# Online Application Title: Mart Inventory

Team : Palash Kardhekar

&

Bhavesh Verma

# SRS Document:

Purpose of Project:

This document contains the complete software requirements for Mart Inventory and describe the design decisions, architectural design needed to implement the system. It provides the visibility in design and provide

information needed for software support. Its new, reliable and fast Mart inventory management which helps for managing daily routines.

Mart Inventory is a web application which provides a platform that will automate various tasks associated with handling product catalog along with execution of proper billing system and Better organizing the stored information and optimum performance there by Helping the business to ensure smooth working of the process.

# Scope of project:

# This System allows Clients to maintain their products for adding or removing from catalog based on their availability.

# Manager will be able to review orders history and may able to cancel order, place order.

# The System will be able to show live Business Operation statistics trends through Customized dashboard for manager/owner.

# The system will enable to do transactions and produce bills for customers.

# Definition:

MI: Mart Inventory

QA: Quality Assurance

Portal: Personalized Online Web Application

MIS: Management Information System

CRM: Customer Relation Management

BI: Business Intelligence

KPI: Key Performance Index

# Overview:

Mart Inventory is a web application. It provides a platform that will automate various tasks associated with handling product catalog and Execution of proper billing system better organizing the stored information and optimum performance. Helping the business to ensure smooth working of the process that provides the function and features to authenticate and identify the sales person.

It provides then with easy, intuitive, personalized and user-customizable web-interface for facilitating access to information and services that are of primary relevance and interests to the sales person and manager.

.

# Functional Requirement:

This section provides requirement overview of the system.

Registration: The sales person as well as manager needs to be registered to the application.

Login: The sales person and the manager need to be logged in onto the application for accessing the products or updating any products and do the transactions.

Logout: After accessing the requirement or updating of information the user has to logout from the system.

# Non-Functional Requirement:

* Secure access to user’s confidential data.
* 24X7 availability.
* Better designing to get best performance.
* Various non-functional requirements are:
* Security.
* Reusability.
* Maintainability.
* Reliability.
* Portability.
* Extensibility.

# List of features:

* Employee (Sales person and manager) management.
* Bill management.
* Customer management.
* Products management.

# ER listing:

* + Products:

ID

Product Name

Product Type

Discount

MRP

Product Quantity

Product Type

Product Items

* + Product Location:

ID

Product Location

* + Category:

ID

Category Name

Description

Quantity

* + Customer Bill:

ID

Total cart price

Total Items

* + Customer:

ID

Customer Name

Customer Mobile

Customer Address

* + Bill Details:

ID

Created On

GST

Total Cart Price

Price

Total Amount

* + Shopping Cart:

ID

# **Application Architecture:**

Application = Logic + data

Logic =(UI Logic + Business Logic + DataAccess Logic) Data =( structured data , Non Structured data)

# Online Application:

* + Web based Application: deployed on web and accessed by users from anywhere.

[Mart Inventory application------Web application-- used remotely by end users (sales persons, mart owners).]

# **Logic:**

* + UI Logic:

Web Pages + HTML controls + Web Components (React) Navigation: (UI Routing) HTML Links, Routing mechanism Data Binding: DOM + JSP tags (JSTL).

* + Event Binding: action handler.

HTTP Request: GET Doget

POST: -------------Dopost

Put:

Delete:

* + Client Side UI----------------HTML, CSS, JavaScript, bootstrap

UI (Client side UI Framework) React.

* + Web Logic: (Server side processing)

Server UI---------------- JSP, servlet, (classical java web technology)

SpringBoot API CRUD REST API

ORM Technique: Hibernate (ORM) JPA

Client-side state management

cookies, query string, form collection, hidden variables local storage, session storage, Web SQL, Server side state management session, Cache, database

* + Business Logic: Java console application will be used to test your business Logic

**Advanced Java:** will contain

* + 1. **Business query processing**
    2. **Business operation management**
    3. **Business data manipulation**

# **From Mart Inventory application point of view**

* + **Modules:**
* Registration: Sales person Registration, Manager Registration.
* Products: Add products, update products, delete products, view products
* Customer: Customer Catalog(Add, Retrieve, Update, Delete)
* Transaction: Adding products in bill, Final billing
* Security: Authentication, Authorization (get email Id, get PRN, create password, change password)

# **DATA:**

* Structured Data
  + - 1. RDBMS
      2. fields
      3. tables
      4. constraints
      5. Not null, auto increment, PK, FK, Unique, check

Add some dummy records in your newly created database.

Write reusable SQL queries against those database tables to check business queries

Test those SQL queries on existing dummy database you built.

* List of Tables :
* **Product :**

**Fields** : ID, Product Name, Product Type, Discount, MRP, Product Quantity, Product Type, Product Items (Primary Key : ID)

* **Product Location:**

**Fields :** ID, Product Location (Primary Key : ID)

* Category:

**Fields :**  ID, Category Name, Description, Quantity (Primary Key : ID)

* Customer Bill:

**Fields :** ID, Total cart price, Total Items (Primary Key : ID)

* Customer:

**Fields :** ID, Customer Name, Customer Mobile, Customer Address (Primary Key : ID)

* Bill Details:

**Fields :** ID, Created On, GST, Total Cart Price, Price, Total Amount

* Shopping Cart:

**Fields :** ID

* + Create .sql file
* ddl.sql--->file will contain DDL commands for Table creation
* dml.sql--->file will contain insert commands for filling dummy data to tables which we have created
* bqyery.sql-->file will conatin SQL queries mapped for business queries