Smart Steel

Final Year Project Report

Student: Zihan Huang
Student ID: 20108869

Supervisor: Richie Lyng

School of Engineering
South East Technological University

Table of Contents

1. Introduction

This system is a comprehensive solution aimed at small and micro businesses. The main purpose of developing it is to improve my parents' work environment by transitioning from handwritten invoices to computer-printed ones, with all data visualized. Therefore, the system must be simple and easy to use, as my parents' generation generally has a lower level of education, and they may not be able to handle complex operations, which could lead to abandoning the system. I hope this system can help people like them.

My parents' business involves stainless steel pipe wholesale and stainless steel fittings. Stainless steel comes in several types: 201, 304 and so on, with many different specifications. As they get older, they often make mistakes or forget the quantities of certain specifications. This system aims to address these issues and improve operational efficiency and reducing errors.

2. Similar Systems

2.1 Odoo

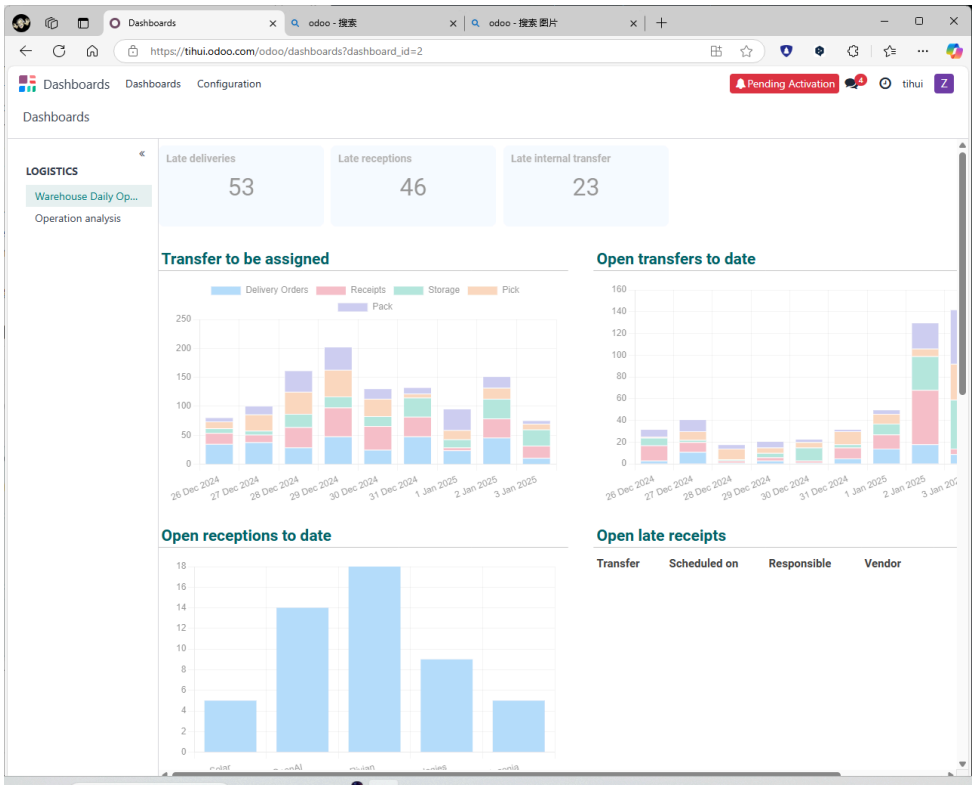

Figure 1: Odoo Dashboard Overview

Odoo is an open-source enterprise management software platform offering a suite of integrated applications to manage all aspects of your business, including customer management (CRM), finance, inventory, human resources, e-commerce, and project management. It has a modular design, and enterprises can select functions according to their actual needs. It also possesses open-source customizability and a user-friendly interface that is fit for enterprises of any scale to optimize operational productivity. [1]

2.2 Megaventory

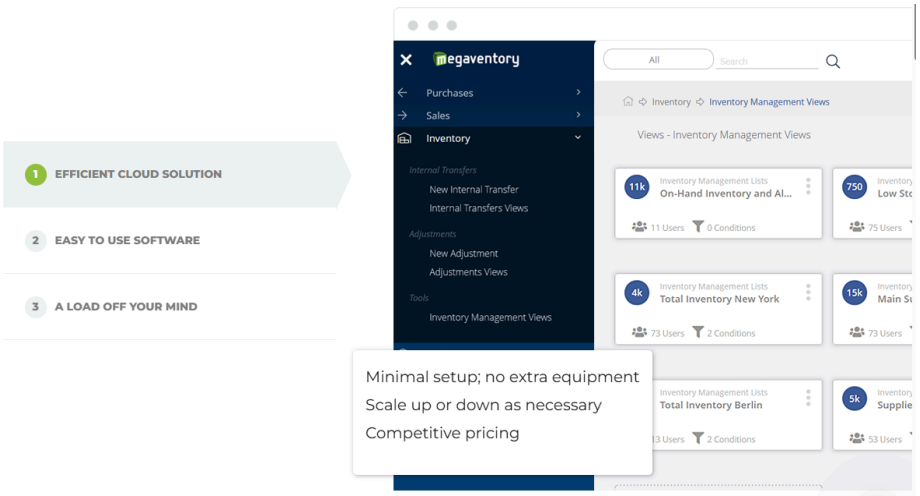

Figure 2: Megaventory Dashboard Overview

Megaventory is a cloud-based inventory and order management solution for small and medium-sized enterprises. It offers multi-warehouse inventory tracking, sales and purchase order processing, the ability to help with production planning, and real-time reporting capabilities while also supporting e-commerce and accounting tool integration. Wholesalers, retailers, and manufacturers use stamps and systems adaptability to manage challenging supply chains and improve operational processes in a simple user interface. [2]

3. Requirements of the system

3.1 Functional requirements

These are about what the system will do in some detail focusing on the system's features and behaviour.

