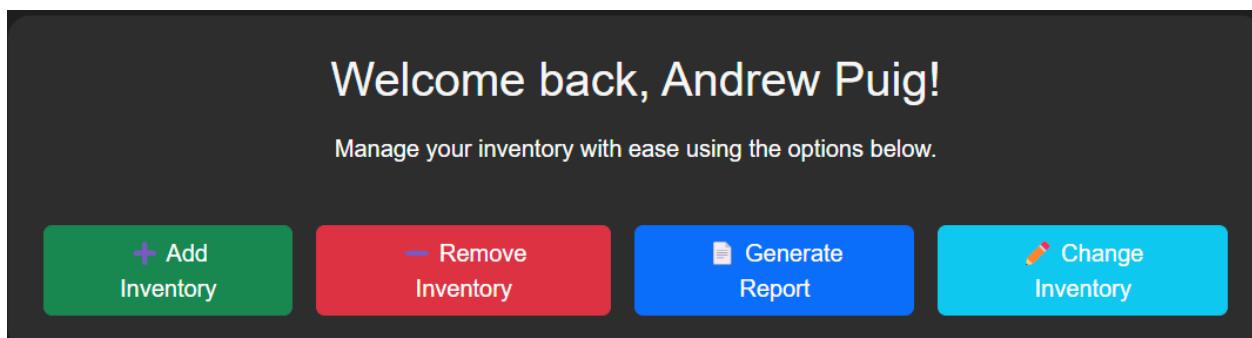
Login

## Navigating the Dashboard

- Use the top navbar to access Inventory, Orders, Reports, Help, and Logout



- Then you can use the dashboard card to access the add inventory, remove inventory, generate Report, and change inventory functions



## Adding Inventory (Admin Only)

- Go to Add Inventory
- Fill in SKU, Name, Category, Description, Price, Stock, Status, Low Stock Warning
- Click "Submit"

Category
Description
Low Stock Warning
Product Name
Sales Price
SKU
Select Status
Stock
<b>Add Inventory Item</b>

## Editing Inventory

- Use the Search or Inventory List to find the item
- Select Edit → modify price, stock, status, or low stock value
- Save changes

Select Item
-- Choose an Item --
-- Choose an Item --
kitkat
chicken
Hammer
Screwdriver
Select Item
chicken
<b>Update Sales Price</b>
4.68
<b>Update Stock</b>
1000
<b>Update Status</b>
Reserved
<b>Update Low Stock Warning</b>
5
<b>Submit Changes</b>

## Removing Inventory

- Navigate theo Inventory → check the select box for every item you wish to Delete.

### Current Warehouse Inventory

Select	SKU	Name	Category	Description	SalesPrice	Stock	LowStockWarning	Status
<input checked="" type="checkbox"/>	1234	kitkat	candy	gfsd	3.99	0	1	Dropped
<input checked="" type="checkbox"/>	123456798	chicken	meat	must be cooled	4.99	90	5	In-Stock
<input type="checkbox"/>	45612	Packers Moving Crate	Crates	10 x 10 Crate	86.49	500	10	In-Stock
<input type="checkbox"/>	SKU001	Hammer	Tools	Heavy duty hammer	15.99	30	5	Backordere
<input type="checkbox"/>	SKU002	Screwdriver	Tools	Flathead screwdriver	7.49	80	5	active
<input type="checkbox"/>	SKU003	Drill	Tools	Electric drill	49.99	40	3	active
<input type="checkbox"/>	SKU004	Pliers	Tools	Multi-purpose pliers	12.99	60	5	Ordered
<input type="checkbox"/>	SKU005	Wrench	Tools	Adjustable wrench	10.99	50	5	active
<input type="checkbox"/>	SKU123	Widget A	Hardware		2.5	10	5	BackOrdere
<input type="checkbox"/>	SKU124	Widget B	Software		5	5	5	Active
<input type="checkbox"/>	SKU125	Widget C	Hardware		1.75	20	5	Active

[Delete Selected Items](#)

[Delete All Items](#)

