Clothes Store Management System

Purpose:

The system provides the management for its business and workers affairs.

It has already facilitate the process of buying and selling and store the clothes.

It’s also controls the Vendor’s updates and make the things go easy.

Scope:

This software provide keep track of the stock of different types of clothes available in the shop, including the quantity and price.

The system should be able to generate bills for customers, including the details of the clothes purchased, the price, and the payment method.

The system should maintain a record of the customer's details including phone number.

The system should maintain records of the employees' details including their names and salary and bonus on total sales for each employee.

The system should be secure and have a backup plan in case of data loss or system failure. It should also be able to restrict access

to sensitive information and provide access only to authorized personnel.

Objective:

The management system should help the shop owner to efficiently manage the stock of clothes, ensuring that the shop always has the right amount of stock available, reducing the chances of stockouts or overstocking.

The system should provide enhanced security and data backup features, ensuring that the shop's data is secure.

The system should be scalable and flexible enough to accommodate the shop's growth and expansion plans, easily integrating with other systems and technology solutions as needed as there is a reliable technical service.

This software is aims to save the cost and time very much as it everything is doing automatically and efficiently at fastest time possible.

Feasibility Study:

A feasibility study is an analysis of the viability of a business idea. This study will focus on the economic feasibility, technical feasibility, and operational feasibility of opening a clothing store.

Technical feasibility:

The development of a management system for a cloth shop is technically feasible as there are many programming languages, frameworks, and libraries available to support the development of such systems.

This system will be developed using Desktop-based technologies such as .net framework using VB.net language, making it accessible from any device with an internet connection.

Integrating the system with other software or hardware such as barcode scanners or point of sale systems is also possible.

Economic feasibility:

The development of a management system for a cloth shop may require a significant investment of time and resources, including hiring developers, acquiring hardware and software, and ongoing maintenance costs.

However, the potential benefits of the system, such as improved efficiency, increased sales, and profitability, may outweigh the costs in the long run.

The system can also be designed to be scalable, allowing the shop to expand its operations without incurring significant additional costs.

Operational feasibility:

The management system should be designed to meet the specific needs of the cloth shop, including inventory management, sales and billing, customer management, employee management, accounting and financial management, and reporting and analytics.

The system should be easy to use and accessible to all employees, reducing the amount of time and effort required to train new employees.

The system should also be reliable and secure, reducing the risk of data loss or theft and ensuring the shop's operations are not disrupted.

The Requirements:

Functional Requirements:

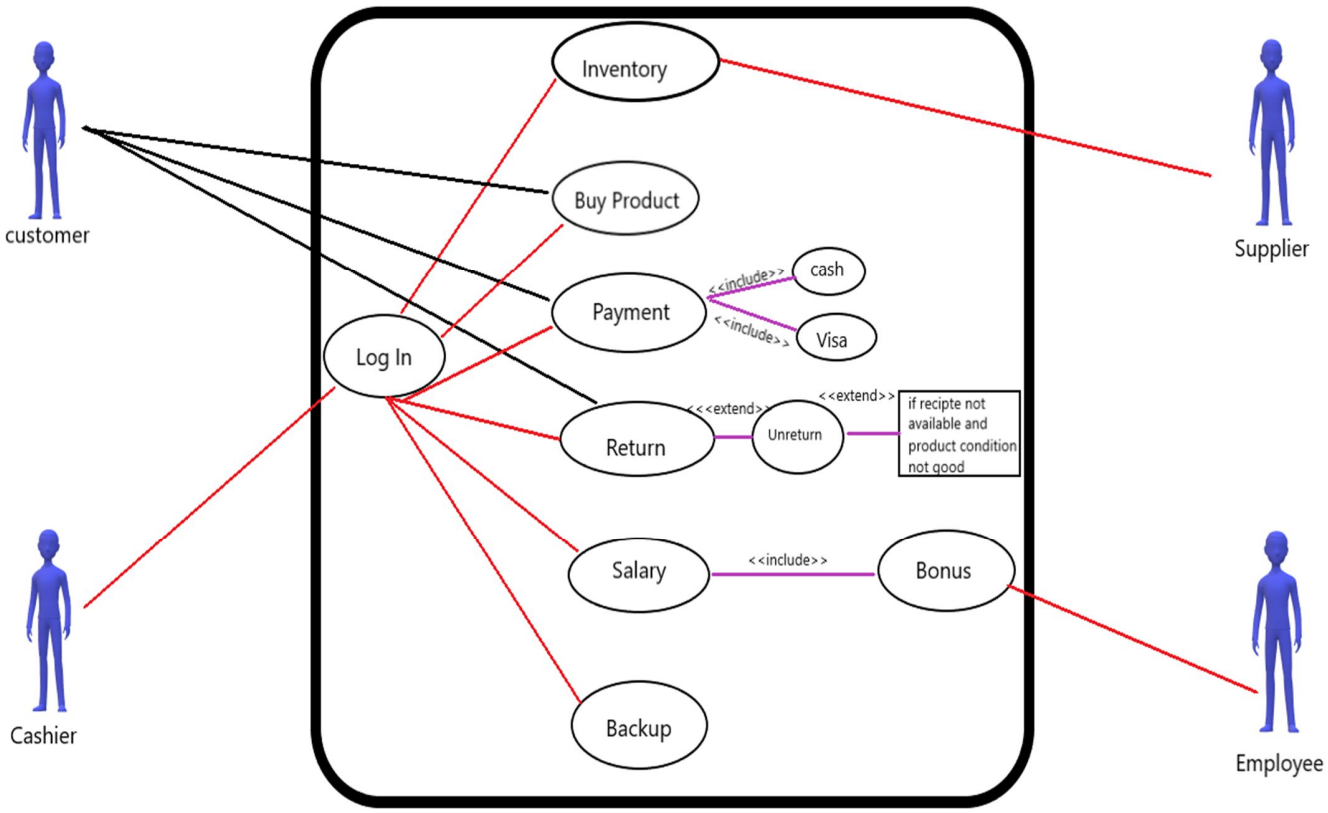
- The system is very easy for the new user to learn how to work on it.
- The number of the products in any model is specified.
- The system has all the function required for their work as the can enter their products with the serial number.
- There is also the option to type the name for all suppliers and traders and the rest of dues for each one. The system is connected to scanner than scan the barcode for each product when the customer buys it and calculate the price and if there is any discount on it.
- When any product is sold its automatically removed from the system.
- It helps very much with the process of inventory and make is much easier and accurate.

Non-Functional Requirements:

- The speed of the system is great and have no lags.
- Every user logging with the username and password and have a specified privileges that cannot overrides it.
- It's very flexible.
- It has a time limitation and after that time the system is closed and cannot accept new customers at the end of the shift.
- The customer service is available for any problem related to the system to repair it from the manufacture company.

Use Case:

Use Case Diagram:



Actors:

Customer ,Employee , Cashier, Supplier.

Use Cases:

Log in ,inventory, Buy Product, payment ,cash ,visa , Return, unreturn , Salary, bonus , Backup.

Flow of Event 1:

Use case: Log in

Actors: Cashier “Manager”

Description: Cashier logs into the system with the username and password valid for him, Cashier and Manager have the same privileges to the system so they can add and remove users “employee”.

Flow of Event 2:

Use case: Stores To Inventory

Actors: Cashier, Supplier.

Description: When there is a lack or shortage of products the supplier is contacted based on his data on the system by the cashier, then he provide the required new items and it’s stored in the inventory, then cashier calculates the fees for the products and give it to him and record it in the system.

Flow of Event 3:

Use case: Buy Product.

Actors: Cashier, Customer, Employee.

Description: When the Customer arrives at the shop. The Employee in charge helps him to find his needs and when customer chooses what he wants, he goes to the cashier who start the scan the product and get it into the system, then he takes customer’s information, take out the alarm from the product and give him the receipt and the item.

Flow of Event 4:

Use case: Payment, Cash, Visa.

Actors: Cashier, Customer.

Description: When the customer is completing his buying process, Cashier asks him about his payment method (Cash Or Visa) to finish the process.

Flow of Event 5:

Use case: Return, Unreturn.
Actors: Cashier, Customer.
Description: When the Customer came and want to return any item, he goes to the cashier who check:

1) Receipt availability.
2) The item's condition that must be the same as when he bought it.
3) The Validity of the Return duration (14 days).

If all these conditions are well met, then the item is returned and it's re-added to the system, and if they didn't met then item is not returned.

Flow of Event 6:

Use case: Salary, bonus.
Actors: Cashier, Employee.
Description: At the end of every month, the basic salary for each employee is calculated and added to the bonus on the total items he sold. And the cashier takes his basic salary as well.

Flow of Event 7:

Use case: Backup.
Actors: Cashier.
Description: The cashier can take backup of the system's whole data every 2 weeks to make sure that nothing can be missed and save the system from any failure can happen in the future.

Include:

- To do anything in the system the cashier should first login by his username and by the password to open his account.
- If he doesn't login by his account, he cannot do any of the process of return, buy product and payment.
- To calculate the total salary, we should add the bonus of the month to the base salary.

Extend:

- If the customer wants to return or exchange any product he must go to the cashier with the reset and the product.
- The reset must be available, and the product must be in the same condition it was bought on it.
- Any absence of these terms he cannot return or exchange the product.
- If the terms are available, then the cashier scans the bar code of the product and get it back on the system again and but the alarm on it.
- Then the customer gets his money back or exchanges the product with any other product he wants.

Observation:

- The security of the shop is very high as he uses:
    - Security cameras with mic.
    - Alarm on every clothes piece.
    - The system gives specific privileges for each user so the user cannot exceed the privileges granted for him.
- The system has flexibility to be extended in the future.
- It has a backup feature that provide the safety and functionality to the system.

Interview:

• Open Ended Questions:

هل بتقدر تتعامل مع السيستم لو حصل مشكلة؟ ولو..هل بتستعين بحد مختص؟
ما هي مميزات وعيوب السيستم اللي شغال بيه؟
هل السيستم شامل كل احتياجاتك ولا شايف حاجه محتاجة تتضاف ليه؟
شايف نفسك مميز عن المنافسين في ايه؟
هل في شراء اونلاين؟ وهل السيستم بيدعمه؟
عندك نية إنك تفتح فرع تاني؟ ولو اه..شايف انك تقدر توفر وقت لالتنين؟
علي أي أساس يعتمد اختيار الموظفين؟
لو حصل مشكلة بين الموظفين والعملاء بتتصرف ازاي؟

• Closed ended Questions:

قيم جودة وفاعلية السيستم بالنسبه ليك.

قيم قابلية وسهوله التعامل مع السيستم للموظفين.

هل بتعتمد علي الموظفين ولا لازم تكون موجود بنفسك؟

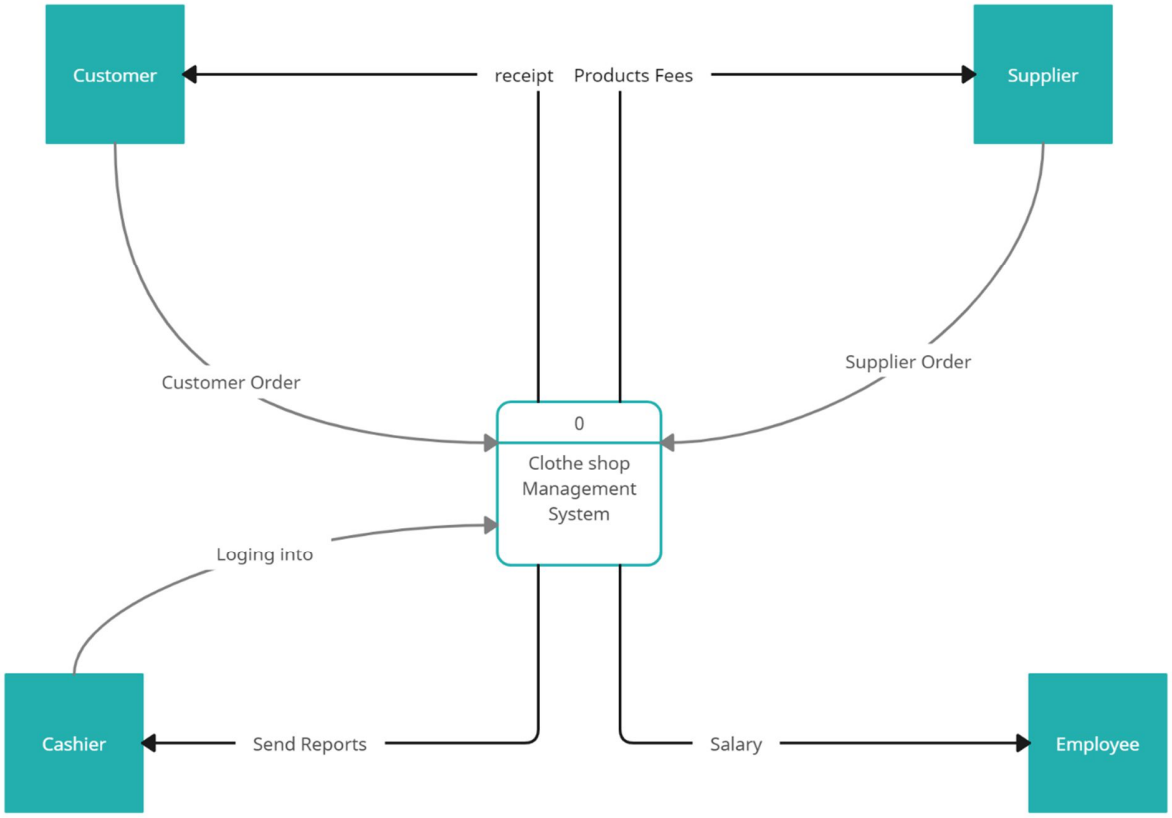
قيم معاملة الموظفين مع العملاء؟

هل مسموح للموظف بأخذ إجازات طارئة

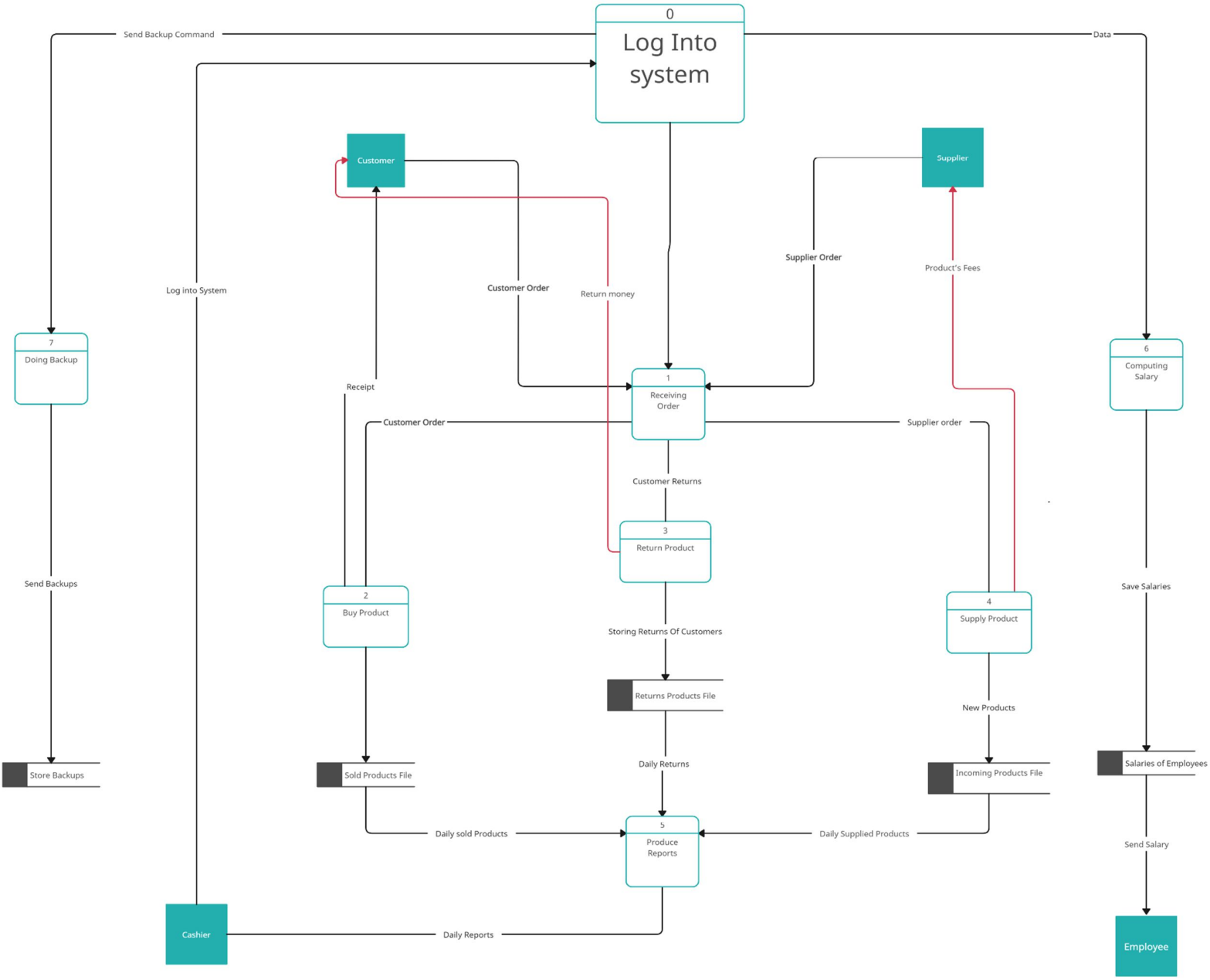
هل فيه زيادة ثابتة في الرواتب؟

Data Flow Diagram (DFD):

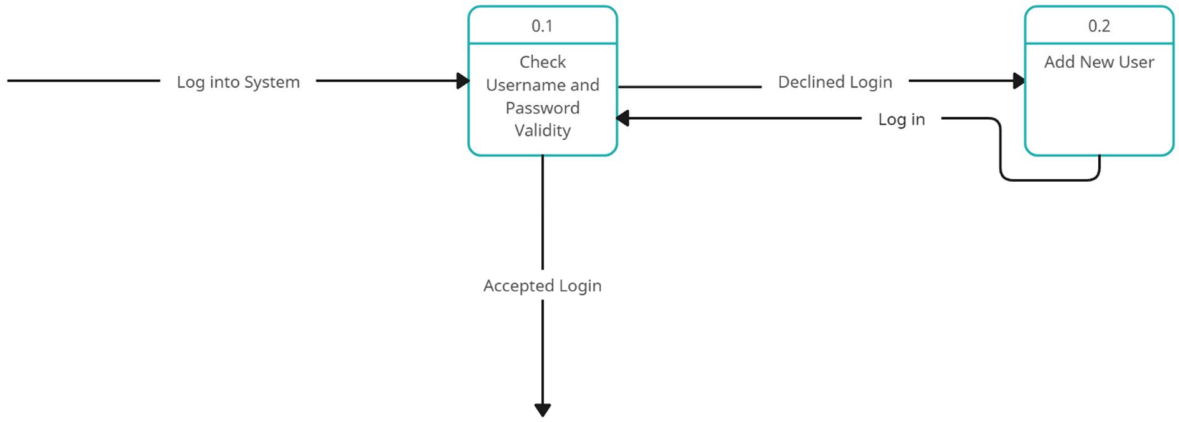
Level Context:



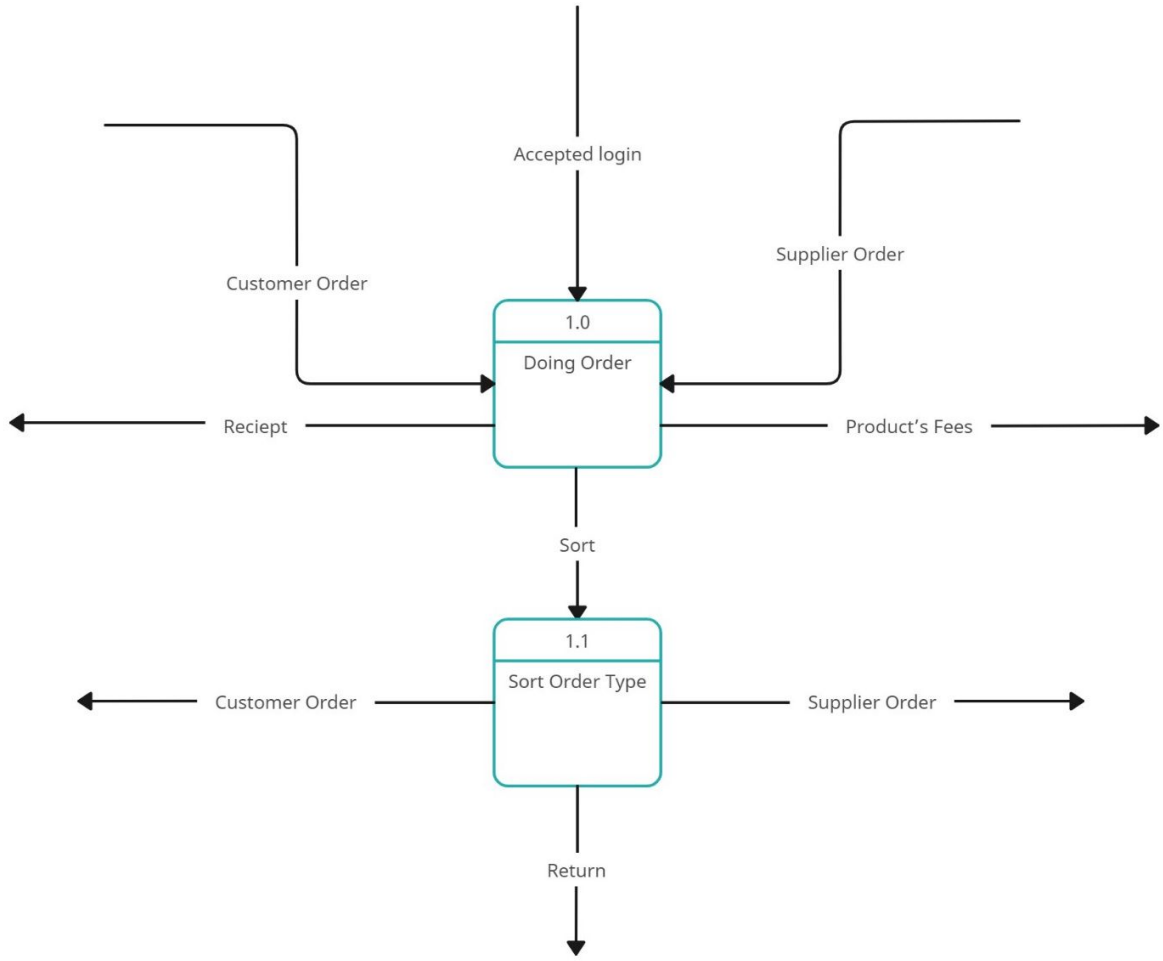
Level 0:


0
Log Into system
Send Backup Command
Data
Customer
Supplier
Log into System
Customer Order
Return money
Supplier Order
Product's Fees
7
Doing Backup
6
Computing Salary
Receipt
1
Receiving Order
Customer Order
Supplier order
Customer Returns
3
Return Product
Send Backups
2
Buy Product
4
Supply Product
Save Salaries
Storing Returns Of Customers
Returns Products File
New Products
Store Backups
Sold Products File
Daily Returns
Incoming Products File
Salaries of Employees
Daily sold Products
5
Produce Reports
Daily Supplied Products
Send Salary
Cashier
Daily Reports
Employee

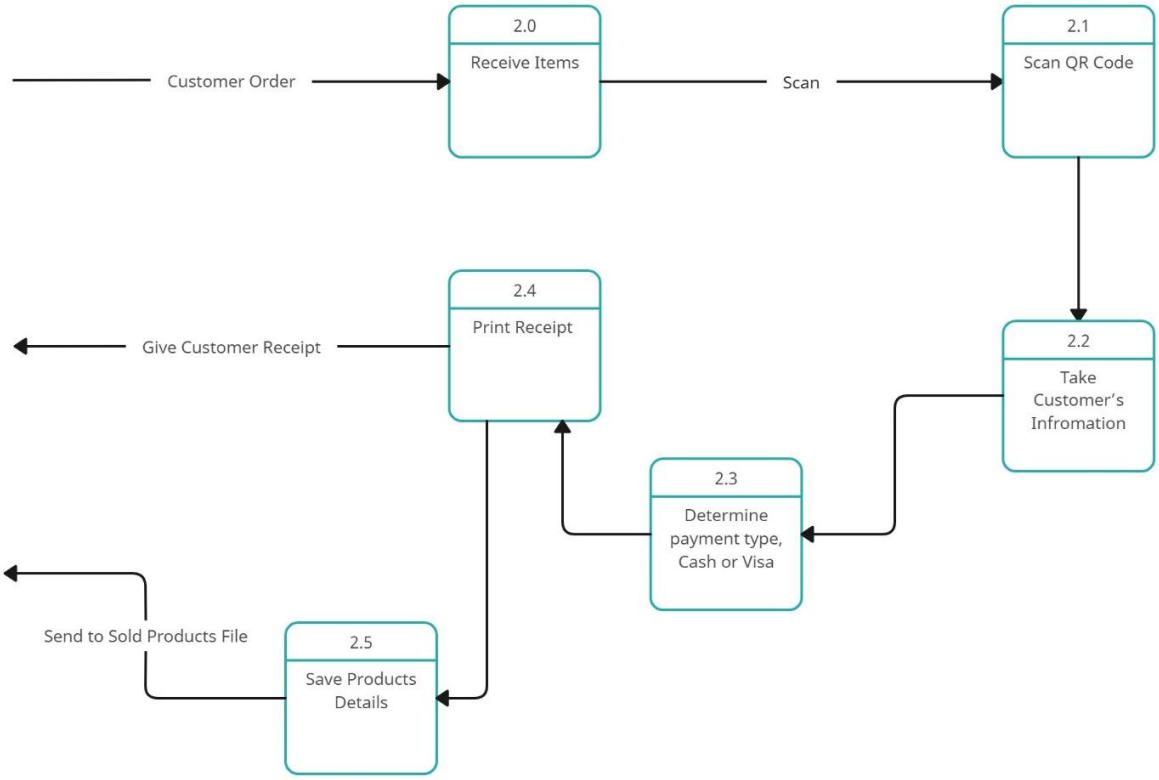
Level 1 Process 0:



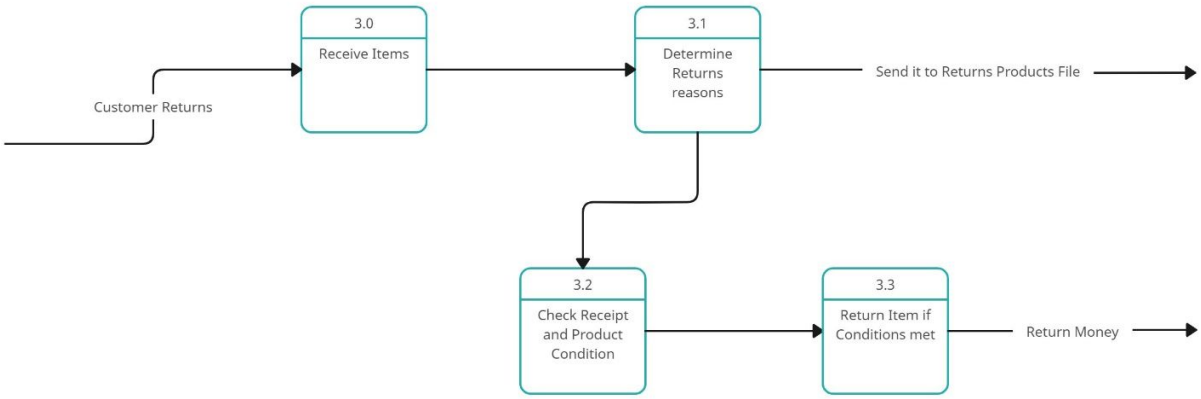
Level 1 Process 1:



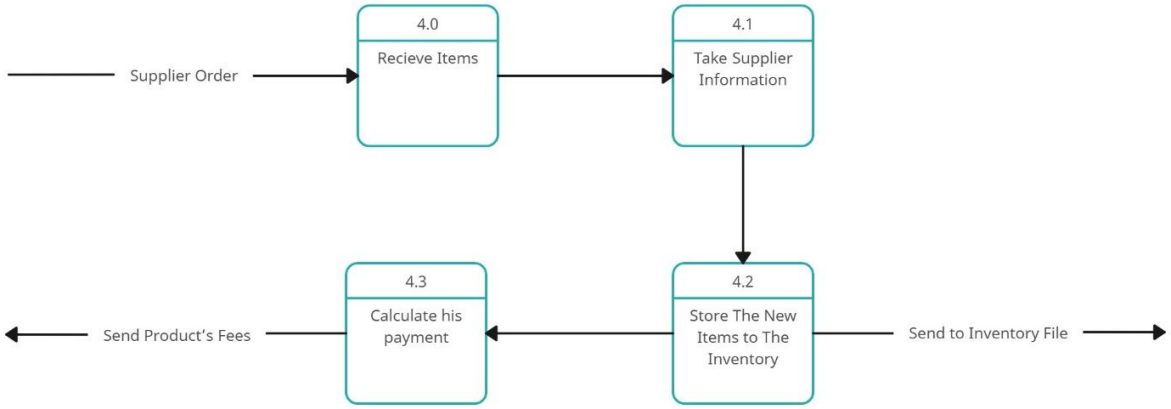
Level 1 Process 2:



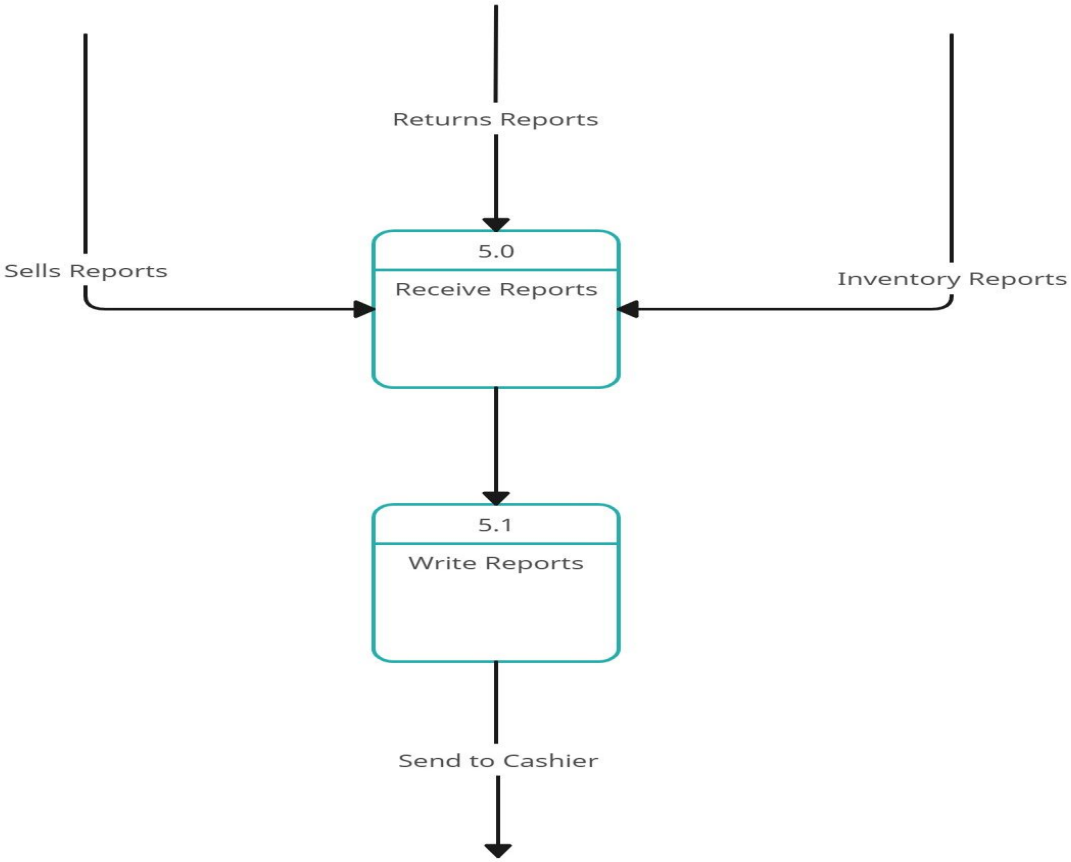
Level 1 Process 3:



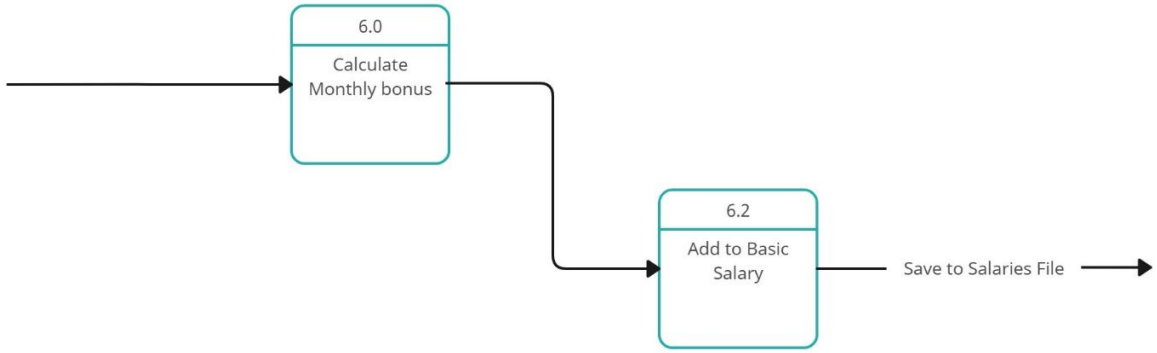
Level 1 Process 4:



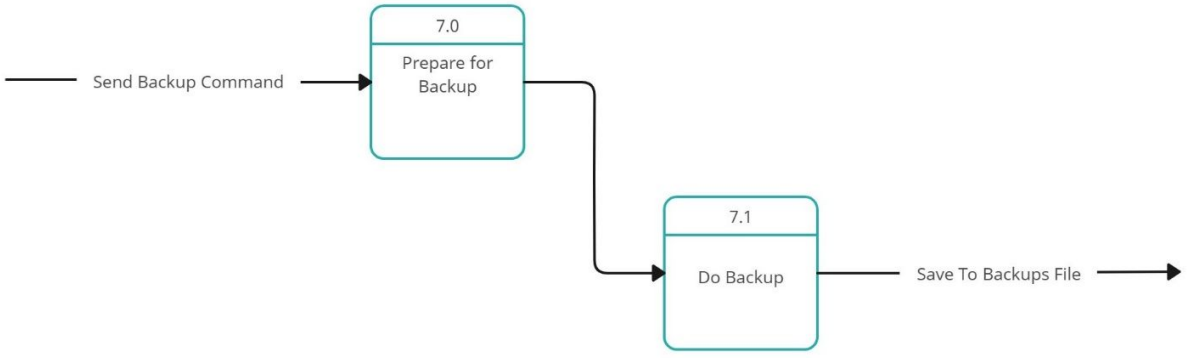
Level 1 Process 5:



Level 1 Process 6:



Level 1 Process 7:



Logical Modeling:

Process 0.0: Log into System (English Structure):

Do

Enter username, password.

if the username exists and the password is right:

Log in done.

if the username doesn't exist:

Add a new user

Provide access to users.

UNTIL End-Of-File

Process 1.0: Receive Order (English Structure):

Do

If Accepted login Process:

Receive products from the supplier.

Receives buying request from the customer.

Sort orders type (customer, supplier, returns).

UNTIL End-Of-File

Process 2.0: Buy product (English Structure):

Do

Receive buying order.

Scan the bar code of the product.

Take the customer's information.

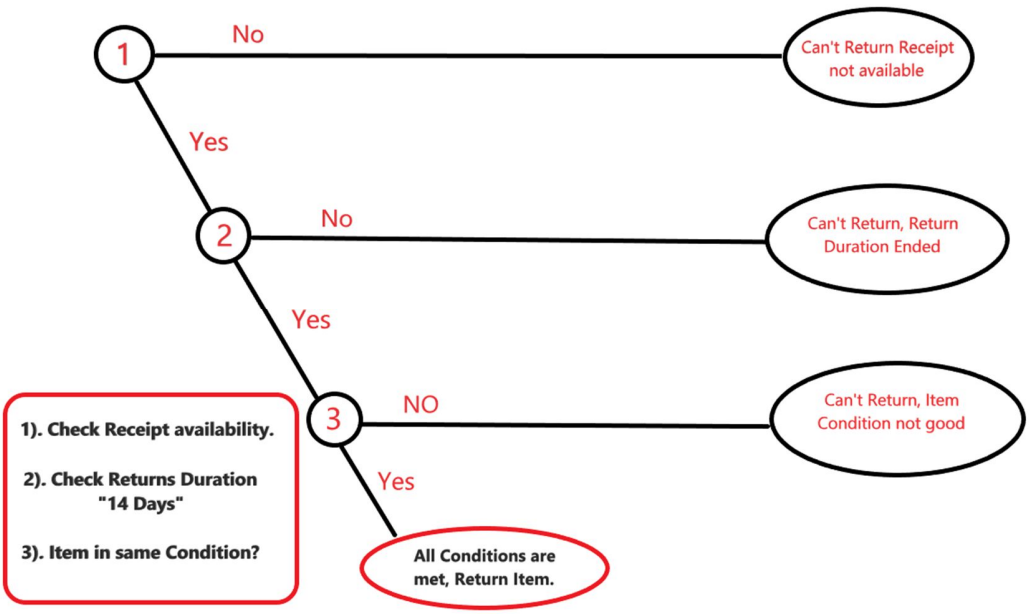
Receive payment via cash or visa.

Send the receipt.

Store the product's details in (sold products file).

UNTIL End-Of-File

Process 3.0: Return product (Decision Tree):



Process 4.0: Supply Products (English Structure):

Do

If new products is needed:

Contact Supplier.

Receive incoming products.

Store the supplier's information.

Store the product in the inventory (incoming products file).

Calculate the product fees.

Send the product fees.

UNTIL End-Of-File

Process 5.0: Receive Reports (English Structure):

Do

If True:

Receives sells reports.

Receives returns reports.

Receives incomings reports.

Send all the reports for the cashier.

UNTIL End-Of-File

Process 6.0: Computing Salary (Decision Table):

	Conditions /Courses of Actions	Rules		
Conditions Stubs		1	2	3
	Employee Type	C	E	E
	Target	–	< 20,000	> 20,000
Actions Stubs	Take Basic Salary	X	X	X
	Take Bonus			X

Process 7.0: Doing Backup (English Structure):

Do

If 2 weeks passed:

Receive backup command.

Start the backup.

Save the backup in (backup file).

UNTIL End-Of-File