**AUTOMATED STORE REPLENISHMENT USING DISTRIBUTED OBJECTS**

1. **Table of Contents**

**2** **Introduction**

**2.1** **Purpose**

**2.2** **Scope**

**2.3** **Definitions, acronyms and abbreviations:**

**2.4** **References**

**2.5** **Overview**

**3** **Overall description**

**3.1** **Product perspective**

3.1.1 Existing System

3.1.2 Proposed System

**3.2** **Product functions**

3.2.1 Hardware Requirements

3.2.2 Software Requirements

3.2.3 Communication Requirements

3.2.4 Operations

3.2.5 User characteristics

**3.3** **Constraints**

3.3.1 Hardware Limitations

**3.4** **Assumptions and Dependencies**

3.4.1 Assumptions

3.4.2 Dependencies

**4** **Specific requirements**

**4.1** **Functional requirements**

4.1.1 Login

4.1.2 Stock report generation

4.1.3 Request for Replenishment

4.1.4 View inventory

4.1.5 View transaction logs

4.1.6 Replenishment analyzer

**4.2** **Non functional Requirements**

# Introduction

Replenishment is the process of refilling stock in a store from a warehouse or a vendor, to avoid stock-out. Typically, one would want to maintain “Minimum Shelf Quantity’, i.e. the minimum number of items on a shelf which is required to maintain a good presentation of the store aisles.

For this project, we are going to consider the replenishment operations of a retail chain called ‘ABC’. The retail chain has been split into 3 regions, and each region contains many of its stores and warehouses. Each store typically contains thousands of items, which have to be replenished periodically, according to needs of the store. The products, which have to be replenished, are shipped to the stores from the warehouses. This is a top level abstract about the replenishment process of a store.

## Purpose

This application is built to assist ‘Store Managers’ with the replenishment process of the retail chain ‘ABC Mart’. Currently, replenishment for ABC Mart happens manually, and this application aims at improving the efficiency and accuracy of the replenishment process by automating the process.

The application would provide intelligent and accurate suggestions to the Store Managers about the replenishment needs, which when followed would increase the revenue of ABC mart by avoiding out of stock situations.

This document is intended to describe the features and functionality that the software must implement and is targeted at the development team.

## Scope

The application will be called ‘AutoRep’

**Scope of this product:**

1. To provide a detailed report of the replenishment needs for a ABC Mart store. These details will include product name, number of products to be purchased, location to be purchased from.
2. The product will allow a store manager to manage multiple stores.
3. Only the suggestions will be provided. Store manager would still have to approve the suggestions.
4. Will efficiently communicate between different distributed systems to make the prediction process fast even when the user load is high.

**Not included in scope:**

1. We assumed that products will be delivered to a store instantly after the store manager approves the replenishment recommendation. Shipping time is not taken into account.
2. Replenishment of the warehouses from the vendors is not in the scope of this application.

## Definitions, acronyms and abbreviations:

**ABC Mart** - ABC Mart is the fictional retail chain for which this distributed system is being developed.

**Replenishment** - Replenishment is the process of refilling stock in a store from a warehouse or a vendor, to avoid stock-out.

**Minimum Shelf Quantity** - Minimum Shelf Quantity is the minimum number of items on a shelf, which is required to maintain a good presentation of the store aisles.

**Regions** - A region is defined as a geographic location, which consists of warehouses and stores for the ABC mart.

**AutoRep** - Acronym for Auto Replenishment.

**MSQ** - Minimum Shelf Quantity

**WLAN** - Wireless Local Area Network

## References

1. IEEE Std 830-1998 (Revision of IEEE Std 830-1993)
2. Cloud computing synopsis and recommendations - NIST document - Special Publication 800-146
3. Oracle Java documentation – (<http://docs.oracle.com/javase/tutorial/>)
4. Spring reference – (<http://spring.io/docs>)
5. Python reference – (<http://docs.python.org>)
6. MongoDB – (<http://www.mongodb.org/>)
7. Apache Tomcat – (<http://tomcat.apache.org/>)
8. MySQL – (<http://www.mysql.com/>)

## Overview

The rest of this document consists of a description of the product in terms of the functionality that it provides. It also consists of a detailed overview of the hardware requirements and limitations that the system has as well as a similar listing of software requirements and limitations.

