



SUPERMARKET AUTOMATION SOFTWARE

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

Vatsalya Chauhan
12CS10053

Kulkarni Bhushan
12CS30016

CONTENTS

1.0. Introduction	3
1.1. Purpose	3
1.2. Scope of Project	3
1.3. Descriptions	3
1.4. Environment Characteristics	4
1.4.1. Hardware	4
1.4.2. People	4
2.0. Feasibility Study	4
2.1. Problem and its Scope	4
2.2. Stakeholders	5
2.3. Alternatives	5
2.3.1. Technology used	5
2.3.2. Hardware infrastructure	5
2.3.3. Security	5
2.4. Criteria to evaluate	5
2.5. Assessment of Unusual circumstances	6
2.6. Evaluation of alternatives	6
2.6.1 Technology used	
2.6.2. Hardware infrastructure	
2.6.3. Security	
2.7. Report	6
3.0. Requirements Specification	7
3.1. External Interface Requirements.....	7
3.1.1. User Interfaces	7
3.1.2. Hardware Interfaces	7
3.1.3. Software Interfaces	7
3.2. Functional Requirements.....	7
3.2.1. Register sales.....	7
3.2.1.1. Read Bar Code.....	7
3.2.1.2. Weigh.....	7
3.2.1.3. Register sold items	8
3.2.1.4. Generate bill	8
3.2.2. Modify Inventory	8
3.2.3. Check Inventory	8
3.2.4. Print sales-statistics	8
3.2.5. Update price	8
3.2.6. Add New Employee	8
3.3. Data Flow:	9

3.4. Data Dictionary	12
3.5. Detailed Non-functional Requirements	12
3.6. Report	13
4.0. Detailed Design	13



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. Feasibility Study

2.1. Problem and its Scope

This software addresses the needs of a supermarket management. It helps in the management of transactions, employees, and available Stock. It can be used for the management of any store small or large.

2.2. Stakeholders

The stakeholders are:

- A) Supermarket manager: User class with highest rights.
 - 1) Can add items to store.
 - 2) Can add / remove new employees
 - 3) Change the price of Items
 - 4) Check inventory details
 - 5) Can view the statistics of transaction
- B) Sales clerks working at Supermarket
 - 1) Can add supply to inventory
 - 2) Get items from customers to print the receipt

2.3. Alternatives

This software can be web based where customers will be able to have a look at the availability and price of various items available in the supermarket and support online shopping.

Store management can also be done manually.

2.3.1. Technology used

In this software, we are using JAVA IDE for coding and MySQL to store the Databases. We can also use C / C++ coding language and other database systems as SQL server, Oracle, Sybase, Access, etc.

2.3.2. Hardware infrastructure

This software is designed to run on Linux and Windows operating systems. It can also be designed to run compatibly with MAC OS systems.

We are currently using internal memory of system. We can also use external hard-disk of any size for its implementation.

Currently SAS is designed to run on 64 bit machine, it can be alternatively designed to run on 32 bit machines.

2.3.3. Security

The SAS can be designed for various levels of security for different supervisors.

Instead of using password protection for login into the system, we can also incorporate Face recognition, a Bar Code scanning system, or a thumb impression recognition system for login of the important stakeholders.

2.4. Criteria to evaluate

The primary criteria which are to be kept in mind while evaluating the alternatives are:

- 1) Cost of technology
- 2) Cost of infrastructure
- 3) Lifetime of technology
- 4) Stability of technology

2.5. Assessment of Unusual circumstances

The software should take care of the fact that the data is not lost in any case.

The design should take care of the fact that the data is not lost in any case, be it software or a hardware failure, system going down or any unusual circumstances that might intervene in between the smooth functioning of the RRTS. For this, we can design a MASTER system which stores the backup of all the data which is fed into the SAS time to time. This master system would allow us to retrieve data at any point of time and restore the database to its original state.

2.6. Evaluation of alternatives

2.6.1 Technology used

Since we do not deal with algorithm intensive techniques in SAS, we need not go for programming languages like C++ and Java would suffice.

2.6.2. Hardware infrastructure

Using an external hard disk to save the database will indirectly imply a backup of the database along with the master system which can be retrieve the data at any point of time desired. Thus, this is better than using the system hard disk to store the data. This also ensures that the software does not any space on the system hard disk. The SAS can be designed for a 32-bit system as well as a 64-bit system. Designing the SAS for a 64-bit system should be preferred for the graphical interface concerns.

