
Software Requirements Specification

for

Warehouses Management System

Version 1.0 approved

Prepared by //TODO: Team Name

Elijah Rey Espiritu

Hoyeon Moon

Binkang Yu

September 21, 2020

1. Introduction

1.1 Purpose

This Warehouse Management System is designed for a computer system. The users are the manager of this application and will perform appropriate actions in the warehouse to achieve certain changes or updates. This can include actions such as adding new products, replenishing stock, or displaying the current inventory. The warehouse(s) and the salesperson(s) who work at the store can be added and updated over time.

1.2 Document Conventions

Different levels of priorities will be applied to each requirement.

1.3 Intended Audience and Reading Suggestions

This document is intended for potential marketing staff and programmers. In the second half of the document, it also contains details on the development of this program.

Reading order may vary among different readers. For programmers, the document can be read in normal order. However, if there are potential marketing staff or persons who have less knowledge in this area, section two or four can be skipped.

1.4 Product Scope

The Warehouse Management System is designed for everyday use. The user can use this application to manage warehouses and execute specific instructions related to a warehouse, product, or salespersons. For example, When a customer orders an item, this application allows the user to create an invoice for the customer and if the invoice isn't fully paid within 30 days, the system will apply compounded interest to the invoice. This application also allows for easy logging of things such as inventory, salespersons, and invoices.

1.5 References

[Use Cases - //TODO: Team Name - 09/16/2020 - v.1.0](#)
[RFP - //TODO: Team Name - 09/16/2020 - v. 1.0](#)

2. Overall Description

2.1 Product Perspective

The goal of this software is to assist with the running of a store by allowing for menial tasks such as checking/replenishing inventory to be performed more efficiently. This application is a new self-contained product that operates on any system that can run Java.

2.2 Product Functions

Password protected with the ability to change it
Add, replenish, ship, or display products
Open, close and display invoices
Add salespersons, show their performances, or set or pay their commissions

2.3 Operating Environment

The Java program will run in any computer system with at least 1GB RAM and 300MB storage.

2.4 Design and Implementation Constraints

Since this program will be developed in Java, any store using this software will need the software required to run Java programs (such as the Java Development Kit).

This software will have a timing requirement in that it will need to be completed before the end of the 2020 semester.

The customers organization will be responsible for maintaining the delivered software but it is designed to make that more tolerable (such as the ability to add warehouses).

2.5 User Documentation

Specific instructions tips will be provided to the user when the application is first used.

2.6 Assumptions and Dependencies

One assumption is that the computer using this software will be capable of running Java. Another assumption is that the program will have enough performance as well as memory to run the program. Memory use will also increase over time since adding new products, warehouses, etc. will increase the memory use. In the scenario where one or more of these assumptions are incorrect, the program may not run well or at all.

3. External Interface Requirements

3.1 User Interfaces

Management APP

Enter password here

Login

Main Menu

Warehouses status

Invoice

All Products

Salespersons

Password

Next page

Check Stock Add Products

All Products

1. Televisions
2. Home Appliances
3. Home Audio
4. Cell Phones
5. Tablets
6. Headphones
7. Digital Cameras
8. Portable Speakers
9. Car Electronics & GPS
10. Drones & Collectibles

1 2 3 ... 15 go

Televisions

SN	Product	Stock
1.	Samsung TV	7
2.	LG TV x	8
3.	TCL TV	7
4.	Sony TV x	10
5.	Sony TV y	5
6.	LG TV y	6
7.	TCL TV y	5
8.	TCL TV z	10
9.	Samsung TV	5
10.	TCL TV h	5

1 2 3 ... 5 go

Add products

*Product Name:

*Selling Price:

*Cost Price:

*Store to which warehouse:

*Total Quantity added:

[Additional information](#) 

