Hong Kong Institute of Vocational Education (CW)

Department of Information Technology

HD in Software Engineering

System Analysis and Design Report

Date: 4/4/2024

Group No. 1

Student	Contribution to the project (%) (Total 100%)	Signature
Tse Man Fung(230026964)	40%	AT
Chung Chun Kit(230118649)	40%	ricky
Sze Wang Lik(230027008)	20%	lik
Wong Wing Wong(230091344)	0%	www

We declare that this is a group project and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgement is made explicitly in the text, nor has any part been written for us by another person.

1. Abstract

This report presents an analysis and proposed design for a central computerized management system to address operational challenges at Legend Motor Limited. The system aims to streamline key processes such as order processing, inventory management and customer service through the digitization and integration of business functions. A feasibility study indicates that such a centralized digital solution can help optimize resource utilization, minimize errors, and enhance decision making through real-time information sharing across departments. Implementation of the proposed system is expected to improve overall efficiency at Legend Motor Limited while delivering a better experience for customers.

2. Table of Content

1. ABSTRACT 2

2. TABLE OF CONTENT 3

3. INTRODUCTION 12

3.1. BACKGROUND 12

3.2. PURPOSE OF THIS PROJECT 12

3.3. PURPOSE OF THIS REPORT 13

3.4. SCOPE OF THIS PROJECT 13

3.5. OVERVIEW 13

4. PROBLEM FINDINGS 14

4.1. PROBLEM AND SOLUTION 14

4.2. PROBLEM-FUNCTION TABLE 18

5. FUNCTIONAL REQUIREMENTS 19

5.1. FOR DEALER AND LM 19

5.1.1. Login function 19

5.1.2. Data center 19

5.1.3. Electronic version 19

5.2. FOR DEALER 20

5.2.1. Order create function 20

5.3. FOR LM 20

5.3.1. Record outstanding order 20

5.3.2. DI set generation function 20

5.3.3. Auto ordering function 20

5.3.4. Report 21

5.3.5. QR-code 21

6. NON-FUNCTIONAL REQUIREMENTS 22

6.1. SYSTEM QUALITY 22

6.1.1. Security 22

6.1.2. Serviceability 22

6.1.3. Maintainability 22

6.1.4. logging 22

6.2. INTERFACE AND USER EXPERIENCE 23

6.2.1. Language 23

6.2.2. User-friendly 23

6.2.3. Operational 23

6.3. PERFORMANCE AND SCALABILITY 23
6.3.1. Network 23
6.3.2. Scalability 24
6.3.3. Database server 24
6.3.4. Performance 24
6.3.5. System Design 24
7. SYSTEM ARCHITECTURE 26
7.1. HARDWARE CONFIGURATION 26
7.2. SOFTWARE CONFIGURATION 27
7.3. NETWORK CONFIGURATION 28
8. HOW CAN AN ORGANIZATION BENEFIT FROM THIS CENTRAL COMPUTERIZED MANAGEMENT SYSTEM? 30
8.1. IMPROVE OPERATIONAL EFFICIENCY 30
8.2. PROVIDE BETTER INFORMATION MANAGEMENT AND DATA ANALYSIS CAPABILITIES 30
8.3. OPTIMIZE ITS PROCESSES 30
8.4. REDUCES HUMAN DELAYS AND ERRORS 30
8.5. REDUCING WRITE-OFF 31
8.6. ENHANCE COLLABORATION AND COMMUNICATION 31
8.7. IMPROVED RISK MANAGEMENT AND SECURITY 31
8.8. BETTER CUSTOMER SERVICE 31
8.9. COMPETITIVE ADVANTAGE 31
8.10. SUMMARY 32
9. CONSTRAINTS AND LIMITATIONS 33
9.1. DATA INTEGRITY AND CONSISTENCY 33
9.2. COMPLIANCE WITH COMPANY INTERESTS AND POLICY REGULATIONS 33
9.3. DEPENDENCY ON THIRD-PARTY STANDARDS 33
10. ACTOR DESCRIPTION 34
11. USE CASE 36
11.1. UNUSUAL 36
11.1.1. Login function 36
11.1.2. forget password 36
11.2. ORDER PROCESSING SYSTEM 38
11.2.1. Diagram 38
11.2.2. create list for delivery 39
11.2.3. Browse catalogue 39
11.2.4. Place An Order 40

11.2.5. Check Order Detail and Review Order Status *41*
11.2.6. Check More Detail *42*
11.2.7. Download Invoice PDF *43*
11.2.8. View Invoice and Sign *43*
11.2.9. Check a DI set *44*
11.2.10. Generate DIC and DID *45*
11.2.11. Collect Spare *45*
11.2.12. Use Scanner *46*
11.2.13. create list for collect spare *47*
11.2.14. edit order *48*
11.2.15. edit order item *49*
11.2.16. edit order number *50*
11.3. STOCK MANAGE SYSTEM 51
11.3.1. Diagram *51*
11.3.2. Generate Annual Report *52*
11.3.3. Add New Spare *52*
11.3.4. Re-Stock Order *53*
11.3.5. Re-Stock Order Receive *54*
11.3.6. Add spare to re-stock list *55*
11.3.7. Scrap Spare *55*
11.3.8. Set Stock Level *56*
11.3.9. Edit Spare *57*
11.3.10. read and update stock *58*
11.4. ACCOUNT MANAGEMENT SYSTEM 59
11.4.1. Diagram *59*
11.4.2. Account management *60*
11.4.3. Dealer account management *60*
11.4.4. LM staff account management *61*
11.4.5. Create Account *62*
11.4.6. Dealer Account create *63*
11.4.7. LM Staff Account create *63*
11.4.8. Edit Supplier *64*
11.4.9. Add Supplier *65*
11.4.10. Add SaleArea *66*
11.4.11. Edit SaleArea *66*
11.5. ACTOR-USE CASE TABLE 71

12. CLASS DIAGRAM 74

13. SEQUENCE DIAGRAM 75

13.1. LOGIN 75

13.2. BROWSE CATALOGUE 75

13.3. CHECK ORDER DETAIL AND REVIEW ORDER STATUS 76

13.4. CREATE LIST FOR COLLECT SPARE 77

13.5. CREATE LIST FOR DELIVERY 78

13.6. DI SET 79

13.7. EDIT ORDER 80

13.8. INVOICE 81

13.9. MAIN MENU 81

13.10. PLACE AN ORDER 82

13.11. RE-STOCK 83

13.12. TIMER 84

13.13. ADD DEPARTMENT 85

13.14. ADD NEW SPARE 86

13.15. ADD SALEAREA 86

13.16. ADD WAREHOUSE 87

13.17. COLLECT SPARE 88

13.18. CREATE ACCOUNT 89

13.19. EDIT DEPARTMENT 90

13.20. EDIT SALEAREA 91

13.21. GENERATE ANNUAL REPORT 92

13.22. SCRAP 92

13.23. SET STOCK LEVEL 93

13.24. ACCOUNT MANAGEMENT 94

13.25. ADD SUPPLIER 95

13.26. EDIT OR DELETE SPARE 95

13.27. EDIT SUPPLIER 96

13.28. EDIT WAREHOUSE 97

14. STATE MACHINE DIAGRAM 98

14.1. ACCOUNT STATE 98

14.2. ORDER STATE 99

14.3. ORDERITEMTOFOLLOW STATE 100

14.4. RESTOCKORDER STATE 101

14.5. INVOICE STATE 102

14.6. TIMER 103
14.7. LOGIN 104
14.8. BROWSE CATALOGUE AND PLACE AN ORDER 105
14.9. CHECK ORDER DETAIL AND REVIEW ORDER STATUS 106
14.10. MAIN MENU 107
14.11. RE-STOCK 108
14.12. EDIT ORDER 109
14.13. COLLECT SPARE 110
14.14. EDIT WAREHOUSE 111
14.15. EDIT DEPARTMENT 112
14.16. EDIT SALEAREA 113
14.17. SET STOCK LEVEL 114
14.18. GENERATE ANNUAL REPORT 115
14.19. SCRAP 116
14.20. SUPPLIER MANAGEMENT 117
14.21. ADD DEPARTMENT 118
14.22. ADD SALEAREA 119
14.23. ACCOUNT MANAGEMENT 120
14.24. ADD WAREHOUSE 121
14.25. EDIT SPARE 122
15. USER INTERFACE 123
15.1. OVERVIEW 123
15.2. UI DESCRIPTION 124
15.2.1. Main Bar 125
15.2.2. Language 125
15.2.3. Login 127
15.2.4. Main Menu 129
15.2.5. Add Sale Area 131
15.2.6. Edit Sale Area 132
15.2.7. Add Department 134
15.2.8. Edit Department 136
15.2.9. Add Warehouse 138
15.2.10. Edit Warehouse 139
15.2.11. Add New Spare 140
15.2.12. Edit Spare 141
15.2.13. Scrap 142

15.2.14. Set Stock Level *142*
15.2.15. Re-stock (Re-order) *143*
15.2.16. Re-stock (Re-order) Check *144*
15.2.17. Placing Order *145*
15.2.18. Report *147*
15.2.19. Supplier management(New Supplier) *148*
15.2.20. Edit Order *150*
15.2.21. Collect Spare *153*
15.2.22. Order Record *155*
15.2.23. User and Account Create *159*
15.2.24. Change Password *164*
15.2.25. Modify Information *164*
15.3. REPORT DESIGN 165
15.4. API 177
15.5. FORM-ACTOR TABLE 178
16. ENTITY RELATION DIAGRAM 180
17. DETAILED DATABASE DESIGN 181
18. TEST DATA 194
18.1. BUILD DATABASE 194
18.2. DATA INSERT 194
18.3. DROP TABLE 208
19. TEST CASES 209
19.1. LOGIN 209
19.1.1. Test Procedure 1 ..... *209*
19.1.2. Test Procedure 2 ..... *209*
19.2. ADD NEW SPARE 209
19.2.1. Test Procedure 1 ..... *209*
19.2.2. Test Procedure 2 ..... *209*
19.3. EDIT SPARE 210
19.3.1. Test Procedure 1 ..... *210*
19.3.2. Test Procedure 2 ..... *210*
19.4. GENERATE SPARE QE-CODE 210
19.4.1. Test Procedure 1 ..... *210*
19.4.2. Test Procedure 2 ..... *210*
19.4.3. Test Procedure 3 ..... *211*
19.5. SCRAP 211

19.5.1. Test Procedure 1 ..... *211*
19.5.2. Test Procedure 2 ..... *211*
19.6. PLACE ORDER 211
19.6.1. Test Procedure 1 ..... *211*
19.6.2. Test Procedure 2 ..... *211*
19.6.3. Test Procedure 3 ..... *212*
19.6.4. Test Procedure 4 ..... *212*
19.6.5. Test Procedure 5 ..... *212*
19.7. ORDER EDIT 212
19.7.1. Test Procedure 1 ..... *212*
19.7.2. Test Procedure 2 ..... *212*
19.7.3. Test Procedure 3 ..... *213*
19.7.4. Test Procedure 4 ..... *213*
19.8. ORDER RECORD 213
19.8.1. Test Procedure 1 ..... *213*
19.8.2. Test Procedure 2 ..... *213*
19.9. COLLECT SPARE 213
19.9.1. Test Procedure 1 ..... *213*
19.9.2. Test Procedure 2 ..... *214*
19.9.3. Test Procedure 3 ..... *214*
19.9.4. Test Procedure 4 ..... *214*
19.10. SET STOCK LEVEL 214
19.10.1. Test Procedure 1 ..... *214*
19.10.2. Test Procedure 2 ..... *215*
19.10.3. Test Procedure 3 ..... *215*
19.11. RE-STOCK 215
19.11.1. Test Procedure 1 ..... *215*
19.12. RE-STOCK RECORD 215
19.12.1. Test Procedure 1 ..... *215*
19.12.2. Test Procedure 2 ..... *216*
19.13. SUPPLIER MANAGEMENT 216
19.13.1. Test Procedure 1 ..... *216*
19.13.2. Test Procedure 2 ..... *216*
19.13.3. Test Procedure 3 ..... *216*
19.13.4. Test Procedure 4 ..... *217*
19.14. USER MANAGEMENT 217

19.14.1. Test Procedure 1 ..... *217*
19.14.2. Test Procedure 2 ..... *217*
19.15. ADD SALE AREA 217
19.15.1. Test Procedure 1 ..... *217*
19.15.2. Test Procedure 2 ..... *217*
19.15.3. Test Procedure 3 ..... *218*
19.16. EDIT SALE AREA 218
19.16.1. Test Procedure 1 ..... *218*
19.16.2. Test Procedure 2 ..... *218*
19.17. ADD DEPARTMENT 218
19.17.1. Test Procedure 1 ..... *218*
19.17.2. Test Procedure 2 ..... *219*
19.18. EDIT DEPARTMENT 219
19.18.1. Test Procedure 1 ..... *219*
19.18.2. Test Procedure 2 ..... *219*
19.18.3. Test Procedure 3 ..... *219*
19.19. ADD WAREHOUSE 219
19.19.1. Test Procedure 1 ..... *219*
19.20. EDIT WAREHOUSE 220
19.20.1. Test Procedure 1 ..... *220*
19.20.2. Test Procedure 2 ..... *220*
19.20.3. Test Procedure 3 ..... *220*
19.21. REPORT 220
19.21.1. Test Procedure 1 ..... *220*
19.22. CHANGE PASSWORD 221
19.22.1. Test Procedure 1 ..... *221*
19.22.2. Test Procedure 2 ..... *221*
19.22.3. Test Procedure 3” *221*
19.23. MODIFY INFORMATION 221
19.23.1. Test Procedure 1 ..... *221*
19.23.2. Test Procedure 2 ..... *222*
19.23.3. Test Procedure 3 ..... *222*
20. USER GUIDE 222
20.1. INSTALLATION AND SET UP 222
20.1.1. Server and MYSQL *222*
20.1.2. client application *225*

20.2. FAQ 230
21. IMPLEMENT PLAN 231
21.1. PROJECT SCHEDULE 231
21.1.1. Planning Table 231
21.1.2. PERT Chart 232
21.1.3. Gantt Chart 233
21.2. COST ESTIMATE 235
21.2.1. Server Requirements: 235
21.2.2. Operating System Requirements: 235
21.2.3. Client Device Requirements: 235
22. CONCLUSION 236
23. PROJECT LOG 237
23.1. MEETING 237
23.2. JOB DISTRIBUTION 238
24. REFERENCE 240
25. APPENDIX 241
25.1. GLOSSARY 241
25.2. PERT CHART 242
25.3. GANTT CHART 243
25.4. DROP_ALL.SQL 247
25.5. BUILD.SQL 248

3. Introduction

3.1. Background

Legend Motor Limited is a subsidiary of Legend Motor Company, a public limited liability company that specializes in the assembly and distribution of motor cars in the People's Republic of China (PRC) market. While Legend Motor Limited primarily focuses on providing spares to its parent company, it also sells spares to approximately 50,000 main dealers across various provinces and cities in the PRC.