2.6.3. Security

If the SAS uses security systems like thumb impression recognition, the cost of technology increases but the security increases. As the cost of production is a major concern for the SAS design, the security alternatives should be given the least priority. Also, the RRTS is less prone to hacking and security threats.

2.7. Report

In the feasibility study, we went through the complete details of the problem. The objectives of SAS have been laid out and the various scopes have been discussed in detail. Firstly we understood the complete problem and found various functions that the SAS performs. We discussed rights of Manager of supermarket and Job of Sales clerk.

Various alternatives were developed keeping in mind the cost and the lifetime of the components the alternative brings with it and hence the advantages and disadvantages were highlighted. These alternatives included the hardware, software, security and technical aspects which form integral part of the software and which could be incorporated in the SAS if desired. The primary criteria for evaluation were expected lifetime, cost, stability, and instability of the technology. The unusual circumstances like loss of data due to hardware or software failure or hacking were taken care of by certain concepts of data backup, cryptography etc.

At last, all the alternatives proposed earlier were analyzed in depth and their advantages and boon to the SAS were clearly mentioned. A very vivid comparison was made between the SAS development without the alternatives and with the alternatives if incorporated in the software.

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

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.

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 weighing 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

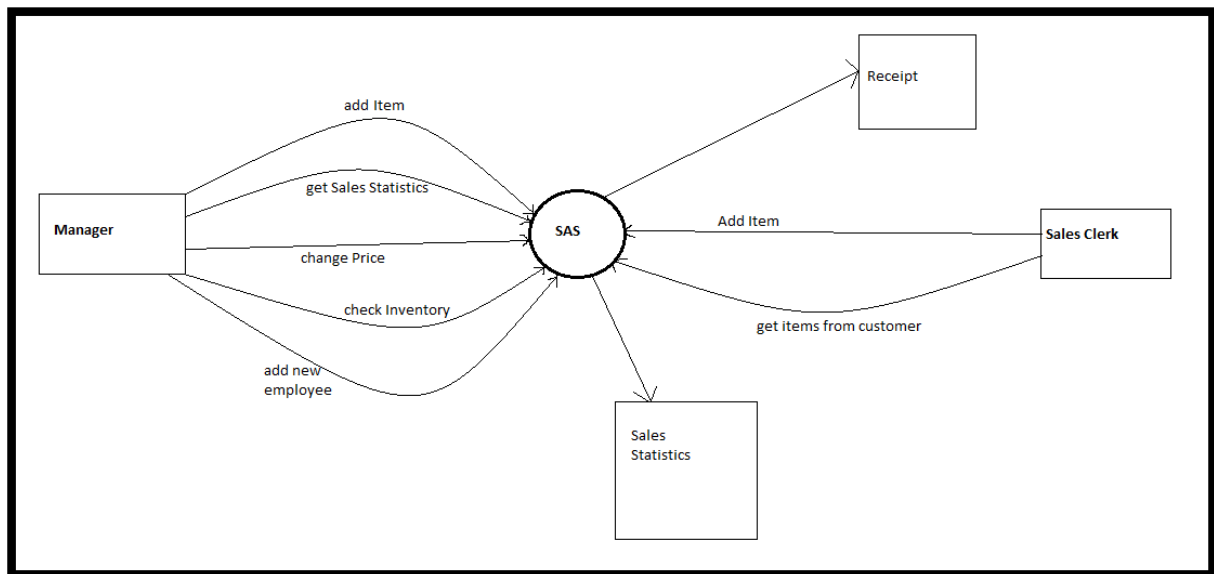
Description:

Manager will be able to add new employee profile

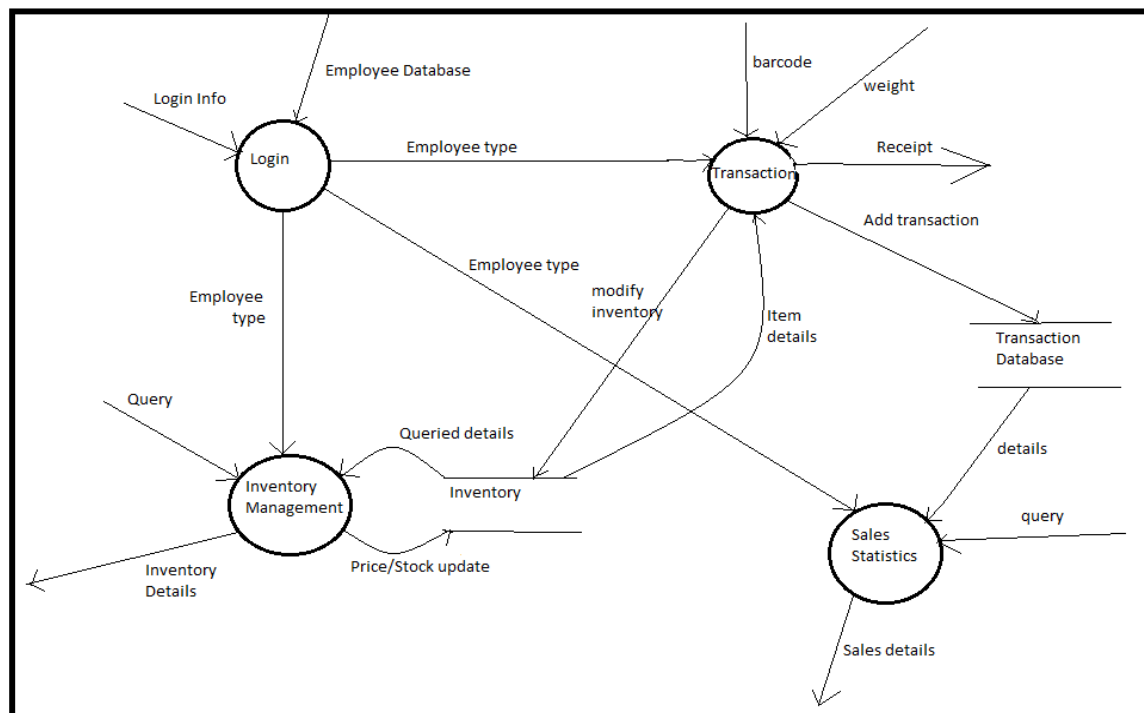
Input: Employee details such as name, address, contact number etc.

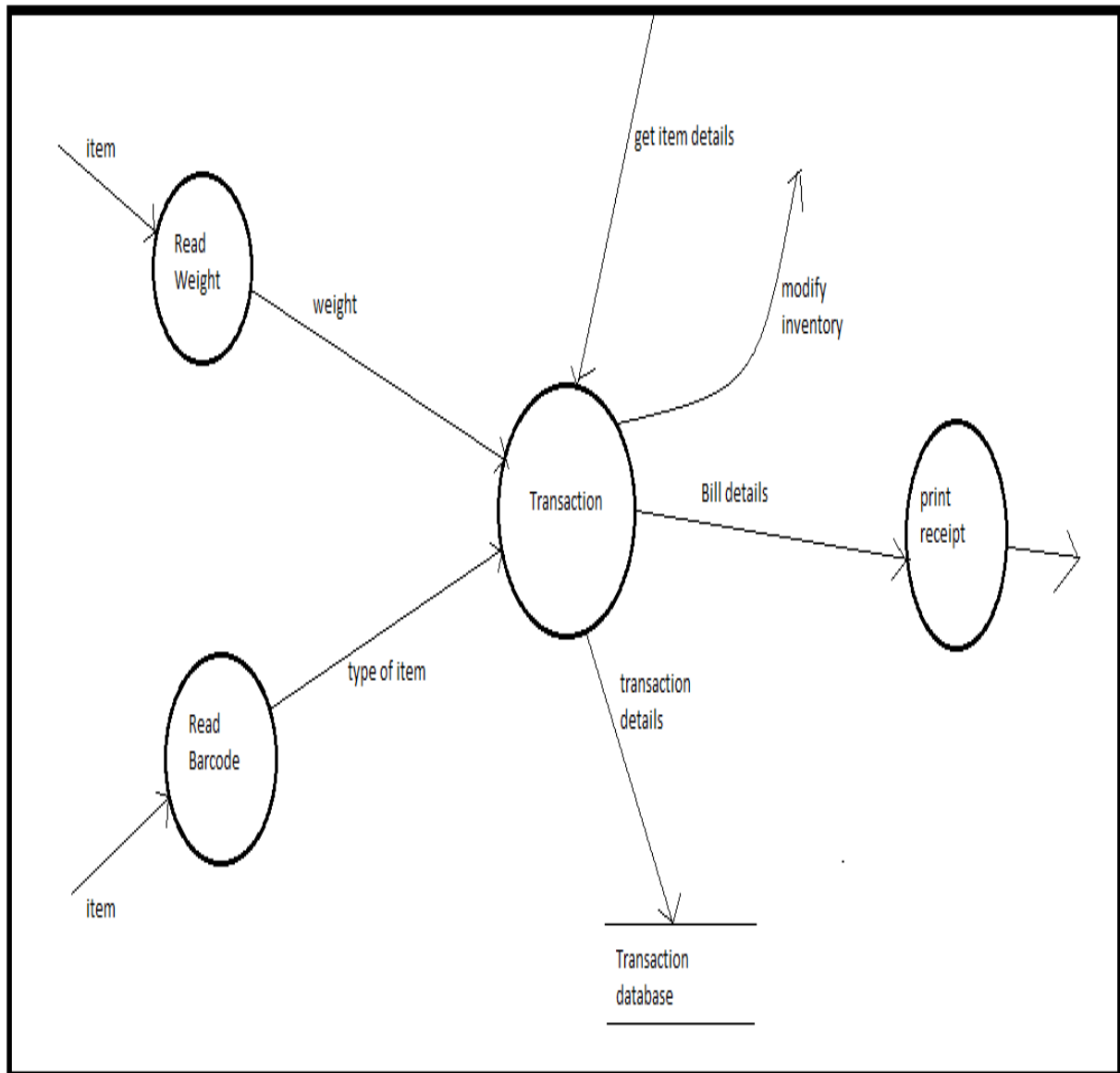
3.3. Data Flow:

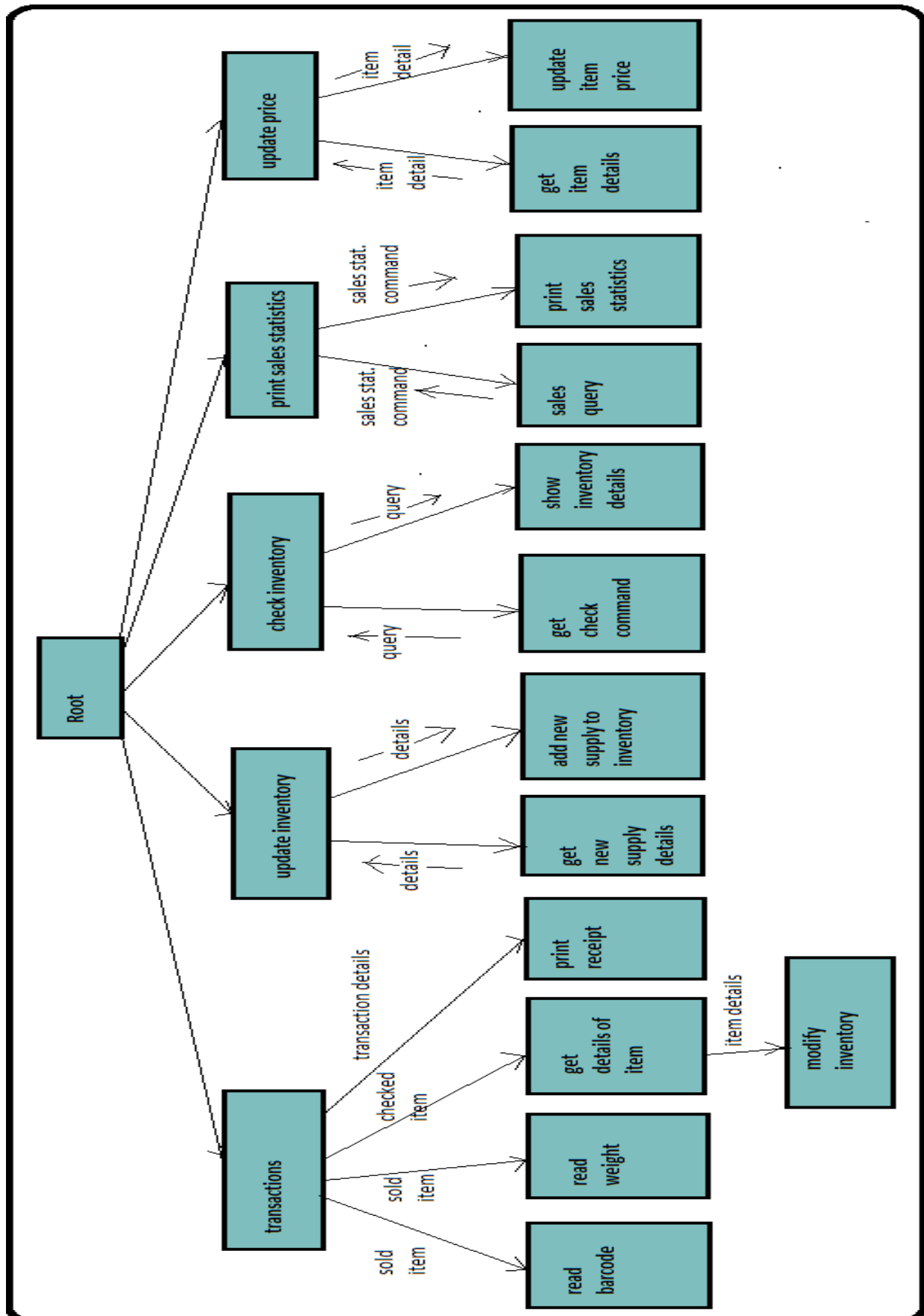
3.3.1 Context Diagram of SAS :