1. User Authentication and Authorization

- Register/ Login and Logout of the user
- Users can be verified by some admin & they are given special permission
- Session management and API access secured using JWT

2. Customer Management

- Add new customers with information (name, phone, address and notes)
- Attaching the fuzzy search for your search customers
- Modifying customer information (deletion requires admin approval)

3. Order Tracking

- Track paid and unpaid orders for a specific customer

4. Order Management

- Create Orders (Purchase or Quotation)
- See all orders (paid, unpaid, incomplete and quotation order)
- Create distinct order numbers automatically based on the schema `YYYYMMDD+00`
- Mark as paid or fulfiled orders
- Export orders as PDFs or images

5. Inventory Management

- Add materials to the inventory

- Material stock levels and specifications updates
- Consume materials upon creating an order or update an existing one
- Load inventory from an Excel file
- Browse inventory by material classification and grade

6. Steel Price Management

- Automatically retrieve and show daily price values for stainless steel and carbon steel
- Keeping historical prices for later analysis

7. Employee Management

- Decline employees, and update their profile
- Update leave and overtime hours for employees
- Employee CRUD operations (View, Edit, Delete)

8. Reporting and Analytics

- Create and download as a PDF report for unpaid orders
- Process a monthly sale to appear as bar charts on the dashboard
- Display real-time information about outstanding sums (sum of unpaid orders)

9. Notifications

- Alert the users when inventory is low or not sufficient to place an order
- As a customer has overdue payments, notify the users

3.2 Non-functional requirements

1. Performance requirements

- Support a certain amount of concurrency (set according to the actual user scale) to ensure normal response under high concurrency.
- For batch operations such as order generation and inventory import, a faster response speed or a feasible background processing mechanism is required.

2. Security

- Login uses JWT Token verification to ensure communication security and reduce the risk of attack.
- Administrator permissions or key confirmation are required for sensitive operations (such as deleting orders, paying orders, importing inventory, registering new users, etc.).
- HTTPS is recommended for data transmission (especially login, order payment, etc.).

3. Availability

- The system interface is friendly and the operation process is simple, which is easy to get started quickly.
- Provide necessary error prompts, exception capture processing and log records to reduce the difficulty of use.

4. Maintainability

- Adopt modular design to develop and maintain login modules, order modules, inventory modules, customer modules, employee modules, etc. relatively independently.
- The data structure is clear and can be managed uniformly through database tables or documents.

5. Scalability

- Reserved interfaces are convenient for subsequent integration (such as docking with other financial systems, SMS reminders, third-party payments, etc.).
- The logic of quotation calculation and inventory update is configurable to meet the expansion requirements of more materials or more pricing methods in the future.

6. Fault tolerance

- There should be reasonable verification, prompts and fallback mechanisms for insufficient inventory, repeated customers, network failures, Excel import format errors, etc.

7. Logs and audits

- Key operations (such as order payment, order deletion, inventory modification, etc.) need to record operation logs to facilitate auditing and tracking problems.

8. Deployment and environment requirements

- The system can run in common server environments (Windows) and supports conventional databases (MySQL).
- It is recommended to use containerized deployment (Docker) to facilitate version management and horizontal expansion.

4. Database Design

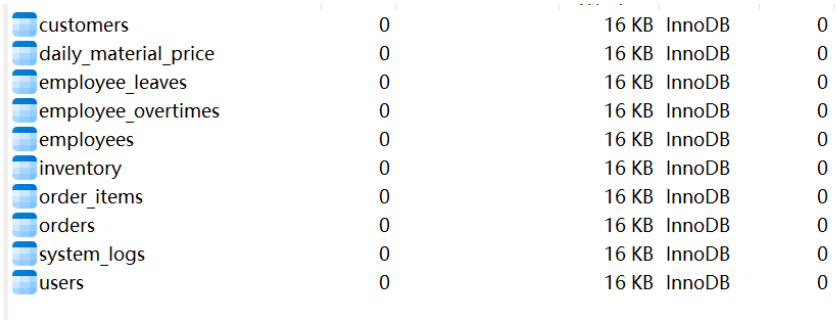

Figure 3: Database Design

4.1 Tables

In this system I will use the mysql DB to store the data. And below is the tales of the DB which are created according to the main feathers of the system.

- User and Permissions : `users`
- Steel Prices: `daily_material_price`
- Order Management : `orders`, `order_items`
- Customer Management: `customers`
- Inventory Management : `inventory`
- Employee Management: `employees`, `employee_leaves`, `employee_overtimes`

4.2 Modules Division

- User and Permissions

  Handel the process of login and register and administrator review

- Steel Prices

  Records of steel prices of different materials provide reference for quotation or ordering.

- Order Management

The order master table and order detail table are linked by foreign keys to perform one-to-many storage. Can distinguish between `sales orders (SALES)` and `quotation orders (QUOTE)`

- Customer Management

  Stores basic customer information and is associated with the order table through a foreign key.

- Inventory Management

  Maintain inventory information (material, specification, quantity, specific gravity, etc.), and deduct inventory accordingly after an order is generated.