The document is arranged in a way such that the product characteristics are defined first. It starts with the perspective of the product in the real world with regard to where it fits in with respect to the ABC Mart’s existing infrastructure and then goes on to describe the various product interfaces such as the hardware, software, communication and user interfaces to name a few. The document then goes on to talk about the profiles of the end users of this system and continues with a discussion of the constraints that the system is subject to and the assumptions that have been made during the drafting of this document. Finally, there is an overview of the functional and non-functional requirements of the system.

# Overall description

## Product perspective

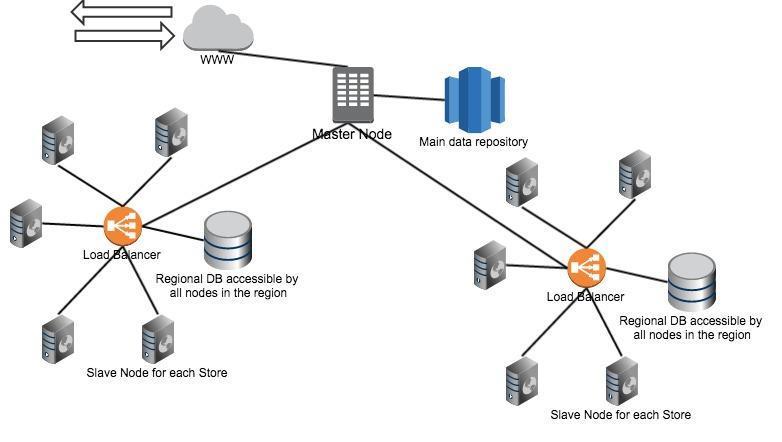
### Existing System

The refilling of stores with item quantities is being done manually. In each store, representatives have to look for available items and its quantities. The quantity of each item in store should be above minimum quantity threshold value to present the items to customer. If the quantity of an item falls below the minimum threshold value, the store manager would make a manual request to replenish that item from warehouse.

### Proposed System

The objective of the proposed system is to automate the store replenishment process with less human intervention. When an item quantity falls below to minimum value, the system should notify the store manager. For each product, minimum threshold value should be retained. At the end of sale, an automatically generated report will be sent to the warehouse. In addition to that, products available in the warehouse are being monitored by system parallely. If any of the products fall below the threshold limit of warehouse, an automated request will be sent to region warehouse.

The following diagram depicts the low level design on the proposed system.



## Product functions

### Hardware Requirements

1. 1 Gigahertz or faster x86 (32-bit) or x64 (64-bit) CPU
2. 1 Gigabyte of RAM
3. Internet access
4. 1024 x 768 or better resolution display

### Software Requirements

1. Spring 3 MVC Framework: Traditional J2EE is cumbersome to use and does not provide an easily usable view for application code. So, as the project keeps growing, it will become difficult for 4 developers to simultaneously expand and manage the codebase if tradition J2EE is used. Spring 3 MVC is a lightweight framework, which provides rich functionality for building robust web applications. Spring 3 MVC is highly configurable, hence code management and expansion becomes easy. Spring 3 also provides excellent support and APIs for working with RESTful web services, which is going to be a mode of communication for the distributed objects. Hence, the choice of a web framework (Spring 3 MVC), which provides the infrastructure for rapidly building robust web applications, is important.
2. Python scripts will be used to run timed automated processes on the database such as a batch process. Python is chosen because of its ease of use. It is possibly the most readable languages around and that makes it easy to maintain and extend and also, it makes it ideal for internal use.
3. MongoDB: NoSQL database for flexible aggregation for the centralized system. This is a solution that uses its own custom Binary JSON format known as BSON. This solution is ideal for analyzing log data. The NoSQL solution will host the transaction logs of the system. If in the future, there is a requirement for data mining on these logs, two dimensional joinless table scanning is required. Furthermore, for large amounts of data, distributing processing using a system such as Map/Reduce makes sense. MongoDB has its own Map/Reduce implementation built in to it and can be accessed using the simplified interactive JavaScript Shell that comes with it.
4. MySQL database for processing relational data for regional database. In local store, quantity of store items will be changed depending on customer’s purchase, to maintain stock quantities adhere to threshold value purchase details are manipulated with help of relational database. MySQL supports ACID transaction properties and suitable for high-speed transactional processing of a web application that services huge queries in peak transaction hours.
5. Apache - Tomcat 6 Server is used for ABC Mart which implements the servlet 2.5 and Javaserver pages 2.1 and includes many additional features that make it a useful platform for developing and deploying web applications and services.

