# Software Engineering Project
# Inventory Management System

# Contents

# 1 Introduction

## 1.1 Purpose

To build an abstract inventory management system that suits the needs of any public institution, such as tracking resources and identifying discrepancies.

## 1.2 Document Conventions

This Software Requirements Specification (SRS) follows the following conventions:

- **Priority Levels:** Requirements are assigned priority levels using the following scale:
  - **MAX** - Critical features essential for core functionality
  - **Mid-Max** - Important features that enhance system capabilities
  - **Min-Mid** - Optional features for future iterations
- **Requirement Notation:** Individual requirements within features are labeled as REQ1, REQ2, etc., and inherit the priority level of their parent feature unless otherwise specified.
- **Text Formatting:**
  - **Bold text** indicates feature titles, priority levels, and emphasis on key terms
  - *Italic text* is used for technical terms and system components
  - Monospace font represents code, commands, or technical identifiers
- **Acronyms and Abbreviations:**
  - IMS - Inventory Management System
  - CRUD - Create, Read, Update, Delete
  - EOQ - Economic Order Quantity
  - TBD - To Be Determined
  - SRS - Software Requirements Specification
- **Placeholder Text:** Sections marked as "Write" or "TBD" indicate content to be completed in future document revisions.

## 1.3 Intended Audience

Blank

## 1.4 Product Scope

Storium will allow its wielders to track and record their inventory placements from geographic locations down to the microscopic shelf spots. Automate routines and visualise supply chains, tradeoff between understock and overstock, and make the institution run like clockwork with the IMS being the escapement that yields the implicit profits.

### 1.5 References

Blank

## 2 Overall Description

### 2.1 Product Perspective

Our product is a new independent software which will be in the hands of its wielder with a slight dependence on cloud technology for backup storage and necessary computation power.

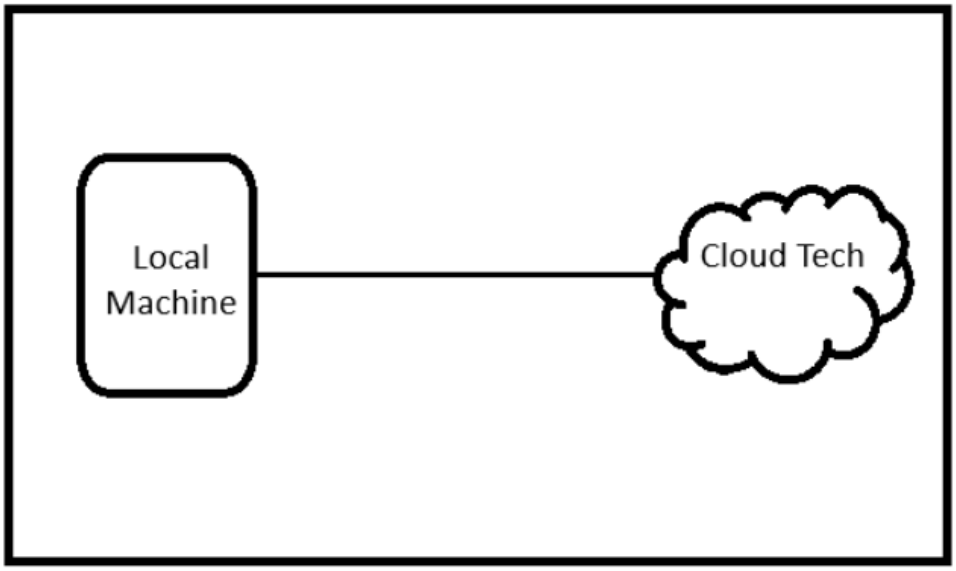


Figure 1: Product Perspective

### 2.2 Product Functions

1. Physical Structure Hierarchy
2. Logging actions & transactions between position
3. Alarming the institution of any pending expiry, overstock or understock
4. Identifying external sources of importation
5. Storing snapshots of inventory records, potentially use them to study the data and infer any decision making
6. CRUD operations on the inventory

### 2.3 User Classes and Characteristics

There are 2 scenarios as to where the image of users within one system is headed. User authentication will be done via tokens.

