

Software Design Document

Point of Sale Simulator

Version 0.1

Team:

Aadithya H Rao – PES1UG21CS005

Aaryan Hemant Badyal –
PES1UG21CS009

Akanksha Sendhil – PES1UG21CS055

Akshay Anand – PES1UG21CS064

Organisation: PES University

Revisions

Version	Primary Author	Description of Version	Date Completed
0.1	Aadithya, Aaryan, Akanksha, Akshay	Initial Release	27/10/2023

Table of Contents

1. Introduction	4
1.1 Purpose	4
1.2 Scope	4
1.3 Definitions, Acronyms and Abbreviations	4
1.4 References	4
2. System Overview	5
3. System Components	6
3.1 Decomposition Description	6
3.2 Dependency Description	7
3.3 Interface Description	8
3.4 Module Interfaces	8
3.5 User Interfaces (GUI)	8
4. Detailed Design	11
4.1 Module Detailed Design	11
4.2 Data Detailed Design	12
4.3 RTM	13

1. Introduction

1.1 Purpose

This software requirement specification document describes all the features and requirements of the Point of Sale system for version 1.0 release. This document serves as a guide to the members and managers of the project team, and should help anyone understand the implementation. The overall design of the project must be clearly mentioned here.

1.2 Scope

The design document describes the implementation of the POS system. The POS system has different functionality for Transaction History, Billing Management and Inventory Management. All of these components will connect to database and will go through the login page. Each of the above components and systems will be described in the design documentation.

1.3 Definitions, Acronyms and Abbreviations

Acronym	Meaning
POS	Point of Sale
SDD	Software Design Document
OS	Operating System
API	Application Programming Interface

1.4 References

- <https://nextjs.org/docs>
- <https://react.dev/learn>
- <https://www.postgresql.org/docs/>

2. System Overview

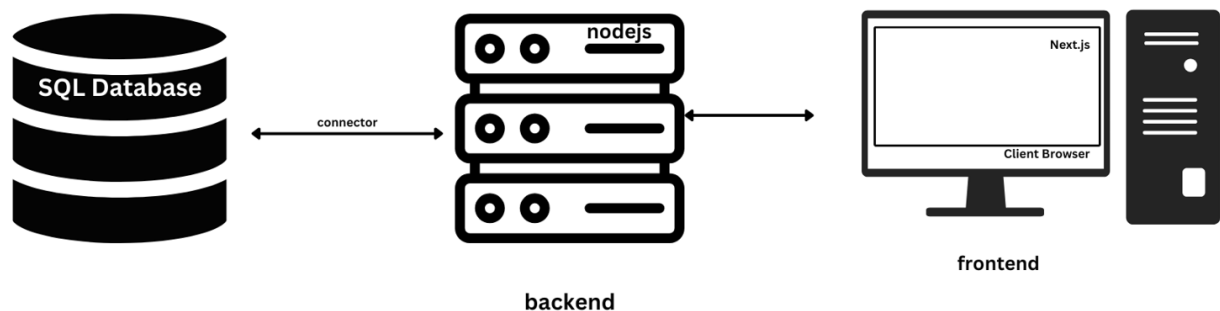


FIGURE 1

The above figure shows the overall architecture that the POS system will be using. The backend will provide different functionalities depending on the role of the user. For the Inventory manager, the backend will query the database for the inventory stock present and those that need to be restocked. For employees, the backend will only take input of the billing items, deduct the required amount and update sales history. While for managers, it will query the database for the sales and transaction history.

The database will also store all the relevant information regarding sales and inventory management. Secured access will be provided based on the user roles mentioned.

The front-end initially provides a login page. This login page is used by different users and provides the roles to the users with proper authentication. The login page then redirects the user to the respective page depending on the role of the user. These pages then make the appropriate connections to the back-end.

3. System Components

3.1 Decomposition Description

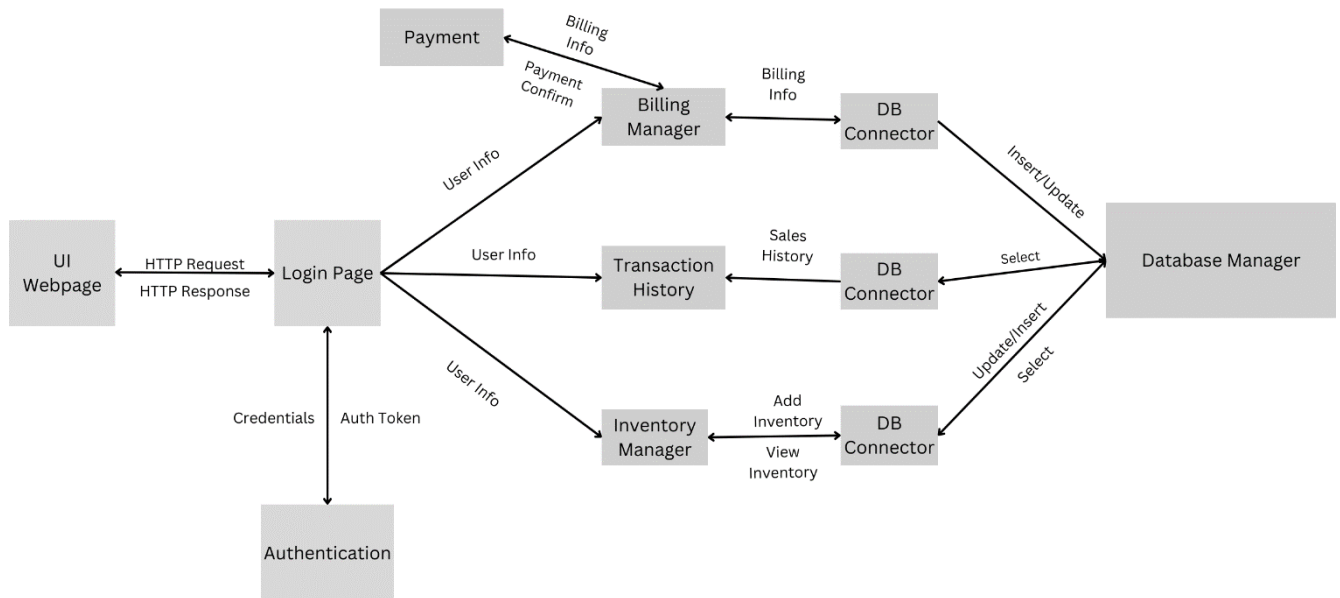


FIGURE 2

The above figure depicts the various components present in our system and how they interact. It also shows what data is sent from one component to another. The UI webpage on the front-end connects and sends messages to the backend using HTTP messages. Login authentication is the first event which occurs when users reach the website. The login page also transfers important data regarding the role of the user. The other components also connect to the database to query or write the information using database connectors.

The Billing Manager generates the bill based on the inputs scanned. It connects to the payment gateway and updates the inventory stock by updating the amount. The Billing Manager also generates a sales report which is written into the database. The Transaction Manager can read the database for the data written and the Inventory Manager can read and update the inventory stock when required.

3.2 Dependency Description

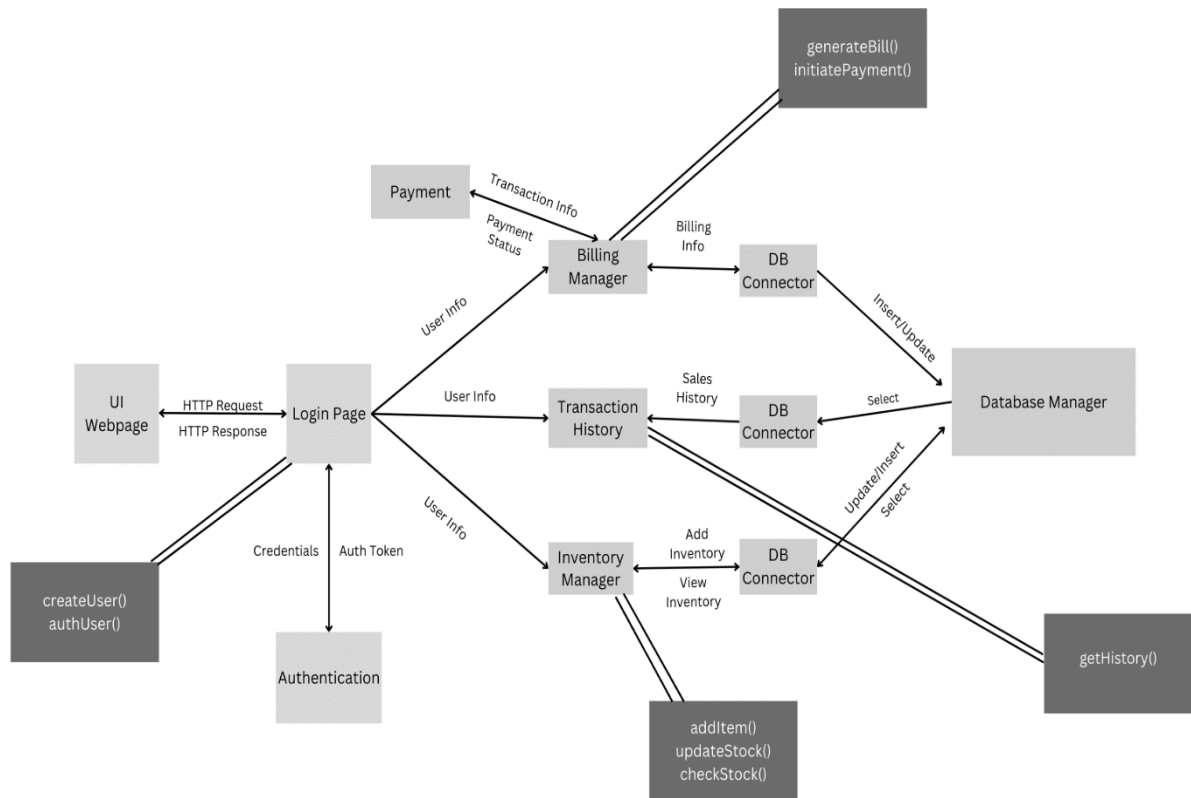


FIGURE 3

The above figure shows the dependency among the components and their functionality. The figure also gives information regarding the different functions or API calls that can be made by each component. This diagram is very useful for developers to implement the system.

Login component uses functionality to create and authenticate users. The different components for different roles also have their own implementation which allows them to use the database connectors to read or write the required information from the database.

3.3 Interface Description

The various components present in the diagrams above are the interfaces of the system. The interfaces send messages to each other and are largely independent in their internal working.

3.4 Module Interfaces

The inner working and functionality of all the interfaces was described earlier under the decomposition diagram and serves as a good reference for any implementational queries.

3.5 User Interfaces (GUI)

The first page in the website is the login page which asks for the required credentials and then allows the user to enter the appropriate page based on their role.

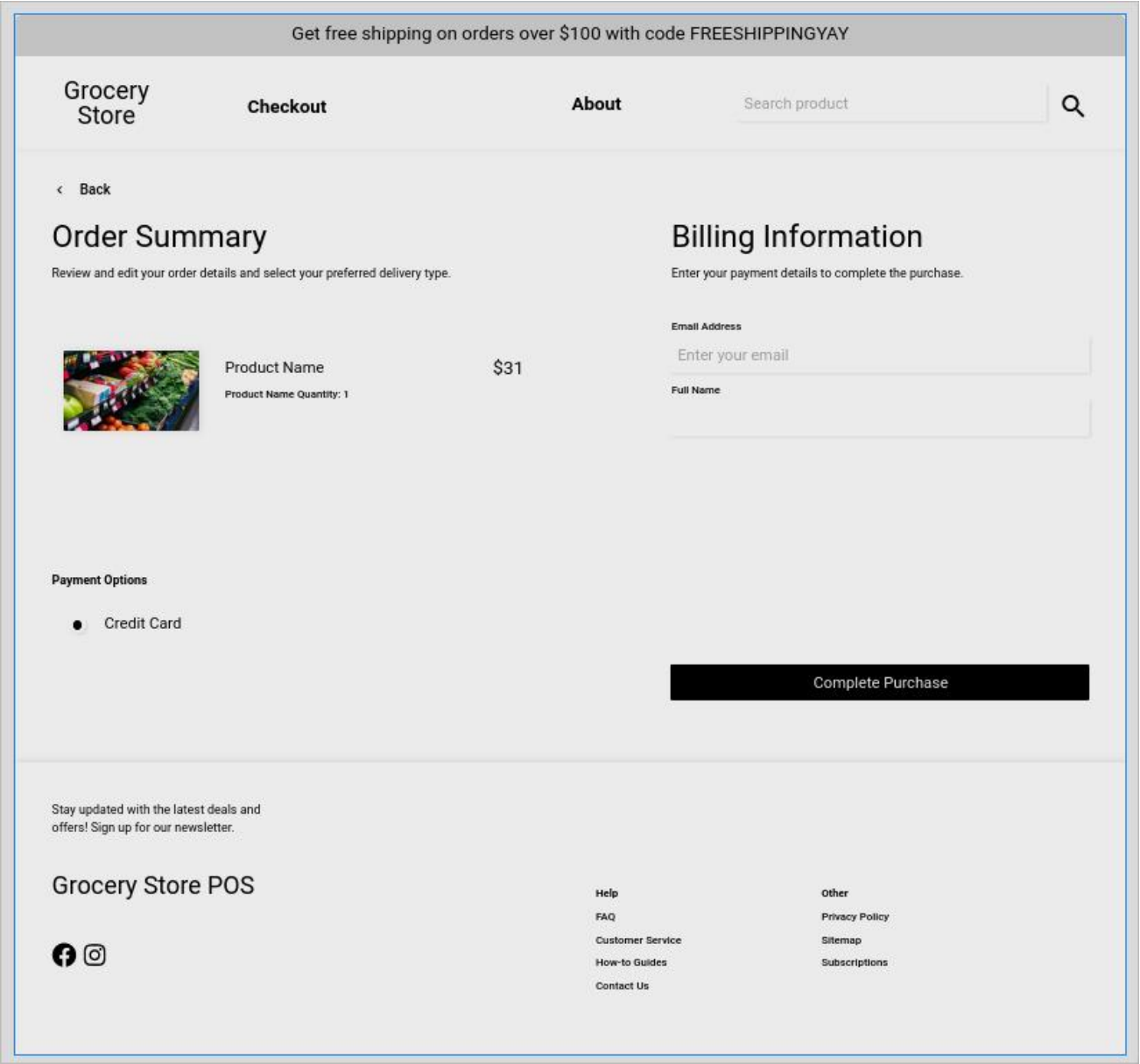
Don't have an account? [Sign up](#)

Sign In

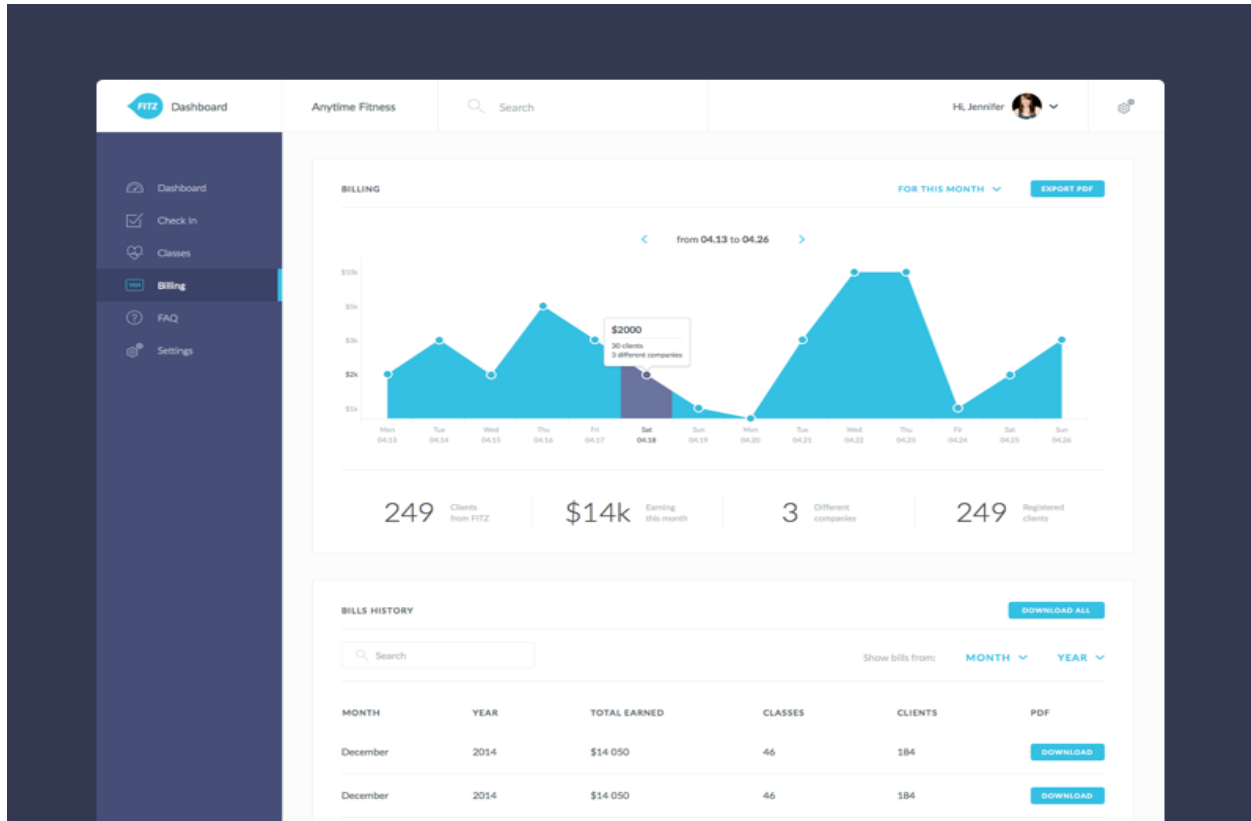
[Or sign in with your email](#)

This app is protected by reCAPTCHA and the app's Privacy Policy.

The second page is the billing page which is used by employees to scan input. This automatically generates the bill and updates the database.

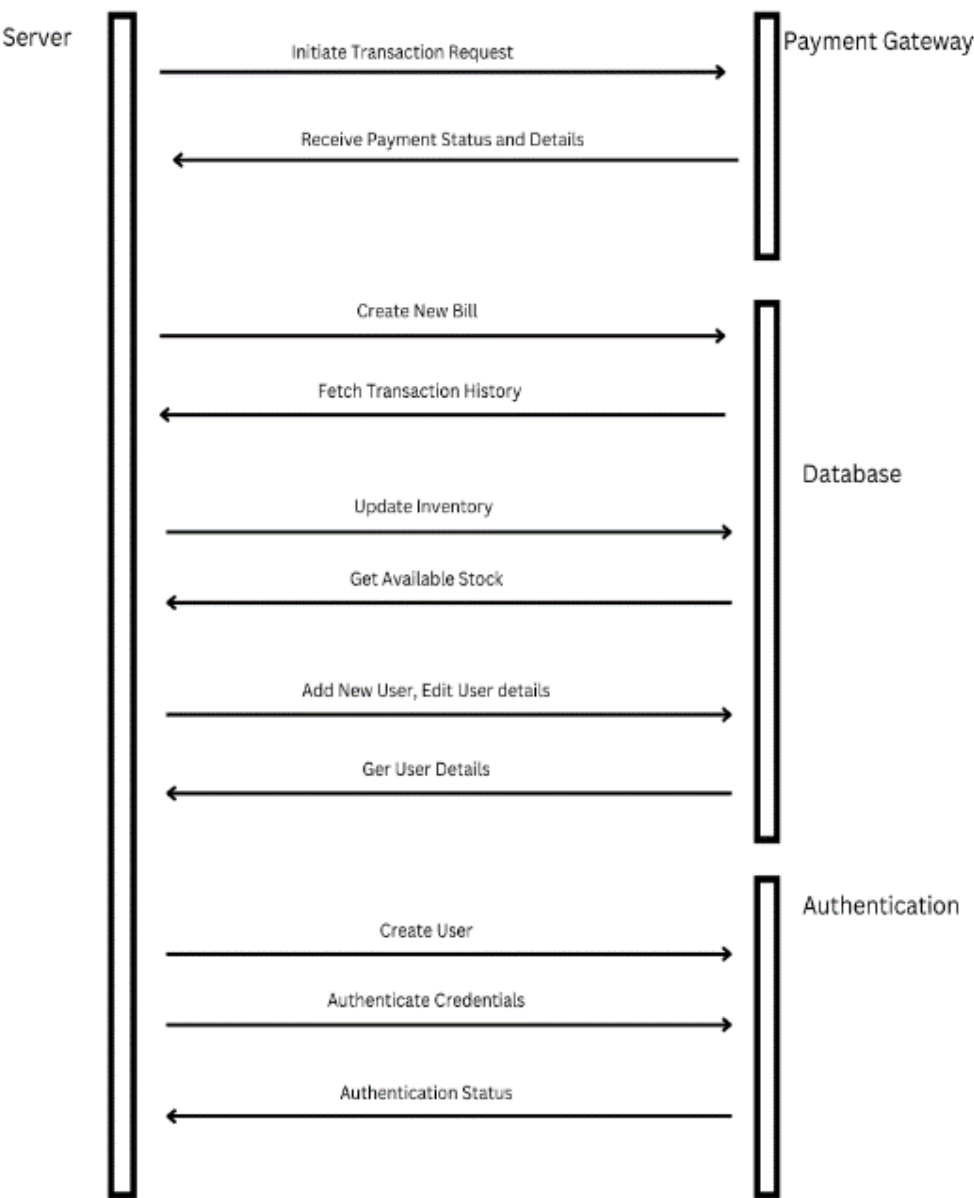


Another page is the transaction or sales history page used by managers to keep track of successful employees.

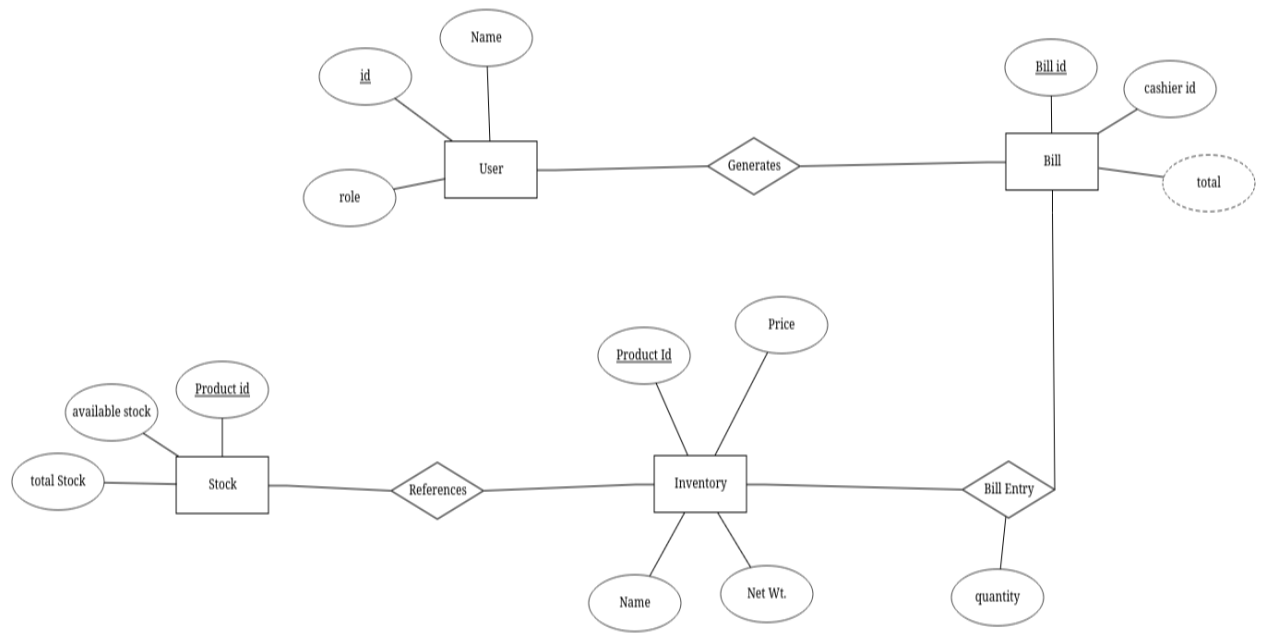


4.Detailed Design

4.1 Module Detailed Design



4.2 Data Detailed Design



ER Diagram

4.3 RTM

Requirement-ID	Requirement Description	Design Component	Test-Case #
5.1	User access to website	UI Webpage	20
4.1.1 , 4.1.2 ,4.1.3, 5.3 ,5.1	Register a profile, access registered account	Login Page	1,2,3,9
4.2.1 ,4.2.2 ,4.2.3	Payment, inserting bill info to Database and subtracting item sold from inventory.	Billing Manager	7,8,10,11,12
4.3.1 ,4.3.2 ,4.3.3,5.5	View Transaction history	Transaction History	4,5,6,13
4.4.1 ,4.4.2 ,4.4.3, 5.3 ,5.5	View and add to inventory	Inventory Manager	14,15,16,17, 18,19
5.3 ,5.5	Stores Transaction history, Inventory information and User information.	Database	