- Employee Management

  Employee basic information form, as well as employee leave form and overtime form, are used for attendance management.

4.3 ER Diagram

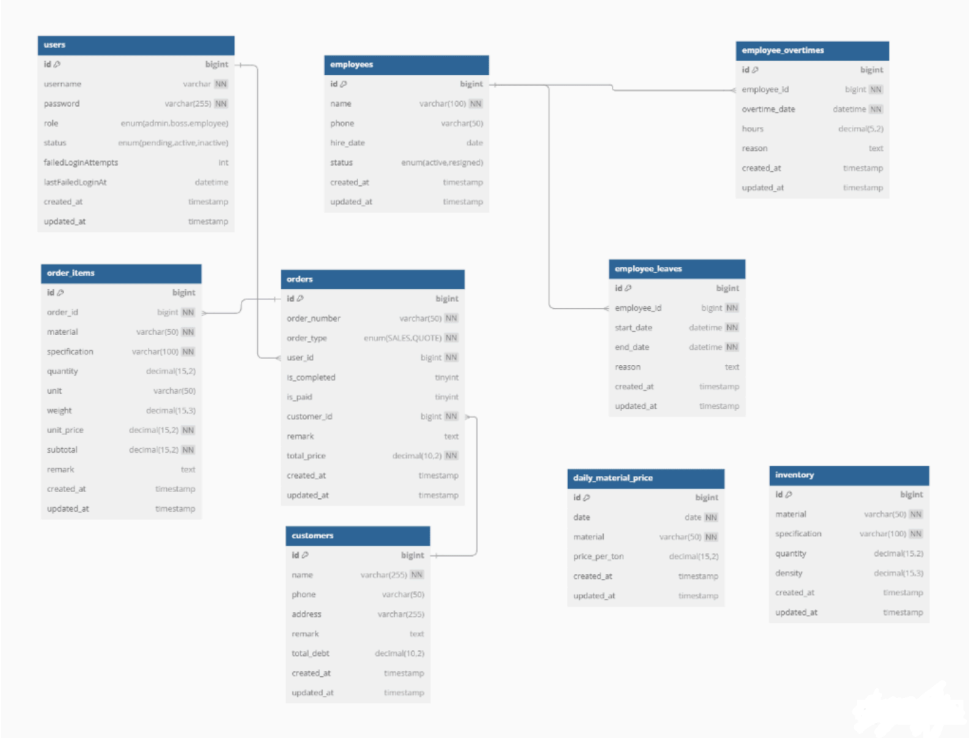

Figure 4: ER Diagram

1. User and System Logs

   One-to-Many: A user can generate multiple system logs (`system_logs.user_id` foreign key points to `users.id`).

2. Customer and Orders

   One-to-Many: A customer can have multiple orders, and each order corresponds to only one customer (`orders.customer_id` foreign key points to `customers.id`).

3. Order and Order Items

   One-to-Many: An order can contain multiple order details (`order_items.order_id` foreign key points to `orders.id`).

4. Employees and Leaves / Overtimes

One-to-Many: An employee can have multiple leave records or overtime records (`employee_leaves.employee_id`, `employee_overtimes.employee_id` foreign keys point to `employees.id` respectively).

5. Inventory

   Inventory is stored independently, and the outbound operations of different orders will be updated in the business layer.

6. Daily Material Price (Steel Daily Price)

   The prices of different materials and dates are stored independently. The unit price of the day can be queried based on the date and material, providing data support for order quotations or statistics.

