SUPERMARKET AUTOMATION SOFTWARE

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

Vatsalya Chauhan
12CS10053

Kulkarni Bhushan
12CS30016

CONTENTS

1.0. INTRODUCTION

The Software Requirements Specification is designed to document and describe the agreement between the customer and the developer regarding the specification of the software product requested. Its primary purpose is to provide a clear and descriptive “statement of user requirements” that can be used as a reference in further development of the software system. This document is broken into a number of sections used to logically separate the software requirements into easily referenced parts.

This Software Requirements Specification aims to describe the Functionality, External Interfaces, Attributes and Design Constraints imposed on Implementation of the software system described throughout the rest of the document. Throughout the description of the software system, the language and terminology used is unambiguous and consistent throughout the document.

1.1. Purpose

The software system being produced is called Supermarket Automation Software or SAS. It is being produced for supermarkets interested in automating their checkout counters. This system is designed to “provide automation support” for the process of selling items and updating inventory details.

1.2. Scope of Project

This software system will be a Supermarket Automation Software System. This system will be designed to maximize the ease of keeping records of register Sales, Stock status etc. The automation property of the system will make the working system very simple, fast and free of Human-errors. By maximizing the user’s work efficiency the system will meet the user’s needs while remaining easy to understand and use.

1.3. Description

This section includes details about what is and is not expected of the SAS system in addition to which cases are intentionally unsupported and assumptions that will be used in the creation of the SAS system.

The Supermarket Automation Software will allow a manager to automate its sales and inventory details. The manager will have the option to view inventory and sales details, update details about a particular item and create new account for new employees. The SAS also allows a manager to manage the inventory with full create, retrieve and update functionality with regards to items in the system.

Definitions	Meaning
SAS	Supermarket Automation Software
Barcode	A unique identifier assigned to single items
Button	A user interface element that allows a user to click and inform the system to take an action
Checkout	The process a Customer goes through to purchase an Item
Customer	A customer is a person who purchases items from the supermarket
Database	Refers to the collection of items inside the supermarket
Item	An individual entity in the inventory which has several descriptive attributes
Manager	A single person that has the ability to create, retrieve, update and delete items in the store. This person cannot simultaneously act as a Sales Clerk and Manager
Sales Clerk	The person who operate the software product.
SRS	Software Requirements Specification
User	Referring to the manager or sales clerk

1.4. Environmental Characteristics

1.4.1. Hardware

The software requires a pc either running on windows or Linux and should have java IDE installed. There are no hardware or software requirements beyond these including, but not limited to, memory or specific software packages that need to be utilized nor software packages that need not be utilized.

1.4.2. People

This software can be used by any user having basic skills in operating a computer since the user interface is very simple.

2.0. Overall Description

2.1. Functional Requirements

Functional Requirements are those that refer to the functionality of the system, i.e. what services it will provide to the user. Nonfunctional requirements pertain to the other information needed to produce the correct system and are detailed separately.

2.2. Use Cases

In software and systems engineering, a **use case** is a list of steps, typically defining interactions between a role (known in Unified Modeling Language (UML) as an "actor") and a system, to achieve a goal. The actor can be a human or an external system.

In systems engineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals.

2.2.1. Use Case: Login

Actors: Manager, Sales Clerk
Type: Primary and essential
Description: Initiated when a user tries to access his account. The user is then Prompted to enter in their username and password in order to Proceed.
Includes: None
Use-Cases: None

2.2.2. Use Case: Print Receipt

Actors: Printer
Type: Primary and essential
Description: Allows the sales clerk to print the bill
Includes: Get Item details
Use-Cases: The Login use case must be completed.

2.2.2.1. Use Case: Get Item Details

Actors: Sales Clerk, Manager
Type: Primary and essential
Description: Allows the sale clerk to get the details of the items bought by the customer.
Includes: Read Barcode, Weight Item, Modify Inventory
Use-Cases: The Login use case must be completed.

2.2.2.2. Use Case: Read Barcode

Actors: Barcode Reader
Type: Primary and essential
Description: Allows the sale clerk to get the details about type of item bought by the customer
Includes: None
Use-Cases: The Login use case must be completed.

2.2.2.3. Use Case: Weight Item

Actors: Weighing Machine
Type: Primary and essential
Description: Allows the sale clerk to get the details about quantity of item bought by the customer
Includes: None
Use-Cases: The Login use case must be completed.

2.2.2.4. Use Case: Modify Inventory

Actors: none
Type: Primary and essential
Description: decrements sold items' quantities from Inventory.
Includes: None
Use-Cases: The Login use case must be completed.

2.2.3. Use Case: Check Inventory

Actors: Manager
Type: Primary and essential
Description: Allows the manager to view the items present in our database.
Includes: None
Use-Cases: The Login use case must be completed.

2.2.4. Use Case: Change price

Actors: Manager
Type: Primary
Description: Allows the manager to change the price of a particular item
Includes: None
Use-Cases: The Login use case must be completed.

2.2.5. Use Case: Show Sales Statistics

Actors: Manager
Type: Primary and essential
Description: Allows the manager to view the sales statistics
Includes: None
Use-Cases: The Login use case must be completed.

2.2.6. Use Case: Update Inventory

Actors: Manager and Sales clerk
Type: Primary and essential
Description: to change Details of newly purchased items in Inventory.
Includes: None
Use-Cases: The Login use case must be completed.

2.2.7. Use Case: Add new Item

Actors: Manager and Sales clerk
Type: Primary and essential
Description: Allows an employee to add new items to the database
Includes: None
Use-Cases: The Login use case must be completed.