Submit

Products Information

Product Name: Samsung 85" Class TU8000 Series
Crystal UHD 4K Smart TV

Selling Price: \$1,797.99

Cost Price: \$1,500

Total Quantity On Hand: 20

Quantity Sold: 7

Total Sales: \$12,585.93

Total Cost: \$35,959.8

Total Profit: \$2085.93

Total Profit Percent: 16.6%

<p><u>password</u></p> <p>Old password <input type="text"/></p> <p>New password <input type="text"/></p> <p><input type="button" value="Submit Change"/></p>	<p><u>Check Stock</u></p> <table><thead><tr><th>SN</th><th>Product</th><th>#</th></tr></thead><tbody><tr><td>1.</td><td>Apple Watch</td><td>2</td></tr><tr><td>2.</td><td>GPS X</td><td>4</td></tr></tbody></table> <p><input type="button" value="Replenish"/></p>	SN	Product	#	1.	Apple Watch	2	2.	GPS X	4	<p><u>Add Salesperson</u></p> <p>*Name: <input type="text"/></p> <p>*Gender: <input type="text"/></p> <p>*DOB: <input type="text"/></p> <p>Start Date: <input type="text"/></p> <p>Schedule: <input type="text"/></p> <p>Commision: <input type="text"/></p> <p><input type="button" value="Confirm"/></p>
SN	Product	#									
1.	Apple Watch	2									
2.	GPS X	4									
<p><input type="button" value="Add"/></p> <p><u>Salesperson</u></p> <table><tbody><tr><td>1.</td><td>Dave</td><td><input checked="" type="checkbox"/></td></tr><tr><td>2.</td><td>Jack</td><td><input type="checkbox"/></td></tr><tr><td>3.</td><td>Alice</td><td><input type="checkbox"/></td></tr></tbody></table> <p><input type="button" value="Check"/></p> <p><input type="button" value="Pay Commission"/></p>	1.	Dave	<input checked="" type="checkbox"/>	2.	Jack	<input type="checkbox"/>	3.	Alice	<input type="checkbox"/>	<p><u>Pay Commision</u></p> <p>• You have selected (Dave)</p> <p><input type="text" value="Enter amount"/></p> <p><input type="button" value="Confirm"/></p>	<p><u>Staff Information</u></p> <p>Name: Dave Gender: Male DOB: 07/01/1992 Commision rate: 2% Sales made: 16 Start Date: 02/25/2020 Schedule: M/W/F 9:00AM - 3:00PM</p> <p><input type="button" value="Make Changes"/></p>
1.	Dave	<input checked="" type="checkbox"/>									
2.	Jack	<input type="checkbox"/>									
3.	Alice	<input type="checkbox"/>									

<p>Invoice</p> <p>Show Open Invoice</p> <p>Show Closed Invoice</p> <p>Enter</p>	<p>Closed Invoice</p> <ol style="list-style-type: none"> 1. Mark 09/20/2019 2. Josh 08/20/2020 3. Abby 09/30/2020 4. Abby 09/31/2020 	<p>Open Invoice</p> <ol style="list-style-type: none"> 1. Bill 1400/1500 2. Ellie 200/700
<p>Customer</p> <ol style="list-style-type: none"> 1. Mark <input type="checkbox"/> 2. Josh <input type="checkbox"/> 3. Abby <input checked="" type="checkbox"/> 4. Bill <input type="checkbox"/> 5. Ellie <input type="checkbox"/> <p>Check History</p>	<p>Mark Inactive</p> <p>Customer History</p> <p>Name: Abby Gender: Male DOB: 07/01/1992 Address: 1250 N Bellflower Blvd City: Long Beach State: CA Purchase Total: 700 Unpaid Total: 0</p> <p>Go Back</p>	<p>Item</p> <p>Name: Samsung TV x Serial Number: 002 Date of Production: 07/01/2020 Status: Sold Shipped: Yes <input checked="" type="checkbox"/> No <input type="checkbox"/></p> <p>Make Changes</p>

3.2 Hardware Interfaces

This mobile application should be able to run for any OS that can run Java. They should have enough storage to run the application and to handle increasing data as the user adds information.

3.3 Software Interfaces

All parts of the system should be connected to each other since they are dependent on each other. For example, if a customer buys a product, it will affect the warehouse and the quantity of that product should be updated when the user displays all products. Data that the user modifies will modify other components of this software. Interfaces are saved connected to a database so that users can easily manage all data.

