**SHIP INVENTORY MANAGEMENT SYSTEM**

Contents

[System Overview 2](#_Toc204938563)

[Requirements 2](#_Toc204938564)

[Functional Requirements 2](#_Toc204938565)

[Non-Functional Requirements 3](#_Toc204938566)

[Technical Specifications 3](#_Toc204938567)

[System Architecture 3](#_Toc204938568)

[Database Schema 4](#_Toc204938569)

[Dependencies 4](#_Toc204938570)

[User Manual 5](#_Toc204938571)

[Login 5](#_Toc204938572)

[User Management (Super User only) 5](#_Toc204938573)

[Department Management (Super User only) 5](#_Toc204938574)

[Inventory Operations 5](#_Toc204938575)

[Reports and Analytics 6](#_Toc204938576)

[Deployment Guide 6](#_Toc204938577)

[Troubleshooting 7](#_Toc204938578)

# System Overview

The Ship Inventory Management System is a barcode-based tracking solution designed specifically for managing ship-related spare parts. It provides a comprehensive platform for:

* Tracking inventory levels
* Managing departments and hierarchical structures
* Recording check-in/check-out transactions
* Generating reports and analytics
* User access control with role-based permissions

The system features a web-based interface built with Streamlit and uses SQLite for data storage, making it lightweight yet powerful for shipboard inventory management.

# Requirements

## Functional Requirements

1. **User Authentication**
   * Role-based access (Super User, Admin, User)
   * Secure password storage with PBKDF2 hashing
   * Department assignment for users
2. **Inventory Management**
   * Barcode generation and validation
   * Part tracking with detailed attributes
   * Check-in/check-out functionality
   * Last piece Level alerts where Qty is 1
   * Low stock alerts based on Re-order Level and Re-order Qty
3. **Department Management**
   * Hierarchical department structure
   * Parent-child department relationships
4. **Reporting**
   * Stock level reports
   * Transaction history
   * Demand forecasting
   * Export capabilities
5. **Bulk Operations**
   * CSV import/export
   * Batch processing

## Non-Functional Requirements

1. **Performance**
   * Responsive interface for 1000+ inventory items
   * Quick barcode lookup (<1 second)
2. **Security**
   * Secure authentication
   * Data validation
   * Session management
3. **Usability**
   * Intuitive web interface
   * Barcode scanner compatibility
   * Mobile-responsive design
4. **Reliability**
   * SQLite database transactions
   * Error handling and recovery

# Technical Specifications

## System Architecture

* Frontend: Streamlit (Python web framework)
* Backend: Python 3.11
* Database: SQLite
* Barcode Handling: python-barcode library

## Database Schema

Key tables:

1. **Departments**
   * id, code, name, parent\_id (hierarchy)
2. **Spare\_parts**
   * id, part\_number, name, description, quantity
   * department\_id, barcode, location info
   * min\_order\_level, min\_order\_quantity
3. **Transactions**
   * part\_id, transaction\_type, quantity
   * timestamp, reason, remarks
4. **Users**
   * username, password\_hash, salt
   * role, department\_id, isactive

## Dependencies

* "barcode>=1.0.4",
* "numpy>=2.2.3",
* "pandas>=2.2.3",
* "pillow>=11.1.0",
* "plotly>=6.0.0",
* "python-barcode>=0.15.1",
* "streamlit-option-menu>=0.4.0",
* "streamlit>=1.43.1",
* "trafilatura>=2.0.0"

# User Manual

## Login

1. Access the system URL (http://localhost:8501/)
2. Enter credentials
   1. For role as Super User : SuperUser / super@123
   2. For role as Admin: Admin / admin123
   3. For role as User: Test / test
3. System shows role-based interface after login
4. **Default Roles**:
   1. Super User: Full access
   2. Admin: Limited admin access
   3. User: Department-specific access

## User Management (Super User only)

1. Add/Edit/Deactivate users
2. Assign roles and departments
3. Reset passwords

## Department Management (Super User only)

1. Create parent/child departments
2. Edit department hierarchy
3. View department structure

## Inventory Operations

1. **Adding Parts** 
   * Manual entry or bulk CSV import
   * Automatic barcode generation
   * **Barcode Format**: 3 letters - 1 letter - 4 digits (e.g., ABC-D-1234)
   * Department assignment
2. **Checking Items In/Out**
   * Barcode scanning interface
   * Manual selection option
   * Reason and remarks tracking
3. **Viewing Inventory**
   * Filter by department
   * Search by various attributes
   * Low stock alerts

## Reports and Analytics

1. **Stock Levels**
   * Last Piece Level
   * Current vs minimum levels
   * Department-wise views
2. **Transaction History**
   * Date range filtering
   * Export to CSV
3. **Demand Forecasting**
   * Moving averages
   * Reorder point calculation

# Deployment Guide

1. **Local Development**
   * pip install python-barcode
   * streamlit run main.py
2. **Containerized Deployment**
   * Use included .devcontainer configuration
   * Port 8501 exposed for Streamlit

# Troubleshooting

| **Issue** | **Solution** |
| --- | --- |
| Barcode not scanning | Verify format matches ABC-D-1234 pattern |
| Login failures | Check username/password or contact admin |
| Slow performance | Reduce inventory view scope or optimize queries |
| Database errors | Check file permissions for inventory.db |
| Missing features | Verify user role has required permissions |

For additional support, consult the system logs or contact the development team.

Appendix

* **Data Backup**: Regularly backup inventory.db file

This documentation covers version 1.0 of the Ship Inventory Management System. Refer to the source code comments for implementation details and future enhancement plans.