**Delete Selected Items**

- Then press the delete selected items button

**Delete All Items**

### Importing Orders

- Upload CSV or JSON file via Order Management page
  - First you must reach the order management page by clicking the navigation bar
  - Then if you will be met with the order management page which allows you to
- 
- view your order history you import order will be the basic history shown for export view by but be selected and changed to export order view

Order ID	Order Date	Status	Delivery	Date Completed	Amount	Type	Actions
ORD6800B48277499	2025-04-17 03:57:54	pending	2025-04-24	In Process	\$85.00	import	<button>Confirm Delivery</button> <button>Edit Order</button> <button>Delete</button>
							<button>Export as... ▾</button> <button>Export</button>
ORD68005B279D303	2025-04-16 21:36:39	complete	Completed	2025-04-16 21:45:37	\$85.00	import	<button>Delete</button> <button>Export as... ▾</button> <button>Export</button>
ORD68005B0640BA7	2025-04-16 21:36:06	pending	2025-04-23	In Process	\$0.00	import	<button>Confirm Delivery</button> <button>Edit Order</button> <button>Delete</button>
							<button>Export as... ▾</button> <button>Export</button>
ORD6800591183B34	2025-04-16 21:27:45	pending	2025-04-23	In Process	\$2,763.35	import	<button>Confirm Delivery</button> <button>Edit Order</button> <button>Delete</button>
							<button>Export as... ▾</button> <button>Export</button>
ORD680057FE3820B	2025-04-16 21:23:10	pending	2025-04-23	In Process	\$0.00	import	<button>Confirm Delivery</button> <button>Edit Order</button> <button>Delete</button>
							<button>Export as... ▾</button> <button>Export</button>
ORD67FF728439AAF	2025-04-16 05:04:04	pending	2025-04-23	In Process	\$85.42	import	<button>Confirm Delivery</button> <button>Edit Order</button> <button>Delete</button>
							<button>Export as... ▾</button> <button>Export</button>
ORD67FF6F5F8A2E1	2025-04-16 04:50:39	complete	Completed	2025-04-16 04:51:05	\$2,763.35	import	<button>Delete</button> <button>Export as... ▾</button> <button>Export</button>
ORD67FF4B3A2F3AB	2025-04-16	complete	Completed	2025-04-16	\$2,763.35	import	<button>Delete</button> <button>Export as... ▾</button> <button>Export</button>

- confirm deliveries- which make orders officially go through and adding them to your inventory

**Confirm Delivery**

- edit orders- where you can change any field from the order items, you can also fully delete items using the remove button.

## Edit Order ORD6800B48277499

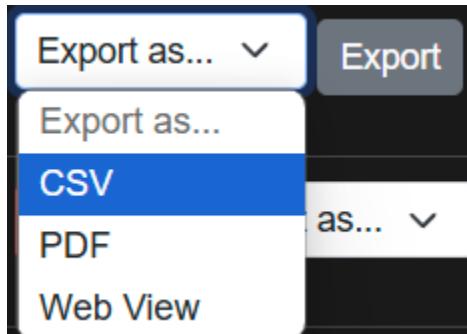
Order Status: pending

SKU	Name	Category	Description	Sales Price	Stock	Low Stock Warning	Status	Action
SKU123	Widget A	Hardware		2.50	10	5	Active	<button>Remove</button>
SKU124	Widget B	Software		5.00	5	5	Active	<button>Remove</button>
SKU125	Widget C	Hardware		1.75	20	5	Active	<button>Remove</button>