3.2. Purpose of this project

To improve the efficiency of its operations and increase its market share, the Head of the IT Department at Legend Motor Limited has proposed the development of a central computerized management system. This system aims to enhance the existing order processing and stock recording procedures within the organization.

Currently, Legend Motor Limited faces challenges in terms of managing its stock and meeting dealer demand. The existing systems developed in different offices and departments are independent of each other, leading to a lack of integration and coordination. This situation hampers the company's ability to efficiently utilize the information obtained from the computer systems in all sites.

To address these challenges and enhance operational efficiency, the company aims to develop a central computerized management system. This system will streamline the order processing and stock recording procedures, improve inventory management, and enhance the overall customer experience. By centralizing the management of operations, Legend Motor Limited will be able to optimize its resources, reduce manual errors, and improve decision-making processes. The objective is to enhance the efficiency of daily operations, improve stock management, and ultimately enhance customer satisfaction.

3.3. Purpose of this report

The purpose of this report is to analyze the current challenges faced by Legend Motor Limited in managing their stock and meeting dealer demand, and to propose the development of a central computerized management system that can address these issues. The report outlines the functional and non-functional requirements of the system, assesses its feasibility, highlights the potential benefits for the organization, and provides comprehensive recommendations for the design and implementation of the proposed system.

3.4. Scope of this project

The scope of this project is to develop a centralized computerized management system that can streamline Legend Motor Limited's key business processes such as order processing, stock management, supplier relationship management and inventory control. The system will focus on integrating the existing independent systems used by various departments, digitizing manual paper-based processes, automating workflows and providing real-time information sharing across the organization. The overall aim is to enhance operational efficiency, inventory accuracy, customer service levels and business decision making through centralized digital transformation of core business functions.

3.5. Overview

In the following sections, we will delve into the current operational problems faced by Legend Motor Limited in more detail, outline the functional and non-functional requirements of the proposed central computerized management system, conduct a feasibility analysis of the system, and explore the potential benefits that the new system can bring to LM through improved operational efficiency, optimized inventory management, and enhanced customer experience.

4. Problem Findings

4.1. Problem and Solution

Problem 1: Lack of Integration The current systems used by different offices and departments are independent and not integrated with each other. This lack of integration hinders the efficient flow of information and data between departments, leading to potential inefficiencies and data inconsistencies.
Solution: Standardized communication • Now all departments in the company will have the same Stock Recording System. As the system is updated in real time, access to information among departments will be facilitated.
Problem 2: Inefficient Order Processing and Stock Recording Procedure The existing order processing and stock recording system is manual and relies on the Sales Office manager to process dealer orders and issue instructions for dispatch. This manual process can be time-consuming and prone to errors.
Solution: Stock Recording System • Most of the order processing will be handled by the new system and the manual input will be greatly reduced. • The re-ordering process will be automated by the new system, eliminating the need to manually check stock levels. Each reorder will be recorded to prevent anomalies. • Invoices will be converted to electronic invoices to improve efficiency. • Implementing an automated system.
Problem 3: Lack of real-time stock updates. The stock recording procedure relies on manual entry of stock quantities, which can lead to delays in updating stock records.
Solution: real-time stock management system

- Implementing a real-time stock management system that automatically updates stock quantities when items are dispatched or received can provide more accurate and up-to-date stock information.
- Using a Stock Recording System allows different departments to share stock data to ensure consistency of information.

Problem 4: Lack of Standardized Processes

The existing processes and policies in the sales division are based on practicality rather than best practices. The case mentions that there is no standard order request form for dealers or for production requests from LMC. This can lead to confusion and errors in processing orders.

Solution: Implementation of standardized processes.

- Standardizing processes and implementing industry best practices can improve efficiency, consistency, and overall performance.
- Implementing a standardized order request form can streamline the process and ensure that all necessary details are provided.

Problem 5: Limited information technology base

Many offices and departments have limited knowledge of current systems and IT departments are only responsible for IT usage at HQ and some key offices and sites. This can lead to a lack of technical knowledge and ability to upgrade and support new systems.

Solution: Offer suitable interfaces and tools.

- Use interfaces and tools that are easy to understand and operate to decrease reliance on IT knowledge.
- Use Easy-to-understand language, diagrams and examples are used to explain technical concepts and operational procedures. This will provide reference for users and help them better understand and apply information technology.

Problem 6: The company's business is huge.

The company has 50,000 main dealers and 300 main sales areas. The company has to process a lot of information at the same time. This may lead to some order processing problems.
Solution: Systems with massive throughput • Large throughput systems are able to process large amounts of requests, transactions or data quickly.
Problem 7: Displeased with Spares Despatch Procedure When the Spares Despatch Department receives a dispatch instruction set, an invoice set is issued with invoice number and with partly completed, include pass to the invoicing, advice note and post to the dealer.
Solution: Set Electronic version of invoice set and change to electronic signature • The company can reduce unnecessary expenditure (e.g. paper). • Reduce the chance of loss
Problem 8: Huge write-offs As the continuous stock check proceeds, the actual stock figures are entered on the record card in the 'stock check, actual figure' column. Discrepancies are reported to the accountant where any exceeding 1 per cent or $100, whichever is the greatest, are investigated. The write-off for last year was $9,000,000.
Solution: QR-code Scanner with Stock Management System • A QR-code scanner with a stock management system can greatly speed up the stock counting process, enabling companies to track the quantity and value of goods faster. • Stock information is used for budgeting, future sales visions, supply chain decisions and other business critical directions, and rapid stock counts enable companies to make stock decisions quickly and frequently. To reduce write-offs.

Problem 9: Too many stock level Although different cards are sent when the stock level (re-order level, Danger Level and out-of-stock level) of a particular spare part reaches different levels, it is also possible that several different levels are reached at the same time, resulting in multiple sends of different cards and repeat orders.
Solution: Set up a one-time order at ROL • Only create a re-order card at ROL • Combine three levels and three cards into ROL and electronic re-order card. • Purchase the stock at once during reaches ROL.

4.2. Problem-function table

function \ problem	P1	P2	P3	P4	P5	P6	P7	P8	P9
Login function	✓								
Data center		✓	✓			✓			✓
Order create function		✓		✓		✓			
Record outstanding order		✓				✓			
DI set generation function		✓				✓			
Auto ordering function		✓				✓			✓
Electronic version				✓		✓	✓		
Report					✓	✓			
QR-code					✓	✓		✓	

5. Functional Requirements

5.1. For dealer and LM

5.1.1. Login function

- The username and password will be store in database.
- The password will be encrypted by Hash function.
- New accounts will be created by IT department of LM with the approval of manager department.
- A random password will be created when create account.
- If user forget the password, user must call IT department to reset password.
- An account will be lock when user input wrong password many times.
- Usernames can only include letters and numbers.

5.1.2. Data center

- Implement the database using MySQL.
- Implement 3NF.
- User can only access the permitted data through the application.
- The number of goods will be store to database and change the number when goods inward item s and goods outward items.
- Staff can add/edit/delete records of goods in the database using the GUI app.

5.1.3. Electronic version

- Electronic version of invoice set(EIV) and change to electronic signature.
- Automatically generates 2 different EIV for LM and dealers.
- For LM, EIV with item price.
- For dealers, EIV without item price and send a pdf version to dealer.

5.2. For dealer

5.2.1. Order create function

- The system will allow dealers create orders and send the order.
- The order form has its own format and strict input requirements.
- The user will be given a copy of the order when it is delivered.
- A standard order form is provided in the app for dealers to complete.
- Dealer can know the stock and add items to cart in app.
- Out-of-stock items cannot be ordered but can be added to the "in-stock notification".

5.3. For LM

5.3.1. Record outstanding order

- When an order cannot be completed, it will be placed in a special table in the database.
- After each time purchase the stock, will check this table, if an order containing this item is in the table, will automatically generate DI set, this order has the highest order priority.

5.3.2. DI set generation function

- Automatically generate a DID for each item on the order.
- Automatically generate a DIC for each order.
- Put DIDs and DIC to a DI set.

5.3.3. Auto ordering function

- When the stock level reaches ROL, a re-order card(pdf) is automatically generated and sent to the purchasing department -> manual confirmation -> send to spare parts shipping department via app.
- Staff can stop the automatically generated of re-order card.
- Provide API for linking to auto-reorder system.

5.3.4. Report

- An evaluated list (at standard purchase price) is required for all stock balances.
- Reports (e.g. sales reports, stock analysis, and performance monitoring, etc.) are available to staff/dealers with a click of a button.

5.3.5. QR-code

- Quick and accurate identification of the ordered items.
- Each bin location is identified by a part number.
- QR-code will contain the part number.

6. Non-Functional Requirements

6.1. System quality

6.1.1. Security

- All data and passwords will be encrypted.
- Different data requires corresponding access rights.
- All data will be backed up.
- The system will have account functionality and each account will have different permissions depending on the user.
- Systems are protected from malicious attacks.
- All Distributors can only see their own orders.

6.1.2. Serviceability

- Automatically sends an email or text message to a pager in the event of an abnormal event. Ensures that relevant personnel are automatically notified when an abnormal event occurs in the system.
- System with network monitoring function.
- System designed for efficient hardware upgrades with minimal downtime.

6.1.3. Maintainability

- All systems need to be maintained including bug fixes, enhancements, removal of obsolete features.
- Optimal adaptation to changes in the environment.

6.1.4. logging

- There will be a logging function to record all activities on the system.
- By logging system events, it is possible to track security events in the system, such as login failures, unauthorized access or other potential security risks.
- Log records can provide information about system performance, such as processing times, resource usage, and bottlenecks. It helps to optimize system performance.

- The system logs will be stored in a separate folder. Only the system administrator can open this file.

6.2. Interface and user experience

6.2.1. Language

- Users can choose between Traditional Chinese, Simplified Chinese and English.

6.2.2. User-friendly

- The system has an intuitive and easy-to-understand interface that allows users to get started quickly.
- Users can easily learn and master the functions of the system.
- The system is able to respond quickly to user actions, avoiding long waiting times.
- The display should be compatible with different screens.

6.2.3. Operational

- All orders are received from the order processing system.
- Most of the processing of stock data will be carried out in the new system.
- All data will be integrated into the new system.

6.3. Performance and scalability

6.3.1. Network

- The network speed will be able to maintain the daily operation of the company.
- The network will be highly stable and reliable.
- The network is highly secure and only special people can know the password.
- Highly scalable network to meet future business needs.

6.3.2. Scalability

- The database system is capable of handling an ever-increasing volume of data.
- Different functional modules of the system can be extended or modified independently without affecting other modules.
- The system will be distributing network traffic equally.

6.3.3. Database server

- Data is properly encrypted for transmission and storage.
- The database can continue to run or reboot in the event of a hardware or software failure.
- Fast database query speed.
- Tools and mechanisms are available to monitor database performance.
- There is plenty of storage space in the database.

6.3.4. Performance

- The speed of the network will be monitored.
- Data updates are immediate.
- The system can support multiple simultaneous users.
- The system has high throughput and can handle a certain number of simultaneous user requests.
- Extremely high storage capacity.