2.2.8. Use Case: Add new Employee

Actors: Manager
Type: Primary and essential
Description: Allows the manager to hire new employee in the supermarket.
Includes: None
Use-Cases: The Login use case must be completed.

2.2.9. Use Case: Logout

Actors: Manager, Customer
Type: Primary and essential
Description: The customer should have the option to logout from his account when he is not going to be active for some time.

Includes: None
Use-Cases: The Login use case must be completed.

2.3. Non-Functional Requirements

There are requirements that are not functional in nature. Specifically, these are the constraints the system must work within.

3.0. Requirements Specification

3.1. External Interface Requirements

3.1.1. User Interfaces

Only mouse and keyboard are required for interacting with system.

3.1.2. Hardware Interfaces

Hardware:	Personal Computer
Operation System:	WindowsXP or more , linux
Internet Connection:	Either LAN connection or Wi-Fi connection

3.1.3. Software Interfaces

The software will be coded in JAVA IDE using NETBEANS 7.0.no other software interface required.

3.2. Functional Requirements

The set of functionalities that are supported by the system are documented below-

3.2.1. Register sales

Description:

Whenever any item is sold from the stock of the supermarket, this function will prompt the clerk to pass the item over a bar code reader and an automatic weighing scale, the data regarding the item type and the quantity get automatically registered then. After the end of a sales transaction it will print the bill containing the serial number of the sales transaction, the name of the item, code number, quantity, unit price, and item price. The bill should indicate the total amount payable.

3.2.1.1. Read Bar Code

Input: sold-items are passed over the reader.

Processing: barcode of the item is read and registered automatically

3.2.1.2. Weigh

Input: sold-items are passed over the automatic weighting scale

Processing: weight of the sold-items is automatically get registered

3.2.1.3. Register sold items

Input: automatically registered data about the item along with its quantity
Processing: the sold-item gets registered

3.2.1.4. Generate bill

Input: automatically generated “generate bill command”
Output: the transaction bill containing the serial number of the sales transaction, the name of the item, code number, quantity, unit price, and item price is printed. The bill also mentions the total amount payable.

3.2.2. Modify Inventory

Description:

In order to support inventory management, this function decreases the inventory whenever an item is sold. Again, when there is a new supply arrival, an employee can update the inventory level by this function.
Input: new supply (when arrives) or registered sold-items
Processing: whenever new supply arrives or items are sold this updates the inventory.

3.2.3. Check Inventory

Description:

The manager upon invoking this function can issue query to see the inventory details. In response, it shows the inventory details.
Input: query from the manager
Output: display the inventory

3.2.4. Print sales-statistics

Description:

Upon invoking this function, it will generate a printed out sales statistics for every item the supermarket deals with for any particular day or any particular period.
Input: generate sales-statistics command
Output: printed out form of the sales statistics for every item sold for any particular period.

3.2.5. Update price

Description:

The manager can change the price of an item by invoking this function.
Input: change price command along with the new assigned price.
Processing: updates the price of the corresponding item in the inventory.

3.2.6. Add New Employee

Description:

Manager will be able to add new employee profile
Input: Employee details such as name, address, contact number etc.

3.3. Detailed Non-functional Requirements

The set of non-functional requirements can be stated as follows:

3.3.1. Bill Format

1.) The bill should contain the serial number of the sales transaction, the name of the item, code number, quantity, unit price, and item price.

2.) The bill should indicate the total amount payable.

3.3.2. Sales-statistics Report Format

The sales statistics report should indicate the quantity of an item sold, the price realised, and the profit.

3.3.3. Data must be saved properly

3.3.4. Correct data must be given

3.3.5. The software should be protected from customers and non-employees of the supermarket.

3.3.6. The latest version of java IDE is installed in the computer in which it is going to be run.

4.0. Constraints

Security is not a concern for this system. The database may store passwords in “serializable” file and there doesn't need to be a password recovery feature nor lockout after numerous invalid login attempts. As such, the system may not work correctly in cases when security is a concern. We are not forcing users to have “strong password”. A strong password is a password that meets a number of conditions that are set in place so that user's passwords cannot be easily guessed by an attacker. Generally, these rules include ensuring that the password contains a sufficient number of characters and contains not only lowercase letters but also capitals, numbers, and in some cases, symbols.

5.0. Assumptions

5.1 The customer cannot buy more items than is available in the supermarket.

5.2 The manager cannot be sales clerk.

5.3 The employees cannot change their account details.

5.4 The sales clerk cannot view the items stored in our database.

6.0. Limitations of Project

As stated, security is not a concern of this project. As such, it is beyond the scope of this system to encrypt personal user data and information, prevent unauthorized login attempts, or any other concern of this nature. Additionally, the system is not responsible for the following:

(1) Verifying if the information given is correct.

(2) Storing data about the customers.

7.0. Data Structure

7.1. Item has these attributes

7.1.1. Serial no.
7.1.2. item name
7.1.3. quantity available
7.1.4. ID
7.1.5. unit price
7.1.6. profit per unit

7.2. Employee has these attributes

7.2.1. Name
7.2.2. Address
7.2.3. Contact Number
7.2.4. username
7.2.5. Password
7.2.6. Type – Manager or Sales Clerk

7.3. Transaction has these attributes

7.3.1. Serial no
7.3.2. date
7.3.3. Items
7.3.4. Quantities
7.3.5. Unit price
7.3.6. profits

Thank you…