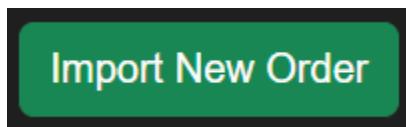
Save Changes to Order

Back to Order Management

- export as pdf/csv, get a web view

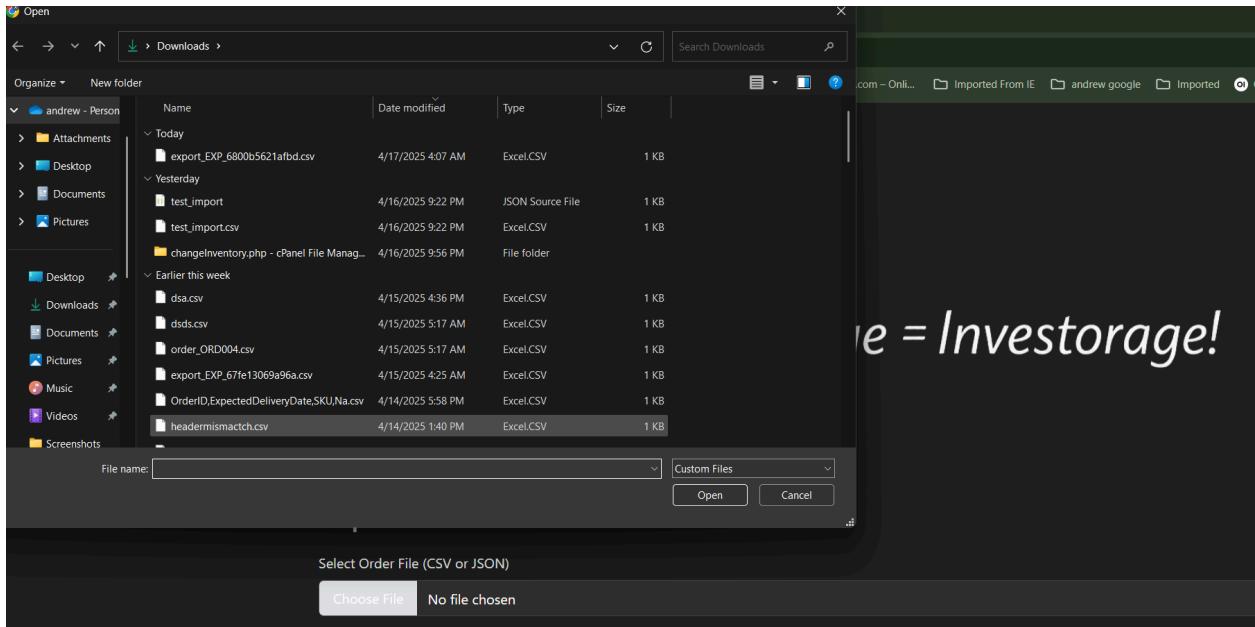


- To begin click import new order

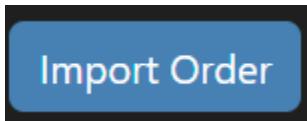


- From this page you click choose file which will open your file explorer where you select a supported file





- Then click import order



Working JSON example of order

```
[  
  {  
    "SKU": "SKU123",  
    "Name": "Widget A",  
    "Category": "Hardware",  
    "Quantity": 10,  
    "SalesPrice": 2.5  
  },  
  {  
    "SKU": "SKU124",  
    "Name": "Widget B",  
    "Category": "Software",  
    "Quantity": 5,  
    "SalesPrice": 5.0  
  },  
  {  
    "SKU": "SKU125",  
    "Name": "Widget C",  
    "Category": "Hardware",  
    "Quantity": 20,  
    "SalesPrice": 1.75  
  }]
```

## Working CSV order

SKU,Name,Category,Description,SalesPrice,Stock,LowStockWarning,Status

SKU001,Hammer,Tools,Heavy duty hammer,15.99,50,5,InStock

SKU002,Screwdriver,Tools,Flathead screwdriver,7.49,40,5,InStock

SKU003,Drill,Tools,Electric drill,49.99,20,3,InStock

SKU004,Pliers,Tools,Multi-purpose pliers,12.99,30,5,InStock

SKU005,Wrench,Tools,Adjustable wrench,10.99,25,5,InStock

## Exporting Orders (Admin Only)

- Click export button

Export Inventory Order

- Select items and quantity to export
- Assign a destination and confirm export

Inventory Export to Location						
Export Location:						
Store 207						
Select	SKU	Name	Category	Description	Stock Available	Quantity to Export
<input type="checkbox"/>	1234	kitkat	candy	gfsd	0	
<input checked="" type="checkbox"/>	123456798	chicken	meat	must be cooled	90	20
<input checked="" type="checkbox"/>	45612	Packers Moving Crate	Crates	10 x 10 Crate	500	100
<input checked="" type="checkbox"/>	SKU001	Hammer	Tools	Heavy duty hammer	30	5
<input checked="" type="checkbox"/>	SKU002	Screwdriver	Tools	Flathead screwdriver	80	1