6.3.5. System Design

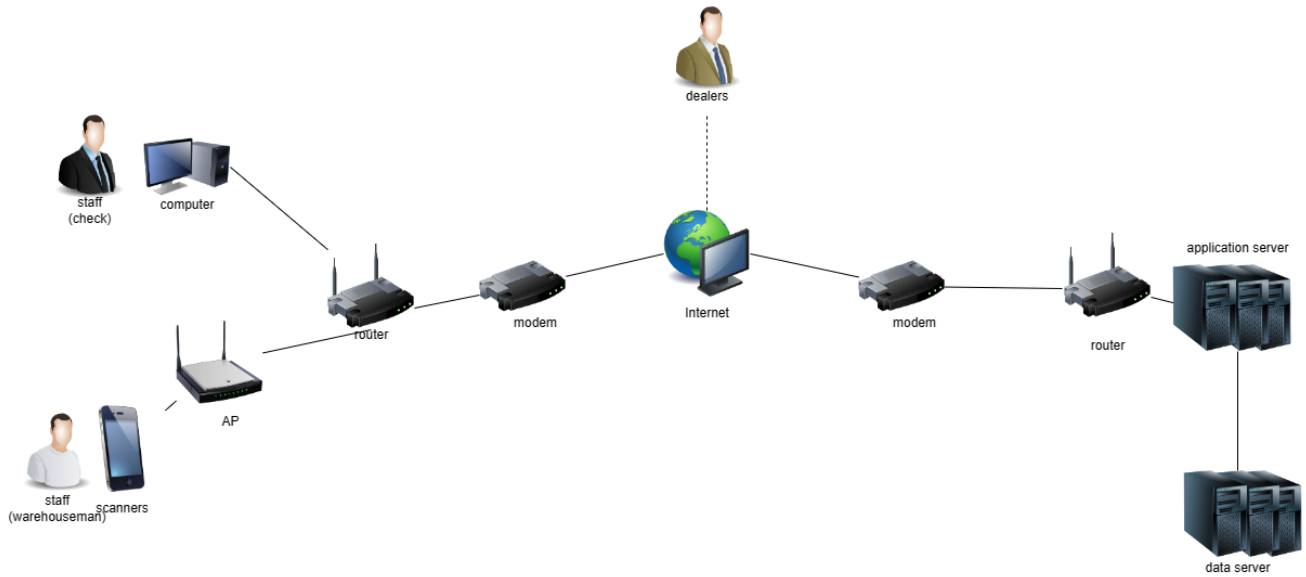
- C# language will be used for development.
- System should be able to handle UTF-8 characters, as customer names could be in Chinese.
- The design and formatting should align with the existing company profile. The developer shall design the interface with the above theme in mind.
- The user interface of the system is developed with ease of use as the main core, and the interfaces for different functions have the same style.

- The system has instruction manuals or instructional workflows so that employees understand how to use the interface and tools.

7. System Architecture

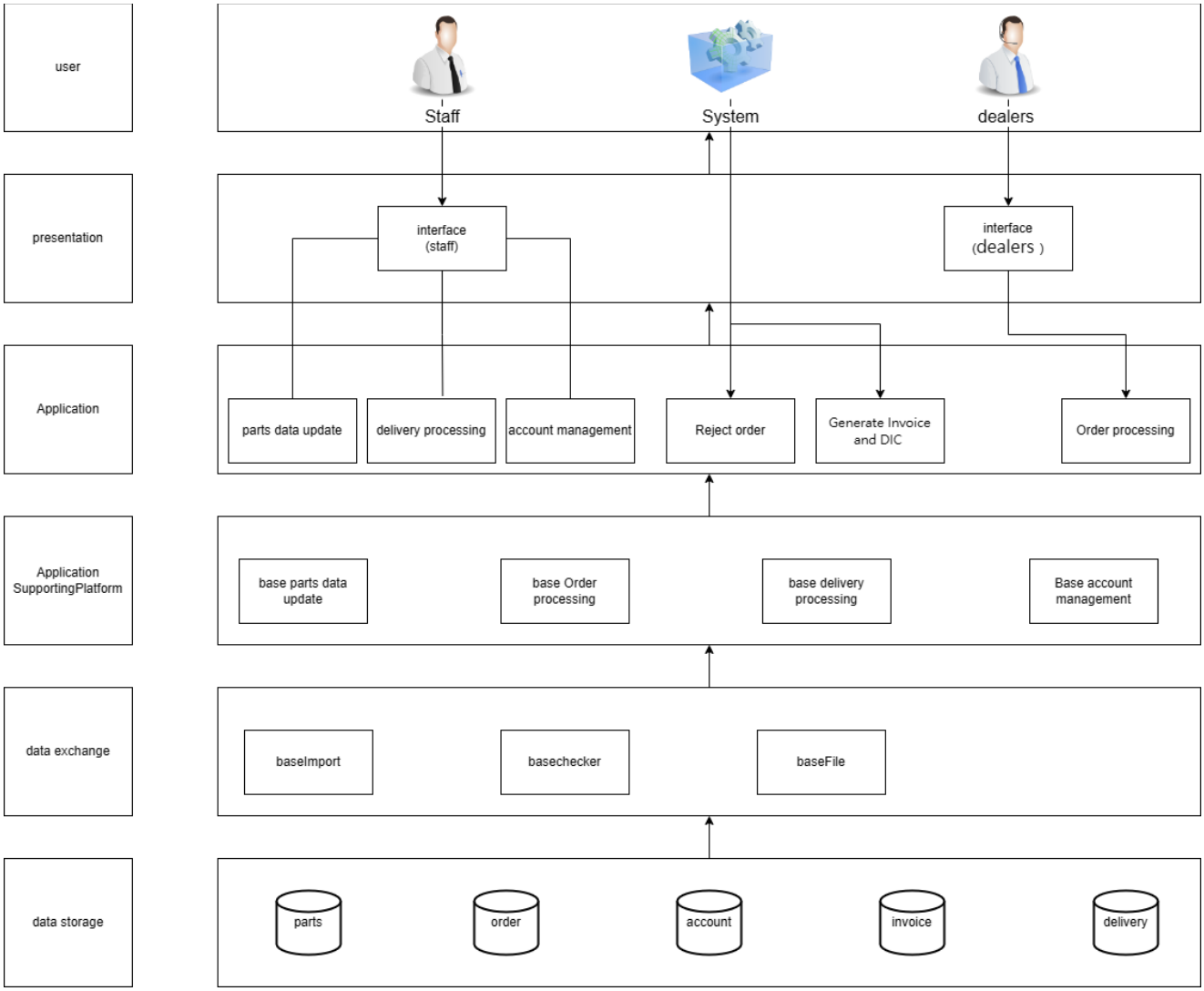
7.1. Hardware Configuration

Based on the business requirements, a three-tier thin client-server architecture is recommended as it allows the company to easily scale the system as needed.



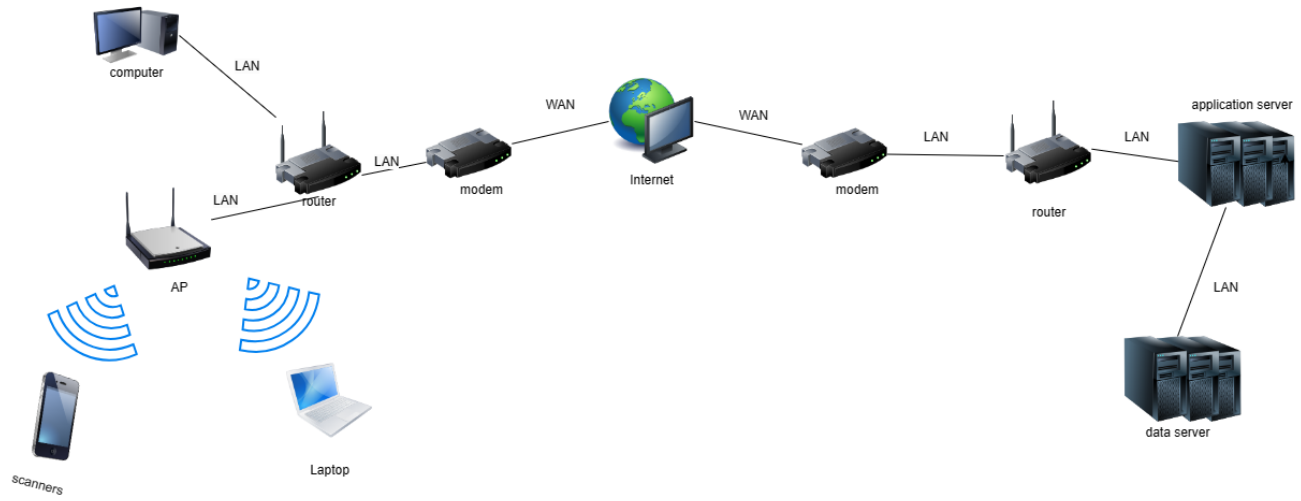
7.2. Software Configuration

The company is currently using Windows Server 2022 datacenter as the operating system. Users need Windows OS to use our applications.



7.3. Network Configuration

The system will rely on the Internet for its operation. Access to the server will be restricted to users with system accounts.



Specifically, dealers and staff will log in using an application that will be developed by our company. The database server will store the system's database, and the application server will use the data from the database server to process specific things on a regular basis.

As mentioned above, a database server, an application server and a client server are required, and the following are their respective specifications.

	Delivery worker / Warehouse worker	Another user	application server	Database Server
Operation System	Windows	Windows	Windows Server 2022 datacenter	Windows Server 2022 datacenter
Special Software	--	--	--	Mysql
Hardware	scanner	--	--	--
Network	--	--	--	--

8. How can an organization benefit from this central computerized management system?

A central computerized management system can bring many benefits to LM.

8.1. improve operational efficiency

The system can improve operational efficiency. By automating and centrally managing individual business processes, organizations can process tasks such as orders, stock management and shipping more quickly and accurately, saving time and resources.

8.2. provide better information management and data analysis capabilities

A centralized system can provide better information management and data analysis capabilities. By centrally storing and managing data, organizations can obtain more comprehensive and accurate information. It provides the opportunity to LM that can make more informed decisions.

8.3. optimize its processes

Implementing a central computerized management system allows LMC to optimize its processes and improve efficiency, resulting in cost savings. By automating processes such as order processing, stock management, Manual labor requirements and related costs can be decreased with LMC. The system can optimize stock levels, prevent stockouts and overstocking, and minimize the risk of obsolete or expired stock, leading to cost reductions and improved financial performance.

8.4. reduces human delays and errors

The central system reduces human delays and errors, improving operational efficiency. Manual processes are prone to errors and delays, which can result in additional costs and customer dissatisfaction. By automating tasks, standardizing workflows, and providing real-time information, the system minimizes the risk of human errors and ensures timely and accurate processing of orders, reducing delays and improving customer satisfaction.

8.5. reducing write-off

The system helps minimize losses by enhancing stock control and reducing write-off. With real-time visibility into stock levels and automated tracking, LMC can accurately monitor stock levels, identify potential issues, and take proactive measures to minimize losses. By optimizing stock turnover and reducing product obsolescence, LMC can improve profitability and mitigate financial risks.

8.6. enhance collaboration and communication

A centralized system can enhance collaboration and communication within an organization. Various departments and teams can share data and information through the system to achieve closer collaboration and coordination. This helps reduce information silos and communication barriers, and improve work efficiency and teamwork capabilities.

8.7. improved risk management and security

A centralized system can improve risk management and security. Centralized systems can enhance the protection of data and information, ensuring its security and confidentiality. At the same time, through centralized management and monitoring, organizations can better control and manage risks and reduce potential errors and security breaches.

8.8. better customer service

This system can also provide an organization with better customer service. At the same time LM can provide more personalized and responsive service. Customers can more easily check order status, obtain product information, etc., thereby improving customer satisfaction and loyalty.

8.9. competitive advantage

By leveraging these advantages, LMC gains a competitive edge in the market. The streamlined operations, improved stock management, and reduced errors and delays allow LMC to deliver products and services more efficiently than competitors relying on manual or less efficient systems. positioning LMC as a

preferred choice for customers and giving the company a competitive advantage.

8.10. summary

A central computerized management system can bring organizations increased, operational efficiency, enhanced information management and decision-making capabilities, Cost savings, enhanced stock control and waste reductio, increased risk management and security, enhance collaboration and communication and better customer service and Reduction of errors and delays. Also, these benefits can help organizations better cope with challenges, become more competitive, and achieve sustainable development.

9. Constraints and Limitations

9.1. Data Integrity and Consistency

Suppliers must use this application for ordering and employees must use it for processing orders to ensure data integrity and consistency. All data is stored in the main database at the company headquarters. However, since the company does not have backup servers, if data cannot be backed up in the main database, the previous data before the backup will be lost. Additionally, if there are issues with the ordering or order processing process of this application, employees will have to manually record their changes and subsequently submit the corresponding reports.

9.2. Compliance with Company Interests and Policy Regulations

The development process must comply with the company's interests and policy regulations. To meet these requirements, the development process cannot use solutions that exceed the allocated budget. Moreover, due to the extensive policy regulations of the company, the range of choices in the development process is limited. Furthermore, compliance with local laws and regulations is required during the development process, including data protection, intellectual property, and other relevant legal provisions. For example, ensuring that the system complies with the respective legal requirements when collecting, processing, and storing data.