3.4 Communications Interfaces

It is very important that all parts of the system communicate with each other since they depend on each other. The system is going to be secured through the login system and is going to use HTTP protocol for communication with the internet. Also, this software will use a shipping system to get

customer's addresses. Some type of synchronization mechanism is needed to share memories between processes and keep the information safe. For example, using a file locking mechanism will keep all information safe by restricting access to the system.

4. System Features

Ability to create and change password

Products

- Add new product
- Show by quantity

Invoices

- Display open invoices
- Display closed invoices
- Open a new invoice
- Set delivery fee
- Mark product shipped

Salespersons

- Add salesperson
- Show performance
- Set commission rate
- Pay commission

Warehouses

- Replenish stock
- Add warehouse
- Show products with quantity ≤ 5
- Show all products and their information (such as quantity sold or selling price)

Mark a customer inactive

Check payment history

4.1 Use Case 1

Use Case 1	Login to System
Actors	User
Pre-Conditions	No user is logged in
Flow on Control	<ul style="list-style-type: none"> ● Fill out password field ● Submit information <ul style="list-style-type: none"> ○ Validate input ○ Display user options

Post-Conditions	The system now displays several options the user can interact with such as displaying invoices, adding new inventory, checking current inventory, etc.
Error-Condition	Input is invalid <ul style="list-style-type: none"> • The password is incorrect
Non-Functional Requirements	System should process the instruction within 5 seconds.

4.2 Use Case 2

Use Case 2	Change Password
Actors	User
Pre-Conditions	No user is logged in
Flow on Control	<ul style="list-style-type: none"> • User selects the “change password” option <ul style="list-style-type: none"> ○ User enters a new password ○ User re-enters their password for confirmation • Submit information <ul style="list-style-type: none"> ○ Validate input
Post-Conditions	The user now has a new password that will be used to login
Error-Condition	Input is invalid The user’s passwords do not match The user left the field empty
Non-Functional Requirements	System should process the instruction within 5 seconds.

4.3 Use Case 3

Use Case 3	Display All Items
Actors	User
Pre-Conditions	Password is entered correctly There is at least one item in a warehouse

Flow on Control	<ul style="list-style-type: none"> User selects the “Display all items” option <ul style="list-style-type: none"> All items are displayed in a list
Post-Conditions	Shows the user all products as well as their selling price, cost price, total quantity on hand, quantity sold, total sales, total cost, total profit and total profit percent in decreasing order of profit percent
Error-Condition	There are no items in any warehouse
Non-Functional Requirements	System should process the instruction within 5 seconds.

4.4 Use Case 4

Use Case 4	Show Products with ≤ 5 Quantity
Actors	User
Pre-Conditions	Password is entered correctly There is at least one item with ≤ 5 quantity in a warehouse
Flow on Control	<ul style="list-style-type: none"> User selects the “Show products with ≤ 5 Quantity” option <ul style="list-style-type: none"> All items are displayed in a list
Post-Conditions	Shows items that have 5 or fewer in the warehouse(s) sorted in increasing order based on quantity
Error-Condition	There are no items with ≤ 5 quantity in any warehouse
Non-Functional Requirements	System should process the instruction within 5 seconds.

4.5 Use Case 5

Use Case 5	Show Quantity for All Products
Actors	User
Pre-Conditions	Password is entered correctly There is at least one item in a warehouse

Flow on Control	User selects the “Show quantity for all products” option. <ul style="list-style-type: none"> • All items are displayed in a list
Post-Conditions	Shows the user the quantity of each item by warehouse
Error-Condition	There are no items in any warehouse
Non-Functional Requirements	System should process the instruction within 5 seconds.

4.6 Use Case 6

Use Case 6	Show Open Invoices
Actors	User
Pre-Conditions	Password is entered correctly At least one customer has opened an invoice
Flow on Control	<ul style="list-style-type: none"> • User selects the “Show open invoices” option <ul style="list-style-type: none"> ◦ All items are displayed in a list
Post-Conditions	Displays open invoices in increasing order based on date
Error-Condition	There are no open invoices
Non-Functional Requirements	System should process the instruction within 5 seconds.