5. UML Diagram

5.1 Use Case Diagram

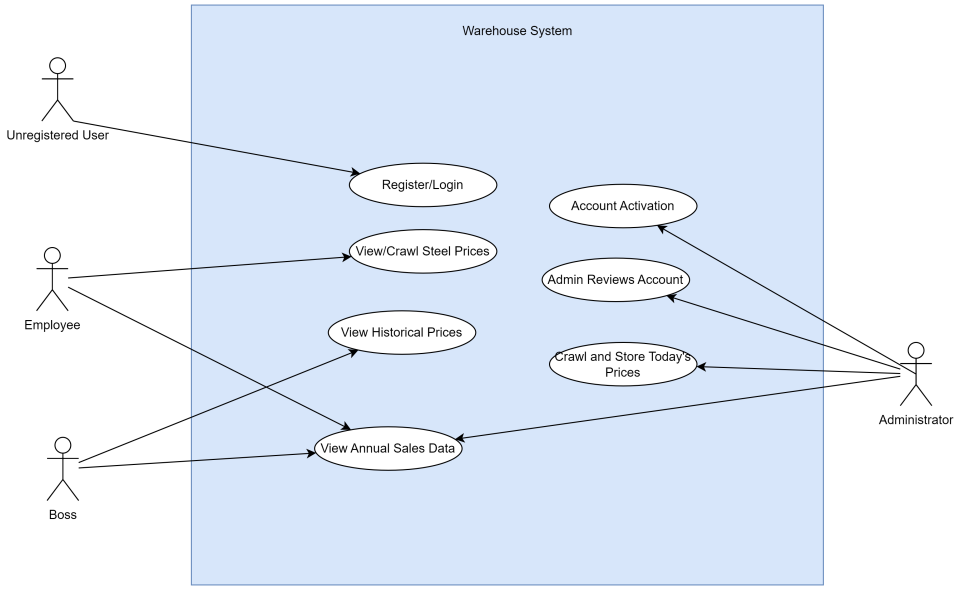

Figure 5: Use Case Diagram

5.2 Flowchart

1. User Registration & Administrator Review Process

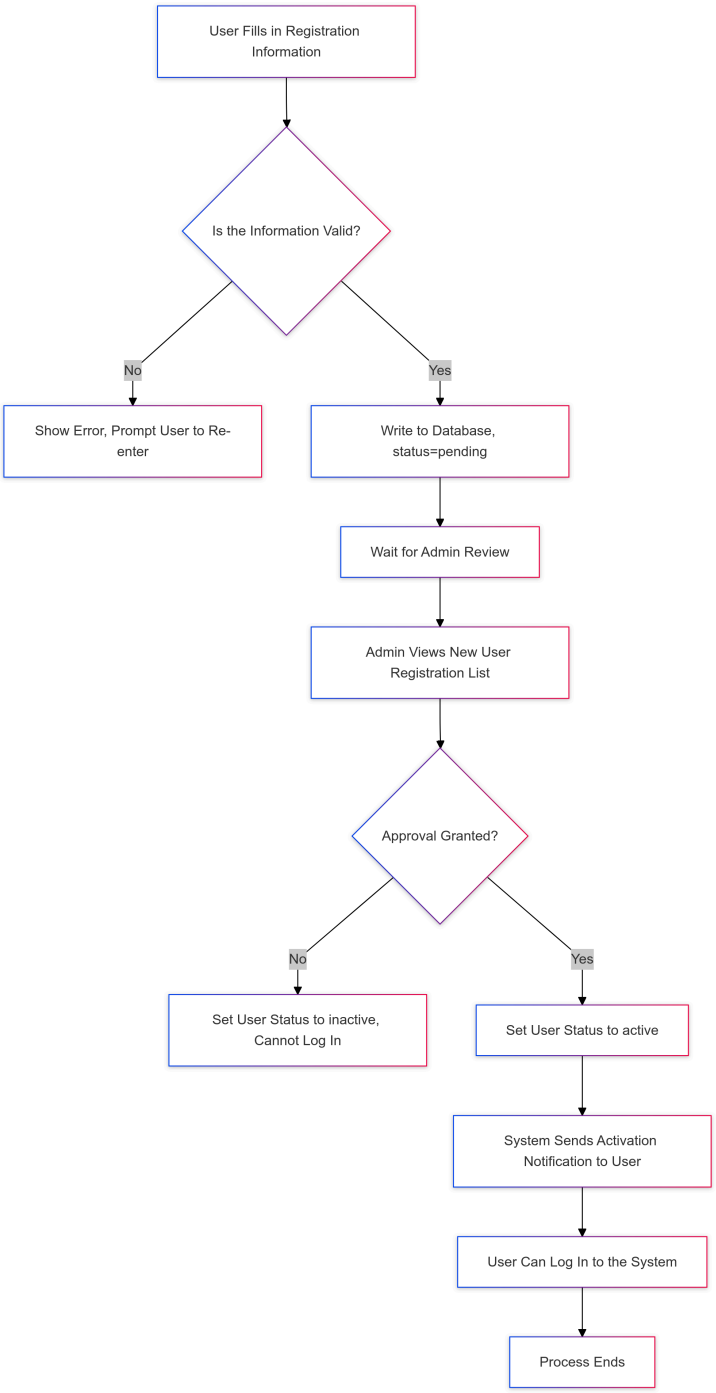

Figure 6: User Registration & Administrator Review Process

This flowchart shows the process of user registration:

1. User registration: The user fills in the registration information.
2. Information verification:

   If the information is invalid, the system prompts the user to re-enter;

   if the information is valid, the system saves it to the database and sets it to “pending review”.
3. Administrator review: The administrator views the new user registration list:

4. If rejected, the user status is set to “inactive” and cannot log in.
5. If approved, the user status is set to “activated” and an activation notification is sent.
6. Process completion: After receiving the notification, the user can log in to the system and complete the registration process.

2. Create order (sales/quotation) process

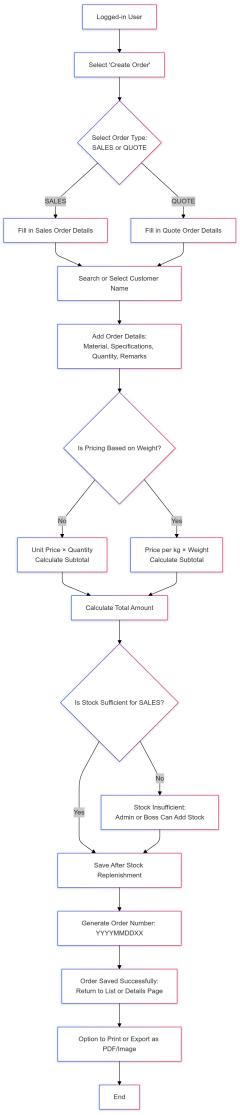

Figure 7: Create Order (Sales/Quotation) Process

This flowchart shows the process of a user creating an order (sales order or quotation) in the system. Here is a brief explanation of the process:

1. User login the system
2. Choose the `create order`option
3. User choose the type of the order (sale or quotation)
4. Search the customer name
5. Add the detail of the order such as materials, specifications, quantities and remarks
6. Determine the way to caculate the

- calculate the subtotal as unit price × quantity.
- calculate the subtotal as unit price per kilogram × weight.

7. Calculate the total price
8. Check whether the inventory is sufficient to meet sales demand
9. After saving the order, the system will generate a unique order number (in the format of YYYYMMDDXX).
10. Show that the order has been saved successfully and return to the order list .

3. Mark order payment process

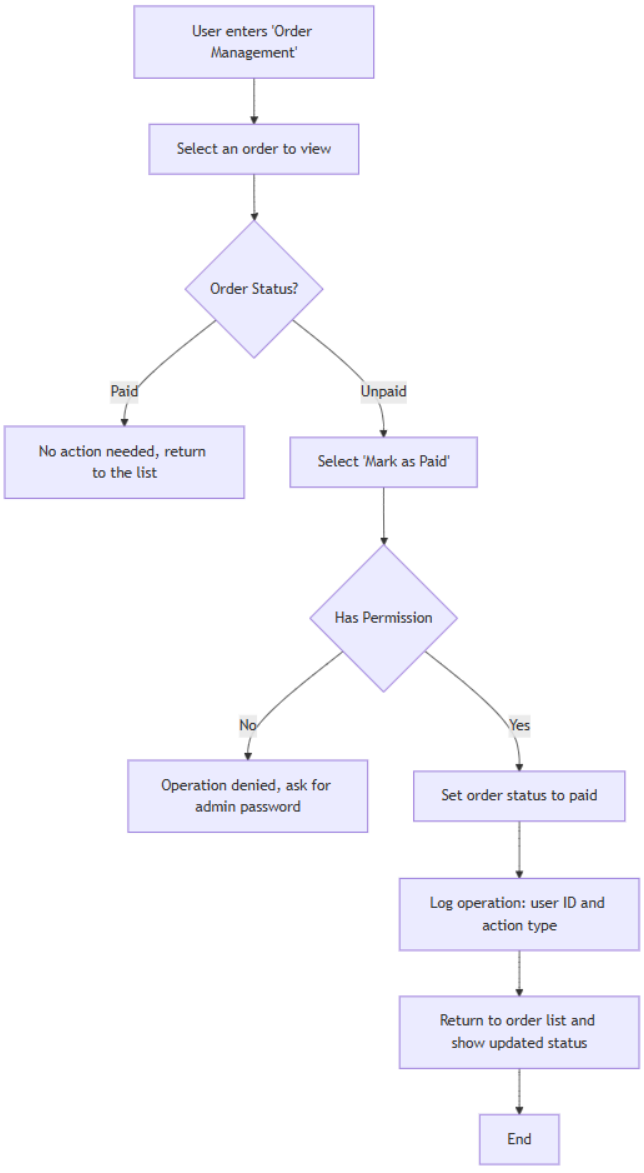

Figure 8: Mark Order Payment Process

The flowchart shows the process of order management:

1. Enter order management: The user enters the “Order Management” page and can select an order to view.
2. Order status judgment:

If the order has been paid, no operation is required and it returns to the order list directly.

If the order has not been paid, the user can select “Mark as paid”.

3. Authorization verification: The system determines whether the user has the authority to perform the operation:

   If there is no authority, it prompts that the operation is denied and requires the administrator password to be entered.

   If there is authority, the order status is set to paid and the operation log (including user ID and operation type) is recorded.

4. Complete the operation: Return to the order list and display the updated status, and the process ends.

5.3 Sequence Diagram

1. Login Sequence Diagram

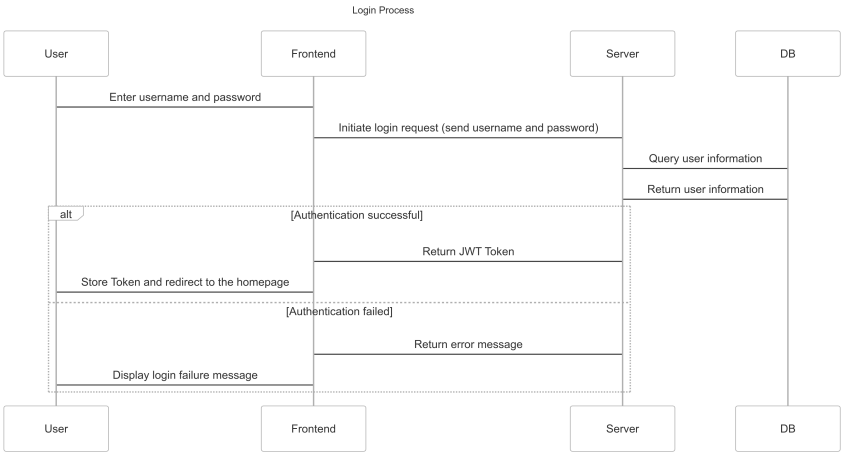

Figure 9: Login Sequence Diagram

User input the username and password, system will verify the information if success will return the JWT Token. If fail the system will return the error message

2. Register Process

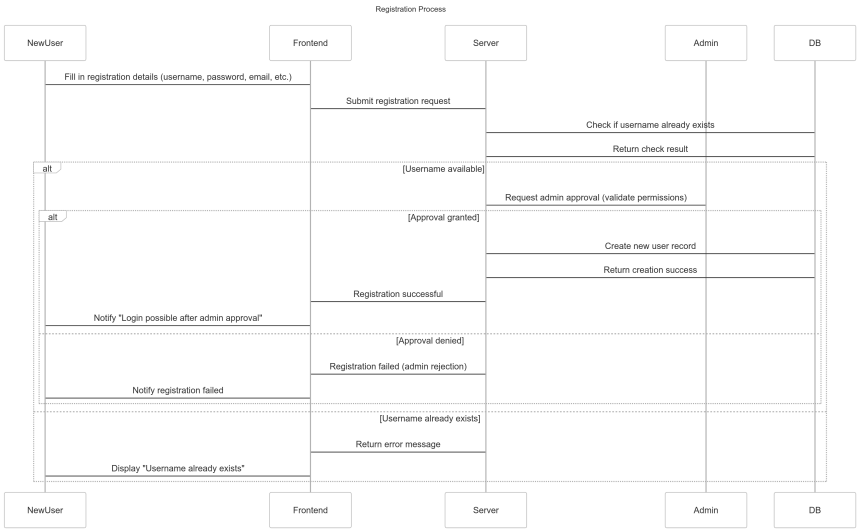

Figure 10: Register Process

Register process needs the admin to audit the request. If the admin approve the register the status of the user will set as the `active` .