**Scenario 1: Only One User [Warden Schema]**

The warden will entail all privileges & permissions to act on its will within the system. Nothing stands above it.

**Scenario 2: Multi User Hierarchy [Pyramid Schema]**

Inspired from the Linux user hierarchy system, multiple users will operate on the same inventory with predefined privileges, the root [used to be the warden] will now overtake the permissions' management.

## 2.4 Operating Environment

- **Operating system:** cross platform
- **Hardware:** A compatible computer
- **Software:** NodeJS runtime environment

## 2.5 Design & Implementation Constraints

Specifications: TBD

## 2.6 User Documentation

Documentation will hopefully be delivered in the form of:

1. An offline manual
2. Tutorials and guides on the online website

## 2.7 Assumptions and Dependencies

**Dependencies:**

Techstack to be used:

**Frontend:**

- React

**Backend:**

- NodeJS Runtime
- Express
- MySQL Database

**Encapsulated inside of:** NeutralinoJS / Electron ⇒ Desktop App

# 3 External Interface Requirements

## 3.1 User Interfaces

Write

## 3.2 Hardware Interfaces

Write

### 3.3 Software Interfaces

Write

# 4 System Features

## 4.1 Feature 1: Physical Hierarchy Structure

**Title:** Physical Hierarchy Structure
**Priority:** MAX
Defines a layered structure from geographical locations down to depots and rack slots.

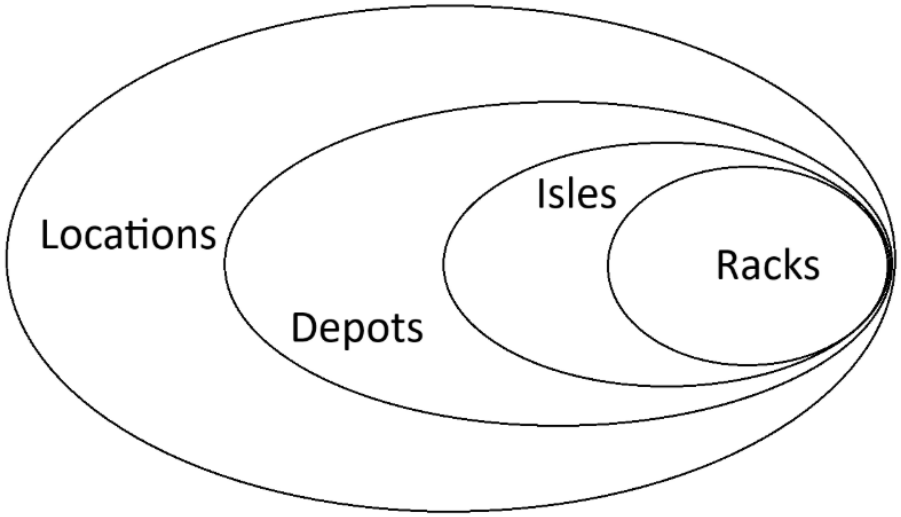


Figure 2: Physical Hierarchy Structure

As the figure above suggests, the software implementation will be in the form of a nested family of different classes, the user will traverse back and forth between these layers to manipulate said inventory.
**REQ1:** Navigation between the different layers
**REQ2:** CRUD operations on instances of the classes

## 4.2 Feature 2: Defining Sources Of Importation

**Title:** Defining Sources Of Importation
**Priority:** MAX
Certain inventories outsource products that do not belong to the institution, defining the sources helps with making the system more modular to observe, study and simulate.
**REQ1:** Setting constant or variant rates of sourcing the products (EOQ)
**REQ2:** CRUD operations on instances of the classes

## 4.3 Feature 3: Tracking & Studying Records of Inventory Snapshots

**Title:** Tracking & studying records of inventory snapshots

**Priority:** MAX

The inventory snapshots which vary in information with respect to time define a time series that allows us to observe trends, patterns and infer decisions from said observations.