9.3. Dependency on Third-Party Standards

Dependency on third-party standards. According to customer requirements, the system will be developed in C# using Visual Studio. Although C# is a stable and mature language, it is a proprietary programming language developed by Microsoft. Additionally, employees must use Windows for this system. If Microsoft later decides to stop supporting this language or charge subscription fees to users, additional work or costs will be required to ensure the smooth operation of the program.

10. Actor Description

Actor	extend	Description
Manager	LM officer	Management of the Company. Responsible for account management, account creation and other administrative matters.
accountant	User	LM accountant the main user of the report feature.
Storeman	User	The company's employees are mainly responsible for collecting and packaging spares, receiving spares and so on.
Warehouse clerk	User	Responsible for managing the department and warehouse.
Dealer	User	Merchants who buy spares from the company.
Timer	/	System Automation. Responsible for the regular handling of matters within the system.
Delivery worker	User	Company's shipping staff.
automatic re-ordering system	/	An automatic re-ordering system has been installed in the warehouse.
Stock record clerk	User	The stock record clerk is responsible for reordering. They are also responsible for Scrap Spare and Set Stock Level.
User	/	All system user

LM officer	User	Staff working in LM office. Responsible for managing the company's data
order process system	/	Order process system read and update Stock.

11. Use Case

11.1.Unusual

11.1.1.Login function

Use case name:	Login function
Use case ID:	OPSRS-50007
Primary actor:	user
Secondary actor(s):	
Brief description:	User login to account
Preconditions:	User who has an account in the system.
Flow of events:	1: User input account and password 1.1: System display successful or fail message 2: When user click Forget password, execute use case “forget password”.
Postconditions:	The user has successfully logged in to the system.
Alternative flows and exceptions:	When user forgot password, should click on the “forgot password?” button below the login button. The user enters an invalid login name or password, this use case restarts.
Non-behavior requirements:	The system should operate 24 hours a day.

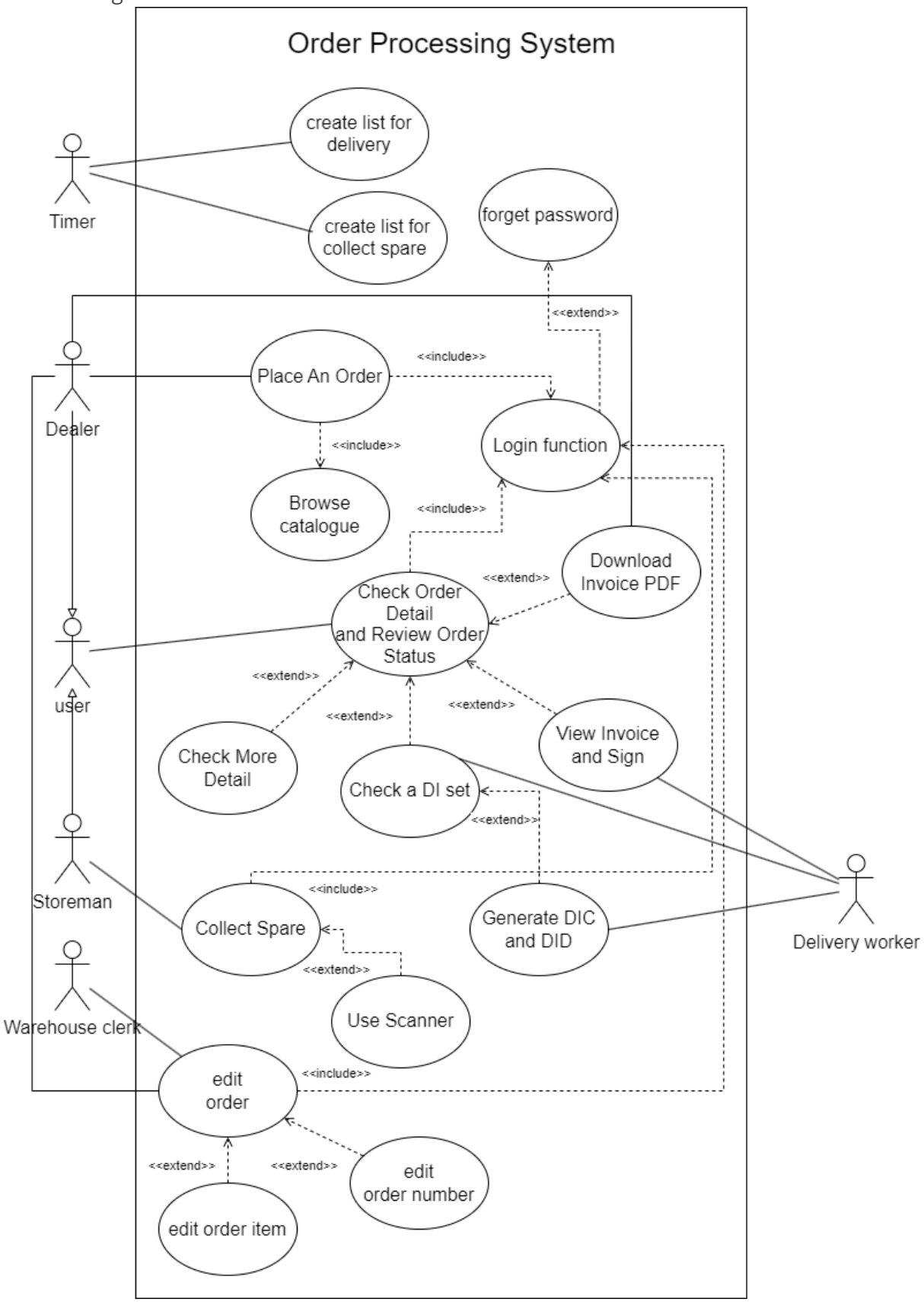
11.1.2.forget password

Use case name:	forget password
Use case ID:	OPSRS-50061
Primary actor:	user
Secondary actor(s):	
Brief description:	User reset password

Preconditions:	User click Forget password
Flow of events:	1: click Forget password 1.2: System display the required verify identity 2: User input identity information 2.1: System display success and new password or fail.
Postconditions:	User reset password
Alternative flows and exceptions:	
Non-behavior requirements:	

11.2.Order Processing System

11.2.1.Diagram



11.2.2.create list for delivery

Use case name:	create list for delivery
Use case ID:	OPSRS-50056
Primary actor:	Timer
Secondary actor(s):	
Brief description:	Timer creates a list for delivery.
Preconditions:	Today is working day.
Flow of events:	1: get the order which state is T 1.1: return the order which state is T [for each order get by 1.1] 2: get invoice by this order 2.1: return invoice by this order 2.2: get all items in table "ActualQuantityDespatched" about invoice 2.3: return 2.2 want 3: group all item will be despatched to a dealer 4: ask for Truck 4.1: return Truck 5: random assign delivery to truck, when their SaleArea is same
Postconditions:	
Alternative flows and exceptions:	Step 2-2.3 will be repeat for each order by 1.1.
Non-behavior requirements:	Only on working days

11.2.3.Browse catalogue

Use case name:	Browse catalogue
Use case ID:	OPSRS-50005

Primary actor:	Dealer
Secondary actor(s):	
Brief description:	Dealers search for spares on the "Placing Order" page with SpareId, SpareName and Category type.
Preconditions:	Dealers who have an account in the system.
Flow of events:	1: enter "SpareID" / "Spare Name" / " Category Type" 1.1: display matching results 2: select spare and choose quantity 2.1: display a message and clear results
Postconditions:	
Alternative flows and exceptions:	Step 1-2 can be repeated until all choices are completed.
Non-behavior requirements:	All spare data will be immediate.

11.2.4.Place An Order

Use case name:	Place An Order
Use case ID:	OPSRS-50006
Primary actor:	Dealer
Secondary actor(s):	
Brief description:	Dealers enter to the cart to
Preconditions:	Dealers who have an account in the system.
Flow of events:	Invoke use case "Browse catalogue". Invoke use case "Login function" 1: select a spare in cart

	1.1: display the name of selected spare 2: user change the quantity of selected spare 2.1: display the new quantity 3: user click the confirm button 3.1: display order serial or fail message
Postconditions:	A new order will be created.
Alternative flows and exceptions:	Step1-2 can be repeated until all choices are completed.
Non-behavior requirements:	The system should be able to handle 15,000 new orders per day.

11.2.5.Check Order Detail and Review Order Status

Use case name:	Check Order Detail and Review Order Status
Use case ID:	OPSRS-50029
Primary actor:	user
Secondary actor(s):	
Brief description:	Dealer or staff check order detail
Preconditions:	Have an account.
Flow of events:	Invoke use case “Login function.” 1: user go to the order record page 1.1: display the first 100 records in processing and complete 2: when user have more order record in database but not show, execute use case “Check More Detail”. 3: If Dealers want to view and download invoice, execute use case “Download Invoice PDF”.

	4: If LM staff want to view and download invoice, execute use case “View Invoice and Sign”. 5: If user want to view DI set, execute use case “Check a DI set”.
Postconditions:	
Alternative flows and exceptions:	
Non-behavior requirements:	

11.2.6.Check More Detail

Use case name:	Check More Detail
Use case ID:	OPSRS-50050
Primary actor:	user
Secondary actor(s):	
Brief description:	Dealer or staff check order detail
Preconditions:	When the user scrolls to the bottom.
Flow of events:	1 when the user scrolls to the bottom. 1.1: System display the next 100 records in processing and complete.
Postconditions:	display the next 100 records
Alternative flows and exceptions:	
Non-behavior requirements:	

11.2.7.Download Invoice PDF

Use case name:	Download Invoice PDF
Use case ID:	OPSRS-50031
Primary actor:	Dealers
Secondary actor(s):	
Brief description:	Dealers download invoice PDF.
Preconditions:	Have an account and a photocopier.
Flow of events:	1: user click the DI set button in order detail page 1.1: display invoice in pdf and print it.
Postconditions:	An invoice PDF will be downloaded.
Alternative flows and exceptions:	
Non-behavior requirements:	

11.2.8.View Invoice and Sign

Use case name:	View Invoice and Sign
Use case ID:	OPSRS-50030
Primary actor:	Delivery worker
Secondary actor(s):	
Brief description:	Storeman view Invoice.
Preconditions:	Have an account.
Flow of events:	1: user click the DI set button in order detail page 1.1: go to invoice page and display invoice 2: user click the sign here button

	2.1: go to signature page 3: user sign 3.1: System display invoice or fail
Postconditions:	Display invoice.
Alternative flows and exceptions:	
Non-behavior requirements:	

11.2.9.Check a DI set

Use case name:	Check a DI set
Use case ID:	OPSRS-50009
Primary actor:	Delivery worker
Secondary actor(s):	
Brief description:	Users check a DI set.
Preconditions:	1: user click the DI set button in order detail page
Flow of events:	1: user click the DI set button in order detail page. 1.1: System change the page to Despatch Instruction Set page and display DIC and DID 2: If user want to generate DIC and DID as pdf, execute use case “Generate DIC and DID”.
Postconditions:	A DI set pdf will be generated.
Alternative flows and exceptions:	
Non-behavior requirements:	

11.2.10. Generate DIC and DID

Use case name:	Generate DIC and DID
Use case ID:	OPSRS-50051
Primary actor:	Delivery worker
Secondary actor(s):	
Brief description:	Print DIC and DID as pdf.
Preconditions:	User click Export as PDF button
Flow of events:	1: user click Export as PDF button. 1.1: display the DIC and DID in pdf.
Postconditions:	A DI set pdf will be generated.
Alternative flows and exceptions:	
Non-behavior requirements:	

11.2.11. Collect Spare

Use case name:	Collect Spare
Use case ID:	OPSRS-50025
Primary actor:	Storeman
Secondary actor(s):	Timer
Brief description:	Storeman collect spare.
Preconditions:	Storeman have a scanner.
Flow of events:	Invoke use case “Login function.” 1: Users go to the collect spare page 1.1: display the list of spares to be collected

	2: When user Select ID directly. 2.1: user select the target spare ID 2.2: display succeed or fail 3: When user use scanner, execute use case “Use Scanner”. 4: enter the number of bundles and Actual Quantity 4.3: clear all textbox and mark this spare is collected
Postconditions:	
Alternative flows and exceptions:	Steps 2-4.3 can be repeated until all spares have been collected. Steps 2 and 3 will not run at the same time.
Non-behavior requirements:	

11.2.12. Use Scanner

Use case name:	Use Scanner
Use case ID:	OPSRS-50070
Primary actor:	Storeman
Secondary actor(s):	
Brief description:	Storeman use scanner.
Preconditions:	Storeman have a scanner.
Flow of events:	1:user click the scanner button 1.1: display the spare Id or fail
Postconditions:	

Alternative flows and exceptions:	
Non-behavior requirements:	

11.2.13. create list for collect spare

Use case name:	create list for collect spare
Use case ID:	OPSRS-50016
Primary actor:	Timer
Secondary actor(s):	
Brief description:	Generate a list of required spares by order
Preconditions:	Today is working day.
Flow of events:	1: get the order which state is 'C' 1.1: return the order which state is 'C' 2: get all items in this order 2.1: get all items in this order 2.2: pass all items to table "OrderItemToFollow" 2.3: return successful or fail 2.4: change the state of this order to 'P' 2.5: return successful or fail 3: create invoice, DespatchForemanID is null 3.1: return successful or fail 4: get all record from table "OrderItemToFollow"

	4.1: return all record from table "OrderItemToFollow" 4.2: get stock 4.3: return stock 5: put the this order item to ActualQuantityDespatched assign to the nearest warehouse's warehouseID 5.1: return successful or fail 6: put the this order item to ActualQuantityDespatched assign to the warehouse's warehouseID, the max stock of this warehouse 6.1: return successful or fail
Postconditions:	
Alternative flows and exceptions:	Step 2-3.1 can be repeate.
Non-behavior requirements:	Only on weekdays.