3.Home Page

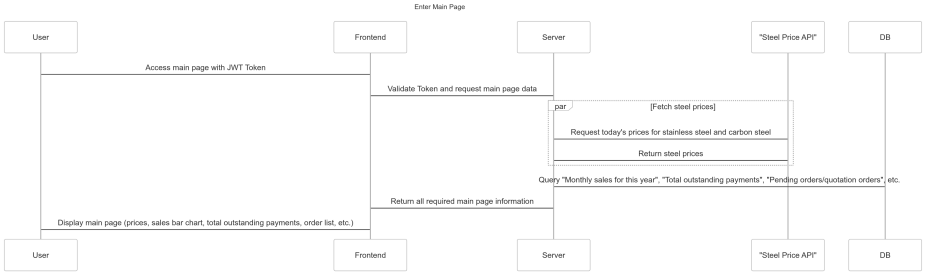

Figure 11: Home Page

When the user login successfully, system will show the steel price,, total outstanding balances and uncompleted orders is displayed on the homepage as a dashboard just like the home page in the odoo system .

4. Production Order Process

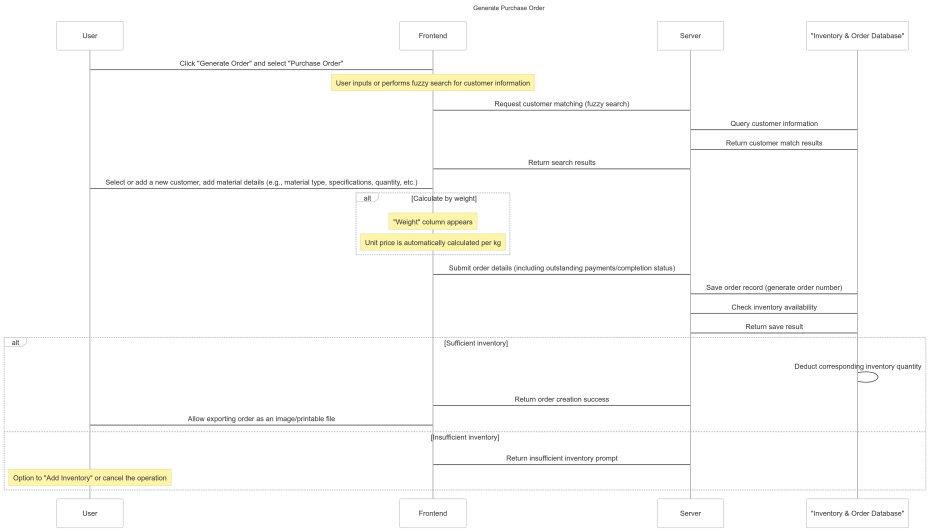

Figure 12: Production Order Process

Production order supports the weight or numbers to calculate the total price, choose customer can use fuzzy searching. If the inventory is insufficient inventory will show a alarm and you can add the inventory in the order page .

5. Quotation Order Process

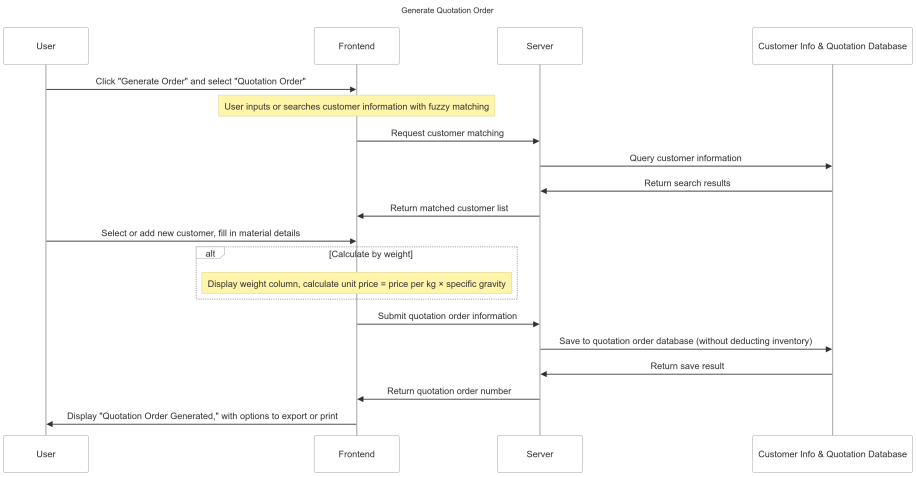

Figure 13: Quotation Order Process

Quotation order will not reduce the inventory immediately which only contain the information of the order in the order list and give a tag “Quatation” in this order.Then you can change the order into the production order in order management system later.

6. Customer Management Process

Figure 14: Order Management Process

Customer Manage can search the information of the customer and their own the historical orders. And it can export the unpaid order as the pdf. And it can pay order or delete the order .