- Choose to download report as PDF, CSV, or view on screen if needed
- Finally press the create export order button

## Create Export Order

- Finally you can view the export order by clicking view by then inventory exports
- Choose to download report as PDF, CSV, or view on screen if needed

## Viewing Reports (Admin Only)

- Go to Generate Reports on the user homepage inside the dashboard card

 Generate Report

- There are different options for a time frame to view from for example inventory changes and order history summaries from the last 7 days, last 30 days, last 90 days, last 180 days, last year, and all time

Last 7 Days

Last 30 Days

Last 90 Days

Last 180 Days

Last Year

View All

- View real-time reports of:
  - Current inventory

### Current Inventory

SKU	Name	Category	Stock	Low Stock Warning	Value
1234	kitkat	candy	0	1	\$0.00
123456798	chicken	meat	70	5	\$349.30
SKU001	Hammer	Tools	25	5	\$399.75
SKU002	Screwdriver	Tools	79	5	\$591.71
SKU003	Drill	Tools	40	3	\$1,999.60
SKU004	Pliers	Tools	60	5	\$779.40
SKU005	Wrench	Tools	50	5	\$549.50
SKU123	Widget A	Hardware	10	5	\$25.00
SKU124	Widget B	Software	5	5	\$25.00
SKU125	Widget C	Hardware	20	5	\$35.00
Totals:			359		\$4,754.26

- Inventory changes

### Recent Inventory Changes

Showing changes from NOW() - INTERVAL 7 DAY to NOW()

SKU	Change Type	Old Value	New Value	Time
SKU002	export	Stock: 80	Stock: 79 (exported 1)	2025-04-17 04:01:38
SKU001	export	Stock: 30	Stock: 25 (exported 5)	2025-04-17 04:01:38
45612	export	Stock: 500	Stock: 400 (exported 100)	2025-04-17 04:01:38
123456798	export	Stock: 90	Stock: 70 (exported 20)	2025-04-17 04:01:38

- Import/export summaries

### Imports & Exports Summary

Within NOW() - INTERVAL 7 DAY to NOW()

Type	# of Orders	Total Items	Total Value
Imports	12	1233	\$19,598.87
Exports	7	217	\$10,049.33

- Export as PDF if needed by clicking Export report as PDF

[Export Report as PDF](#)