11.2.14. edit order

Use case name:	edit order
Use case ID:	OPSRS-50054
Primary actor:	Dealer
Secondary actor(s):	Warehouse clerk
Brief description:	Dealers edit an order.
Preconditions:	Dealer who had placed an order.
Flow of events:	Invoke use case “Login function.”

	1: user go to edit order page 1.1: System change the page to Edit Order. 2: Users select order serial and click edit order item or edit your order number. 2.1: If user click edit order item, execute use case “edit order item” else, execute use case “edit order number”.
Postconditions:	
Alternative flows and exceptions:	Step 5-6.1 can be repeat.
Non-behavior requirements:	

11.2.15. edit order item

Use case name:	edit order item
Use case ID:	OPSRS-50056
Primary actor:	Dealer
Secondary actor(s):	
Brief description:	Dealers edit order items.
Preconditions:	If user click edit order item button.
Flow of events:	1: user click edit order item 1.1: System display spare name, id, and quantity 2: user select the target spare name 2.1: System display the id and quantity 3: user change the quantity and click save 3.1: System display successful or fail
Postconditions:	An order item will be edit.

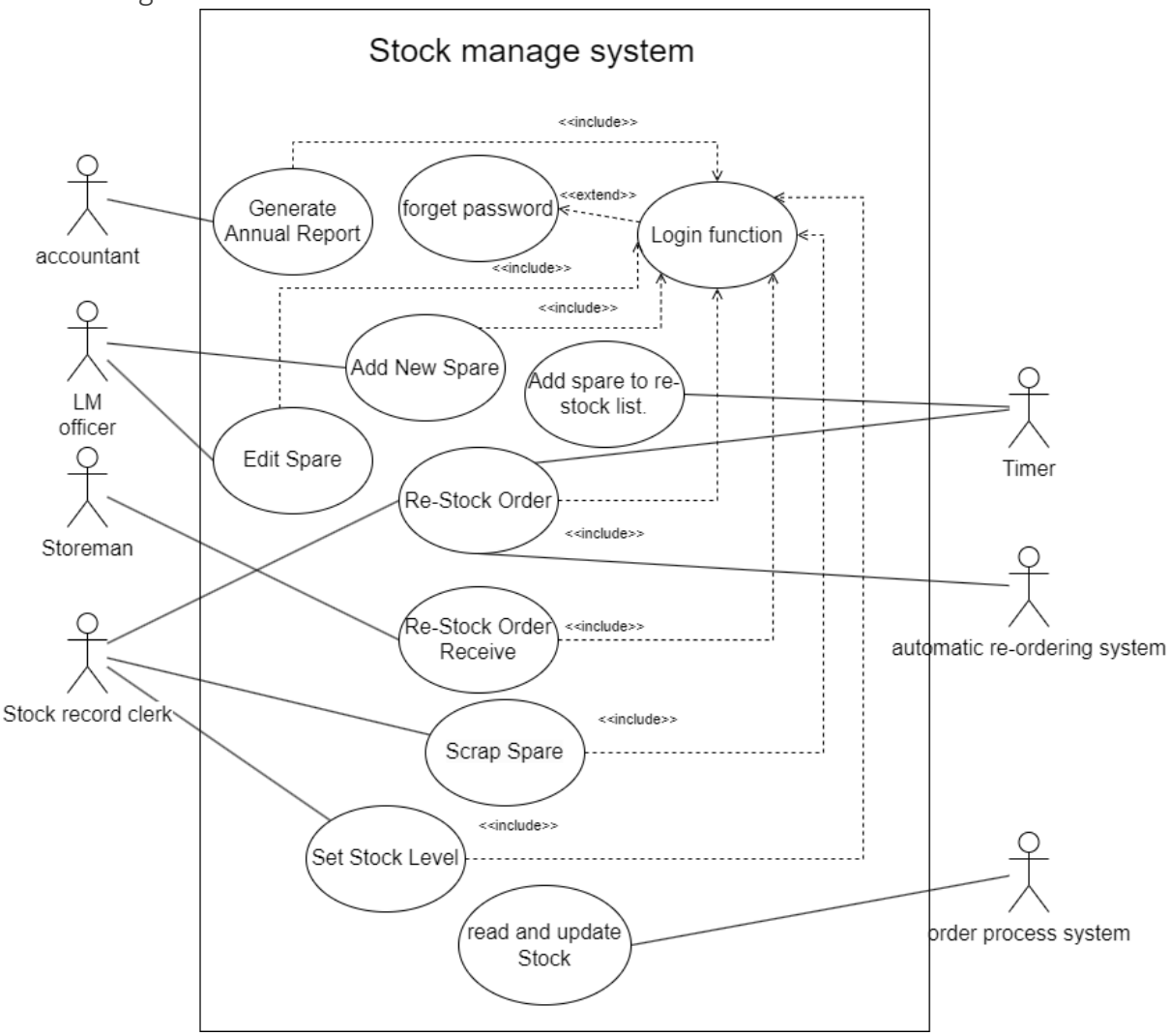
Alternative flows and exceptions:	
Non-behavior requirements:	

11.2.16. edit order number

Use case name:	edit order number
Use case ID:	OPSRS-50055
Primary actor:	Dealer
Secondary actor(s):	
Brief description:	Dealers edit order number.
Preconditions:	if user click edit your order number button
Flow of events:	1: user click edit your order number 1.1 display dealer's order number 2: user edit dealer's order number 2.1: display successful or fail message
Postconditions:	Order number will be change.
Alternative flows and exceptions:	
Non-behavior requirements:	

11.3.Stock manage system

11.3.1.Diagram



11.3.2.Generate Annual Report

Use case name:	Generate Annual Report
Use case ID:	OPSRS-50002
Primary actor:	accountant
Secondary actor(s):	
Brief description:	Accountant using the system Generate a report.
Preconditions:	Accountant who has an account in the system.
Flow of events:	Invoke use case "Login function." 1: user click Report 1.1: System change the page to Report 2: user select period and report type 2.1: System display the preview 3: user click Output button 3.1: print a review
Postconditions:	A report will be generated
Alternative flows and exceptions:	
Non-behavior requirements:	The version of the database used to generate the report is displayed for reference.

11.3.3.Add New Spare

Use case name:	Add New Spare
Use case ID:	OPSRS-50004
Primary actor:	LM officer
Secondary actor(s):	

Brief description:	LM officer add new spare.
Preconditions:	Staff who have an account in the system.
Flow of events:	Invoke use case “Login function.” 1: user click Add New Spare 1.1: System change the page to Add New Spare 2: user enter information of the new spare and click Save button 2.1: System display succeed or fail
Postconditions:	New Spare will be Add.
Alternative flows and exceptions:	
Non-behavior requirements:	Inputs are required to ensure that there are no abnormalities.
Assumptions:	System response time should not exceed two seconds

11.3.4.Re-Stock Order

Use case name:	Re-Stock Order
Use case ID:	OPSRS-50014
Primary actor:	Stock record clerk
Secondary actor(s):	automatic re-ordering system
Brief description:	Stock record clerk send the Re-order card to replenishment.
Preconditions:	Dealer has an account in the system.
Flow of events:	Invoke use case “Login function.” 1: user go to restock 1.1: System display the Re-Stock list

	2: user select a record 2.1: System displays selected 3: user click the arrow 3.1: System update the record to box 4: user click confirm button 4.1: System create a re-stock order.
Postconditions:	A re-stock order will be sent to the supplier.
Alternative flows and exceptions:	Step 2-3.1 can be repeated.
Non-behavior requirements:	

11.3.5.Re-Stock Order Receive

Use case name:	Re-Stock Order Receive
Use case ID:	OPSRS-50061
Primary actor:	Storeman
Secondary actor(s):	
Brief description:	Storeman receive re-stock spare.
Preconditions:	Storeman has an account in the system.
Flow of events:	Invoke use case “Login function.” 1: user go to Re-stock Check 1.1: System display the Re-Stock list 2: user select a record 2.1: System displays selected 3: user click Received 3.1: System display the new state in record.

Postconditions:	The state will update.
Alternative flows and exceptions:	Step 2-3.1 can be repeated.
Non-behavior requirements:	

11.3.6.Add spare to re-stock list

Use case name:	Add spare to re-stock list
Use case ID:	OPSRS-50053
Primary actor:	Timer
Secondary actor(s):	
Brief description:	Append spares to a list.
Preconditions:	[if stock level - follow quantity + restock order quantity <= ROL and autoRestock is true].
Flow of events:	1: append this spare to a list
Postconditions:	A spare will be add to a re-stock list.
Alternative flows and exceptions:	All Step can be repeated.
Non-behavior requirements:	

11.3.7.Scrap Spare

Use case name:	Scrap Spare
Use case ID:	OPSRS-50028
Primary actor:	Stock Record Clerk

Secondary actor(s):	
Brief description:	Stock Record Clerk scrap spares.
Preconditions:	Have an account.
Flow of events:	Invoke use case “Login function.” 1: user click Scrap 1.1: System change the page to Scrap. 2: Users enter quantity, discrepancy, selected spare id and click Record button 2.1: System display succeed or fail message
Postconditions:	The number of spare stocks in database will be decrease
Alternative flows and exceptions:	
Non-behavior requirements:	

11.3.8.Set Stock Level

Use case name:	Set Stock Level
Use case ID:	OPSRS-50060
Primary actor:	Stock Record Clerk
Secondary actor(s):	
Brief description:	Stock Record Clerk set stock level
Preconditions:	Have an account.
Flow of events:	Invoke use case “Login function.” 1: User click Set Stock Level

	1.1: System change the page to Set Stock Level 2: Users select a spare ID 2.1: System display the spare name. If ROL and CSL more than 0, will display them. 3: Users enter re-order level, common stock level, selected spare id and click Save button 3.1: System display succeed or fail message
Postconditions:	A spare's stock level will be set.
Alternative flows and exceptions:	
Non-behavior requirements:	

11.3.9.Edit Spare

Use case name:	Edit Spare
Use case ID:	OPSRS-50080
Primary actor:	LM officer
Secondary actor(s):	
Brief description:	LM officers edit a spare
Preconditions:	Have an account.
Flow of events:	Invoke use case "Login function." 1: user click Edit Spare 1.2: change the page to Edit and Delete Spare 2: user selcet the target spare and click Edit 2.1: display text boxs and drop-down list for enter 3: user enter spare attribute and click Save

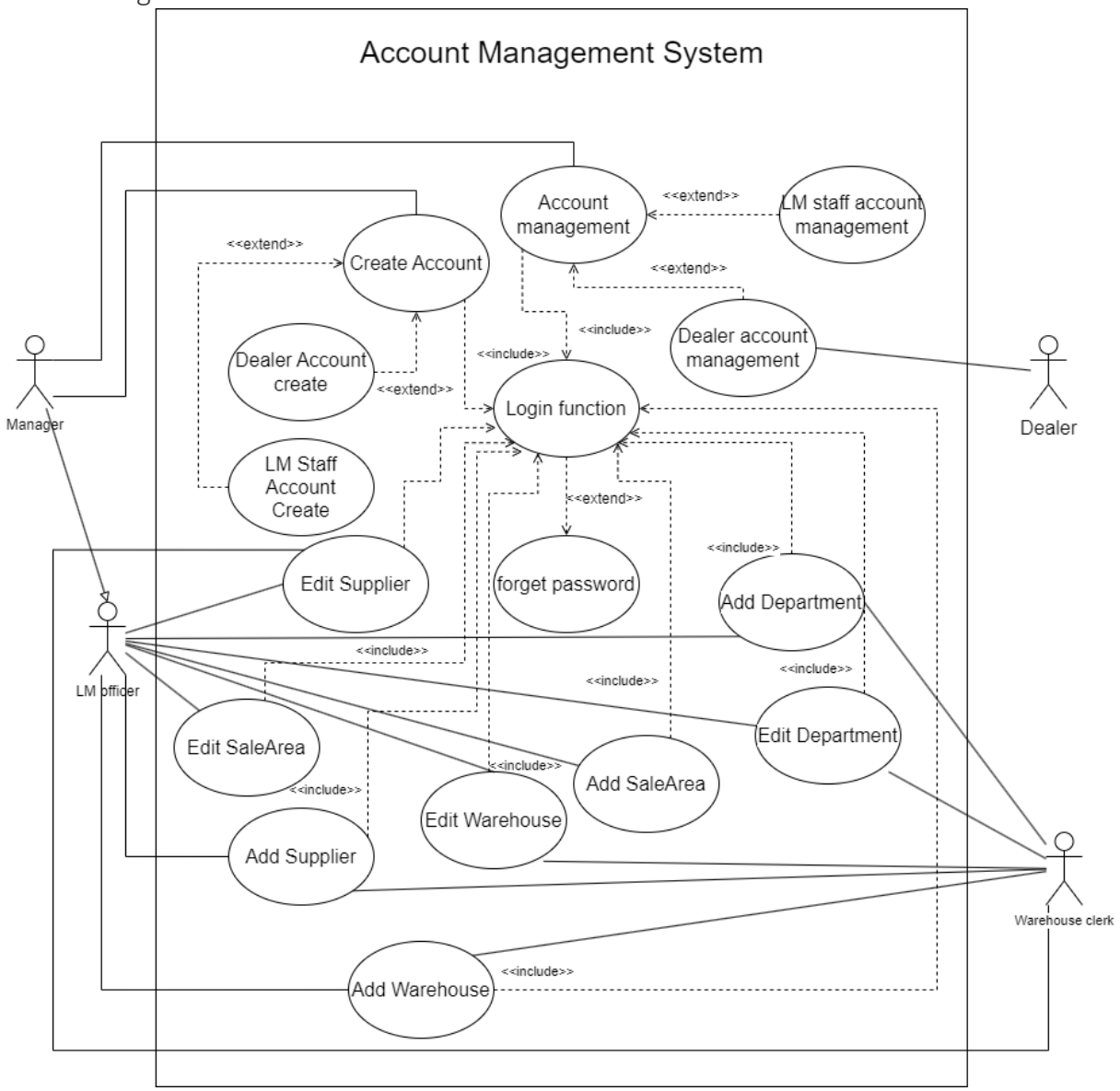
	3.1: display succeed or fail message
Postconditions:	A spare's stock level will be set.
Alternative flows and exceptions:	
Non-behavior requirements:	

11.3.10. read and update stock

Use case name:	read and update stock
Use case ID:	OPSRS-50081
Primary actor:	order process system
Secondary actor(s):	
Brief description:	Order process system read and update Stock.
Preconditions:	
Flow of events:	1: read stock 2: update stock
Postconditions:	Stock record will be
Alternative flows and exceptions:	
Non-behavior requirements:	

11.4.Account Management System

11.4.1.Diagram



11.4.2.Account management

Use case name:	Account management
Use case ID:	OPSRS-50011
Primary actor:	manager
Secondary actor(s):	
Brief description:	Manager Viewing or Changing Staff account Properties
Preconditions:	Manager who has an account in the system.
Flow of events:	Invoke use case “Login function.” 1. User click User Management 1.1: change the page to User and Account Create or reject 2: user click Manage exits account 2.1: change the page to Account Management 3: When user select a user account], execute use case “LM staff account management” 4: When user select a dealer account, execute use case “Dealer account management”
Postconditions:	
Alternative flows and exceptions:	
Non-behavior requirements:	Only valid attributes will be accepted

11.4.3.Dealer account management

Use case name:	Dealer account management
Use case ID:	OPSRS-50010

Primary actor:	Manager, Dealer
Secondary actor(s):	
Brief description:	Manager Viewing or Changing dealer account Properties
Preconditions:	Manager who has an account in the system.
Flow of events:	1: user select Dealer account and click Edit 1.1: System display a list of sale area and text boxes 2: user enter attributes and click Save button 2.1: System display succeed or fail message
Postconditions:	
Alternative flows and exceptions:	Can be left unchanged. Only valid attributes will be accepted
Non-behavior requirements:	

11.4.4.LM staff account management

Use case name:	LM staff account management
Use case ID:	OPSRS-50058
Primary actor:	manager
Secondary actor(s):	
Brief description:	Manager Viewing or Changing LM staff account Properties
Preconditions:	Manager who has an account in the system.
Flow of events:	1: user select Our Staff account and click Edit 1.1: System display drop-down list of sale area, department, position, and text boxes 2: user enter attributes and click Save button

	2.1: System display succeed or fail message
Postconditions:	
Alternative flows and exceptions:	Can be left unchanged. Only valid attributes will be accepted
Non-behavior requirements:	

11.4.5.Create Account

Use case name:	Create Account
Use case ID:	OPSRS-50001
Primary actor:	Manager
Secondary actor(s):	
Brief description:	Managers create a user account for all employees under their position.
Preconditions:	An employee is a full-time employee or an employee without an account.
Flow of events:	Invoke use case “Login function.” 1: user click user Management 1.1: System change the page to User and Account Create 2: user Enter login name, access level, click Dealer or LM staff radio button and click Create a new account button. 2.1: If user click Dealer, execute use case “Dealer Account create”, else execute use case “LM Staff Account Create”

Postconditions:	A new user account is created.
Alternative flows and exceptions:	If an account already exists, the execution of the use case terminates
Non-behavior requirements:	

11.4.6.Dealer Account create

Use case name:	Dealer Account create
Use case ID:	OPSRS-50072
Primary actor:	Manager
Secondary actor(s):	
Brief description:	Managers create a Dealer Account.
Preconditions:	Managers have an account in the system.
Flow of events:	1: System change the page to New Dealer 2: user click Yes and enter attribute or click NO and select a dealer id. Last Click Save. 2.1: System display succeed or fail
Postconditions:	A new Dealer account is created.
Alternative flows and exceptions:	If an account already exists, the execution of the use case terminates
Non-behavior requirements:	

11.4.7.LM Staff Account create

Use case name:	LM Staff Account create
Use case ID:	OPSRS-50073

Primary actor:	Manager
Secondary actor(s):	
Brief description:	Managers create a Dealer Account.
Preconditions:	Managers have an account in the system.
Flow of events:	1: System change the page to New Staff 2: user click Yes and enter attribute or click NO and select a staff id. Last Click Save. 2.1: System display succeed or fail
Postconditions:	A new LM Staff account is created.
Alternative flows and exceptions:	If an account already exists, the execution of the use case terminates
Non-behavior requirements:	

11.4.8.Edit Supplier

Use case name:	Edit Supplier
Use case ID:	OPSRS-50027
Primary actor:	Manager, LM officer
Secondary actor(s):	
Brief description:	Managers edit a supplier account.
Preconditions:	Manager who has a manager account.
Flow of events:	Invoke use case “Login function.” 1: user go to Supplier Management 1.1: System change the page to New Supplier 2: user click Edit exist Supplier

	2.1: System change the page to Edit Supplier 3:user select target supplier id and click Edit 3.1: System display text box for edit. 4: user enter Attributes and click Save 4.1: display succeed or fail message
Postconditions:	
Alternative flows and exceptions:	
Non-behavior requirements:	Only valid attributes will be accepted.

11.4.9.Add Supplier

Use case name:	Add Supplier
Use case ID:	OPSRS-50057
Primary actor:	Manager, LM officer
Secondary actor(s):	
Brief description:	Managers add a supplier.
Preconditions:	Manager who has a manager account.
Flow of events:	Invoke use case “Login function.” 1: user go to Supplier Management 1.1: System change the page to “New Supplier” 2: user enter name, email, contact number, address, and click Create a new supplier button 2.1: display succeed or fail message
Postconditions:	A new supplier will be added.
Alternative flows and exceptions:	

Non-behavior requirements:	Only valid attributes will be accepted.

11.4.10. Add SaleArea

Use case name:	Add SaleArea
Use case ID:	OPSRS-50032
Primary actor:	Manager, LM officer
Secondary actor(s):	
Brief description:	Managers have added a new SaleArea.
Preconditions:	Manager who has a manager account.
Flow of events:	Invoke use case "Login function." 1: user click Add SaleArea 1.1: change the page to Add Salearea 2: user enter location, remark and click Save button 2.1: display succeed or fail message
Postconditions:	A new sale area will be add.
Alternative flows and exceptions:	
Non-behavior requirements:	Only valid attributes will be accepted.

11.4.11. Edit SaleArea

Use case name:	Edit SaleArea
Use case ID:	OPSRS-50033
Primary actor:	Manager, LM officer
Secondary actor(s):	

Brief description:	Managers Edit a SaleArea.
Preconditions:	Manager who has a manager account.
Flow of events:	Invoke use case "Login function." 1: user click Edit SaleArea 1.1: System change the page to Edit Salearea 2: user select the target sale area and click Edit button 2.1: display tow text box for enter location and remark 3: select sale area, enter location, and remark and click Save 3.1: System display succeed or fail message
Postconditions:	A sale area will be edit.
Alternative flows and exceptions:	
Non-behavior requirements:	Only valid attributes will be accepted.

Use case name:	Add Department
Use case ID:	OPSRS-50034
Primary actor:	Manager, Warehouse clerk, LM officer
Secondary actor(s):	
Brief description:	Managers add a new department
Preconditions:	Manager who has a manager account.
Flow of events:	Invoke use case "Login function." 1: user click Add Department

	1.1: System change the page to Add Department 2: user enter name, email of department, and click Yes radio button and select warehouse or click No radio button. Last click Save. 2.1: System display succeed or fail message
Postconditions:	A new department will be added.
Alternative flows and exceptions:	
Non-behavior requirements:	Only valid attributes will be accepted.

Use case name:	Edit Department
Use case ID:	OPSRS-50035
Primary actor:	Manager, Warehouse clerk, LM officer
Secondary actor(s):	
Brief description:	Managers edit department data.
Preconditions:	Manager who has a manager account.
Flow of events:	Invoke use case “Login function.” 1: user click Edit Department 1.1: System change the page to Edit Department 2: user select the target department and click edit 2.1: System display a textbox for enter name and email of department 3: user enter name, email, and click save 3.1: System display succeed or fail message
Postconditions:	A department will be edit.

Alternative flows and exceptions:	
Non-behavior requirements:	Only valid attributes will be accepted.

Use case name:	Add Warehouse
Use case ID:	OPSRS-50036
Primary actor:	Manager, Warehouse clerk, LM officer
Secondary actor(s):	
Brief description:	Managers add a new warehouse.
Preconditions:	Manager who has a manager account.
Flow of events:	Invoke use case “Login function.” 1: user click Add Warehouse 1.1: change the page to Add Warehouse 2: user select the target sale area, manager, and enter location, click save button 2.1: System display succeed or fail message
Postconditions:	A new warehouse will be added.
Alternative flows and exceptions:	
Non-behavior requirements:	Only valid attributes will be accepted.

Use case name:	Edit Warehouse
Use case ID:	OPSRS-50037

Primary actor:	Manager, Warehouse clerk, LM officer
Secondary actor(s):	
Brief description:	Managers edit a warehouse.
Preconditions:	Manager who has a manager account.
Flow of events:	Invoke use case "Login function." 1: user click Edit Warehouse 1.1: System change the page to Edit Warehouse 2: user select the target warehouse and click edit 2.1: System display a textbox for enter location 3: user enter location and click save 3.1: System display succeed or fail message
Postconditions:	Edit Warehouse will be edit or delete.
Alternative flows and exceptions:	
Non-behavior requirements:	Only valid attributes will be accepted.