7. Inventory Management Process

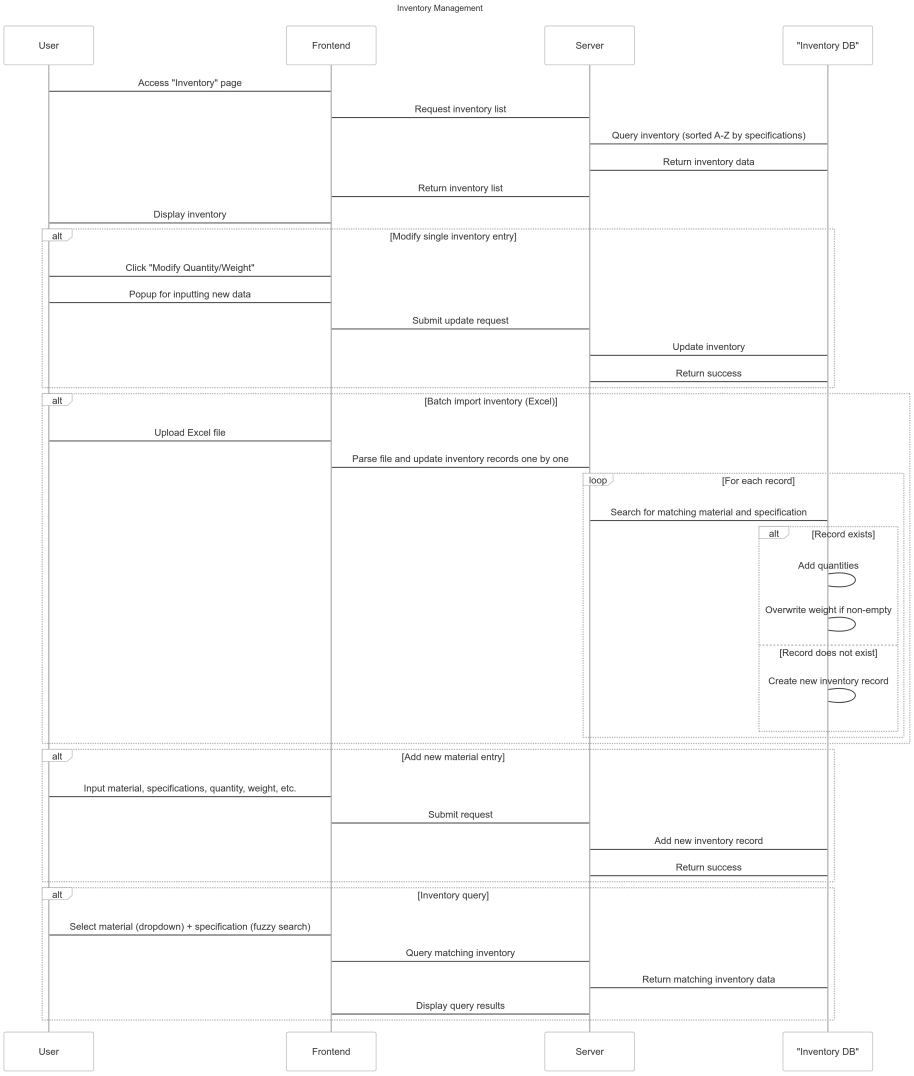

Figure 15: Inventory Management Process

Inventory Manage is used to check the available materials and specifications, and their numbers and weight. What's more the system provide a function to batch import, single item addition or modification functions, and inventory query.

8. Order Management Process

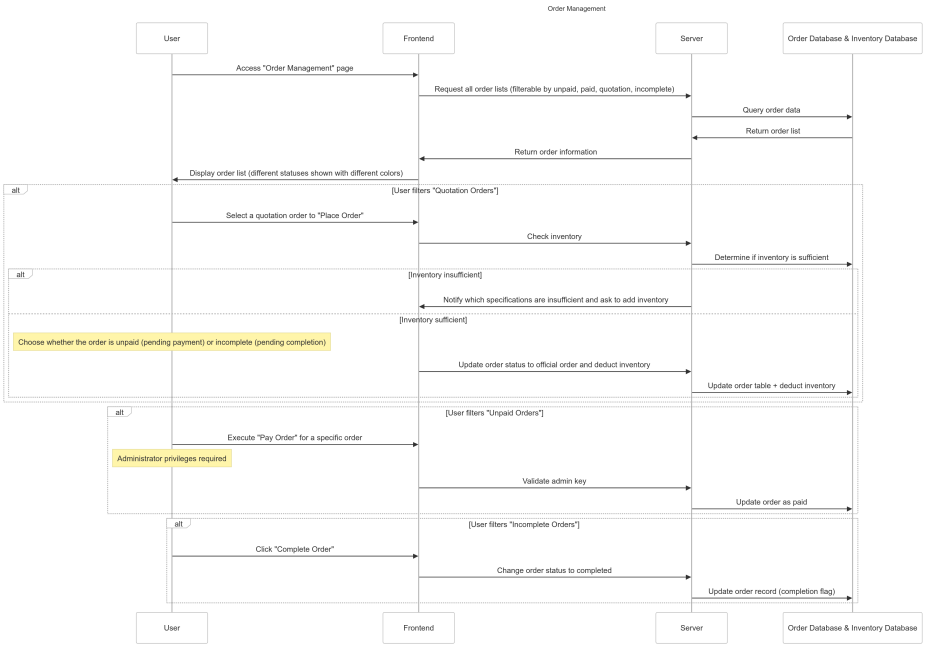

Figure 16: Order Management Process

Order management can check all the orders and sorted by payment status,order type and fullfilled status. Support to transform the quotation order into the formal order and user can pay the order in this model.

