California State University Fullerton

CPSC 462

Object Oriented Software Design Supplementary Specification for the

High Velocity Sales Technology System

Ryan McDonald
Senior Architect
rtmcdonald96@csu.fullerton.edu

Alexander Frederick
Senior Technical Director
afrederick2@csu.fullerton.edu

Benjamin Baesu
Senior Localization SW Designer
bbaesu@csu.fullerton.edu

Revision History:

Version	Date	Summary of Changes	Author
1.0	September 28, 2020	• Initial Release	Ryan McDonald Alexander Frederick Benjamin Baesu
1.1	November 9, 2020	• Added additional Reports and adjusted previous reports to encompass additional functionality	Alexander Frederick
1.2	December 2, 2020	• Various spelling and formatting fixes.	Benjamin Baesu

Table of Contents

1 Introduction and Purpose

This supplementary specification documents the system requirements that are not captured in the use cases and use-case models.

This includes

- Legal and domain rule requirements
- Component and interface requirements
- Quality and functionality requirements

The purpose of this document is to give developers, managers, implementers, testers, etc. a document to refer to for the system requirements that are not clearly drawn and captured from the use cases.

2 Feature Requirements List

- Most features require the use of an internal database
- All features require authentication prior to system access
- Purchase feature requires a 3rd party payment system
- Purchase order feature requires a two-person authentication

3 Functionality (common across UCs) Requirements

3.1 System Event Logging Requirements

System shall log all errors to a persistent storage device within 100 milliseconds of error.

System shall log system authentication to a persistent storage device with 100 milliseconds of authentication.

System shall generate server backups every hour.

3.2 Security Requirements

System shall require authentication prior to system usage.

3.3 Other Requirements

System shall connect to database servers with 100 milliseconds of request.

System shall respond to requests within 100 milliseconds.

4 Quality Attribute Requirements

4.1 Usability Requirements

All system event requests shall be as user friendly as possible. In each feature case there should be a set list of user groups allowed to use that feature. If a user group is not allowed to use that feature, they should not even be aware of its existence.

4.2 Reliability Requirements

The system shall maintain a high level of reliability. A minimum of 23 hours per day of uptime is required.

4.3 Performance Requirements

The system shall be able to perform hundreds to thousands of transactions per minute. Each individual event or transaction should have a definition for minimum reliability.

4.4 Supportability Requirements

The system shall be simple to support and maintain. Each feature should have robust documentation in order for any adjustments to be made by the lowest level of authorized persons.

4.5 Adaptability Requirements

The system shall be adaptable on any high-risk features. Any external requirement shall have robust documentation to define how to adapt to adjustments to original specifications.

4.6 Configurability Requirements

The system shall be configurable for any number of user groups and definitions of those groups. Each feature shall have a defined list of user groups allowed to use; this list shall be configurable at the client level.

5 Implementation Constraint Requirements

- System shall be implemented in a 3-tier layered architecture: Presentation Layer, Domain Layer, and Technical Services Layer.
- System shall be implemented in a C++ environment.
- System shall be able to adjust 3rd party authentication without code base adjustments

6 Purchased Component Requirements

- Tax rate calculator. Shall support custom tax rate for special groups such as: Military, Government, Religious groups, etc. Shall support different global tax rates.

7 Free Open Source Component Requirements

It would be beneficial to maximize the use of free open source technology for this system. At this state in the process it would be premature to create a definitive list of technology to use, however the following software could be good candidates as we go further into development:

- Fluentd
- Nagios
- LOGalyze

8 Interface Requirements

8.1 Noteworthy Hardware and Hardware Interface Requirements

- The system shall have internal databases on site for back up purposes
- The system shall require a networked structure in order to better maintain reliability

8.2 Software Interface Requirements

- The system shall require a networked front-end system such as a web page or mobile application
- The system shall require an internal database software to be decided on at time of implementation

9 Application-Specific Domain (Business) Rule Requirements

ID	Rule (Requirement)	Changeability	Parent Domain Rule ID
RULE1	Customer discount rules. Examples: Wholesale Customer: 10% off Employee: 20% off	High. Depends on rules of company.	RULE7
RULE2	Sales discount rules. All discounts apply to total prior to taxes and S&H. Examples: 10% off order of $25 or more. 15% off order of $50 or more. 20% off order of $100 or more.	High. Depends on rules of company. Sales may be rotated at a monthly, daily or hourly rate.	RULE7
RULE3	Product discount rules. Examples: 10% off bottom wear this week. Buy one shirt get one 25% off.	High. Depends on rules of company. Sales may be rotated at a monthly, daily or hourly rate.	RULE7

10 Legal Requirements

- Open source software is recommended if their licensing restrictions and terms of service allow for the software to be used in a resale environment.
- All tax laws, rules and procedures, shall be applied during each sale. These tax rules and laws can change frequently.

11 Information in Domains of Interest

- System shall maintain original product price in the event of a product price markdown for tax and accounting purposes.
- Because of the nature of the complexity of sales tax. Delegation of sales tax calculations to a third-party software is advisable. Depending on the location of the purchase, tax may be owed to multiple government bodies. Certain items may also be tax exempt depending on the item or the status of the individual purchasing the item.
- HVST needs to support the different identification methods used for the products. Whether that be, UPCs, SKUs, or and item ID.

12 Report Requirements

12.1 Purchase report and Billable Invoice

12.1.1 Description

This is a common sales receipt, but the format will be used to display purchase total prior to payment. This purchase report can also be used to generate a billable invoice in the case of a payment plan or system information loss. The receipt should contain each item ID, item Name, quantity, cost, final tax, final subtotal, and final total.

12.1.2 Format and Content Requirements

Item ID	Item Name	Qty	Cost	Sale

0123456789	Shirt	1	4.99	5.99
			Tax	1.11
			Subtotal	5.99
			Total	7.1

12.2 Current Inventory Report

12.2.1 Description

This is a common inventory report. The report should contain the item ID, item name, quantity on hand, cost, sale cost, markup value, description of item, and any other various descriptors required.

12.2.2 Format and Content Requirements

Item ID	Item Name	Qty	Cost	Sale	Markup	Description	size/color/gender
0123456789	Shirt	1	4.99	5.99	10%	Shirt with design	Various

12.3 Sales Order report

12.3.1 Description

This is a common list of sales by customer. The report should contain each saleID, Date, and total sale for a given customerID sorted by date.

CustomerID	Date	SaleID	Total
Alex	10/9/2020	100920201	4.99
Alex	10/15/2020	101520201	25.99
Alex	11/1/2020	110120201	36.99

13 Packaging

The system shall be packaged in an orderly manner. The system header and CPP files will be named accordingly to their namespace requirements. The system shall then be compressed or built into an executable file on site.

14 Technical Standards

- Robust documentation on all coded files is required
- Namespace allocation is required on a per file basis
- Variable naming will be in a clear and concise camel casing format
- Constant naming will be in a clear and concise all capitals format