Inventory Report																																																																																
Warehouse: test																																																																																
Exported By: Andrew Puig																																																																																
<b>Current Inventory</b>																																																																																
<table border="1"> <thead> <tr><th>SKU</th><th>Name</th><th>Category</th><th>Stock</th><th>Low Stock Warning</th><th>Value</th></tr> </thead> <tbody> <tr><td>1234</td><td>kitkat</td><td>candy</td><td>0</td><td>1</td><td>\$0.00</td></tr> <tr><td>123456798</td><td>chicken</td><td>meat</td><td>70</td><td>5</td><td>\$349.30</td></tr> <tr><td>SKU001</td><td>Hammer</td><td>Tools</td><td>25</td><td>5</td><td>\$399.75</td></tr> <tr><td>SKU002</td><td>Screwdriver</td><td>Tools</td><td>79</td><td>5</td><td>\$591.71</td></tr> <tr><td>SKU003</td><td>Drill</td><td>Tools</td><td>40</td><td>3</td><td>\$1,999.60</td></tr> <tr><td>SKU004</td><td>Pliers</td><td>Tools</td><td>60</td><td>5</td><td>\$779.40</td></tr> <tr><td>SKU005</td><td>Wrench</td><td>Tools</td><td>50</td><td>5</td><td>\$549.50</td></tr> <tr><td>SKU123</td><td>Widget A</td><td>Hardware</td><td>10</td><td>5</td><td>\$25.00</td></tr> <tr><td>SKU124</td><td>Widget B</td><td>Software</td><td>5</td><td>5</td><td>\$25.00</td></tr> <tr><td>SKU125</td><td>Widget C</td><td>Hardware</td><td>20</td><td>5</td><td>\$35.00</td></tr> </tbody> </table>						SKU	Name	Category	Stock	Low Stock Warning	Value	1234	kitkat	candy	0	1	\$0.00	123456798	chicken	meat	70	5	\$349.30	SKU001	Hammer	Tools	25	5	\$399.75	SKU002	Screwdriver	Tools	79	5	\$591.71	SKU003	Drill	Tools	40	3	\$1,999.60	SKU004	Pliers	Tools	60	5	\$779.40	SKU005	Wrench	Tools	50	5	\$549.50	SKU123	Widget A	Hardware	10	5	\$25.00	SKU124	Widget B	Software	5	5	\$25.00	SKU125	Widget C	Hardware	20	5	\$35.00									
SKU	Name	Category	Stock	Low Stock Warning	Value																																																																											
1234	kitkat	candy	0	1	\$0.00																																																																											
123456798	chicken	meat	70	5	\$349.30																																																																											
SKU001	Hammer	Tools	25	5	\$399.75																																																																											
SKU002	Screwdriver	Tools	79	5	\$591.71																																																																											
SKU003	Drill	Tools	40	3	\$1,999.60																																																																											
SKU004	Pliers	Tools	60	5	\$779.40																																																																											
SKU005	Wrench	Tools	50	5	\$549.50																																																																											
SKU123	Widget A	Hardware	10	5	\$25.00																																																																											
SKU124	Widget B	Software	5	5	\$25.00																																																																											
SKU125	Widget C	Hardware	20	5	\$35.00																																																																											
<b>Recent Inventory Changes</b>																																																																																
<table border="1"> <thead> <tr><th>SKU</th><th>Type</th><th>Old Value</th><th>New Value</th><th>Time</th></tr> </thead> <tbody> <tr><td>SKU002</td><td>export</td><td>Stock: 80</td><td>Stock: 79 (exported 1)</td><td>2025-04-17 04:01:38</td></tr> <tr><td>SKU001</td><td>export</td><td>Stock: 30</td><td>Stock: 25 (exported 5)</td><td>2025-04-17 04:01:38</td></tr> <tr><td>45612</td><td>export</td><td>Stock: 500</td><td>Stock: 400 (exported 100)</td><td>2025-04-17 04:01:38</td></tr> <tr><td>123456798</td><td>export</td><td>Stock: 90</td><td>Stock: 70 (exported 20)</td><td>2025-04-17 04:01:38</td></tr> <tr><td>123456798</td><td>export</td><td>Stock: 100</td><td>Stock: 90 (exported 10)</td><td>2025-04-17 03:54:06</td></tr> <tr><td>123456798</td><td>add</td><td>N/A</td><td>Added item: chicken, stock:100</td><td>2025-04-16 22:08:50</td></tr> <tr><td>SKU123</td><td>import</td><td>New item</td><td>Imported qty: 10</td><td>2025-04-16 21:45:37</td></tr> <tr><td>SKU124</td><td>import</td><td>New item</td><td>Imported qty: 5</td><td>2025-04-16 21:45:37</td></tr> <tr><td>SKU125</td><td>import</td><td>New