4.7 Use Case 7

Use Case 7	Show Closed Invoices
Actors	User
Pre-Conditions	Password is entered correctly At least one customer has closed an invoice
Flow on Control	<ul style="list-style-type: none"> • User selects the “Show closed invoices” option <ul style="list-style-type: none"> ◦ All items are displayed in a list
Post-Conditions	Displays closed invoices in decreasing order based on amount paid
Error-Condition	There are no closed invoices

Non-Functional Requirements	System should process the instruction within 5 seconds.
-----------------------------	---

4.8 Use Case 8

Use Case 8	Mark Customer as Inactive
Actors	User
Pre-Conditions	Password is entered correctly There is at least one open/closed invoice
Flow on Control	User selects the “Mark user inactive” option <ul style="list-style-type: none"> • User selects a specific customer from a list • User submits
Post-Conditions	The customer is marked as inactive
Error-Condition	There are no invoices
Non-Functional Requirements	System should process the instruction within 5 seconds.

4.9 Use Case 9

Use Case 9	Set Delivery Fee
Actors	User
Pre-Conditions	Password is entered correctly Customer has confirmed the order Customer has set the delivery method to shipping
Flow on Control	<ul style="list-style-type: none"> • User selects the order <ul style="list-style-type: none"> ◦ Displays the shipping information • User enters delivery fee and submits <ul style="list-style-type: none"> ◦ System asks for confirmation • User confirms the delivery fee
Post-Conditions	Delivery fee for the order updated.
Error-Condition	No active invoices exist

Non-Functional Requirements	System should process the instruction within 5 seconds.
-----------------------------	---

4.10 Use Case 10

Use Case 10	Check Payment History
Actors	User
Pre-Conditions	Password is entered correctly New customer's information is present
Flow on Control	<ul style="list-style-type: none"> ● User selects the "Payment History" option <ul style="list-style-type: none"> ○ System displays all of the payment history ● The user selects the specific invoice <ul style="list-style-type: none"> ○ The system displays all information about that invoice.
Post-Conditions	Payment history are displayed for that order
Error-Condition	The store just opened, and there are no active customer and invoice
Non-Functional Requirements	System should process the instruction within 5 seconds.

4.11 Use Case 11

Use Case 11	Mark Product Shipped
Actors	User
Pre-Conditions	Password is entered correctly Customer has confirmed the order Customer has set the delivery method to shipping
Flow on Control	<ul style="list-style-type: none"> ● User selects the box ● User submits
Post-Conditions	The box gets filled Quantities in warehouses deducted by one
Error-Condition	Product is out of stock

Non-Functional Requirements	System should process the instruction within 5 seconds.
-----------------------------	---

4.12 Use Case 12

Use Case 12	Pay Commission
Actors	User
Pre-Conditions	Password is entered correctly The salesperson has sold at least one product to customers
Flow on Control	<ul style="list-style-type: none"> ● User selects the “Pay Commission” option <ul style="list-style-type: none"> ○ User searches for a salesperson in the search bar ○ The system prompts the user to select a salesperson ● User check on what the salespersons sold previously <ul style="list-style-type: none"> ○ The system retrieves data and calculates their commission depend on their commission percentage ● User confirms
Post-Conditions	Commission is paid to the salesperson
Error-Condition	The store just opened and there are no active salespeople working.
Non-Functional Requirements	System should process the instruction within 5 seconds.

4.13 Use Case 13

Use Case 13	Show Salesperson Performance
Actors	User
Pre-Conditions	Password is entered correctly There are at least one salesperson working in the store
Flow on Control	<ul style="list-style-type: none"> ● User selects the “Salesperson Performance” option <ul style="list-style-type: none"> ○ System shows total sales made by all salespersons ○ System shows commissions earned by each salesperson
Post-Conditions	All salesperson performances are shown

Error-Condition	None
Non-Functional Requirements	System should process the instruction within 5 seconds.