### Communication Requirements

1. IEEE 802.3 (Ethernet)
2. IEEE 802.11 b/g/n (WLAN)
3. Transmission Control Protocol (TCP)
4. Internet Protocol (IP)
5. HyperText Transfer Protocol (HTTP)

### Operations

1. **Login**
   1. User will be prompted to enter credentials on the login page. Upon successful authentication, the user will be taken to the home page. If the authentication is unsuccessful, an error message will be displayed.
2. **View Inventory for a specific store**
   1. The user will be able to view the current inventory for a store, which is linked to him.
3. **View replenishment dashboard**
   1. A dashboard dedicated for replenishment suggestions would display the replenishment needs for every store that the user is linked to. A detailed report for each store would be accessible in the dashboard page.
4. **Approve/Reject replenishment request**
   1. The user will be able approve or reject the replenishment suggestions provided by AutoRep. On approval, the stores will be replenished according to the suggestions provided by AutoRep.

### User characteristics

1. The user will have basic computing skills and will have the knowledge required to operate a modern web browser.
2. The user will be a store manager. Hence, the user will be able to make rational judgments to verify the correctness of the predictions provided by AutoRep.

## Constraints

### Hardware Limitations

Minimum 4GB-RAM is required to create enough instances to deploy the application over the network.

Minimum Internet connection (of bandwidth > 2-Mbps) is needed for transferring data to other instances.

## Assumptions and Dependencies

### Assumptions

Following are the assumptions that hold for this project.

1. The warehouse has fully defined schema and fully populated databases that reflect the quantities in the warehouses’ storage.
2. Users and access control lists of the system are already defined. It should prevent unauthorized access to any anonymous users.
3. External vendors automatically replenish warehouse stores.
4. Once the product replenishment request is made by store manager, the time taken for shipping time is assumed to be zero i.e., products are replenished instantaneously.

### Dependencies

The project depends on below factors,

1. The client and servers communicate over the Internet over a secure connection.
2. All servers must have uninterrupted power supply in order to ensure maximum uptime. In case of failures in the power grid, sufficient backup generators must run in order to supply an uninterrupted power supply.
3. A stable connection must be maintained between the various nodes in the system.

# Specific requirements

## Functional requirements

Following are major functional requirements for this project.

### Login

A valid user should be able to login to the system from the login page. The login page will prompt the user to enter credentials. On successful login, the following will happen:

1. The user will be redirected to the home page.
2. The user will be linked to all the stores he manages.
3. On unsuccessful login, error messages will be displayed.

### Stock report generation

Store managers will be shown a report whenever a product’s quantity is below a critical threshold value. A timed, automated program that performs checks to identify products that need to be ordered again for replenishment will generate this report. This automated program is described in following section of this document. Typically, the automated program will raise flags to the system, which will then generate reports to the store manager for further action. The reports generated by the system are displayed to the store manager. The store manager must then approve the report for the system to initiate a request for replenishment.

### Request for Replenishment

Generated report will be processed and consolidated for making decisions for auto replenishment process. The product that needs to be replenished will be grouped and provided to store manager. Store manager can review the products and send confirmation for replenishment to warehouse.

### View inventory

The user will be able to view inventory details for each store, which he/she manages. This report will contain the store name, product ID, product name, quantity and product expiry date.

### View transaction logs

The user will be able to see a human friendly version of all the replenishment requests that have been sent from the store to the warehouse since the deployment of the system. The user will be able to make judgments about the items that are commonly ordered from the store.

### Replenishment analyzer

An hourly batch process will calculate the replenishment needs for every store of ABC Mart. These suggestions will be shown to the user who logs in to the system. This batch will compute replenishment suggestions based on the business rules. Suggestion lists provided by the replenishment analyzer will be refreshed and updated hourly.

## Non functional Requirements

Consistency - The state of the database should be consistently maintained, regardless of how many times it is accessed or the processes accessing it.

Concurrency - This system should work normally without delay in response time when many users access it simultaneously.

Scalability - The system should be tolerably responsive regardless of the amount of data being processed by the automated program. The system should also be responsive even if a number of store managers simultaneously make requests.

Robustness - The Client systems (Local ABC mart stores) should cope with errors like network failures, System crash by generating reports automatically.

Response Time - The system will not exceed the maximum latency ~5 seconds.