item</td><td>Imported qty: 20</td><td>2025-04-16 21:45:37</td></tr> <tr><td>SKU001</td><td>updateInventory</td><td>Price=15.99, Stock=30, Status=</td><td>Price=15.99, Stock=30, Status=</td><td>2025-04-16 21:39:17</td></tr> <tr><td>SKU001</td><td>updateInventory</td><td>Price=15.99, Stock=30, Status=</td><td>Price=15.99, Stock=30, Status=</td><td>2025-04-16 21:39:00</td></tr> <tr><td>SKU004</td><td>updateInventory</td><td>Price=12.99, Stock=60, Status=</td><td>Price=12.99, Stock=60, Status=</td><td>2025-04-16 21:38:50</td></tr> <tr><td>SKU001</td><td>updateInventory</td><td>Price=15.99, Stock=30, Status=</td><td>Price=15.99, Stock=30, Status=</td><td>2025-04-16 21:38:46</td></tr> <tr><td>1234</td><td>updateInventory</td><td>Price=3.99, Stock=0, Status=Or</td><td>Price=3.99, Stock=0, Status=Or</td><td>2025-04-16 21:38:40</td></tr> </tbody> </table>						SKU	Type	Old Value	New Value	Time	SKU002	export	Stock: 80	Stock: 79 (exported 1)	2025-04-17 04:01:38	SKU001	export	Stock: 30	Stock: 25 (exported 5)	2025-04-17 04:01:38	45612	export	Stock: 500	Stock: 400 (exported 100)	2025-04-17 04:01:38	123456798	export	Stock: 90	Stock: 70 (exported 20)	2025-04-17 04:01:38	123456798	export	Stock: 100	Stock: 90 (exported 10)	2025-04-17 03:54:06	123456798	add	N/A	Added item: chicken, stock:100	2025-04-16 22:08:50	SKU123	import	New item	Imported qty: 10	2025-04-16 21:45:37	SKU124	import	New item	Imported qty: 5	2025-04-16 21:45:37	SKU125	import	New item	Imported qty: 20	2025-04-16 21:45:37	SKU001	updateInventory	Price=15.99, Stock=30, Status=	Price=15.99, Stock=30, Status=	2025-04-16 21:39:17	SKU001	updateInventory	Price=15.99, Stock=30, Status=	Price=15.99, Stock=30, Status=	2025-04-16 21:39:00	SKU004	updateInventory	Price=12.99, Stock=60, Status=	Price=12.99, Stock=60, Status=	2025-04-16 21:38:50	SKU001	updateInventory	Price=15.99, Stock=30, Status=	Price=15.99, Stock=30, Status=	2025-04-16 21:38:46	1234	updateInventory	Price=3.99, Stock=0, Status=Or	Price=3.99, Stock=0, Status=Or	2025-04-16 21:38:40
SKU	Type	Old Value	New Value	Time																																																																												
SKU002	export	Stock: 80	Stock: 79 (exported 1)	2025-04-17 04:01:38																																																																												
SKU001	export	Stock: 30	Stock: 25 (exported 5)	2025-04-17 04:01:38																																																																												
45612	export	Stock: 500	Stock: 400 (exported 100)	2025-04-17 04:01:38																																																																												
123456798	export	Stock: 90	Stock: 70 (exported 20)	2025-04-17 04:01:38																																																																												
123456798	export	Stock: 100	Stock: 90 (exported 10)	2025-04-17 03:54:06																																																																												
123456798	add	N/A	Added item: chicken, stock:100	2025-04-16 22:08:50																																																																												
SKU123	import	New item	Imported qty: 10	2025-04-16 21:45:37																																																																												
SKU124	import	New item	Imported qty: 5	2025-04-16 21:45:37																																																																												
SKU125	import	New item	Imported qty: 20	2025-04-16 21:45:37																																																																												
SKU001	updateInventory	Price=15.99, Stock=30, Status=	Price=15.99, Stock=30, Status=	2025-04-16 21:39:17																																																																												
SKU001	updateInventory	Price=15.99, Stock=30, Status=	Price=15.99, Stock=30, Status=	2025-04-16 21:39:00																																																																												
SKU004	updateInventory	Price=12.99, Stock=60, Status=	Price=12.99, Stock=60, Status=	2025-04-16 21:38:50																																																																												
SKU001	updateInventory	Price=15.99, Stock=30, Status=	Price=15.99, Stock=30, Status=	2025-04-16 21:38:46																																																																												
1234	updateInventory	Price=3.99, Stock=0, Status=Or	Price=3.99, Stock=0, Status=Or	2025-04-16 21:38:40																																																																												