4.14 Use Case 14

Use Case 14	Set Commission Rates
Actors	User
Pre-Conditions	Password is entered correctly There are at least one salesperson working in the store
Flow on Control	<ul style="list-style-type: none"> ● User selects the “Salesperson Information” button <ul style="list-style-type: none"> ○ System listed all salesperson ● User chooses an individual salesperson <ul style="list-style-type: none"> ○ System shows information about the salesperson ● User set commission rates for the salesperson <ul style="list-style-type: none"> ○ System asks for confirmation ● User confirms new changes
Post-Conditions	New commission rates have been applied to the salesperson
Error-Condition	The store just opened and there are no active salespeople working
Non-Functional Requirements	System should process the instruction within 5 seconds.

4.15 Use Case 15

Use Case 15	Open New Invoice
Actors	User
Pre-Conditions	Password is entered correctly The customer is new or has paid off their previous invoices
Flow on Control	<ul style="list-style-type: none"> ● User selects the “Open New Invoice” option <ul style="list-style-type: none"> ○ System asks for user to enter order information ● User fills out the necessary information and submits <ul style="list-style-type: none"> ○ System asks for confirmation ● User confirms new invoice

Post-Conditions	New invoice has been created for the customer.
Error-Condition	Customer has an unpaid invoice.
Non-Functional Requirements	System should process the instruction within 5 seconds.

4.16 Use Case 16

Use Case 16	Add Salesperson
Actors	User
Pre-Conditions	Password is entered correctly
Flow on Control	<ul style="list-style-type: none"> ● User selects the “Add Salesperson” option <ul style="list-style-type: none"> ○ System asks user to fill out information on the new salesperson ● User enters information and submits <ul style="list-style-type: none"> ○ System asks for confirmation ● User confirms
Post-Conditions	New salesperson added to salesperson list
Error-Condition	None
Non-Functional Requirements	System should process the instruction within 5 seconds.

4.17 Use Case 17

Use Case 17	Add New Product
Actors	User
Pre-Conditions	Password is entered correctly
Flow on Control	<ul style="list-style-type: none"> ● User selects the “Add New Product” option <ul style="list-style-type: none"> ○ System asks for product information as well as which warehouse to store it in ● User fills out the information needed and submits <ul style="list-style-type: none"> ○ System asks for confirmation ● User confirms

Post-Conditions	New product is added to the system.
Error-Condition	There is already an existing product which is the same as the product the user wants to add (same product name). All the warehouses are currently full on stock
Non-Functional Requirements	System should process the instruction within 5 seconds.

4.18 Use Case 18

Use Case 18	Add New Warehouse
Actors	User
Pre-Conditions	Password is entered correctly
Flow on Control	<ul style="list-style-type: none"> ● User selects the “Add New Warehouse” option <ul style="list-style-type: none"> ○ System asks for the new warehouse’s basic information and scale ● User enters information and submits <ul style="list-style-type: none"> ○ System asks for confirmation ● User confirms
Post-Conditions	A new and empty warehouse is created in the database
Error-Condition	None
Non-Functional Requirements	System should process the instruction within 5 seconds.

4.19 Use Case 19

Use Case 19	Replenish Stock
Actors	User
Pre-Conditions	Password is entered correctly
Flow on Control	<ul style="list-style-type: none"> ● User selects the “Show products with ≤ 5 Quantity” option <ul style="list-style-type: none"> ○ System displays all the products which need to be replenished in increasing order ● User clicks on a certain item that needs to be replenished

	<ul style="list-style-type: none">• User selects the warehouse<ul style="list-style-type: none">◦ System asks for the amount to replenish• User enters the replenishment amount
Post-Conditions	Quantity on products updated Selected products are replenished
Error-Condition	The item the user is trying to stock does not exist
Non-Functional Requirements	System should process the instruction within 5 seconds.

5. Nonfunctional Requirements

All of the instructions and database updates should process within 5 seconds.

6. Other Requirements

All legal copyrights are reserved by //TODO: Team Name and may not be redistributed or resold without obtaining license from our team.