TABLE OF CONTENTS

0 Preface 1

0.1 Purpose of this document 1

1 Introduction 2

1.1 Purpose 2

1.2 Scope 2

1.3 Overview 2

2 System Overview 3

2.1 System Architecture 3

*3* System Design 4

3.1 Naming conventions 4

3.2 Programming Standards 4

3.3 Software development tools 4

3.4 Assumptions 4

Document Control 5

Document Signoff 5

Document Change Record 5

0 Preface

0.1 Purpose of this document

1. This is a that provides guidance which is intended to assist the relevant management or technical staff, whether client or supplier in developing or maintaining the Message Processing System (JPM‑MP).

# Introduction

1. The Message Processing application is used to read the incoming messages and process them to save required data such that the system can generate reports as and when applicable to monitor the sales of the products.
2. The application will be developed in Windows and the programming language involved is Core Java. As Java is robust and platform independent, we can develop this application once and run later under various platforms to serve the needs. There are no UI or database interactions in the product. If required, the UI/ database integration will be picked as a new design requirement as new enhancement.

## Purpose

The purpose of this document is to understand the Message Processing application and identify the operations involved in order to develop the application to suit the needs of the end user. This document is intended for junior/ senior developers, Solution Architects, Business Analysts and the Project Manager to understand the needs and arrive at the required solution.

## Scope

1. The Scope of the document is to develop a software product using Java to process incoming messages from the sender (file) and save them in memory to generate sales report of the products.
2. This system will accept incoming messages and generate sales report after every 10 sales and a consolidated report after the 50th sale. Upon generating the 50th sale report, the system will not accept any incoming message and all new messages are to be processed as part of next run of the application.
3. The benefits of the application is to track the sales automatically thus eliminating any human error that might have a direct effect on the sales and the pricing of each sale.

## Overview

##### Section 1 is the introduction and includes a description of the project.

##### Section 2 provides a system overview.

##### #3 Section 3describes the system design.

##### #4 Section 4 contains the component descriptions.

# System Overview

## System Architecture

1. The System Architecture of this application is given as a flowchart below.

Start

Generate report of sales made

Generate report of the completed sales

A

End

A

On completion of 50 sales

On completion of 10 sales

Receive Message and process sales data

Yes

Throw error that the file is unavailable.

Is File Present?

Load the file

# System Design

## Naming conventions

1. Package name of the project should always have “com.jpmc.msgproc” followed by the relevant package name. Example – com.jpmc.msgprog.xyz, com.jpmc.msgproc.abc
2. The naming conventions of a Class should be followed as per Java naming conventions – Class name should begin in uppercase and the first character of every word should be written in uppercase. Example – ClassName.java
3. The variable names should be declared in camel case as per Java standards.

## Programming Standards

1. The code should adhere to Java programming standards and follow the standard best practices of Java programming.
2. The code should be modular.
3. Comments should be provided in all classes/ methods as and when required.
4. The code format should be consistent across all classes.

## Software development tools

1. The tools that are required for the development of the Message Processing application are as follows,

Eclipse (Kepler and above)

Maven 3.3.9

SonarLint for Eclipse (Eclipse plugin)

EclEmma for Eclipse (Eclipse plugin)

Version Control - Git

## Assumptions

1. The messages are sent to the application through a file input as no UI is applicable (for exercise purposes)
2. Code will only ever be executed in a single threaded environment.
3. A Sale value should always be in ‘pence’
4. Output will always be displayed as plain text in console and not as Output file (for exercise purposes)
5. System will perform processing for 50 messages per run only.

Document Control

|  |  |
| --- | --- |
| **Title:** |  |
| **Issue:** |  |
| **Date:** | 8 |
| **Author:** |  |
| **Filename:** |  |

Document Signoff

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nature of Signoff** | **Person** | **Signature** | **Date** | **Role** |
| Author(s) | Aravind Rajasekaran |  |  | Project Member |
|  |  |  |  |  |

Document Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Change Details** |
| 17October 2018 | Issue 1 Draft 1 | Aravind Rajasekaran | First complete draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |