Software Requirements Specification

for

Master Invoice

Version 1.0

Prepared by

Group 1:		Group Name: while (1)
Cezan Vispi Damania	230310	cezanvd23@iitk.ac.in
Ch V Sai Koushik	230312	skoushik23@iitk.ac.in
Challa Kethan	230317	kethanc23@iitk.ac.in
Challa Umesh Varun	230318	cumesh23@iitk.ac.in
Chilamakuri Kundan Sai	230330	ckundans23@iitk.ac.in
Gudi Praneeth Sai	230425	gudips23@iitk.ac.in
Kishore Senthil Kumar	230566	kishores23@iitk.ac.in
Sai Saketh Mogillapalli	230895	saisaketh23@iitk.ac.in
Srijani Gadupudi	231033	srijanig23@iitk.ac.in
Vundela Obula Reddy	231178	voreddy23@iitk.ac.in

Course: CS253

Mentor TA: Hemang Mohanlal Khatri

Date: 24-01-2024

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
v1.0	Cezan Vispi Damania Ch V Sai Koushik Challa Kethan Challa Umesh Varun Chilamakuri Kundan Sai Gudi Praneeth Sai Kishore Senthil Kumar Sai Saketh Mogillapalli Srijani Gadupudi Vundela Obula Reddy	The first version of the requirement document	24/01/2025

1 Introduction

1.1 Product Scope

In today's fast-paced world of commerce, intermediaries are the unsung heroes who keep the supply chain moving smoothly. From ensuring products make their journey from factories to store shelves, they play a critical role in connecting manufacturers with retailers. However, traditional methods like jotting down records in notebooks can be time-consuming, prone to errors, and ineffective when it comes to generating invoices or analyzing market trends.

That's where Master Invoice comes in—a smart, intuitive software solution built to simplify the challenges intermediaries face. It streamlines supply chain management, covering everything from inventory tracking and invoice processing to financial record-keeping and market analysis.

By digitalizing workflows, Master Invoice reduces manual errors, eliminates bottlenecks, and boosts efficiency. Key features like inventory management help intermediaries keep a close eye on stock levels, ensuring purchases align with demand. With tools for tracking inward and outward supplies, managing payments, and monitoring pending transactions, users gain financial clarity and control. On top of that, powerful analytics provide actionable insights, enabling intermediaries to spot best-selling products and prioritize key suppliers.

Master Invoice empowers intermediaries to work smarter, not harder. It transforms their operations into a more transparent, productive, and profitable process, helping them stay ahead in a competitive marketplace.

1.2 Intended Audience and Document Overview

1.2.1 Intended Audience

This Software Requirements Specification (SRS) document is designed to cater to various stakeholders involved in the **Master Invoice** project, including:

- **Developers**
  Developers will use this document to understand the functional and non-functional requirements of the system. It provides detailed descriptions of the system architecture, modules, and data flows, enabling effective implementation and integration.
- **Project Managers**
  Project managers will refer to this document to track deliverables, ensure alignment with project goals, and manage timelines and resources. It provides a high-level overview of the system's purpose, scope, and key requirements.
- **Marketing Staff**
  Marketing professionals will find the overview and user-related sections valuable to

understanding the product's unique features, target audience, and competitive advantages. This information supports creating effective promotional strategies.

- **End Users**
  The document includes sections explaining the system's functionality and user interface, providing users insight into how the system meets their needs for managing inventory, transactions, and financial analysis.
- **Testers**
  Testers will rely on this document for defining test cases, understanding system behavior, and verifying that the software meets its specifications.
- **Documentation Writers**
  Writers responsible for creating user manuals or training materials will use this document as a foundation for understanding the system's functionality, terminology, and workflows.

1.2.2 Document Overview

1. **Introduction**
   This section introduces the purpose and scope of the **Master Invoice** system, defines the target audience for the document, and outlines its structure. It also provides relevant definitions, acronyms, and document conventions to ensure clarity. Lastly, references and acknowledgments are listed to credit sources and contributors.
2. **Overall Description**
   This section provides a high-level overview of the system, including its goals and functionalities. It describes how the product addresses user needs, outlines the design an d implementation constraints, and lists assumptions and dependencies that influence system development and operation.
3. **Requirements**
   This section is the core of the document, detailing both functional and non-functional requirements.
   - **Functional Requirements**: Describes the specific features and functionalities, such as inventory management, inward and outward supplies, and financial analysis.
   - **Non-Functional Requirements**: Specifies performance metrics, safety and security needs, and software quality attributes like scalability and reliability.
   - **External Interface Requirements**: Details the interactions with external systems, devices, and users.
   - **Use Case Model**: Includes user scenarios and workflows to demonstrate how the system behaves in different situations.
4. **Additional Requirements**
   This section includes supplementary details that support the core requirements, such as the data dictionary, which defines data elements and structures, and any additional requirements that don't fit into other sections.
5. **Appendices**
   The appendices provide supporting materials, such as a detailed data dictionary in **Appendix A** and a group log in **Appendix B** that documents team activities and decisions throughout the project.

1.3 Definitions, Acronyms and Abbreviations

Term	Definition
CSS	Cascading Style Sheets
FAQs	Frequently Asked Questions
HTML	Hyper Text Markup Language
MRP	Maximum Retail Price
OTP	One Time Password
SRS	Software Requirements Specification

1.4 Document Conventions

Formatting Conventions:

1. The document is written in the **Arial** font size 11 with 1.15 spacing and 1-inch margins.
2. Headings are written in **Arial** font, using bold characters with a font size of 18.
3. Subheadings are written in **Arial** font, using bold characters with a font size of 14.
4. Sub Subheadings are written in **Arial** font, with a font size of 13, and are bold.
5. Bullet point ordering has been used as a listing typesetting tool.
6. Some important words are made bold.

Naming Conventions:

1. User - Our client who uses.
2. Retailer - Customer of our client.
3. Supplier - Supplier to our client.
4. Product - Goods that are sold or purchased by our client.
5. Invoice Bills - Bills that are either given to the client from the supplier or given by the client to the retailer.

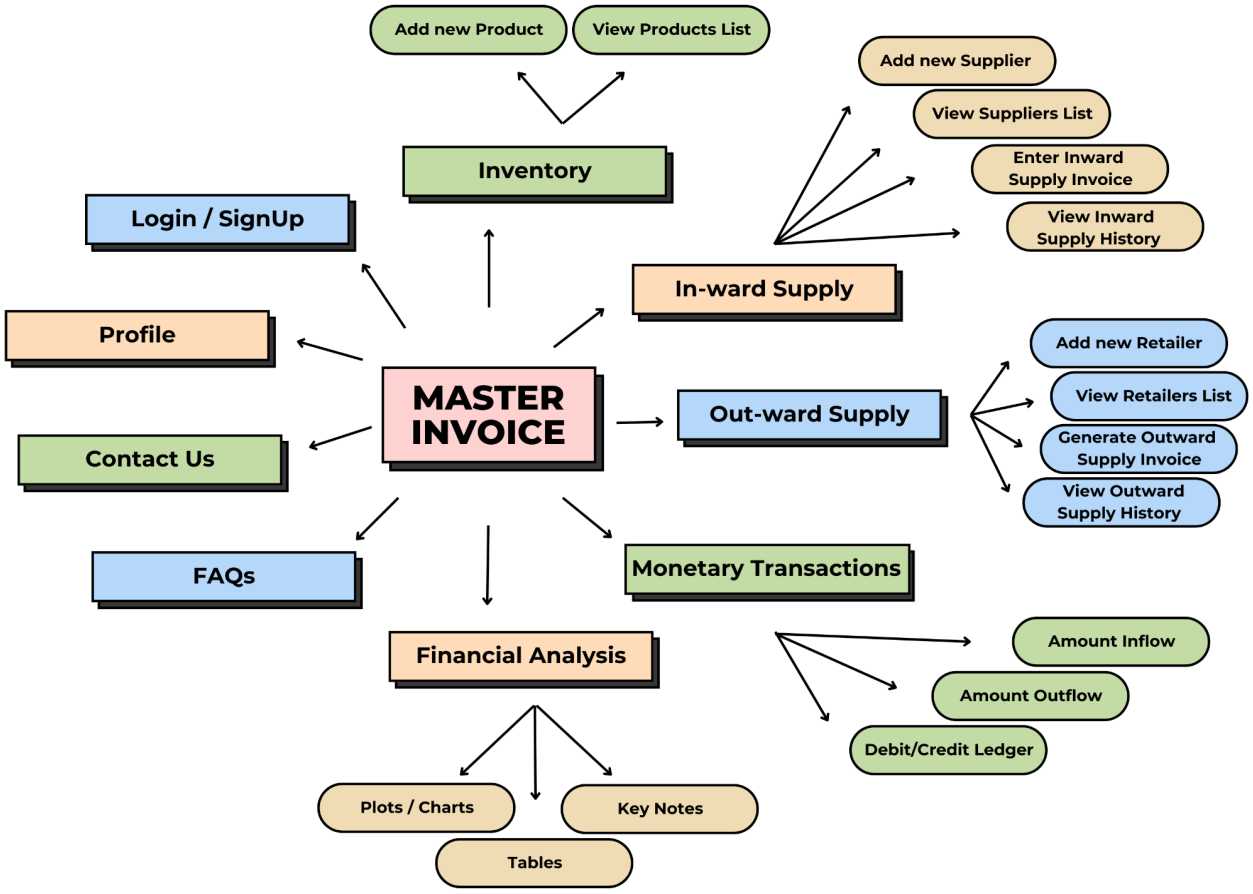
1.5 References and Acknowledgments

1. We utilized Lucid Charts to craft visually compelling images, and flowcharts, effectively translating our ideas into a concise and impactful pictorial representation.
2. Canva was used for the graphical representation of the overall product high-level summary description.
3. We referred to Busy accounting software for references in a few functionalities.

2 Overall Description

2.1 Product Overview

Master Invoice is a self-contained software solution designed to modernize and streamline supply chain operations for intermediaries managing the movement of goods effectively between factories and stores. It replaces outdated manual methods with a centralized digital platform, addressing key challenges such as inventory tracking, invoice management, financial transactions, and market analysis. The software operates independently but uses a database for efficient data storage and retrieval. Acting as the central hub, Master Invoice connects factories for procurement, and retailers for distribution, and integrates seamlessly with external systems, empowering intermediaries to efficiently manage their critical role in the supply chain.



2.2 Product Functionality

This product is intended to act as a management system for supply chain intermediates. It includes the following functionalities:

2.2.1 Login/Sign Up:

A. Login
B. Sign up
   a. OTP generation and verification
   b. Set password
C. Password recovery
D. Change password

2.2.2 Inventory:

A. Add new product
B. View current Inventory
C. Modify product details

2.2.3 Inward Supply:

A. Add new supplier
B. View list of existing suppliers
C. Enter inward supply invoice bill
D. View inward supply invoice bills

2.2.4 Outward Supply :

A. Add new retailer
B. View list of retailers
C. Generate bill to retailer(Invoice Bill, Way Bill,..etc)
D. View outward supply invoice bills

2.2.5 Transactions :

A. Add a payment inflow or payment outflow
B. Transaction history
C. Pending transactions

2.2.6 Analysis :

A. Graphs- Sales amount over time, profits over time, Bills over time
B. Tables - Top products with highest sales, top retailers making highest sales
C. Statistics - Total retailers, total profits, total number of products, total Suppliers, total sales, total bills generated

2.3 Design and Implementation Constraints

- The system must efficiently handle and store data related to inventory, user profiles, and transactions as the database grows.
- Real-time updates for inventory, transactions, and reports are crucial to ensure accurate information for users and administrators.
- The application will initially support only English for all user interfaces.
- The system must simultaneously handle multiple users performing tasks like inventory updates, invoice entries, and financial reporting during peak hours.

2.4 Assumptions and Dependencies

Assumptions:

1. All users of the software are assumed to possess an Email and Phone number.
2. It is assumed that a particular account will be used by a single person at a time (the warehouse would have a single counter).
3. It is assumed that the user should have a stable internet connection while using the software.

Dependencies:

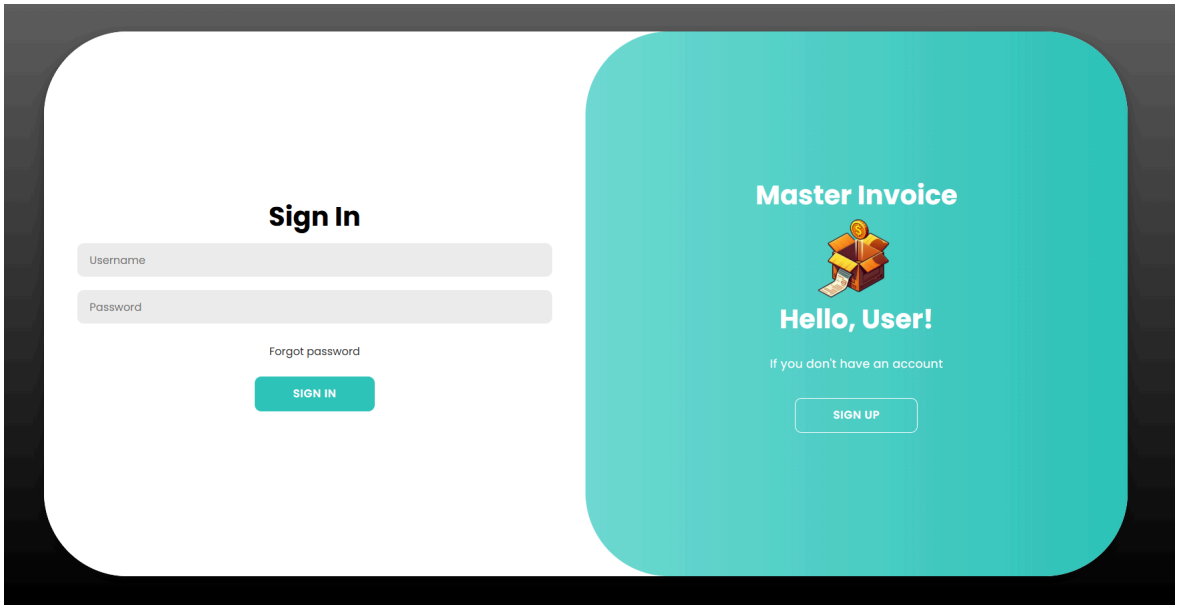
1. **HTML** and **CSS** serves as a critical component for the front end, providing the necessary tools for creating an interactive and dynamic user interface
2. The project relies on various **JavaScript** libraries to enhance functionality and streamline development.
3. The backend is dependent on **Django 4.x** and Python 3.10+.
4. **Django-crispy-forms** is required for rendering and managing form layouts. Crispy Forms is configured to work with **Bootstrap 5**.

3 Specific Requirements

3.1 External Interface Requirements

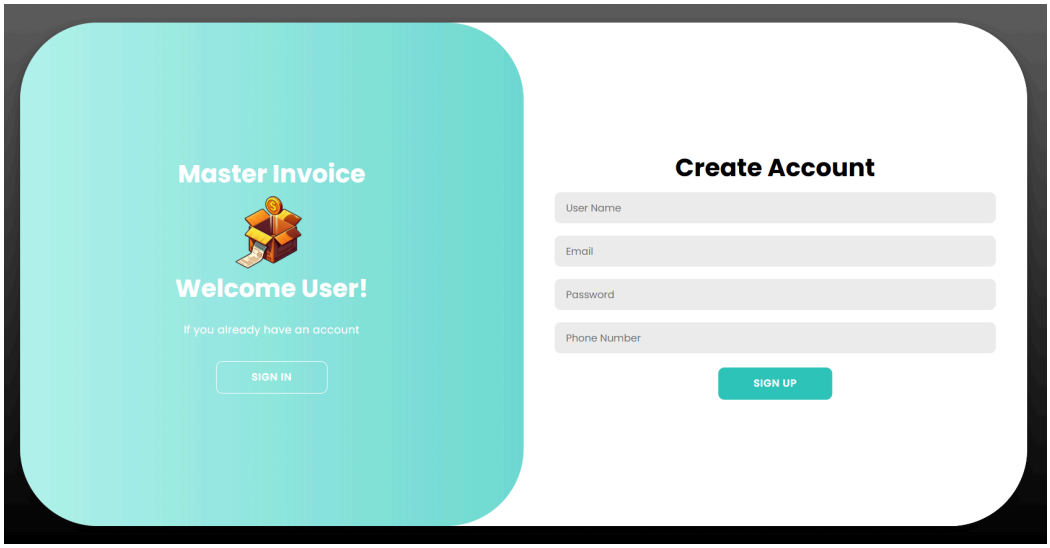
3.1.1 User Interfaces

1) Sign In / Sign Up page



The login window prompts users to enter their email or phone number along with their password. After entering these details into the designated text fields, users can click on the "Sign In" option at the bottom to access their account. In case a user forgets their password, they can utilize the "Forgot Password" option for updation of password via OTP verification.

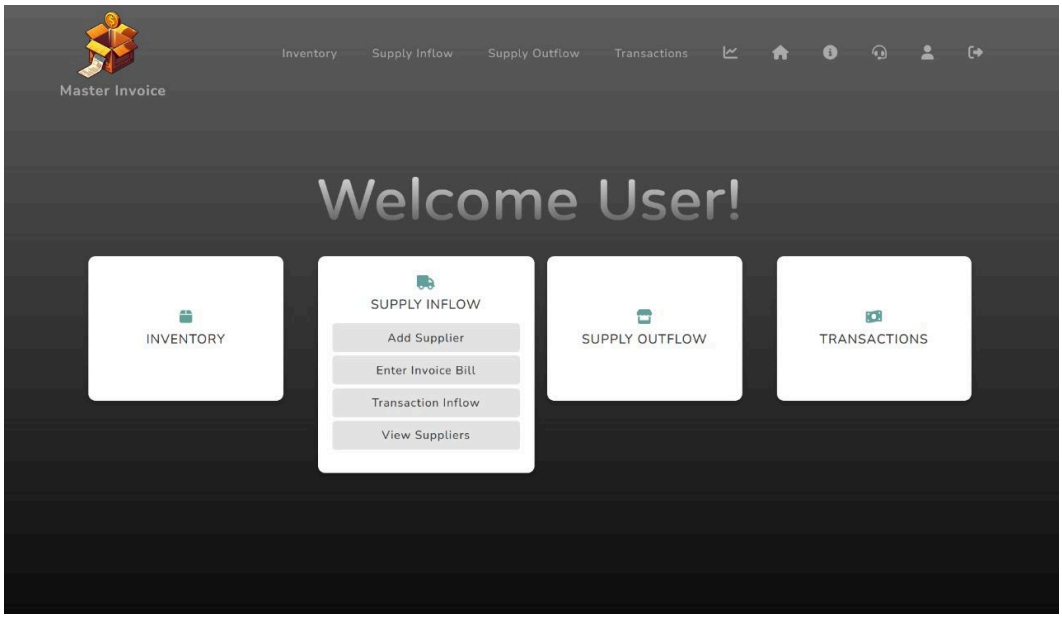
For new users, selecting the "Sign Up" option toggles to a window
where they can create a new account and set a password. The creation of a new account needs a unique username, valid Email, and mobile number. Following this, OTP verification for both email and phone number is conducted. Subsequently, users can create a password adhering to predefined rules and confirm it.



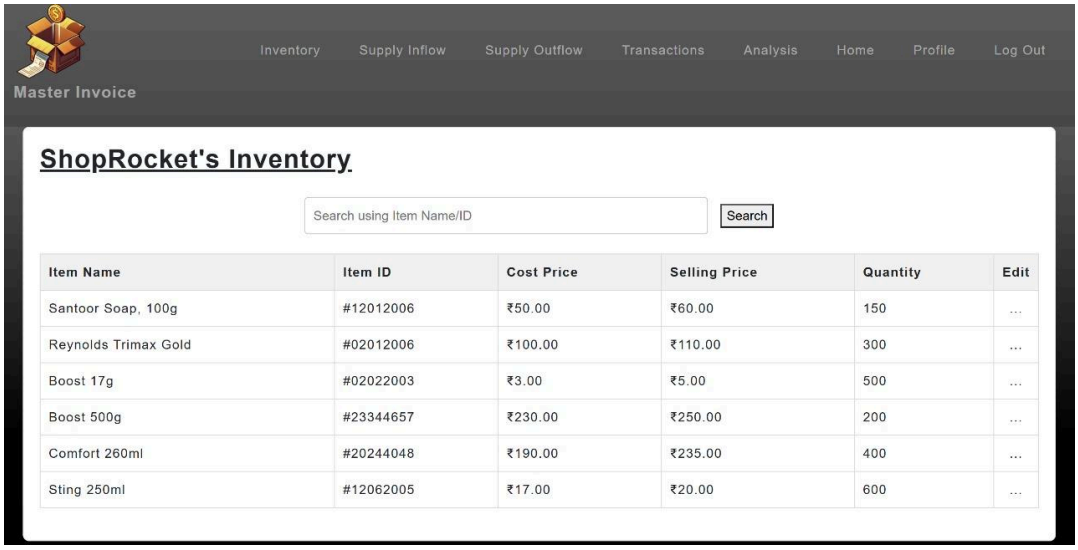
2) Home Page

Upon successfully signing into their account, the users will be directed to the Dashboard/Homepage. The dashboard acts like a directory for all functional aspects. It can direct to functions like **Inventory**(used to manage current stock)**, Inward Supply**(used to manage the inflow of products)**, Outward Supply**(used to manage the outflow of products)**, and Transactions**(used to manage monetary conditions) which hold their functionalities.

The navigation bar holds the above-mentioned functionalities as we move towards the right. There are more functionalities like **Analysis (**which can be used to understand trends)**, Contact Us (a** way to contact the technical team in case of any issue)**, Profile**(to view and edit user details)**, and Logout.**



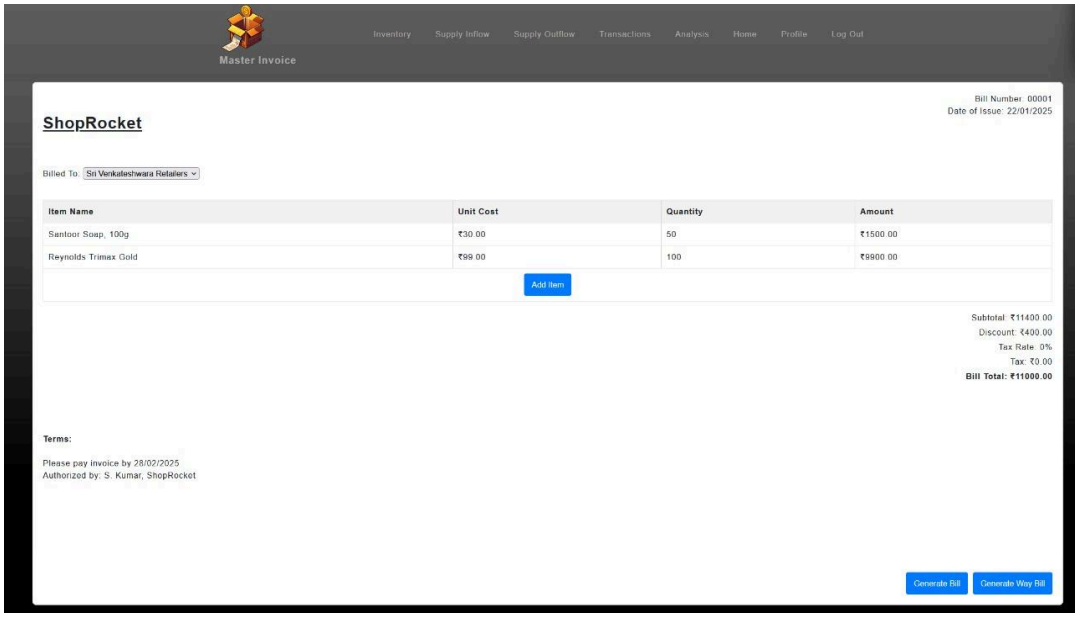
3) Inventory



Inventory acts as a representation of the whole currently available products with the user. Inventory makes the user's understanding of currently available products much better by representing things in a much more simple and neat representation.

Every product consists of details like **ID**(acts like a unique representation for a particular product), **Purchase price**(incoming (or) buying price), and **Sale price** (outgoing (or) selling price) with an **Edit** option that pops out to **update** in case of any changes with the attributes and **delete** in case of deleting the given product.

4) Invoice Generation



This page serves as the platform for **Generation of Bills**. The user begins by selecting the retailer in the designated box. An automatically generated **Invoice number/Bill number** will be displayed in the right top corner. Users have the flexibility to manually input items, specifying the quantity for each item. At the bottom left, a list of terms will be added (the generated bill will be added to Pending Transactions). It's important to note that, for the initial sale with a retailer, a new retailer profile must be established in the records before the generation of the bill. Once all items are added, users can click on the Generate Bill option to review the invoice before finalizing it. Then the bill will be generated in the appropriate format.

3.1.2 Hardware Interfaces

A thermal printer is required to print the Invoice/bill in physical format.

3.1.3 Software Interfaces

The user does not need any special software. Only a web browser and a stable internet connection are needed.

3.2 Functional Requirements

3.2.1 Login/Sign Up:

Sign up:

- The new user will be asked to enter details such as their Username, Email-ID, and Phone number and to set a valid password.
- They will be sent a verification mail or an OTP in case of registration using Email-ID and Phone number respectively.

Login:

- An existing user shall be able to log in using their Username/Email-ID and password.
- An option called ‘Forgot Password’ exists in case the user forgets the password. Authentication will be carried out through their registered Email-ID or phone number. Phone number authentication involves entering an OTP and email ID authentication involves a link that redirects the user to a page where they can reset their password.

3.2.2 Inventory:

- The user may view all the information related to the products that currently exist in their inventory. They may search using different filters and sort based on different parameters. Each product has attributes such as:
    - Serial Number
    - Name of Product
    - Item ID
    - Company Name
    - Quantity
    - Cost Price
    - Selling Price
    - MRP etc.

- **Add Product:**
  The user will be required to fill in data (such as the attributes listed above) related to the product they are adding. The default value of the 'quantity' attribute will be set to 0, this gets updated whenever the user adds an invoice bill with this product. While entering the 'company' attribute, a drop-down menu of already existing companies is displayed. If the user is adding a product from a new company, they may simply enter the name and the system stores it as a new company.

- **Modify Product:**
  The user may change product attributes when required.

3.2.3 Inward Supply:

The user will be able to sort out the management of their inward supply flow of products from various suppliers seamlessly.

- **Add Supplier:**
  The user can add a new supplier by filling in their appropriate details (the above details are required)

- **View Suppliers:**
  The user can view the list of suppliers with the below details:
    - Person Name
    - Firm Name
    - Contact details (Phone number, Email-ID)
    - Address
    - Debt
    - Invoice Bill History

- **Add Supplier Invoice Bill:**

The user will be able to add an invoice bill generated by the supplier using which the system updates the user's inventory and finances instantaneously. The user will manually enter the invoice bill with the below details.

- Bill Number (given by supplier)
- Bill ID (indexing for the user)
- Details of products (product, quantity, cost, etc.)

- **View Supplier Invoice Bills History:**
  The user will be able to view all the invoice bills added from suppliers. The user can sort the bills with various filters (supplier, date, bill number) and they can sort bills with various quantities (debit, date).

3.2.4 Outward Supply

The user will be able to sort out the management of their outward supply flow of products for various retailers seamlessly.

- **View Retailers:**
  The user can view the list of retailers with the below details:
  - Person Name
  - Firm Name
  - Contact details (mobile data, email ID)
  - Address
  - Credit
  - Invoice Bill History

- **Add Retailer**:
  The user can add a new retailer by filling in their appropriate details (the above details are required).

- **Generate Bills to Retailer:**
  The user will be able to generate various required bills in a ready-to-use format.
  - Invoice Bill
  - Way Bill (used for transportation purposes)

  Invoice bill generated for the retailer using which system updates the user's inventory and finances instantaneously. The user will manually enter the invoice bill with the below details.
  - Bill Number (given by supplier)
  - Bill ID (indexing for the user)
  - Details of products (product, quantity, cost, etc.)

- **View Retailer Invoice Bills History:**

The user will be able to view all the invoice bills added by the user. The user can sort the bills with various filters (retailer, date, bill number) and they can sort bills with various quantities (credit, date).

3.2.5 Transactions

- **Add Transaction:**
  - The transactions may be of 2 types: **Payment Inflow** and **Payment Outflow**.
  - The user may add a transaction when it is completed. Each added transaction requires the user to fill in data for attributes like ‘Supplier/Retailer’ and ‘Amount’. The system also records the date on which the transaction has been added and adds it to the ‘Date’ attribute.
  - Once a transaction is added, the pending transaction to that supplier/retailer is automatically updated in the Credit/Debit tables.

- **Pending Transactions:**
  - The user may view a list of all the pending transactions to suppliers/retailers, search using different filters, and sort based on different parameters. Each pending transaction has two attributes: ‘Name of Supplier/Retailer’ and ‘Amount’.
  - The transactions are divided into two tables: ‘Credit’ (represented by red) and ‘Debit’ (represented by green). The ‘Credit’ table displays all the transactions in which the user must pay the supplier/retailer and the ‘Debit’ table displays all the transactions where the user is owed by the supplier/retailer.
  - The pending transactions are updated automatically by the system whenever the user adds invoice data and are modified based on added transactions.

- **View Transaction History:**
  The user may view their transaction history and sort, filter, and search through them.

3.2.6 Analysis

The user can analyze all of his product exchanges and their transactions using various data representation methods(like graphs etc.)

- **Graphical representation:**
  - The user can analyze trends in business in a specific selected period over various useful variable choices.
- **Tabular representation:**
  - The user can analyze the whole data corresponding to products and overall transactions happening.

- **Statistical representation:**
    - The user can analyze data corresponding to a particular product supplier or retailer in a very simple yet effective way of representation.

  Analysis of data makes the user understand trends about how certain products are performing and their role in business and it may help him make new business choices more effectively based on his previous products' performance.

3.2.7 Profile

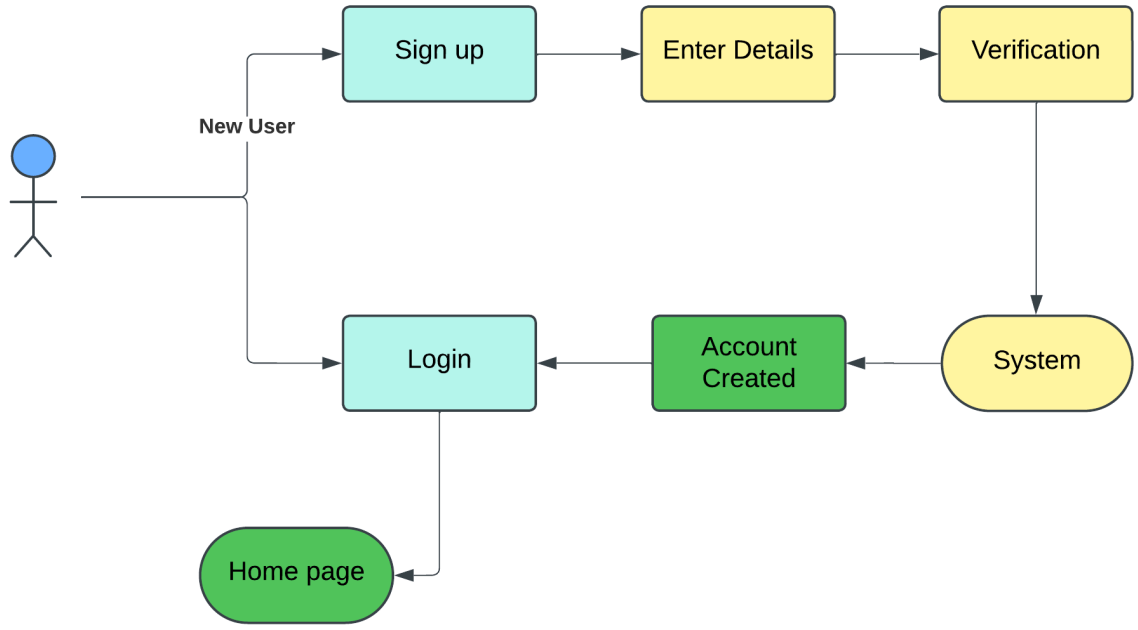
- The user can edit the details in the profile to maintain an up-to-date status with his current status.

3.2.8 Contact Us

- In case of any issue with the application, the user can contact the technical team using contact us.

3.3 Use Case Model

3.3.1 Use Case #1 (Signup/Login)



Author – Team while(1)

Purpose - For users to sign up/register or log in.

Requirements Traceability – Name, a valid Email-ID, and a password setup for sign-up; email ID and password for login.

Priority - High.

Preconditions - Valid email for sign-up.Valid email and password for login.

Postconditions -

For Sign up: A user account is created and the application can be used.

For Login: The user is logged in and can access the dashboard and application.

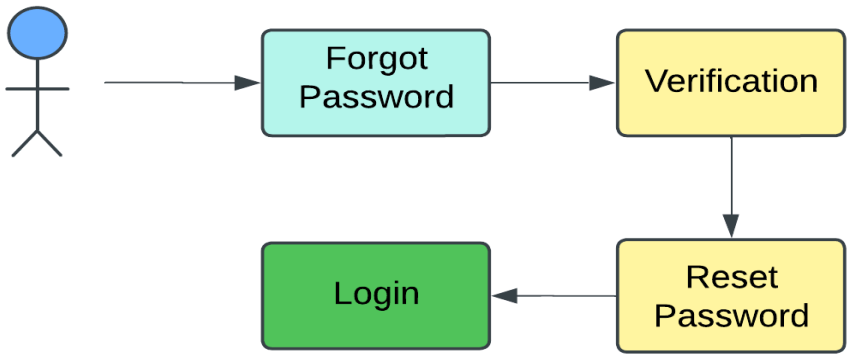
Actors – New User (for Sign up), Registered User (for login).

Exceptions -

For Sign up: Email/phone number must be valid and OTP must be verified within the given time.

For Login: Email/phone number and password must be valid.

3.3.2 Use Case #2 (Forgot Password)



Author – Team while(1)

Purpose - To reset the password if the user has forgotten the password.

Requirements Traceability – Registered Email-ID.

Priority - Low.

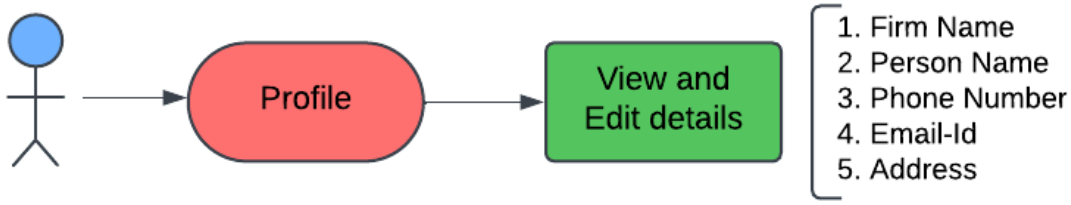
Preconditions - Valid Email-ID of the registered user.

Postconditions - After resetting the password, the user can log in and use the software.

Actors – Humans and systems.

Exceptions - Entered Email-ID should be valid.

3.3.3 Use Case #3 (Profile)



Author – Team while(1)

Purpose - To view the profile of a user.

Requirements Traceability – To store all the details of a user.

Priority - Low.

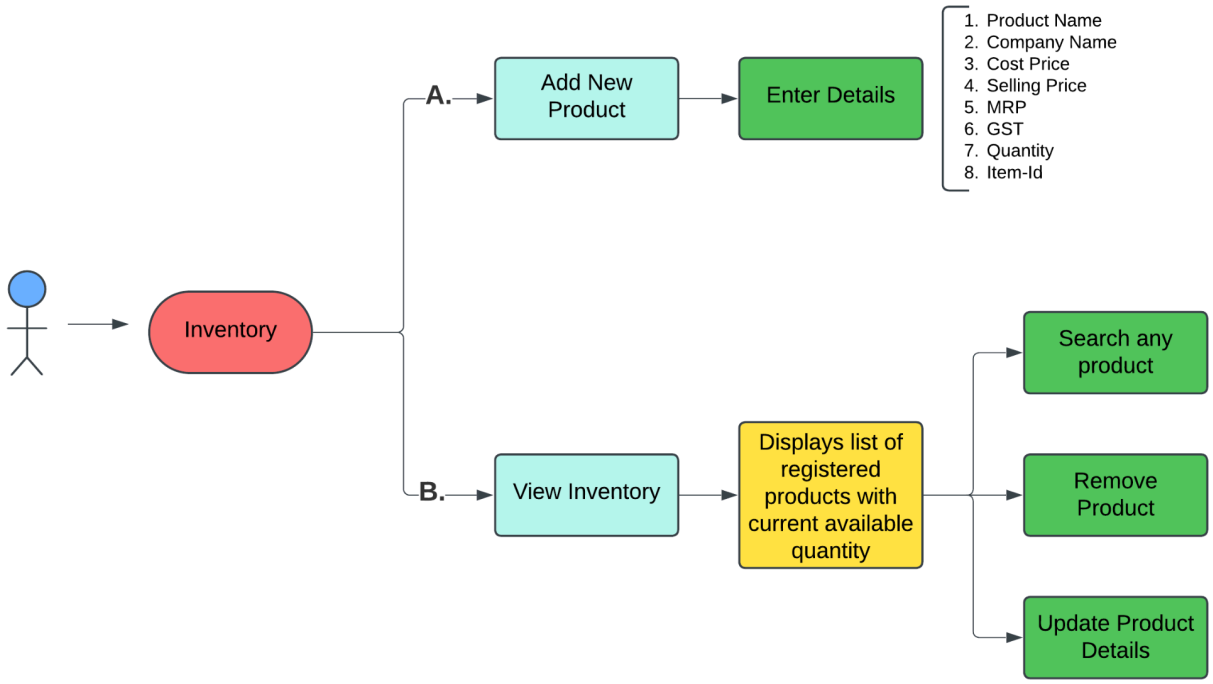
Preconditions - Users should have a profile created before.

Postconditions - The user can view the profile.

Actors – Humans.

Exceptions - None.

3.3.4 Use Case #4 (Inventory)



Author – Team while(1)

Purpose - To add new products and check the existing products in inventory.

Requirements Traceability – Edit and view inventory.

Priority - High.

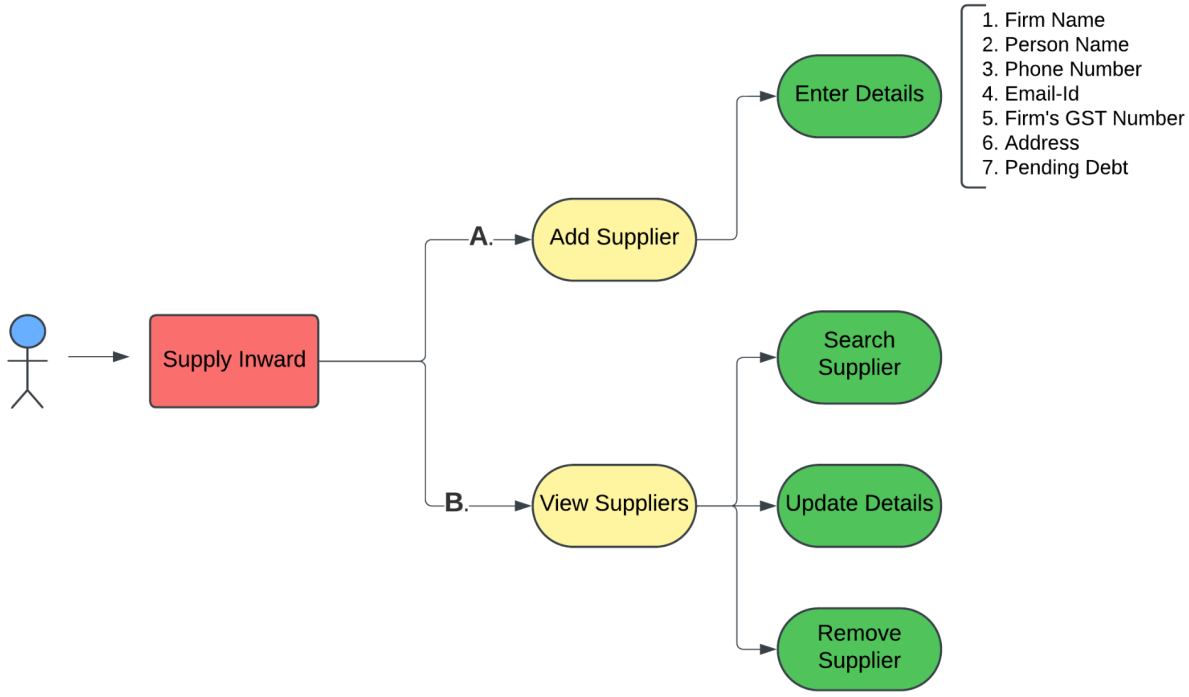
Preconditions - The user must have the necessary access privileges to manage inventory.

Postconditions - The products are successfully added or updated in the inventory.

Actors – Humans.

Exceptions - The numerical value of the quantity of products is a positive integer.

3.3.5 Use Case #5 (Suppliers)



Author – Team while(1)

Purpose - To add or remove suppliers and view suppliers.

Requirements Traceability – None.

Priority - High.

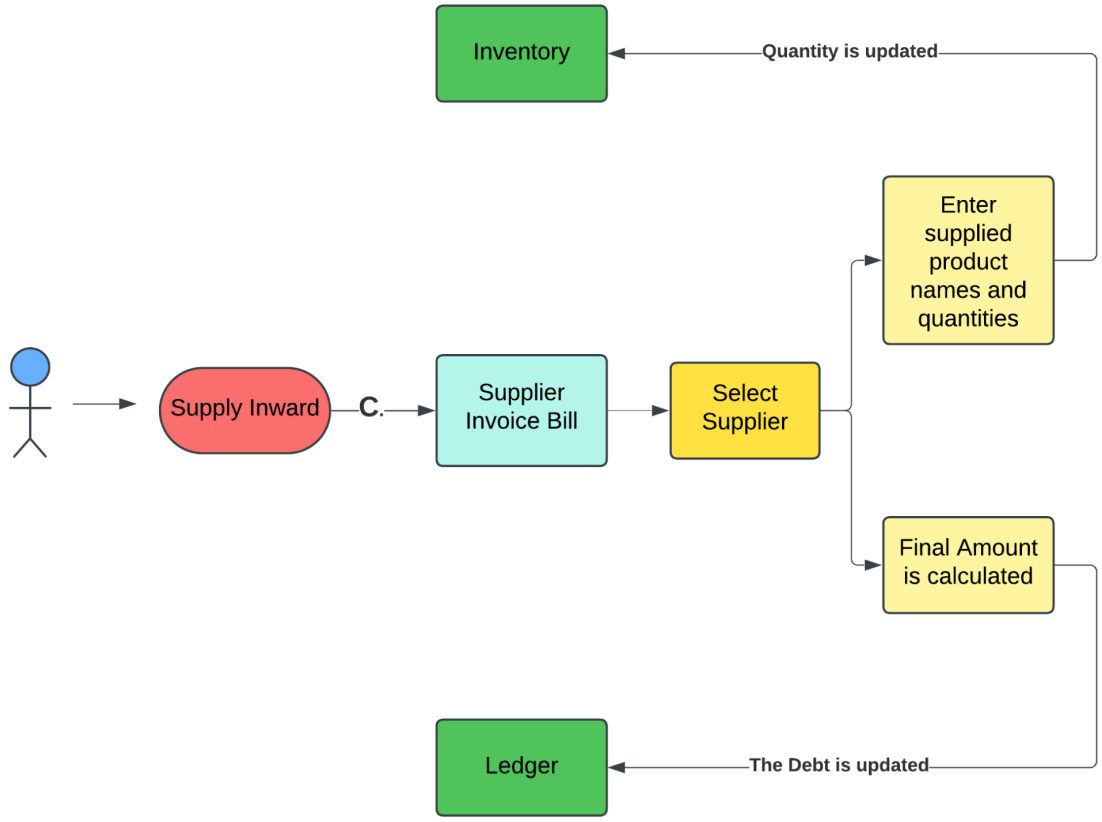
Preconditions - None.

Postconditions - Users can view suppliers' profiles or suppliers are added successfully.

Actors – Humans.

Exceptions - The user should enter the supplier's details in the past.

3.3.6 Use Case #6 (Supplier's Invoice Bill)



Author – Team while(1)

Purpose - To automate updating the quantity of the products in the inventory and update debt.

Requirements Traceability – The data should be entered correctly by the user.

Priority - High.

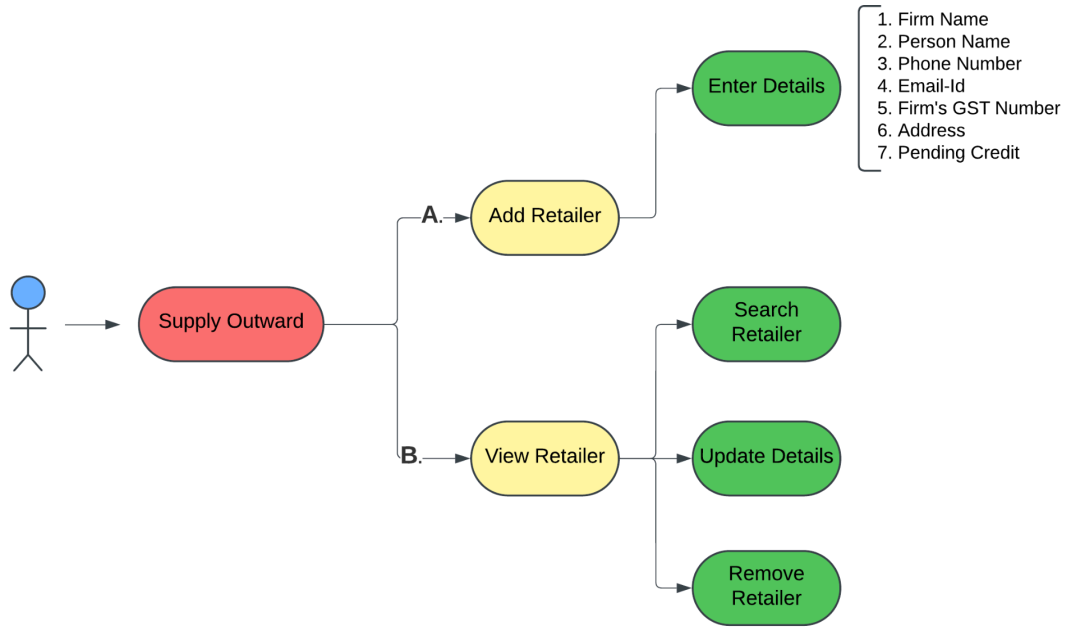
Preconditions - Provision of goods by the supplier.

Postconditions - The final amount is calculated and quantity and debt are automatically updated.

Actors – Humans and systems.

Exceptions - In case of a supply of a new product, we should first add the product to the inventory.

3.3.7 Use Case #7 (Retailers)



Author – Team while(1)

Purpose - To add or remove retailers and view retailers.

Requirements Traceability – None.

Priority - High.

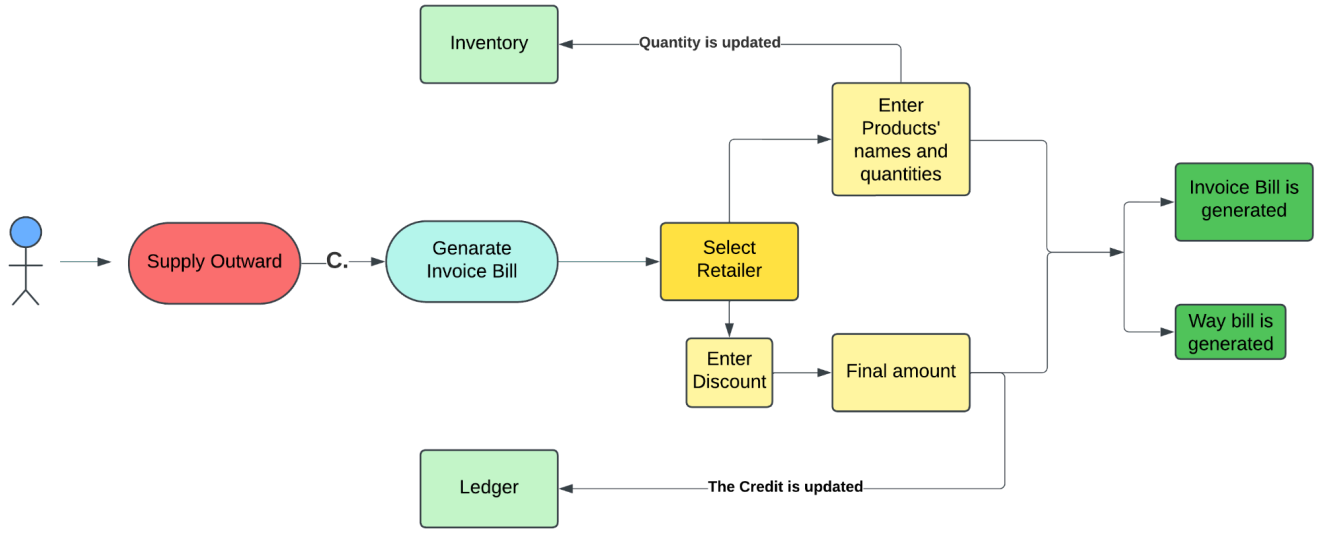
Preconditions - None.

Postconditions - Users can view retailers' profiles or retailers are added successfully.

Actors – Humans.

Exceptions - The user should enter the retailer's details in the past.

3.3.8 Use Case #8(Retailer's Invoice Bill and generating Way-bill)



Author – Team while(1)

Purpose - To automate the process of updating the quantity of the products in the inventory, updating the pending transactions, and generating the bills.

Requirements Traceability – The data should be entered correctly by the user.

Priority - High.

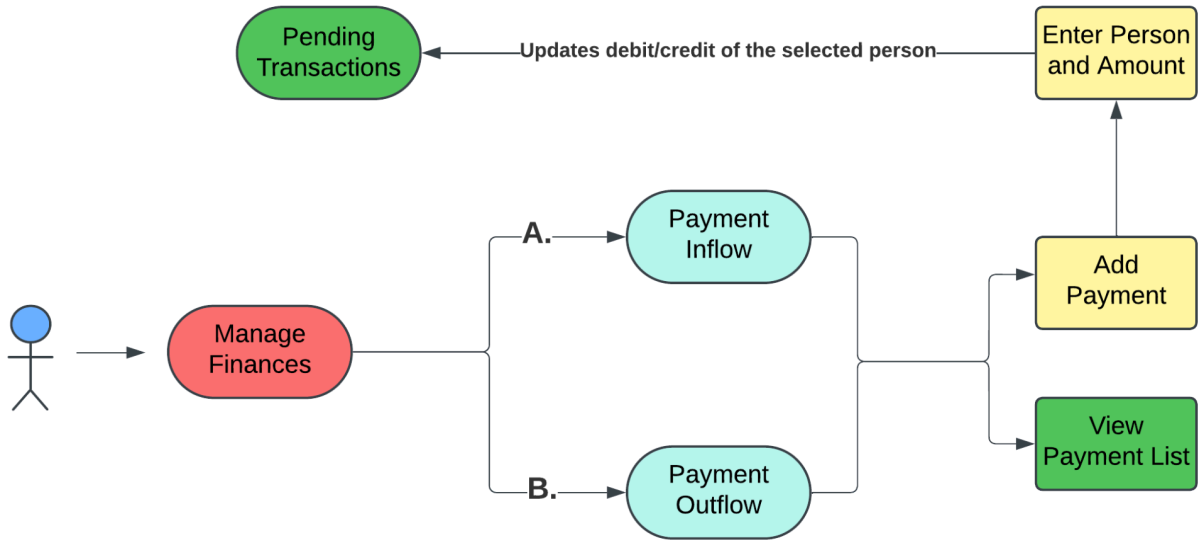
Preconditions - When products are supplied to the retailers.

Postconditions - The quantity is automatically updated in the inventory and a bill is generated.

Actors – Humans and systems.

Exceptions - Cases where retailers did not receive products due to transportation issues are not being considered.

3.3.9 Use Case #9 (Transactions)



Author – Team while(1)

Purpose - To view or store the payments.

Requirements Traceability – To store all the payments and receipts.

Priority - High.

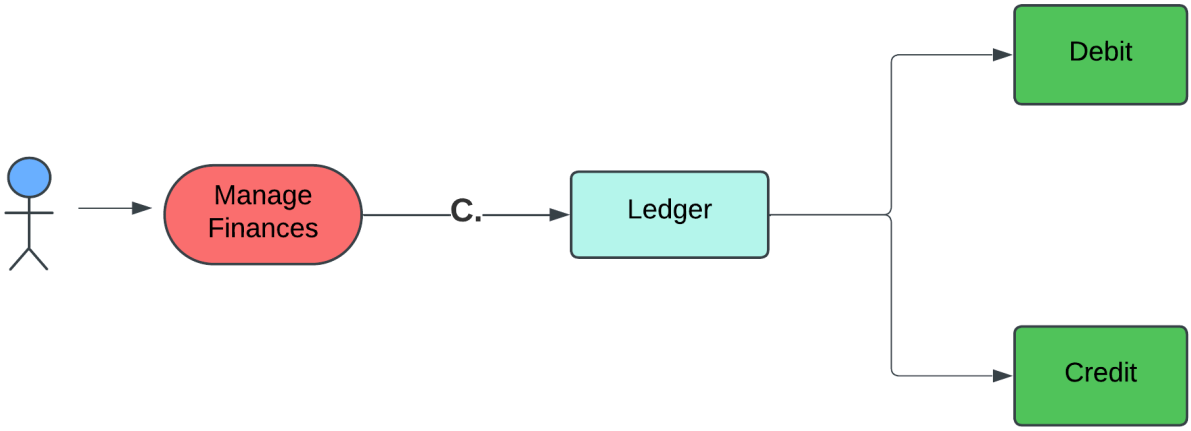
Preconditions - Whenever we make or receive a payment.

Postconditions - All the payments are stored and the debt is updated in the pending transactions.

Actors – Humans

Exceptions - The amount entered.

3.3.10 Use Case #10 (Pending transactions)



Author – Team while(1)

Purpose - To view and update debit or credit transactions associated with a supplier or retailer.

Requirements Traceability – To maintain all the pending transactions.

Priority - High.

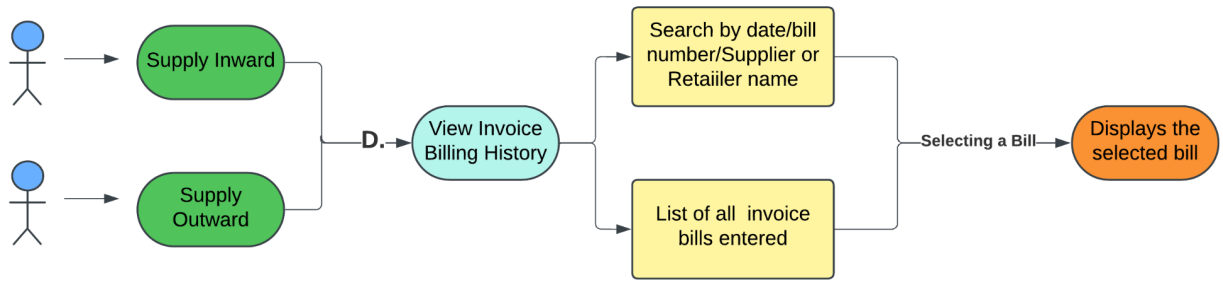
Preconditions - Whenever the user pays or receives money can check all the pending transactions.

Postconditions - The pending payments are successfully updated.

Actors – Humans

Exceptions - If the user forgets to update the payment received or paid.

3.3.11 Use Case #11 (Invoice bills history)



Author – Team while(1)

Purpose - To check all the past transactions (inward and outward) with multiple filtering and sorting options.

Requirements Traceability – To maintain all the **Bill Numbers** and store all the quantity of products in that transaction bill.

Priority - High.

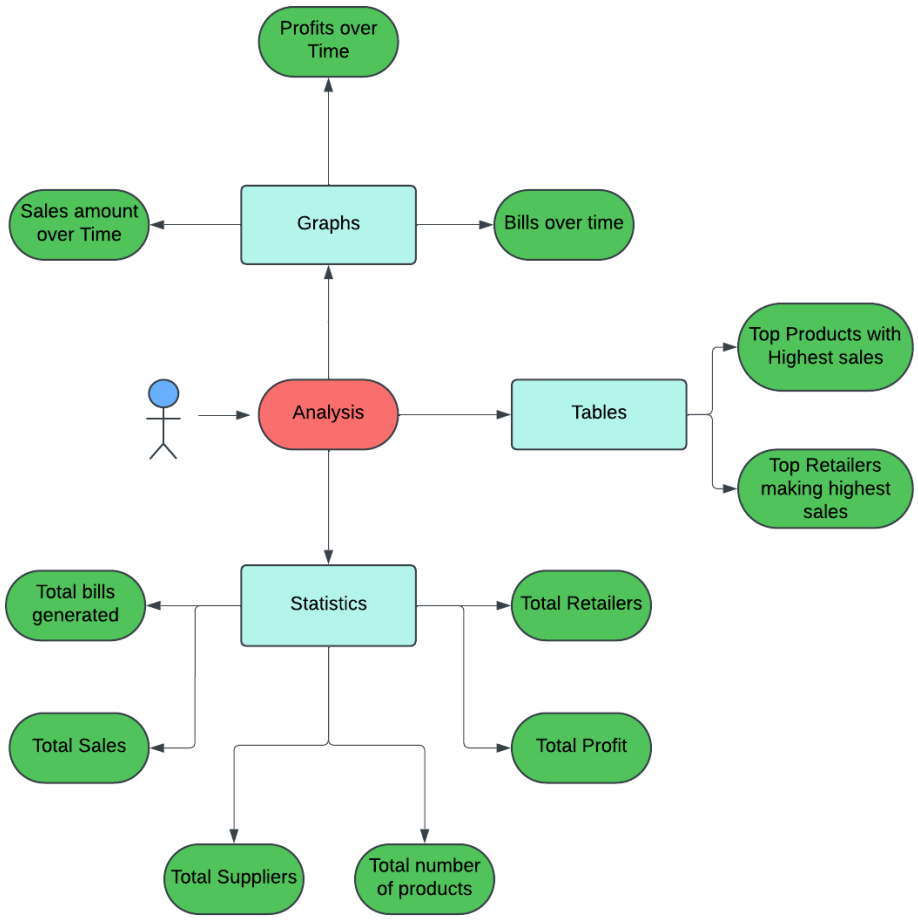
Preconditions - To cross-check past transactions and to check a particular bill.

Postconditions - The user can view all past transactions (both inward and outward) and corresponding bills.

Actors – Humans

Exceptions - If the user forgets to create the bill on the website.

3.3.12 Use Case #12 (Analysis)



Author – Team while(1)

Purpose - To view all the analysis which includes profit, sales, top products, total retailers, total suppliers, etc. within the selected time range.

Requirements Traceability – To store various attributes like time, transaction history, etc.

Priority - High.

Preconditions - The user should update their bills (both inward and outward) up to date.

Postconditions - The user can see various effective ways of data representation.

Actors – Humans

4 Other Non-functional Requirements

4.1 Performance Requirements

Latency

- For web pages, the system's response time should not exceed 100ms to ensure a fast and smooth user experience. Minimizing latency is crucial to avoid user frustration, maintain engagement, and optimize system performance.
- When a user is viewing their inventory, which could span across multiple pages, the application should ensure that loading additional data from the database doesn't result in any noticeable performance degradation. This means the system should handle pagination or infinite scrolling efficiently, without negatively impacting the user experience as more data is requested.

Throughput

- The system should be able to handle up to 1000 concurrent users without any significant performance degradation. It must remain stable and capable of managing a high volume of API calls, ensuring continuous operation and providing a reliable, smooth user experience even under heavy load.

Scalability

- The system should be designed to support both horizontal and vertical scaling, allowing it to efficiently handle increased load. This ensures flexibility and scalability to meet growing demands.

4.2 Safety and Security Requirements

User Data Protection

- User data and personal information must be protected from any security flaws in the software that may have been inadvertently introduced. Strong safeguards should be implemented to prevent unauthorized access.
- The administration should delete users' data upon their request.
- The system should implement backup and disaster recovery plans.
- Cookies must be securely stored and transmitted to prevent unauthorized access or tampering.
- User history should be retained for a minimum of 5 years as this will be more important in analyzing Transactions.

User Authentication

- User authentication should be carried out using either the user's mobile number or email address as the primary identifier. To ensure secure and reliable verification, an OTP (One-Time Password) should be sent via SMS or email. Additionally, users must be required to create a strong password with a minimum of **8** characters, following best practices for password-hashing, to further secure their accounts. The entire authentication process should be seamless and efficient, providing both an added layer of security and a user-friendly experience. All authentication methods must comply with industry-standard encryption protocols to protect user information.
- The user is responsible for keeping his password secret. If the user forgets his password then there should be an OTP sent to both the mobile number and email where it is the user's responsibility to keep his OTP secured and OTP should expire after 5 min.

Transaction Information

- Payment receipts should be accessible only to authenticated users. Unauthorized users should not be able to view payment details or receipts.

4.3 Software Quality Attributes

Usability

- Users should find it easy to update their profiles, manage transaction details with vendors or partners, and efficiently connect with product distributors to streamline the supply process.
- The Search feature should be capable of handling partial matches, auto-suggestions, and advanced filtering to help users quickly find the products they are looking for

Reliability

- The system must operate consistently without errors, ensuring seamless performance under all conditions. It should be resilient to failures and capable of recovering quickly from any issues that arise, providing users with a stable and dependable experience at all times.

Portability

- The application should be portable across different devices with consistent functionality and user experience.
- The front end of the website is designed using Bootstrap, ensuring a responsive layout that adapts seamlessly to various screen sizes and devices.

Maintainability

- The codebase should be structured modularly, ensuring that components are loosely coupled and easily modified without affecting other parts of the system.

5 Other Requirements

5.1 Authentication

Permission will be required for utilizing OTPs for the authentication process as this will require sending automated emails to the user's Email-Id.

Appendix A – Data Dictionary

A.1 User Class

Variable Name	Variable Type	Description	Example
Name	string	Stores the user's name entered during registration.	Tony
Email	string	Stores the user's email address	abc@gmail.com
Phone	string	Stores the user’s phone number	9999999999
Password	string	Stores the password set by the user.	xyz#Ab123
User ID	string	System-generated unique identifier for each user.	abc123ABC

A.2 Inventory Class

Variable Name	Variable Type	Description	Example
Product Name	string	The name of the product.	Amul Milk 250ML
Product ID	string	System-generated unique ID for each product.	123abc
Company Name	string	Name of the manufacturer or supplier of the product.	Amul
Category	string	The category to which the item belongs.	Dairy products
Stock	Positive integer	Stores the available stock of the product.	100

Purchase Price(₹)	Positive integer	Cost at which the product was purchased.	40
Sale Price(₹)	Positive integer	Price at which the product is sold to retailers.	50
MRP	Positive integer	Maximum Retail Price allowed for the product.	65
Batch Expiry	date	The date on which the items of a particular batch expire	12/01/2025

A.3 Supplier/Retailer Class

Variable Name	Variable Type	Description	Example
Supplier Name	string	Name of the supplier	XYZ Suppliers
Supplier ID	string	System-generated unique ID for each supplier	SUP123
Retailer Name	string	Name of the retailer	ABC Retailers
Retailer ID	string	System-generated unique ID for each retailer	RET456
Supplier Address	string	Address of supplier	Street XYZ
Retailer Address	string	Address of retailer	Street XYZ
Supplier no.	10-digit positive number	Mobile no. of supplier	9123456780
Retailer no.	10-digit positive number	Mobile no. of retailer	9123456780

A.4 Invoice Class

Variable Name	Variable Type	Description	Example
Price	Positive Real No.	Cost of each item	₹100
Amount	Positive Real No.	Purchase on each item	₹300
Discount	Positive Real No.	Discount to an item	₹45
Total Amount	Positive Real No.	Total amount purchased on all items	₹500
Quantity	Positive Integer	No items purchased	4
Taxes	Positive Real No.	Taxes	₹35
Customer Name	string	Name of the customer	Tony

A.5 Transaction Class

Variable Name	Variable Type	Description	Example
Sender name	string	Name of user who sent money	John
Receiver name	string	Name of who has done the transaction	Tony
Transaction amount	Positive real number	Amount sent	₹4000
Transaction date	date-time	Date on which transaction occurred	24-01-2025

Appendix B – Group Log

SL. No.	Date	Timings	Venue	Description
1	11/01/2025	10:00 pm - 11:30 pm	Google meet	Brainstormed various possible prospective ideas for the project.
2	14/01/2025	5:00 pm - 6:30 pm	RM building	Finalized the idea for the project and discussed various aspects of it.
3	16/01/2025	2:00 pm - 4:00 pm	RM Building	Studied the SRS template given and discussed various technical doubts related to our software.
4	17/01/2025	10:00 pm - 11:30 pm	Google meet	Clarified doubts regarding the Software Requirements Specification (SRS) and distributed the work amongst the team.
5	19/01/2025	12:00 pm - 4:00 pm	RM Building	Explored some more functionalities for the product and progressed with the SRS document
6	20/01/2025	10:00 pm - 12:00 pm	KD Library	Progressed with SRS documentation work.
7	21/01/2025	10:00 pm - 1:00 am	KD Library	Progressed with SRS documentation work.
8	22/01/2025	9:00 pm - 12:30 am	KD Library	Progressed with SRS documentation work.
9	23/01/2025	12:40 pm - 1:20 pm	KD Library	Redistributed pending work after clarifying doubts.
10	23/01/2025	10:30 pm - 11:00 pm	Google Meet	Discussed remaining doubs with TA and finalized the document.