**Supermarket Automation Software (SAS)**

Software Requirements Specification

Version <3.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <04/02/2025> | <1.0> | SRS 1.0 | Group-11 |
| <11/02/2025> | <2.0> | SRS 2.0 | Group-11 |
| <04/03/2025> | <3.0> | SRS 3.0 | Group-11 |

Table of Contents

1. Introduction
   1. Purpose
   2. Scope
   3. Definitions, Acronyms and Abbreviations
   4. References
   5. Overview
2. Overall Description
   1. Product Perspective
   2. Product Functions
   3. User characteristics
   4. Constraints
   5. Assumptions and Dependencies
3. Specific Requirements
   1. Functional Requirements
   2. Non-Functional Requirements
4. Data Dictionary
   1. Login Table
   2. Item Table
   3. Customer Table
   4. Sales Table
   5. Purchase Table
   6. Vendor Table

5. Data Flow Diagram

5.1. Levels of DFD

5.2. Level 0 DFD Diagram

5.3. Level 1 DFD Diagram

5.4. ER Diagram

5.5. Structure chart

5.6. Use case Diagram

5.7. Class Diagram

5.8. Activity Diagram

# Introduction

## *Purpose*

This document defines the functional and non-functional requirements for the Supermarket Automation Software (SAS). This software aims to automate the sales, inventory management, and reporting processes of a supermarket, ensuring efficient operations and accurate record-keeping. The document will serve as a guide for developers, testers, and stakeholders to understand the system's requirements and functionalities.

## *Scope*

The **Supermarket Automation Software (SAS)** will:

* Automate sales transactions, including item scanning, bill generation, and inventory updates.
* Maintain an up-to-date inventory of all items in the supermarket.
* Provide sales statistics for any given day or period.
* Allow the manager to update item prices and view inventory details.
* Support multiple user roles: Salesclerk, Employee, and Manager.

The software will be developed as a desktop application using Java and Java Swing for the user interface. It will integrate with a database to store inventory, sales, and user data.

## *Definitions, Acronyms, and Abbreviations*

|  |  |
| --- | --- |
| Configuration | Supermarket Automation Software. |
| FAQ | Frequently Asked Questions |
| SAS | Software automation system |
| UI | User Interface. |
| DB | Database |
| Item | A product sold in the supermarket. |
| Inventory | The stock of items available in the supermarket. |
| Transaction | A sales transaction involving one or more items |
| Sales Statistics | A report showing the quantity sold, price realized, and profit for items over a specific period. |

## *References*

The references are:

* IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications.
* Problem statement provided for the assignment.
* Java and Java Swing documentation.

## *Overview*

This document is organized into sections that describe the functional and non-functional requirements of the **Supermarket Automation Software (SAS)**. The functional requirements outline the system's capabilities, such as sales transaction processing, inventory management, and sales statistics generation. The non-functional requirements cover performance, usability, security, reliability, and scalability aspects of the system.

The subsequent sections of this document will provide detailed descriptions of the system's requirements, ensuring a clear understanding for all stakeholders involved in the development and implementation of the software.

# Overall Description

### **2.1** **Product Perspective**

The **Supermarket Automation Software (SAS)** is a standalone desktop application designed to streamline supermarket operations. It integrates with a barcode scanner, weighing scale, and a database to manage sales, inventory, and reporting. The system interacts with users through a graphical user interface (GUI) developed using Java Swing.

### **2.2** **Product Functions**

The key functions of the system include:

1. **Sales Transaction Management**: Process sales, generate bills, and update inventory.
2. **Inventory Management**: Maintain and update inventory levels.
3. **Sales Statistics**: Generate reports on sales and profits.
4. **Price Management**: Allow the manager to update item prices.
5. **User Authentication**: Secure access based on user roles.

### **2.3** **User Characteristics**

* **Salesclerk**: Responsible for processing sales transactions. Basic computer literacy is required.
* **Employee**: Responsible for updating inventory. Basic computer literacy is required.
* **Manager**: Responsible for managing prices, viewing inventory, and generating reports. Intermediate computer literacy is required.

### **2.4** **Constraints**

1. The system must be compatible with barcode scanners and weighing scales.
2. The system must run on Windows and Linux operating systems.
3. The system must support a database with up to 10,000 items and 5,000 transactions per day.

### **2.5** **Assumptions and Dependencies**

1. The barcode scanner and weighing scale will be connected to the system.
2. The database will always be available and accessible.
3. Users will have basic training to operate the system.