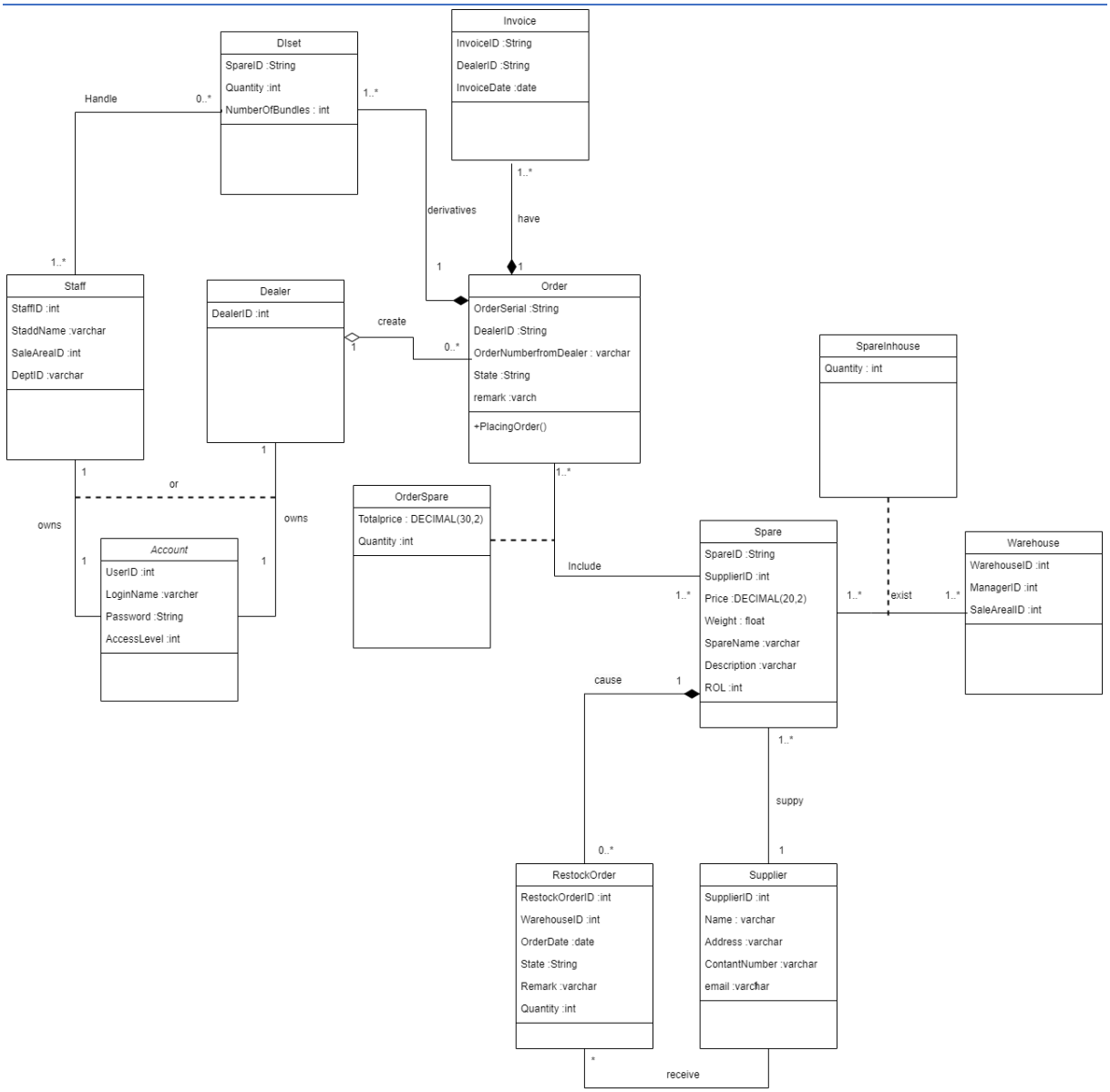
11.5.actor-use case table

	User	LM officer	Manager	accountant	Storemen	Warehouse clerk	Dealer	Delivery worker	Stock record clerk	automatic re-ordering system	Timer	order process system
Login function	✓	✓	✓	✓	✓	✓	✓	✓	✓			
forget password	✓	✓	✓	✓	✓	✓	✓	✓	✓			
create list for delivery											✓	
Browse catalogue							✓					
Place An Order							✓					
Check Order Detail and Review Order Status	✓	✓	✓	✓	✓	✓	✓	✓	✓			
Check More Detail	✓	✓	✓	✓	✓	✓	✓	✓	✓			
Download Invoice PDF							✓					
View Invoice and Sign								✓				
Check a DI set								✓				
Generate DIC and DID								✓				
Collect Spare					✓						✓	
Use Scanner					✓							
create list for collect spare											✓	
edit order						✓	✓					
edit order item							✓					
edit order number							✓					
Generate Annual Report				✓								
Add New Spare		✓	✓									

	User	LM officer	Manager	accountant	Storemen	Warehouse clerk	Dealer	Delivery worker	Stock record clerk	automatic re-ordering system	Timer	order process system
Re-Stock Order									✓	✓		
Re-Stock Order Receive					✓							
Add spare to re-stock list.											✓	
Scrap Spare									✓			
Set Stock Level									✓			
Edit Spare		✓	✓									
read and update Stock												✓
Account management			✓									
Dealer account management			✓				✓					
LM staff account management			✓									
Create Account			✓									
Dealer Account create			✓									
LM Staff Account create			✓									
Edit Supplier		✓	✓									
Add Supplier		✓	✓									
Add SaleArea		✓	✓									
Edit SaleArea		✓	✓									
Add Department		✓	✓			✓						
Edit Department		✓	✓			✓						
Add Warehouse		✓	✓			✓						

	User	LM officer	Manager	accountant	Storemen	Warehouse clerk	Dealer	Delivery worker	Stock record clerk	automatic re-ordering system	Timer	order process system
Edit Warehouse		✓	✓			✓						

12. Class Diagram

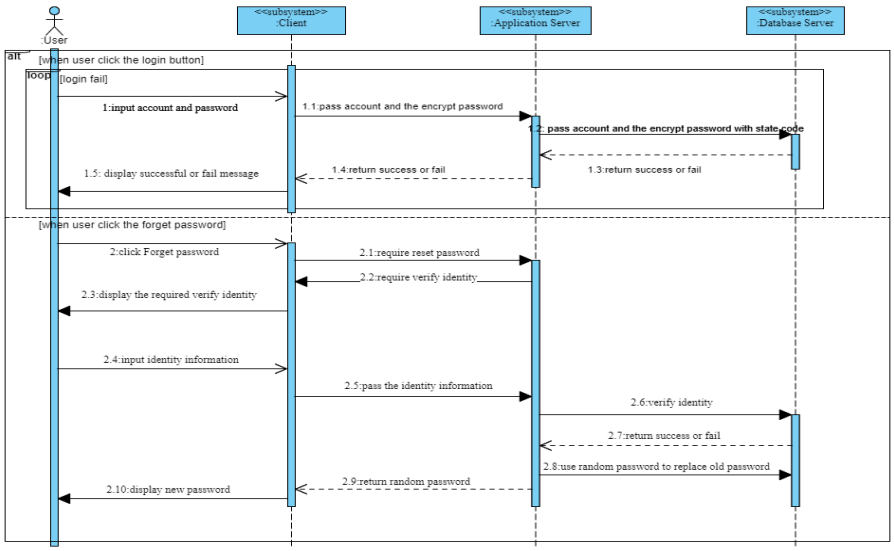


(refer to Class diagram.png)

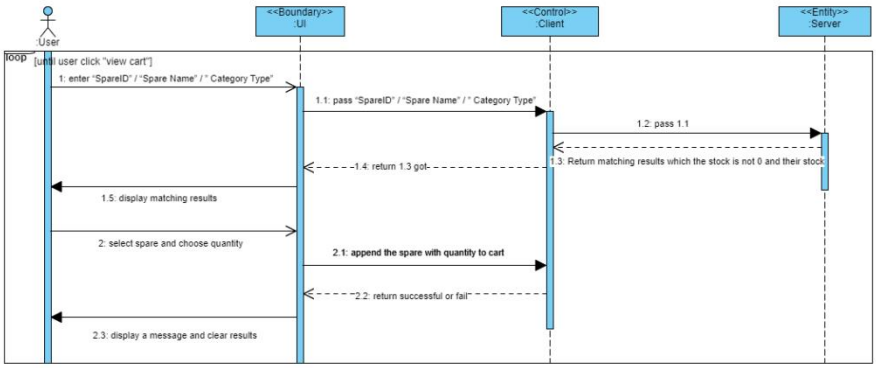
13. Sequence Diagram

(refer to Sequence Diagram.pdf)

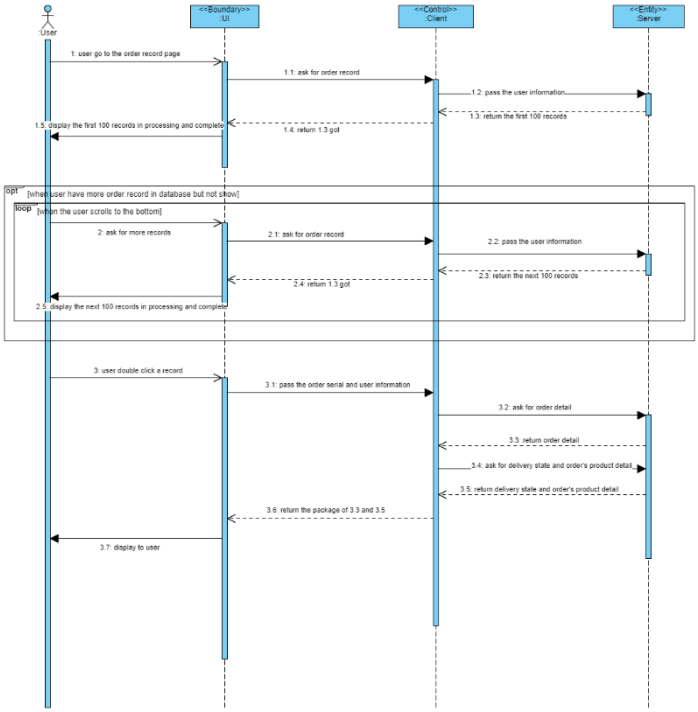
13.1.Login



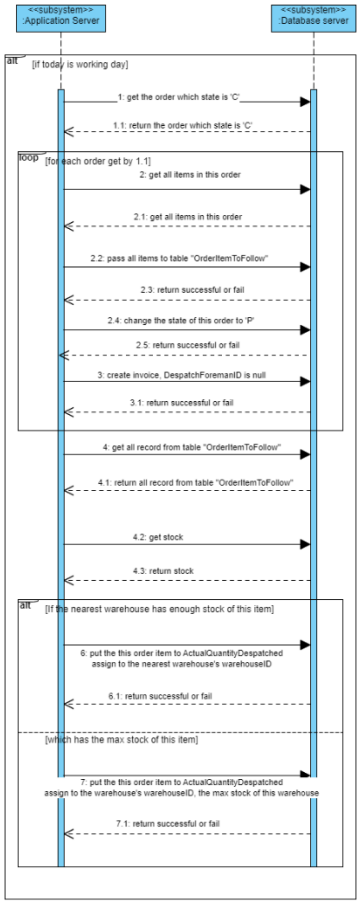
13.2. Browse catalogue



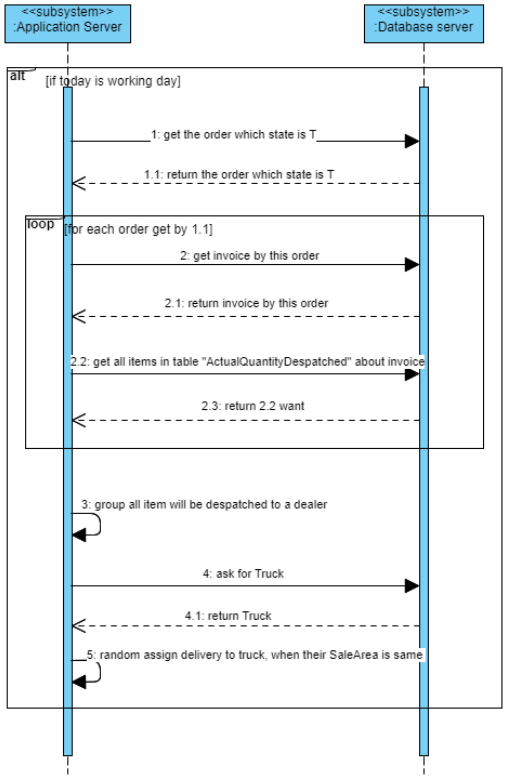
13.3. Check Order Detail and Review Order Status



