# ABC’s Inventory Management

## Architecture Notebook

## Purpose

The purpose of this document is to describe the architecture philosophy, decisions, constraints, justifications, significant elements, and other overarching aspects of ABC’s Inventory Management System that shape the design and implementation of the system. This document will firstly describe the goals and philosophy of the architecture. This will be followed by any assumptions and dependencies effecting the architectural decisions. The document will then describe the architecturally significant requirements, decisions, constraints and justifications made. Different architectural mechanisms driving the design and implementation will also be described in this document. Finally, the document will present the key abstractions, architectural framework and architectural views of the system.

## Architectural goals and philosophy

The main architectural goal of ABC’s Inventory Management System is to provide a basic inventory management tool that can be used by the store and warehouse staff of the company. The system needs to be built from scratch since the business is new and has never used a digitized online inventory management system. The system should also be able to handle at least 1000 requests per hour. Moreover, the system must be strong and avoid crashes as much as possible as this might affect the daily business in store and the warehouse. Similarly, the system must address minor crashes by rebooting automatically and create a report of the crash. The wait time for users for any action should not be more than 5 seconds. Scanners and printers are essential part of the business hence the system, the system should be able to access these devices to scan barcodes, print documents and so on.

## Assumptions and Dependencies

The following are the assumptions and dependencies that drive the architectural decisions.

* All product and current inventory data will be available during the implementation of the system.
* Store and Warehouse employees have basic knowledge about computers and windows operating system.
* Project team members are familiar with the Unified Process.
* Project team members are familiar with core Java, Java EE, XML, JavaScript and MySQL.
* Computers with enough are available to deploy the system.
* Physical devices like scanner and printers are available.

## Architecturally significant requirements

The following are the architecturally significant requirements of the system:

**Performance**

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

**Availability**

The system should be available 24/7. Since store and warehouse staffs might need to use the system to check stock level, stock availability and status at any time, the system should be always available to deliver. If maintenance needs to be made, a notice must be given to all staff and should be done outside business hours so that the business is not highly affected during system maintenance.

**Reliability**

The system can have at most 100 hours of downtime per year. For normal breakdowns, the system must be able to restart and continue normal functions. For major breakdowns, the system must be able to display appropriate message which helps to identify the severity of the breakdown.

**Capacity**

The system must be able to handle 1000s of requests per hour. The business still being of small scale, the system will not be swamped with requests from different stores. 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.

## Decisions, constraints and justifications

The following are the decisions, constraints, and justification regarding the architectural approaches:

|  |  |
| --- | --- |
| Decisions or Constraints | Justifications |
| The system will have a basic UI with minimum UI design. | * Users will find it easy to navigate through the application. * Less time can be spent on beautifying the user interface and more time can be spent on solving the business issues the application is designed to solve. |
| The system will use Glassfish as the application server. | * Allow 1000 requests per hour. |
| Windows OS will be used in the client computers. | * Windows being very popular, simple and easy to use. |
|  |  |
|  |  |
|  |  |

## Architectural Mechanism

The following are the architectural mechanisms that will be used in the system:

Architectural Mechanism 1

## Key Abstractions

The following are the key abstractions of the system:

* Warehouse – This represents the warehouse that distributes the stock to stores.
* Store – This represent the stores where product is sold.
* Product – Items that are sold, requested to be transferred from warehouse and sent through deliveries.
* User – Warehouse and store staff that will be using the system.

## Layers or architectural framework

The following diagram represents the layers of the system:

User Interface Layer

Business Logic Layer

Data Layer

## Architectural Views

**Logical**