# 

# 3. Specific Requirements

## *3.1 Functional Requirements*

#### **Transaction Processing**

#### **Input**

* Product ID scanned via barcode reader
* Weight captured by the weighing scale
  1. **Processing**
* The system retrieves product information from the database and calculates the total amount due.
* A receipt is generated, detailing the purchased items and the total payable amount.
  1. **Output**
* A printed receipt is issued, showing the complete list of items, their prices, and the total amount to be paid. The salesclerk's ID is also included.

1. **Viewing Sales Data**
2. **Input**

* Product ID and the time for which the sales data should be retrieved.

1. **Processing**

* The system queries the database to gather information on each transaction for the selected period. It computes the profit based on the cost and selling price of the product.

1. **Output**

* A detailed profit report is displayed in the requested format, providing the manager with insights into the product's performance.

#### **3** **Price Updates**

1. **Input**

* Product ID and the updated price for the product.

1. **Processing**

* The system locates the product in the database and shows its current details. It then updates the price to the new value provided.

1. **Output**

* The system displays the updated product details, including the new price.

#### **4** **Inventory Management**

1. **Input**

* Product ID and the newly arrived quantity.

1. **Processing**

* The system checks the inventory database to see if the product already exists. If it does, the system updates the quantity. If it does not, a new entry is added with the product details and quantity.

1. **Output**

* A notification confirming that the product and quantity have been successfully updated in the inventory database.

### **3.2 Non-Functional Requirements**

#### **1** **Performance and Reliability**

* The system must be capable of handling a high volume of transactions and large data sets efficiently. Adequate network bandwidth and robust infrastructure are necessary to ensure smooth performance and response times, even under heavy usage.

#### **2** **Security**

* Users will access the system via secure login credentials, requiring a unique ID and password. Only authorized personnel will have access to sensitive data and system features.

**3** **Reliability**:

* The system shall handle up to 1000 transactions per day without failure.
* The system shall include a daily backup mechanism for all data.

**4** **Scalability**:

* The system shall be able to handle an increase in the number of items (up to 10,000 items).
* The system shall support an increase in the number of transactions (up to 5000 transactions per day).

**5** **Maintainability**:

* The system shall be modular, with clear separation of functionalities to facilitate future updates and maintenance.
* The code shall be well-documented for ease of understanding and modification

**4. DATA DICTIONARY**

* 1. ***LOGIN TABLE***

This is a login table which has two fields username and password. Here primary key consists of both username and password fields as for every user there is unique password to access the system.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SR NO. | FIELD NAME | DATA TYPE | SIZE OF DATA  TYPE | DESCRIPTION | CONSTRAINT |
| 1 | username | Varchar2 | 22 | It contains the  Username | Not null |
| 2 | password | Varchar2 | 22 | It contains the  password | Not null |

* 1. ***ITEM TABLE***

This is the item detail table which stores information corresponding to the items that are available in the supermarket.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SR NO. | FIELD NAME | DATA TYPE | DESCRIPTION | CONSTRAINT |
| 1 | Product\_code | Varchar2(20) | For product no. | Primary key |
| 2 | Product\_name | Varchar2(30) | Product name |  |
| 3 | Item\_cat | Varchar2(20) | Category of item |  |
| 4 | Weight | Number (5) | Product weight |  |
| 5 | Mrp\_price | Number (5) | Marked price |  |
| 6 | Quantity\_in\_stock | Number (5) | Quantity of the product |  |
| 7 | Reorder\_qty | Number (6) | Reorder level of  product |  |

* 1. ***CUSTOMER TABLE***

This is the customer detail table which stores information corresponding to the various customers that purchases some items that are available in the supermarket.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SR.NO | Fieldname | Data type | description | constraints |
| 1 | cust\_id | Number (5) | Customer no. | Primary key |
| 2 | cust\_name | Varchar2(30) | Customer name |  |
| 3 | Address | Varchar2(30) | Customer address |  |
| 4 | cust\_no | Number (10) | Customer Phone no. |  |

* 1. ***SALES TABLE***

This is the sale detail table which stores information corresponding to the sale of various items to the customers.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SR NO. | FIELD NAME | DATA TYPE | DESCRIPTION | CONSTRAINT |
| 1 | Bill\_no | Number (2) | Bill number | Primary key |
| 2 | date | date | Bill Date |  |
| 3 | Product\_code | Varchar2(5) | Product code | Reference key |
| 5 | Sale\_price | Number (20) | Selling price |  |
| 6 | Cust\_no. | Varchar2(5) | Customer name | Reference key |
| 7 | Product\_qty | Number (20) | Quantity of product sold |  |
| 8 | discount | Number (2) | Discount on product |  |
| 9 | total | Number (20,2) | Total value of sale |  |

* 1. ***PURCHASE TABLE***

This is the purchase detail table which stores information corresponding to the purchase of various item with their vendor id and product code.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SR NO. | FIELD NAME | DATA TYPE | DESCRIPTION | CONSTRAINT |
| 1 | order\_no | Number (20,2) | Id number |  |
| 2 | date | date | Order date |  |
| 3 | Product\_code | Varchar2(20) | Description of purchased product | Reference key |
| 4 | Product\_qty | Number  (20,2) | Quantity of product  purchased |  |
| 5 | Product\_price | Number (20,3) | Product purchase price |  |
| 6 | discount | Number (2) | Discount on  purchased product |  |
| 7 | Vendor\_id | Number (20) | Vendor no. | Reference key |
| 8 | vat | Number (20) | Vat tax |  |
| 9 | total | Number2(40) | Total purchase value |  |

* 1. ***VENDOR TABLE***

This is the vendor detail form which stores the information related to the vendors that are supplying products to the supermarket.

|  |  |  |  |
| --- | --- | --- | --- |
| Fieldname | Data type | description | constraints |
| vendor\_id | Number (5) | Vendor no. | Primary key |
| vendor\_name | Varchar2(30) | Vendor name |  |
| Address | Varchar2(30) | Vendor address |  |
| Phone\_no | Number (10) | Vendor Phone no. |  |
| Email\_id | Varchar2(30) | Vendor Email\_id |  |

**5. DATA FLOW DIAGRAM**

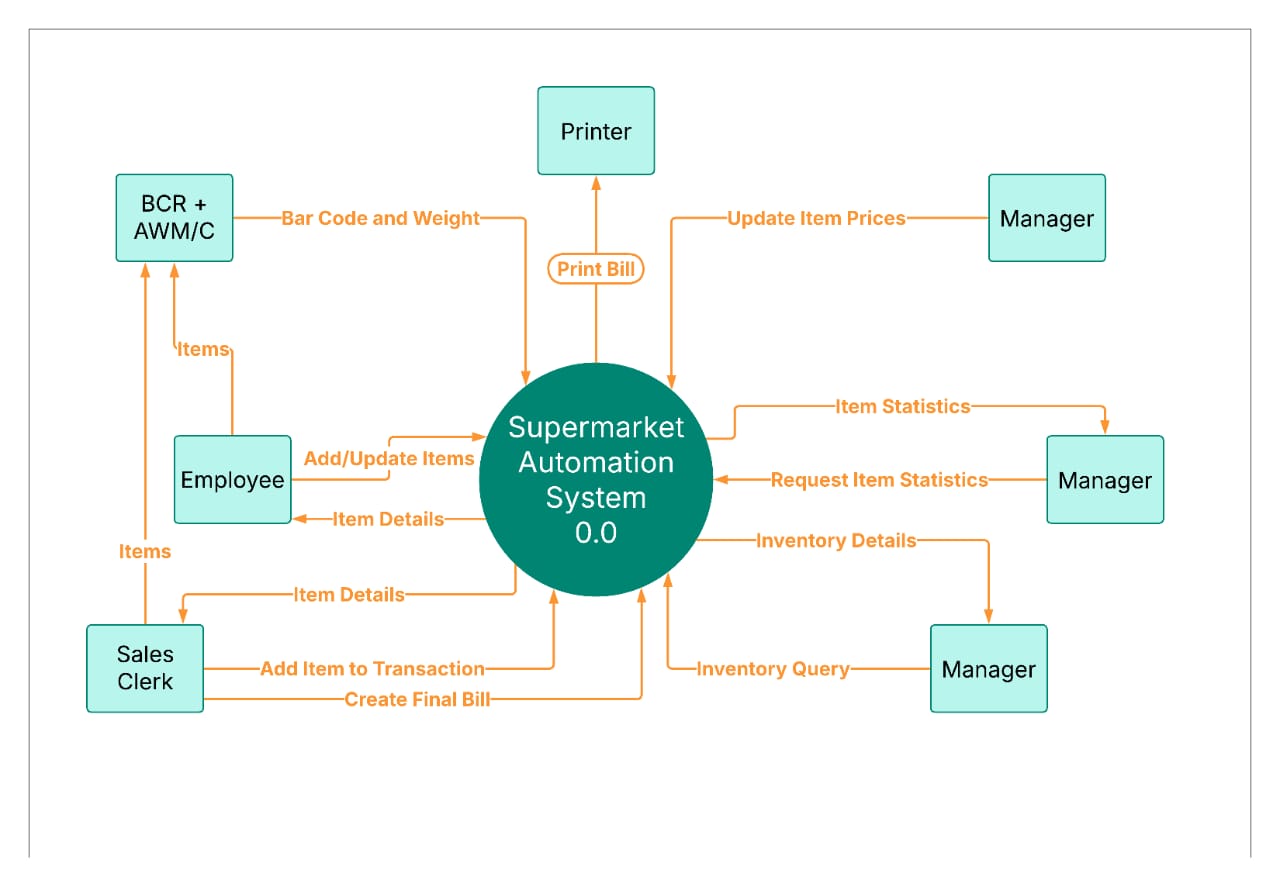
***5.1 Levels of the DFD***

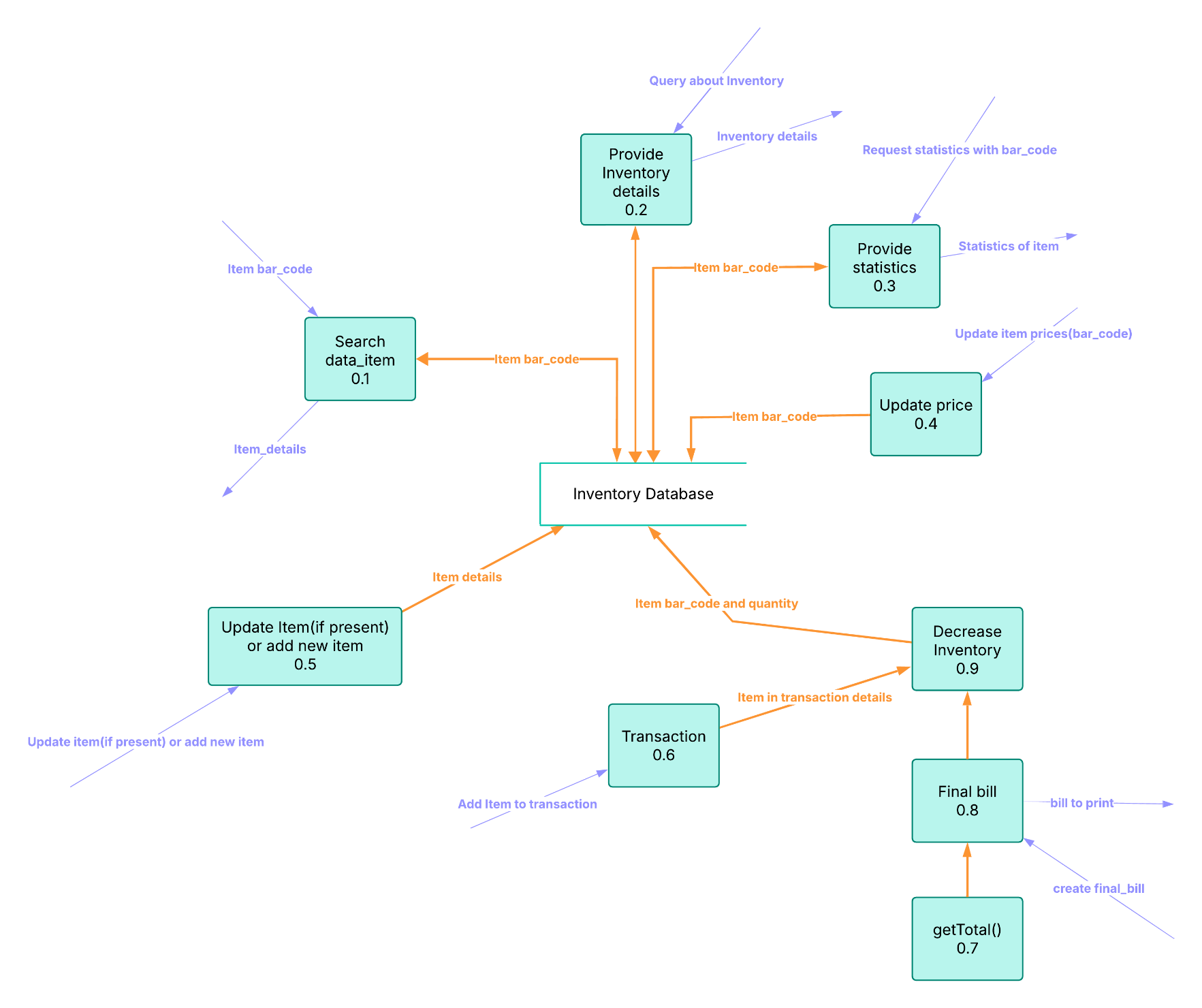
1. **Level 0 (Context Diagram):**

Shows the entire system as a single process interacting with users (Customer, Manager and Supplier).

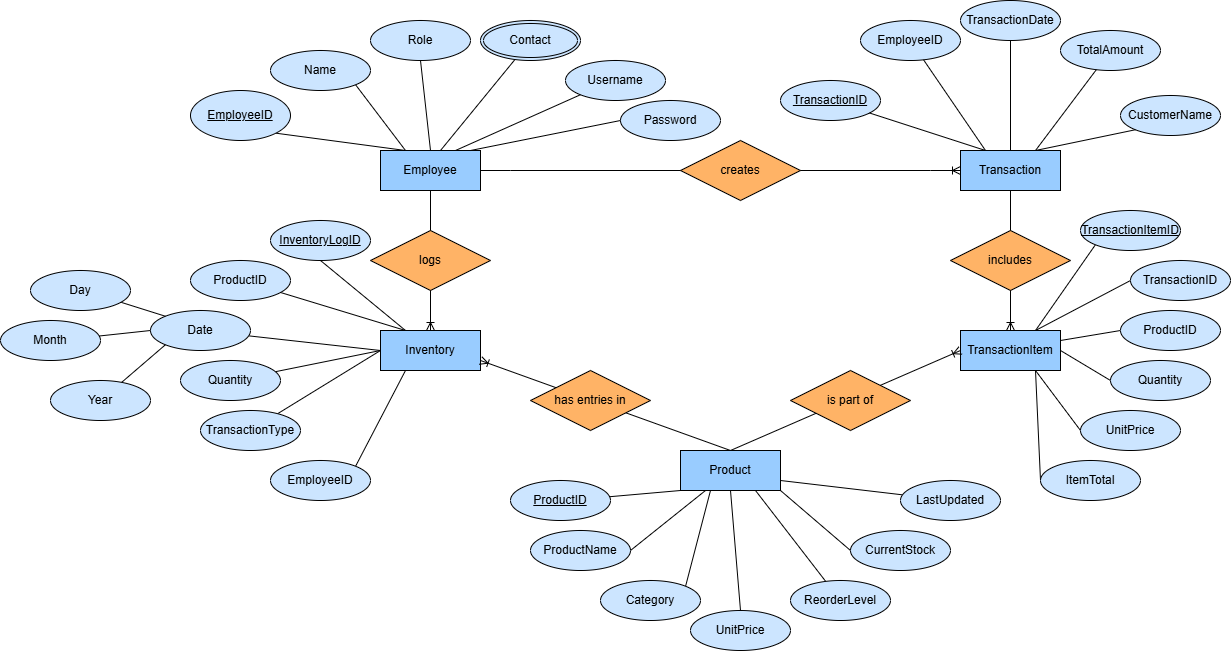
1. **Level 1:**Expands the system into major processes:
   1. Sales Process: Order Processing, Customer Sales Details
   2. Purchase Process: Vendor Orders, Invoice Processing, Payment Handling
   3. Inventory Management: Stock Updates, Supply of Items
   4. User Management: Oversight, Work management.

***5.2 LEVEL 0 DFD DIAGRAM***

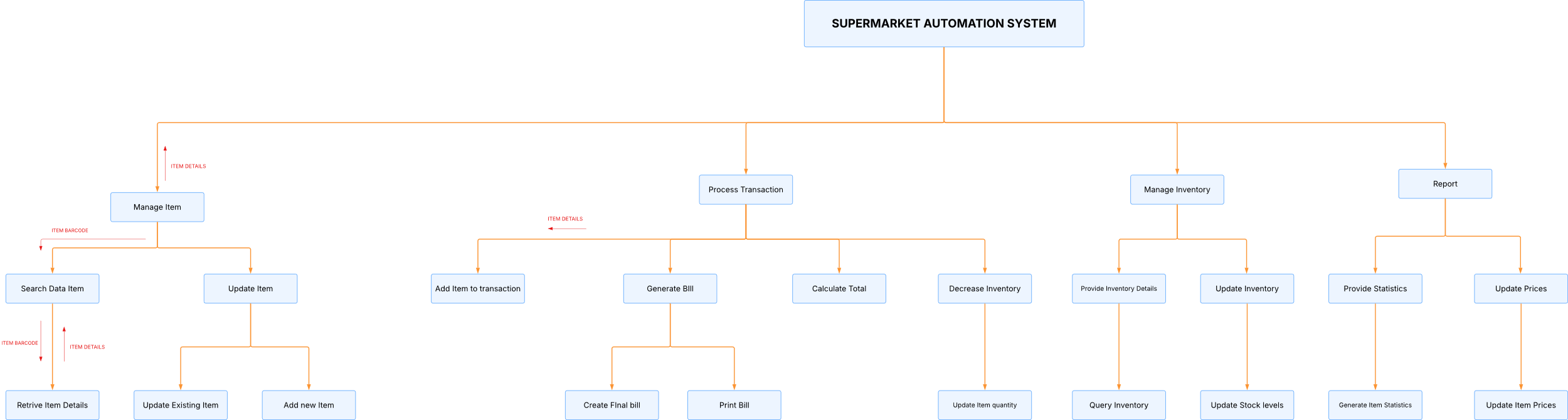


***5.3 LEVEL 1 DFD DIAGRAM***

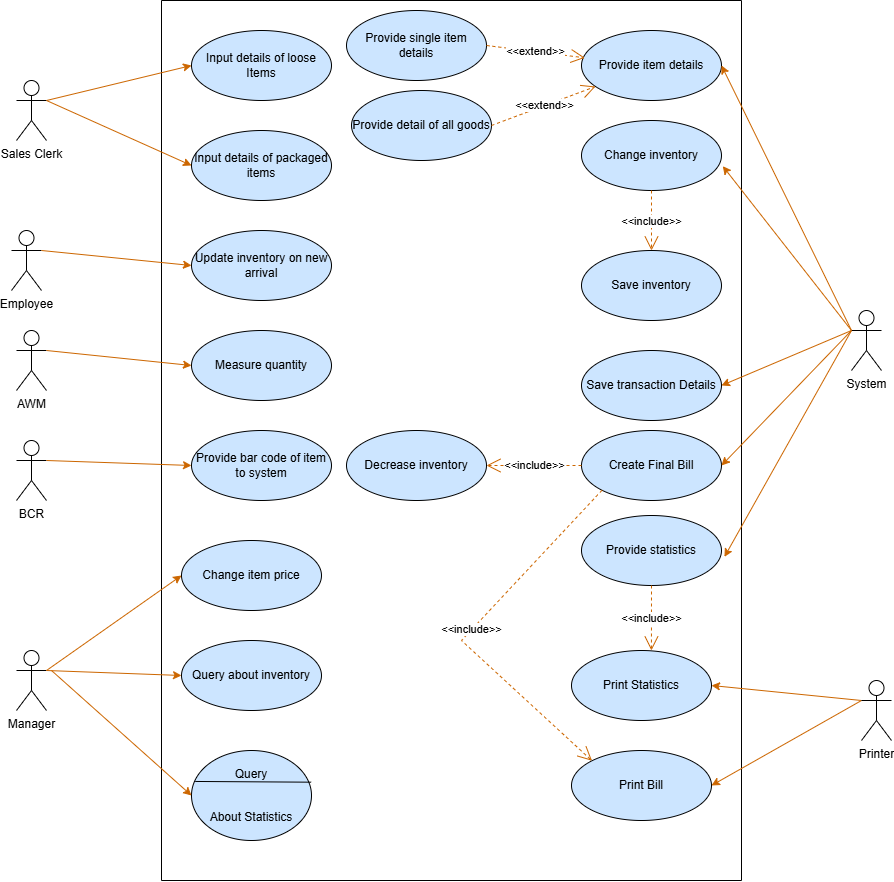
***5.4 ER DIAGRAM***



***5.5 STRCUTURE CHART***



***5.6 USE CASE DIAGRAM***



***5.7 CLASS DIAGRAM***

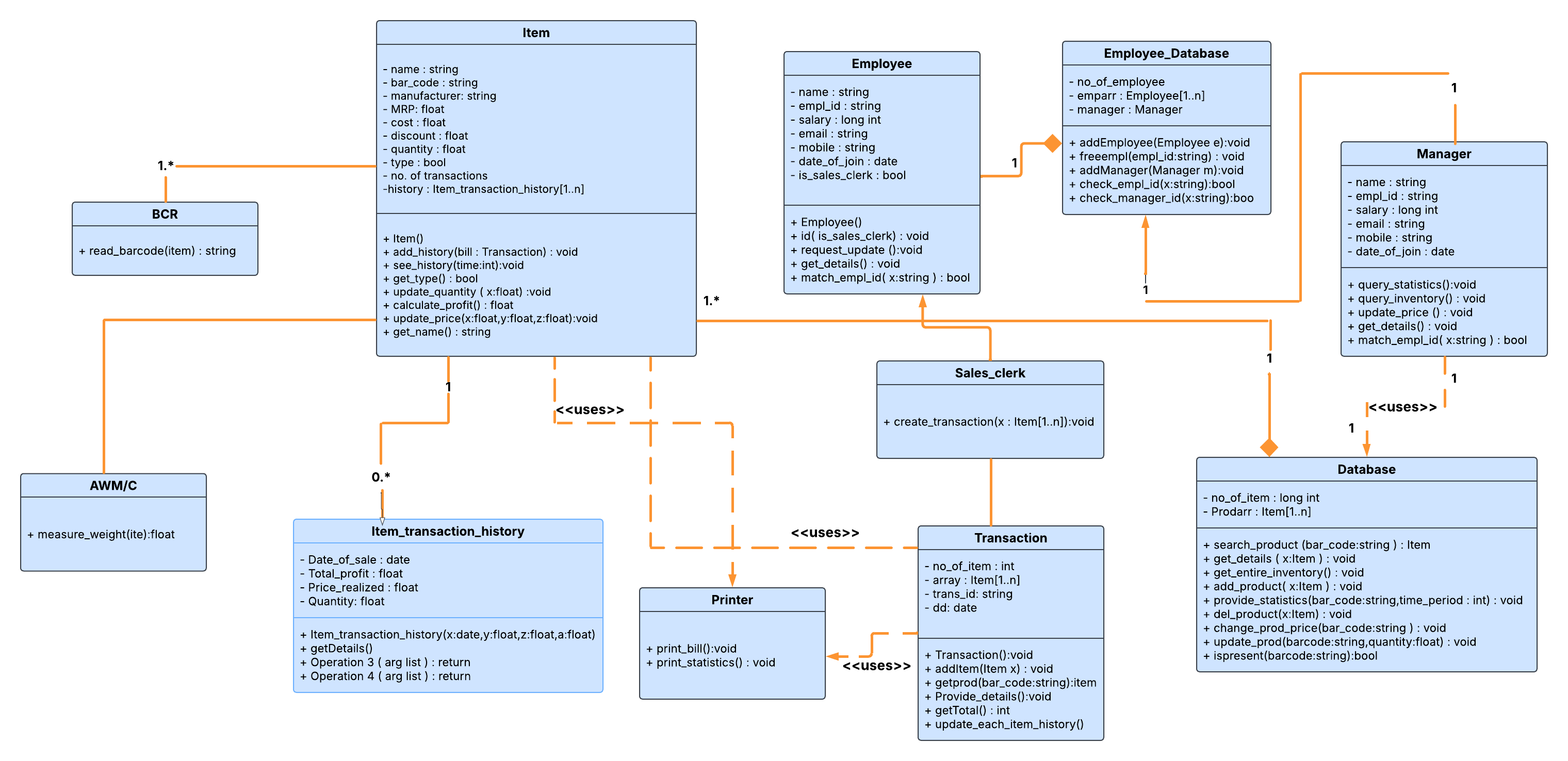
The class diagram is the main building block of object-oriented modelling. It is used both for general conceptual modelling of the systematics of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modeling.[1] The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

### **Brief Description**

There are 11 main classes. Those include

1. Item
2. Employee
3. Sales Clerk
4. Manager
5. Employee Database
6. Database(basically it’s inventory database)
7. Transaction
8. Item transaction history
9. AWM/C
10. BCR
11. Printer

The relationships among various classes have been shown below. The re- quired attributes and methods of each class derived from use case diagram have been clearly mentioned in the use case diagram.



***5.8 ACTIVITY DIAGRAMS***

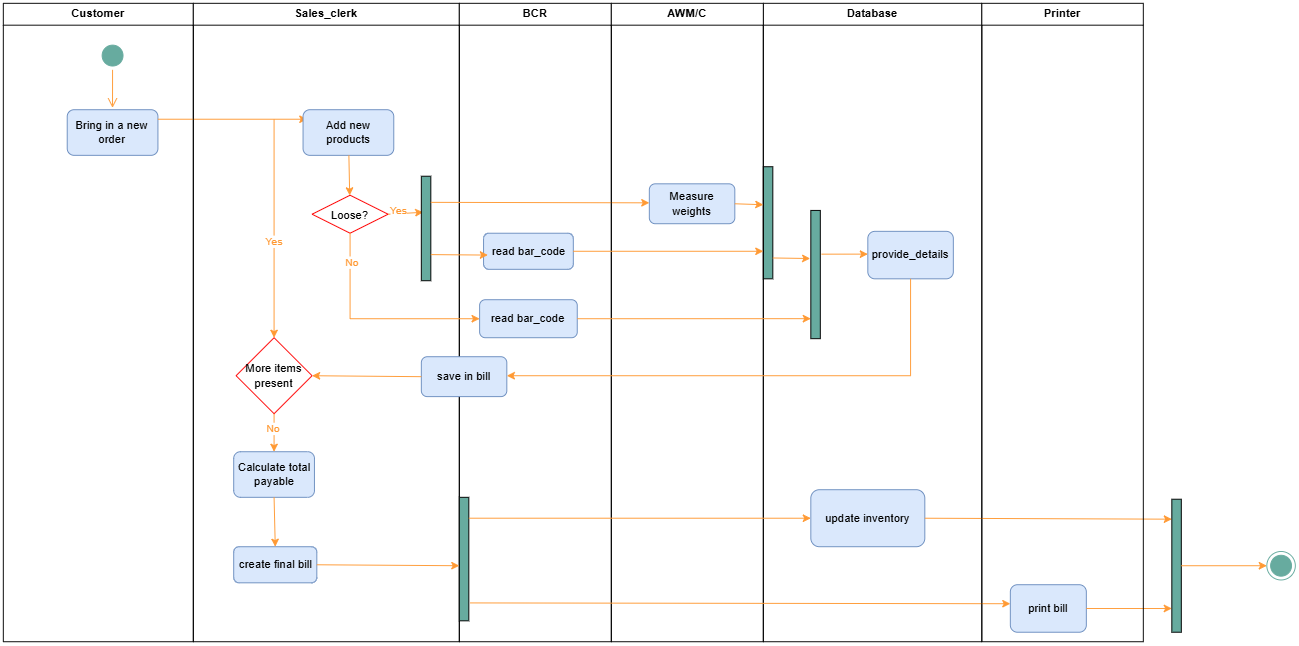
**5.8.1** **ACTIVITY DIAGRAM 1: SALES TRANSACTION**

Basic data flow includes:

* + 1. When the customer brings in order, the sales clerk places items one by one in front BCR and AWM/C (if loose).
    2. The BCR gives the bar code of the item to the system. The system searches for item details in the database and adds those details in the transaction.

3. After doing this for all items, final billing is done and inventory of each good is

decreased by the Required amount.



**5.8.2 ACTIVITY DIAGRAM 2: VIEW SALES STATISTICS**

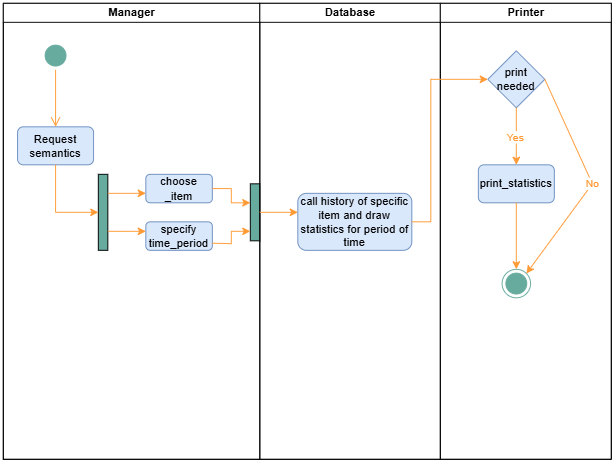
Basic data flow includes:

1. When the manager gives item bar code and time period to the system, the

system Searches the item in its database. .

2. After the item is found, the item’s history is passed on to the screen.

3. Printing is optional.

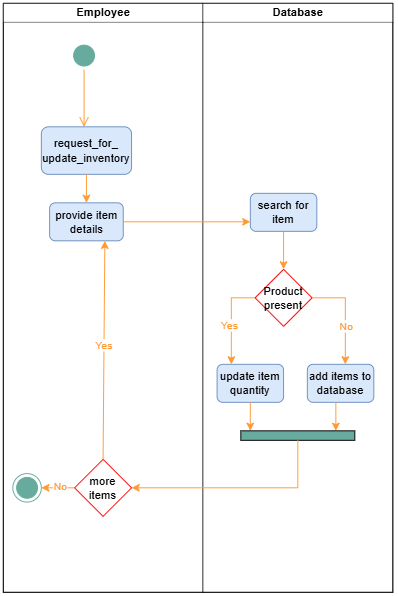


**5.8.3 ACTIVITY DIAGRAM 3: UPDATE INVENTORY**

Basic data flow includes:

1. On new arrival, the employee places the item in front of BCR and AWM/C (if loose). The database searches for the item in the database.

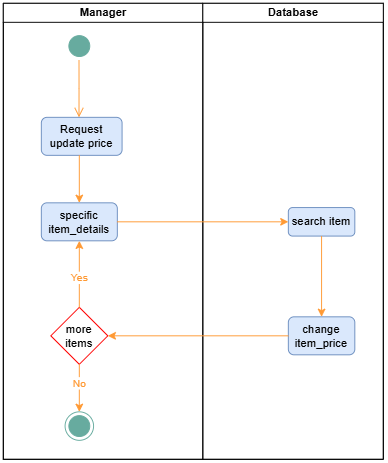
2. If item is present in the database, the system simply adjusts the quantity of item else if the item isn’t present, it adds a new item to the database.



**5.8.4 ACTIVITY DIAGRAM 4: UPDATE PRICE**

Basic data flow includes:

* + 1. The manager first specifies item barcode as input.
    2. Then the system searches for the item in its database.

Next, the system changes the item’s price details as specified by the manager