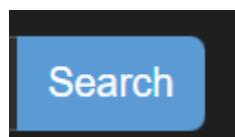
## Searching Inventory

- Use search bar or filters (category, status)
- Is found at top left on navigation bar, click Search(this page should pop up)

Inventory + Storage = Investorage!							
Search Inventory							
Search by name or category				Status			
Category				All Statuses			
<input type="button" value="All Categories"/> <input type="button" value="All Statuses"/>							
SKU	Name	Category	Description	Sales Price	Status	Stock	Low Stock Warning
1234	kitkat	candy	gfsd	3.99	Dropped	0	1
123456798	chicken	meat	must be cooled	4.99	In-Stock	70	5
45612	Packers Moving Crate	Crates	10 x 10 Crate	86.49	In-Stock	400	10
SKU001	Hammer	Tools	Heavy duty hammer	15.99	Backordered	25	5
SKU002	Screwdriver	Tools	Flathead screwdriver	7.49	active	79	5
SKU003	Drill	Tools	Electric drill	49.99	active	40	3

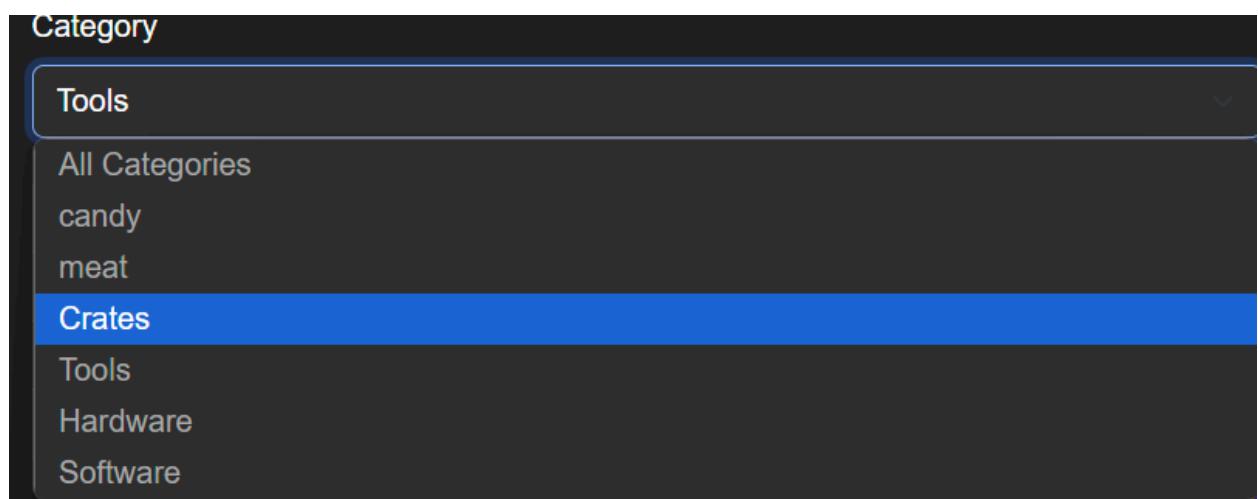
- Here you can view your current inventory

- You can search for specific items by typing in a name then clicking search. The searched item should be the only one showing



Search Inventory								
Drill					Search			
Category			Status					
Tools					All Statuses			
SKU	Name	Category	Description	Sales Price	Status	Stock	Low Stock Warning	
SKU003	Drill	Tools	Electric drill	49.99	active	40	3	

- Sort by categories for example Tools then click search



Category				Status			
Tools				All Statuses			
SKU	Name	Category	Description	Sales Price	Status	Stock	Low Stock
SKU001	Hammer	Tools	Heavy duty hammer	15.99	Backordered	25	5
SKU002	Screwdriver	Tools	Flathead screwdriver	7.49	active	79	5
SKU003	Drill	Tools	Electric drill	49.99	active	40	3
SKU004	Pliers	Tools	Multi-purpose pliers	12.99	Ordered	60	5
SKU005	Wrench	Tools	Adjustable wrench	10.99	active	50	5

