# ABC’s Inventory Management System

## Non-Functional Requirements Specification

## Introduction

This document specifies the non-functional requirements of the system. This document will state the functional requirement of the system followed by the non-functional requirement. Moreover, the system will describe the expectations of the system interface including the user interface and the interfaces to external systems or devices. Finally, the document will describe the system constraint and compliance respectively.

## System-Wide Functional Requirements

**Searching**

The system must be able to search for a product or a product item in each location using both product code and product item code. The product code and product item code must be assigned for each existing product and product item.

**Transferring**

The system must allow users to transfer stock to other locations (different store or warehouse). The system must automatically adjust stock levels in each location when stocks are transferred. The receiving store stock adjustment must only be done once it has been confirmed that stocks have been received.

**Stock Management**

The system must allow users to add new accounts, edit details about existing accounts and delete existing accounts. Moreover, the system should allow only users with admin rights (warehouse staff) to do these functions.

**Requesting**

The system must allow users to request for stocks from other stores or warehouse. If a staff realizes that the stock level for a particular item is low, the system must allow the staff to request for the item or a group of items to other locations.

**Report and reporting**

The system must allow users to create a report of all transfers made dynamically when needed. This would allow the business to use the data collected to make business decisions in the future.

**Notification**

The system must notify users belonging to the receiving location of a stock request. This allows users to know that a location has made a request and act according to the situation. If the requested items still exist and is not needed in that location, stock can be arranged to be sent to the location that has requested for it. If not, the request can simply be ignored.

## System Qualities

**Performance**

The system must have a reasonable response time. Lagging of the program should be avoided. Queries should be processed in a reasonable time (less than 3 seconds). Processing time must be less than 3 seconds. During high traffic, the processing time should be 5 seconds at most.

**Security**

The system must have login requirements for the users. The system should only accept new passwords that are at least 8 characters in length and that have digits or special characters in them. Similarly, the store staff must not have admin functions (high level functionalities like adding new user, delete stock, add stock and so on). The warehouse staff should have more functionalities like manage user and manage stock. The system should logout automatically after 5 minutes of inactivity.

**Availability**

The system should be available 24/7. Store and warehouse staffs will need to use the system to check stock level, stock availability and status at any time, the system should be always available to deliver. The server and database must be deployed to the cloud so that it can be accessed from any location at any time. If maintenance needs to be made, a notice must be sent to all staff and should be done outside business hours so that the business is not highly affected during system maintenance.

**Reliability**

The system must be able to generate real time data. The system must make changes to the database accurately and give out appropriate messages when something goes wrong. The system can have at most 100 hours of downtime per year. After breakdowns, the system must be able to restart and continue normal functions.

**Usability**

The system should have a very basic and consistent UI. The system must have a white or gray background while buttons and links should have black font color so that these elements are very clear to see. The system must be available in English. The system must be displayed full screen when launching the application by default. However, the system should allow users to change the size. The system must display appropriate messages. The system should follow Neilson’s 10 usability principles to improve usability.

**Capacity**

The system must be able to handle 100s of requests per hour. The business still being of small scale, the system will not be swamped with requests from different stores. The system should be able to handle the order, delivery and request handling. However, in case of a business growth in the next 3 years, the system should be able to handle at least 1000 requests per hour. Similarly, the system should be able to store data and information of all products, stores and the warehouse.

## System Interfaces

### User Interfaces

* 1. Look and Feel
* The look and feel of the system should be very basic. Basic colors like black, white and light blue should be used for texts, background and decorations. Any other decorations should be avoided or changed. Animations are not a priority however, it is acceptable. Animations and decorations which help in easy navigation of the system for the users is acceptable.
  1. Layout and Navigation Requirements
* When user logs in the system, the first screen should have an overview of the system, displaying major options user might select like send stock, search stock, create report and so on. When user selects an option, they should have the ability to go back to the previous screen or select menu to navigate to other options.
  1. Consistency
* The system must cover full screen by default. Font color, background color and decorations must be consistent in all screens of the system.
  1. User Personalization and Customization Requirements
* Store user accounts and warehouse user accounts will have the same presentation. However, warehouse user accounts will have more functionalities. Similarly, individual users will not have the ability to customize content displayed or personalize displayed content.

### Hardware Interfaces

The system needs to connect with a printer and a barcode scanner to print a label while sending stock and scan barcode from a label while accepting stocks. The following are the logical structure, physical addresses and expected behavior of each hardware interface:

|  |  |  |  |
| --- | --- | --- | --- |
| Hardware | Logical Structure | Physical Address | Expected Behavior |
| Printer | Java Print Service API  Printer | The physical address (MAC address) of the printer will differ in each location (store or warehouse). Connection to a printer is done using in built Java API which allows a print job to be created, associate it with a printer and print the print job. | Print the assigned print job. |
| Barcode Scanner |  | Barcode Scanner translates a barcode to a readable code. Therefore, if the cursor is in a text field and a barcode is scanned, the readable code is displayed in the text field automatically as if type via a keyboard. Therefore, a specific physical address for a barcode scanner is not required. | Scan a barcode and translate it to readable code in a text field. |

###### System Constraints

Following are the constraints on the system being built:

* Proposed application server technology could not be enough for processing data in case of rapid business growth in the future.
* The system is not available for mobile devices and is focused on the windows operating system.
* The system is dependent on the internet connection in each location.
* Team members do not have expertise on required analysis and design skills like interviewing, design documents and so on.
* Team members do not have expertise on the required programming languages to build the system.
* ABC being a small business, the budget for the current project might not be enough to support the licensing requirements for products that need licensing for building a system that can handle large scale transaction in the future.

###### System Compliance

###### Licensing Requirements

The software does not need any licensing for the tools and products being used to build the system as all tools and products will use tools with permissive software license.

###### Legal, Copyright, and Other Notices

All rights of modification and distribution are reserved to ABC retail company. After signed upon for acceptance of the system, the developers of the system cannot be held liable for any misuse of the system. The developers do not hold any warranties for the use of the system. The developers cannot be hold accountable for any business losses, data leaks or data loss caused due to the misuse of the system.

###### System Documentation

Following the development of the system, a user manual must be developed describing each step to perform a task to accomplish each business needs. The documentation must provide detail description of each steps including a picture of the screen in each step. The documentation must also clearly state what the requirements are for using the system. All developers will be responsible for creating this document. Developers will be expected to collaborate and write this document clearly avoiding jargons making it standard script.