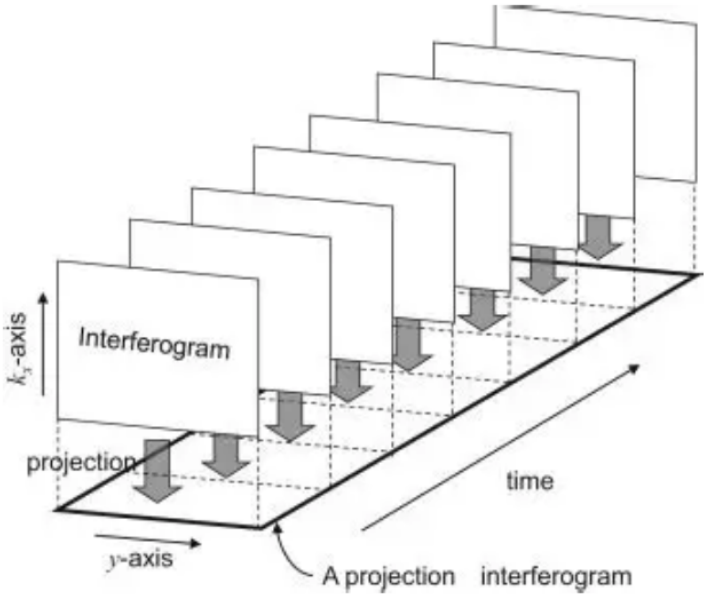


Figure 3: Inventory Snapshots Visualization

**REQ1:** Liberty to choose the precision in days, weeks, months...

**REQ2:** Ability to export the inventory files in tabular format

## 4.4 Feature 4: Data Visualisation

**Title:** Data Visualisation

**Priority:** MID-MAX

The visualisation will vary in forms like charts, pies, bars, graphs etc. And the data will entail the inventory records or the current quantities in stock.

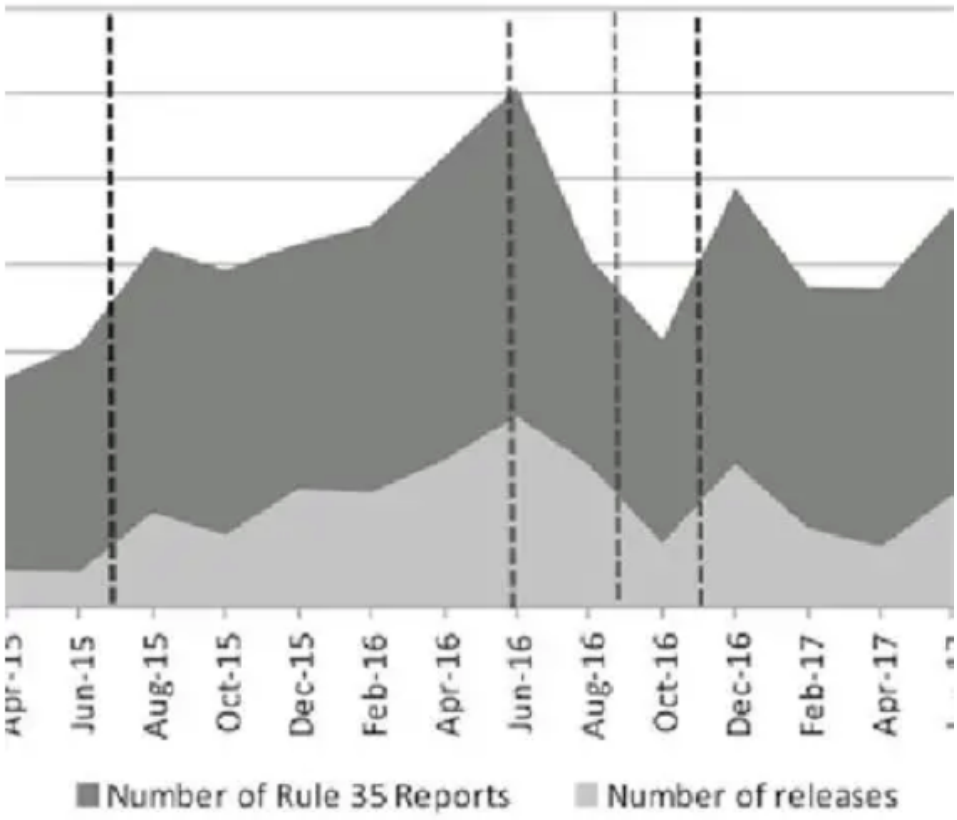


Figure 4: Data Visualization

**REQ1:** Visualisation will be available for any data
**REQ2:** Ability to export the data in:

- CSV, JSON, tabular format

Or visualise them in:

- Pie, bar charts
- Gantt diagrams
- Graphs, Histograms

## 4.5 Feature 5: Tracking Transactions

**Title:** Tracking Transactions
**Priority:** MAX

The transactions in question are the different actions that operate on the inventory's premises, albeit automated like routines or manual like user actions, these transactions will help with traceability and transparency within the institution.

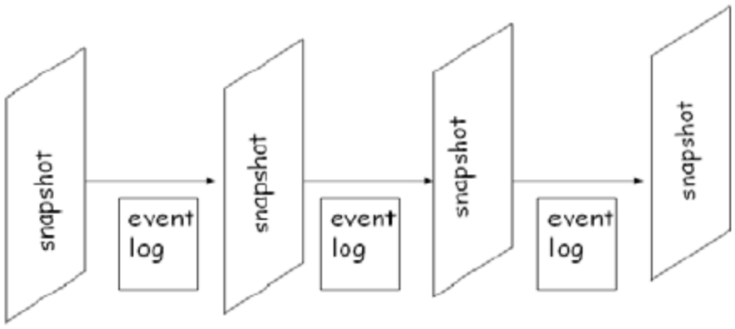


Figure 5: Transaction Tracking

**REQ1:** Filter the transaction logs based on time, responsible user/automated, and affected stock.

**REQ2:** Export the transaction logs in timelines.

## 4.6 Feature 6: Alerts

**Title:** Alerts

**Priority:** MAX

The alerts will notify the users about a pending expiry, overstock or understock.


Figure 6: Data Visualization

**REQ1:** The alert will be configured to notify the institution via an email or a ping in the institution's internal system or simply a push notification.

**REQ2:** The alert will be categorised on how important it is based on user preferences or inferred supply chain annotations.

**REQ3:** The concerned stock will be highlighted indefinitely while the alert's promise is ongoing.

**REQ4:** The alerts will be configured to the user's constraints.

## 4.7 Feature 7: Access Control

**Title:** Access Control

**Priority:** MAX

The user will obtain a product key upon purchase from our website, the product key will then enable the user to access their system. The key will be generated based on the company's credentials.

## 4.8 Feature 8: User Management

**Title:** User Management

**Priority:** Min-Mid

The multi user system will define a hierarchy of privileges and permissions that operate on the system.

## 4.9 Feature 9: Cloud Storage & Backup

**Title:** Cloud Storage & Backup

**Priority:** Mid-Max

The respective institutions will be able to migrate their system definition and back it up in our cloud storage, with additional computation cloud services.

## 4.10 Feature 10: Demand Forecasting & Predictive Analytics

**Title:** Demand Forecasting & Predictive Analytics

**Priority:** Min-Mid

Using the collective data in cloud + the computation service, numerical methods and machine learning can be used to predict certain metrics and behaviours about the inventory to further optimise the actions performed and balance the overstock-understock tradeoff.

## 4.11 Feature 11: Customisation

**Title:** Customisation

**Priority:** Max

The user will customise the system tables, icons, names and coloring to their own liking according to a configuration standard.