- And sort by Status for example In-Stock, active, Dropped then click search

**Status**

All Statuses

All Statuses

InStock

Ordered

Backordered

Reserved

Dropped

15.99	Backordere
7.49	active
49.99	active
12.99	Ordered
10.99	active

SKU	Name	Category	Description	Sales Price	Status	Stock	Low Stock Warning
123456798	chicken	meat	must be cooled	4.99	InStock	70	5
45612	Packers Moving Crate	Crates	10 x 10 Crate	86.49	InStock	400	10
SKU002	Screwdriver	Tools	Flathead screwdriver	7.49	InStock	79	5
SKU003	Drill	Tools	Electric drill	49.99	InStock	40	3
SKU005	Wrench	Tools	Adjustable wrench	10.99	InStock	50	5
SKU124	Widget B	Software		5	InStock	5	5
SKU125	Widget C	Hardware		1.75	InStock	20	5

## Managing Low Stock Alerts

- Alerts appear on dashboard and in low stock report up in the navigation top bar

### Low Stock Report

- Items below or that have reached the threshold have a warning sign next to their name, and items 10 away from the threshold will show up but without a warning sign

### Low Stock Warnings

SKU	Name	Stock	Threshold
1234	kitkat <span style="color: orange;">⚠</span>	0	1
SKU123	Widget A	10	5
SKU124	Widget B <span style="color: orange;">⚠</span>	5	5

## 5. Troubleshooting

### Issue

### Solution

-----

-----

Cannot log in

Check credentials and user role

No items showing

Ensure you have joined the correct warehouse group

Upload failed

Ensure file is in correct CSV/JSON format

PDF doesn't load	Refresh and check for ad blockers
Export button not working	Ensure you selected an export type and have items selected
Error message 'Invalid Group Password'	Double check group password with the group Admin
Sign up fields stay filled after submit	Refresh the page after sign-up confirmation
Report not downloading	Make sure popup blockers are disabled and the file format is supported
Low stock badge doesn't appear	Ensure items are actually below their warning threshold and notifications are enabled
Search yields no results	Clear filters or try a broader keyword term
Inventory report page is blank	Check that inventory changes occurred within selected time filters like 7, 30, or 90 days
Duplicate inventory entries after import	Ensure imported file doesn't contain repeated SKUs or upload is not resubmitted accidentally
Exported CSV file is empty	Make sure at least one item was selected before exporting
Changes to inventory don't save	Ensure you're logged in as Admin or Staff and that session hasn't expired
Time format looks wrong in reports	Ensure server timezone settings are correct and consistent with system clock
Filters not applying	Clear all filters and reapply only one at a time to test filter behavior

## 6.. FAQs

**Q:** Can Staff create warehouse groups? **A:** No, only Admins can create or manage groups.

**Q:** What status values can I use? **A:** InStock, Ordered, Backordered, Reserved, Dropped

**Q:** Can I export individual reports? **A:** Yes, Admins can export full or filtered reports

**Q:** How do I reset my password? **A:** Password resets are not currently supported in the interface, but will be added in the near future. Contact your Admin for assistance.

**Q:** Can Staff import or export inventory orders? **A:** No, only Admins have permission to handle imports and exports.

**Q:** What file types are accepted for imports? **A:** CSV and JSON files formatted with correct fields are accepted. XML is not currently supported.

**Q:** Why do my changes not appear right away? **A:** Refresh the page or check that your session is still active. Some cached views may delay updates.

**Q:** Can multiple staff users use the system at the same time? **A:** Yes, the system supports concurrent sessions within the same warehouse group.

**Q:** Why is the system not allowing me to delete an item? **A:** Only Admins can delete inventory items. Staff users have view and edit permissions only.

**Q:** What if I accidentally submitted the same import file twice? **A:** Check for duplicate entries and use the Remove function to delete unwanted items. Consider adding a unique identifier in your import files to prevent reuploading.

## 7. Contact and Support

For assistance, contact your system administrator or reach out to the development team

Contacts:

carlos@investorage.site

apuig016@fiu.edu

jflor214@fiu.edu

Ccuad002@fiu.edu

ccoll145@fiu.edu