3.3.2. Data Flow Diagram Level-1 of SAS:



3.3.3. Data Flow Diagram Level-2 for Transactions of SAS :**3.3.4. Structural Design of SAS :**



3.4 Data Dictionary:

Databases: Inventory containing Items

Transactions containing Transaction

Employees containing Employee

3.4.1. Item has these attributes

1. Serial no. (int)
2. item name (String)
3. quantity available (float)
4. ID (String)
5. unit price (float)
6. profit per unit (float)

3.4.2. Employee has these attributes

1. Name (String)
2. Address (String)
3. Contact Number (long int)
4. username (String)
5. Password (String)
6. Type – Manager or Sales Clerk (bool)

3.4.3. Transaction has these attributes

1. Serial no (int)
2. date (Date)
3. Items (ArrayList<Item>)
4. Quantities (ArrayList<float>)
5. Unit price (ArrayList<float>)
6. profits (ArrayList<float>)

3.5. Detailed Non-functional Requirements

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

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

3.5.1. 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.

3.5.2. Sales-statistics Report Format

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

3.5.3. Data must be saved properly

3.5.4. Correct data must be given

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

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

3.6. Report

In the requirements analysis section, we started with the functional requirements of the SAS and explained them in detail using data flow diagrams, Structure chart and data dictionary. The data flow diagrams graphically represent the “flow” of data through the information system, modeling the process aspects of the SAS. They are a preliminary step to create an overview of the SAS. Thus the DFDs along with the structure chart and the data dictionary have been used for the visualization of data processing in the SAS, i.e. the structural design of the software being developed. The second part of the requirements analysis deals with the non-functional requirements of the SAS. These include the database requirements, the legal requirements, and the availability of the SAS over a day and the physical security of the software being developed. The various non-functional requirements ensure the delivery of an operable and manageable system which provides the required functionality in a reliable fashion, uninterrupted and with minimal time of interruption even under the unusual circumstances.

4.0. Detailed Design

4.1. Global System Architecture

The overall system architecture is a 2-tier architecture which includes client at one end and the database at the other. There is no server based middle tier in the software being designed.

4.2. Platform

Minimum system requirements:

Hardware Requirements:

Operating system	Windows XP/98 or later versions, Linux
Processor	Pentium II processor or equivalent

Software Requirements:

MySQL, NetBeans

4.3. Software Architecture

Object-oriented architecture forms the basis of the SAS.

In this style data representations and their associated primitive operations are encapsulated in an abstract data type or object. The components of this style are the objects—or instances of the abstract data types. Objects interact through function and procedure invocations. Two important aspects of this style are (a) that an object is responsible for preserving the integrity of its representation (usually by maintaining some invariant over it), and (b) that the representation is hidden from other objects. Thus the aspects of OOA mentioned justify our choice.

4.4. Report

Under the detailed design section of the software design, the global system architecture was discussed. The SAS has a 2-tier architecture comprising of the client and the database with no server.

Then the platform requirements for the SAS was discussed in terms of the operating system, etc.

The software architecture of the SAS was later stated to be of the object-oriented type using JAVA as the core technology. The important aspects of OOD used for the SAS are data abstraction and the preservation of integrity of the software

Thank you...