9. Employee Management Process

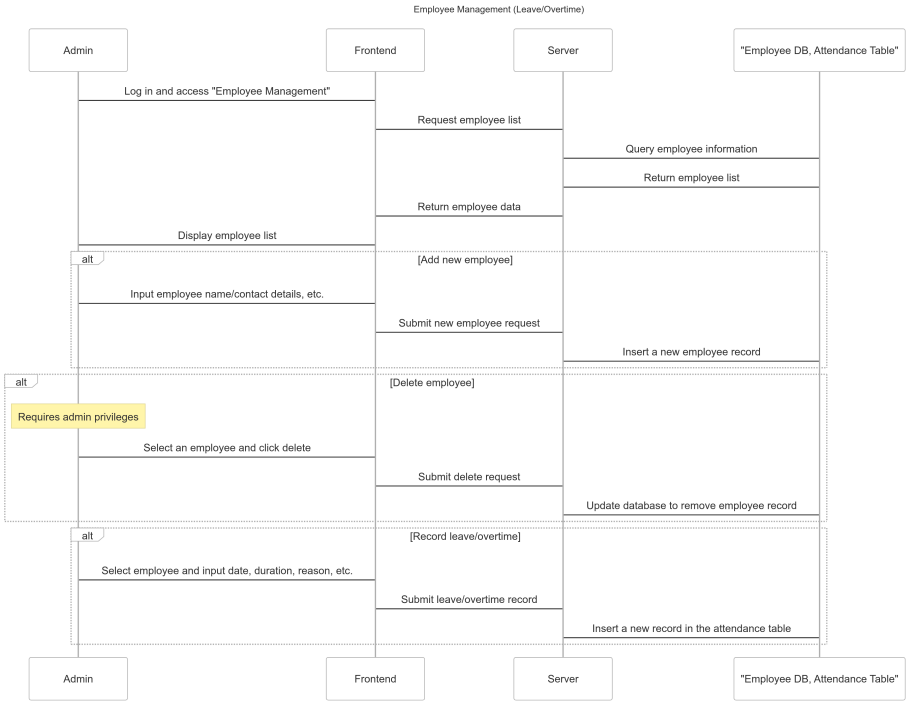

Figure 17: Employee Management Process

Admin can add and delete the employees. Also can manage and record the leave and overtime.

6. Software

- Visual Studio Code:

  version: 1.96.2 (user setup)

- Node.js:

  version: 20.18.1

- Postman:

  version:11.23.3

- Navicat:

  version: 16.0.11 - Premium

- Mysql:

  version: 8.0.35 for Win64 on x86_64 (MySQL Community Server - GPL)

7. Technology Stack

7.1 Frontend Technology Stack

- React - A JavaScript library for building user interfaces
- React Router - For frontend routing management, enabling navigation between different pages in a single-page application
- React Query - For data fetching, caching, and state management
- Material UI (`@mui`) - Provides ready-to-use UI component library and styling system
- Axios - HTTP client for communicating with backend APIs
- React Hook Form - A library that simplifies form handling and validation
- JSPdf and html2canvas - For generating PDF documents on the frontend
- React Slick - A component library for implementing carousel effects

7.2 Backend Technology Stack

- Node.js - A JavaScript runtime environment that allows running JavaScript on the server side
- Express - A web application framework for Node.js, simplifying routing handling and middleware use
- Sequelize - ORM library that maps JavaScript objects to database tables, simplifying database operations
- MySQL (mysql2) - A relational database for storing application data
- JWT (jsonwebtoken) - For generating and verifying tokens, implementing stateless authentication
- bcryptjs - For password hashing and verification, enhancing security
- node-cron - A scheduling library for automated tasks
- Multer - Middleware for handling file uploads
- ExcelJS - For generating and parsing Excel files
- PDFKit - For generating PDF documents on the server side
- Puppeteer - Headless browser for webpage screenshots and PDF generation
- Handlebars - Template engine for generating HTML content
- dotenv - For loading environment variables, managing configurations for different environments
- cors - For handling Cross-Origin Resource Sharing
- express-rate-limit - For limiting API request frequency, preventing abuse

7.3 Architecture

The project adopts a frontend-backend separation architecture, with the frontend building a single-page application and the backend providing RESTful API services.

7.4 Deployment Environment

- Domain:
  - Namesilo
- Frontend Hosting:
  - Vercel - Cloud platform for static sites and serverless functions, providing automatic deployments, global CDN, and scaling
- Database Hosting:
  - MySQL (hosted on AWS RDS) - Managed relational database service providing easy setup, operations, and scaling
- Backend Deployment:
  - AWS Elastic Beanstalk - Easy-to-use service for deploying and scaling web applications, handling capacity provisioning, load balancing, and application health monitoring
- Reverse Proxy/Load Balancing:
  - Nginx - High-performance HTTP server and reverse proxy, providing load balancing, caching, and SSL termination

References

[1] Odoo S.A. (2023). Odoo Inventory Management. https://www.odoo.com/app/inventory

[2] Oracle Corporation. (2023). MySQL Documentation. https://dev.mysql.com/doc/

[3] React Team. (2023). React Documentation. https://reactjs.org/docs/getting-started.html

[4] Express.js. (2023). Express.js Documentation. https://expressjs.com/

[5] Sequelize. (2023). Sequelize ORM Documentation. https://sequelize.org/master/