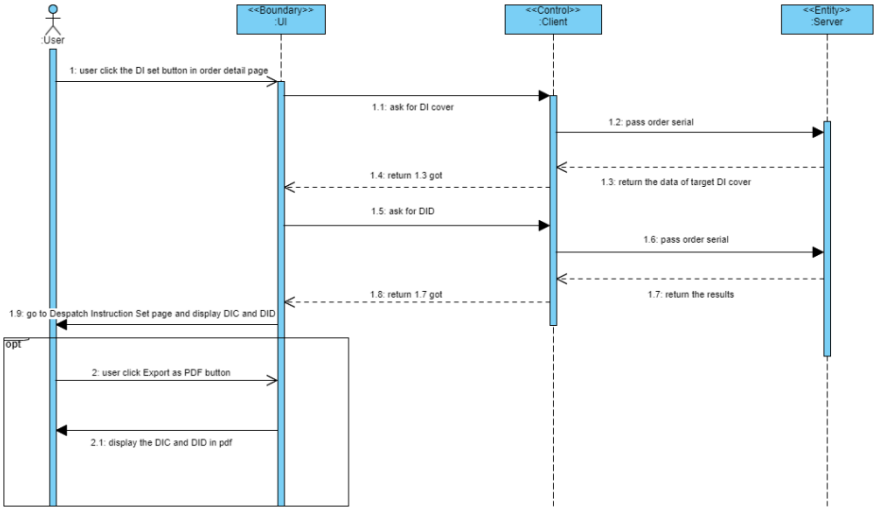
13.4. create list for collect spare



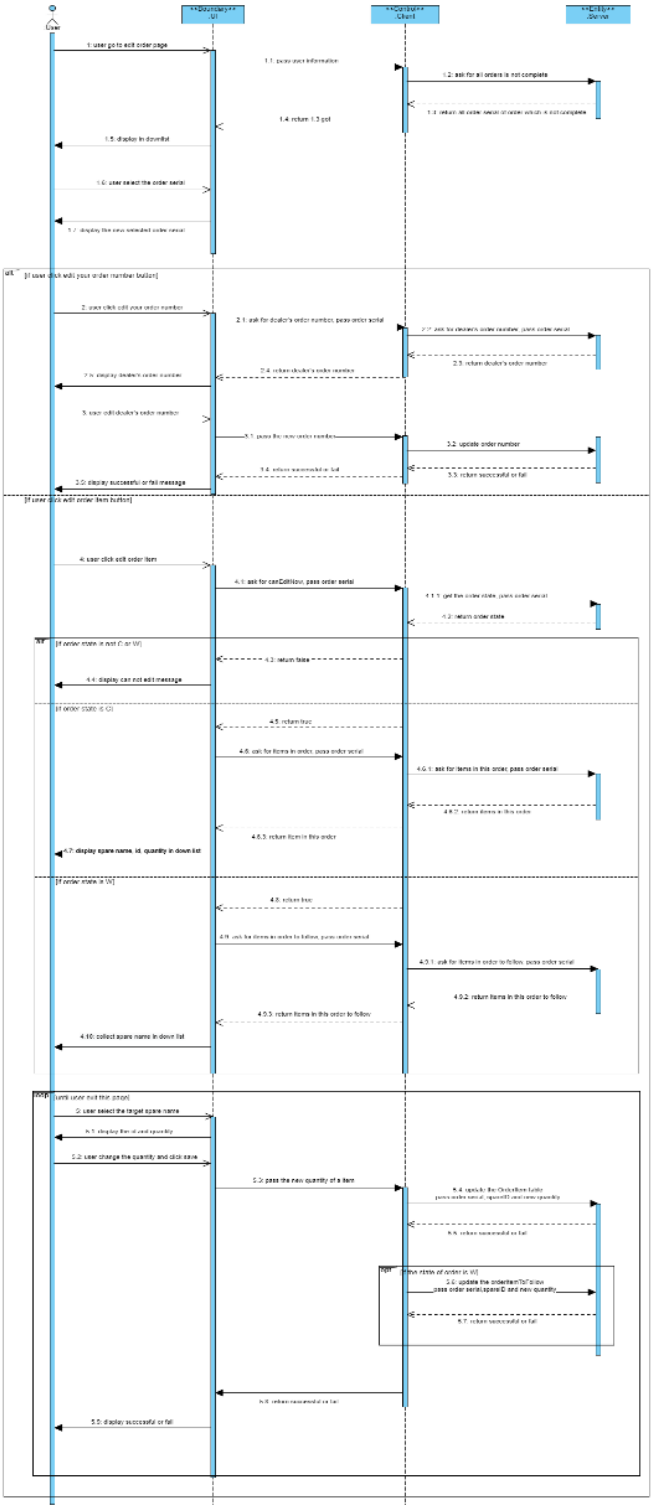
13.5. create list for delivery



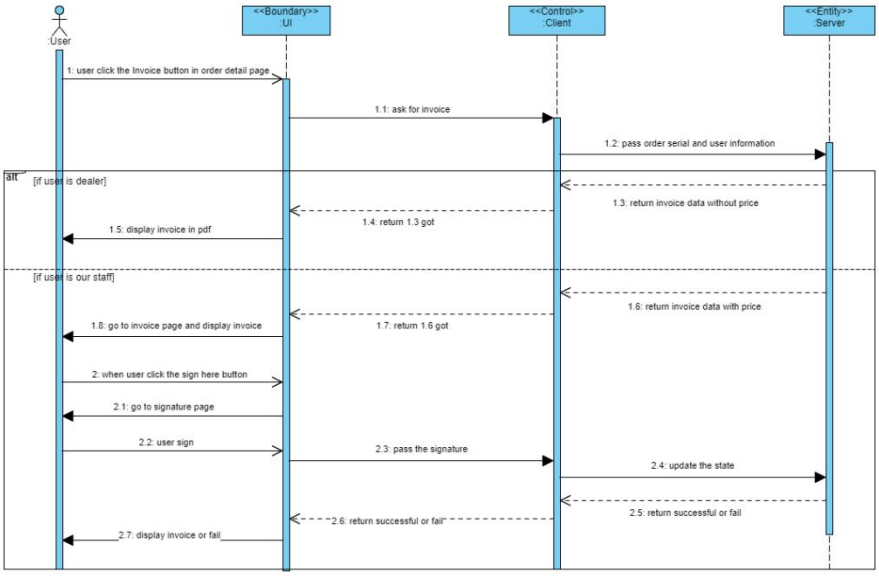
13.6. DI set


:User
<<Boundary>>
:UI
<<Control>>
:Client
<<Entity>>
:Server
1: user click the DI set button in order detail page
1.1: ask for DI cover
1.2: pass order serial
1.3: return the data of target DI cover
1.4: return 1.3 got
1.5: ask for DID
1.6: pass order serial
1.7: return the results
1.8: return 1.7 got
1.9: go to Despatch Instruction Set page and display DIC and DID
opt
2: user click Export as PDF button
2.1: display the DIC and DID in pdf

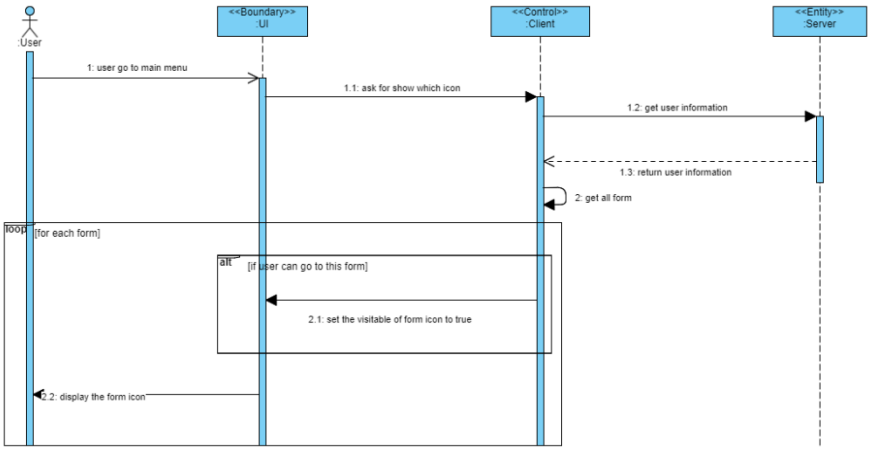
13.7. edit order



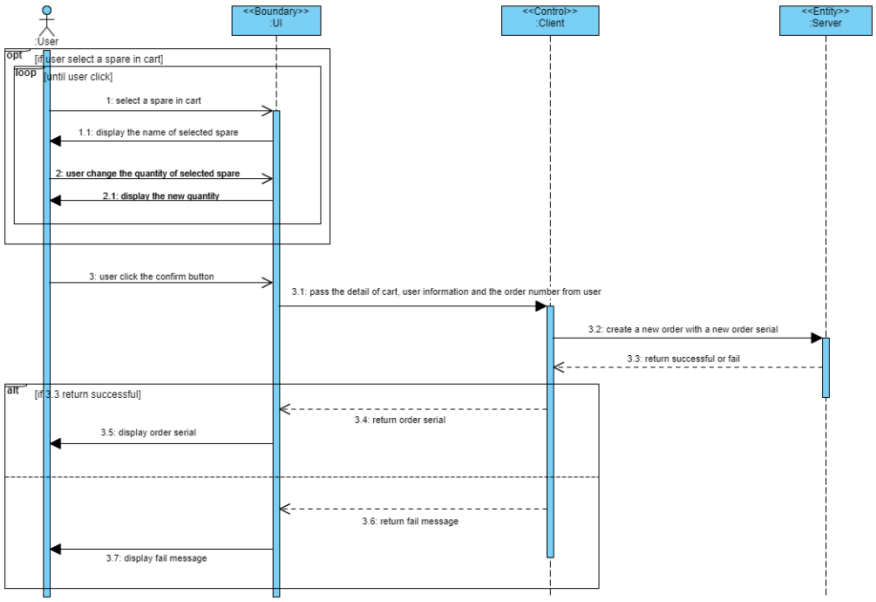
13.8. Invoice



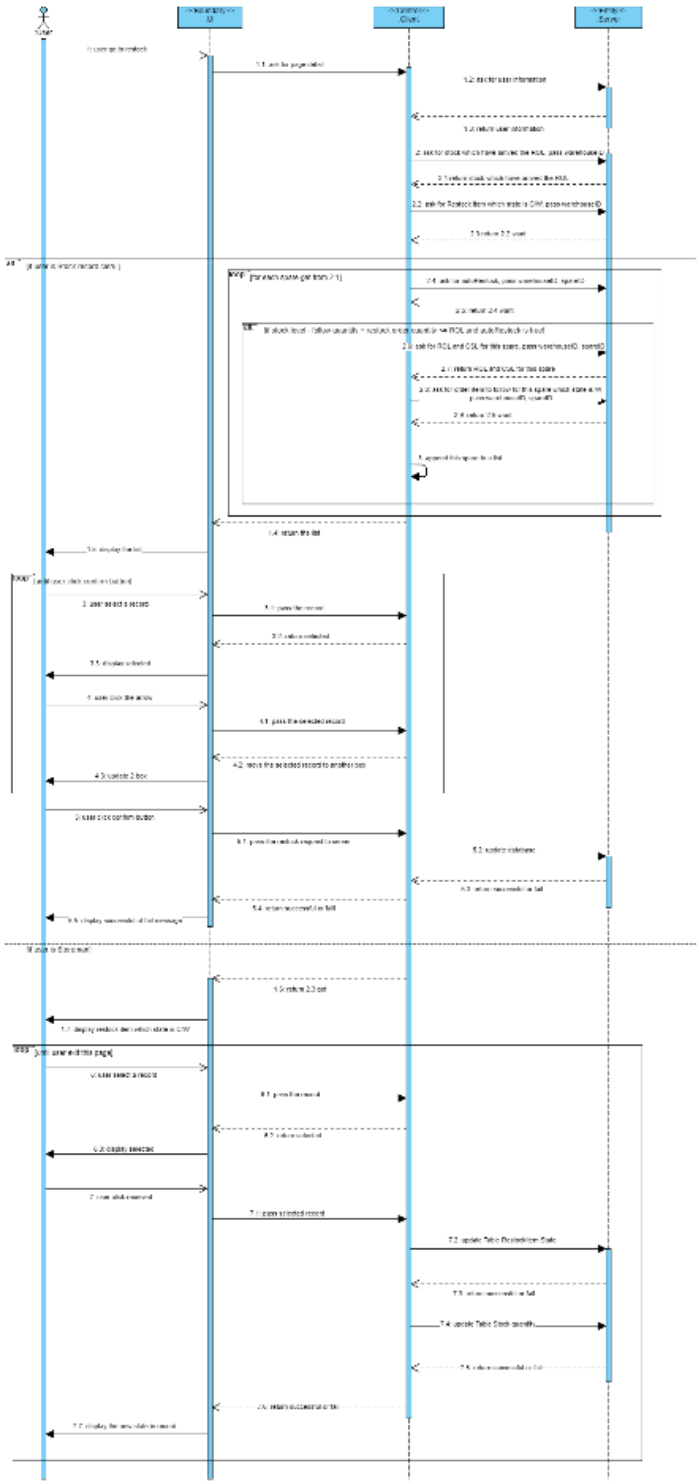
13.9. main menu



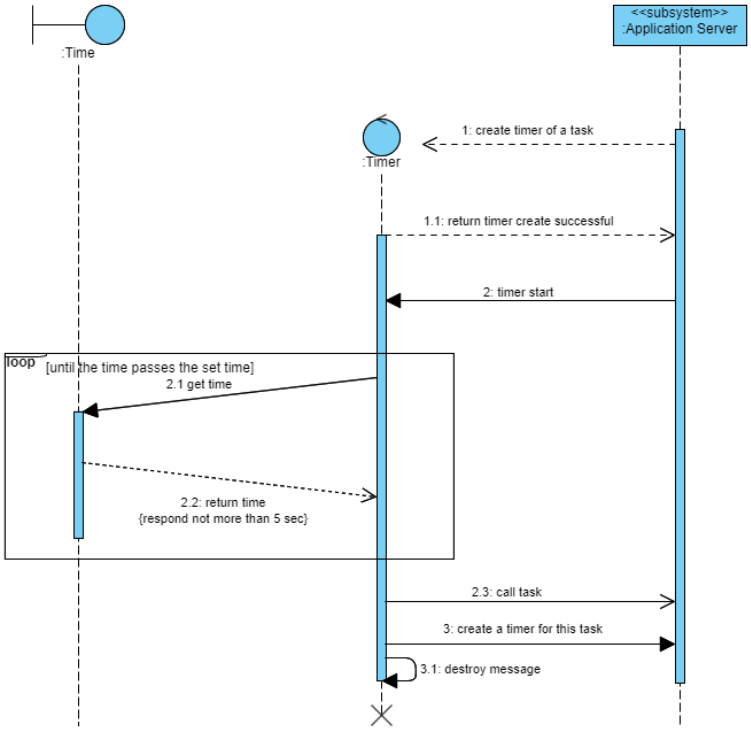
13.10. Place An Order



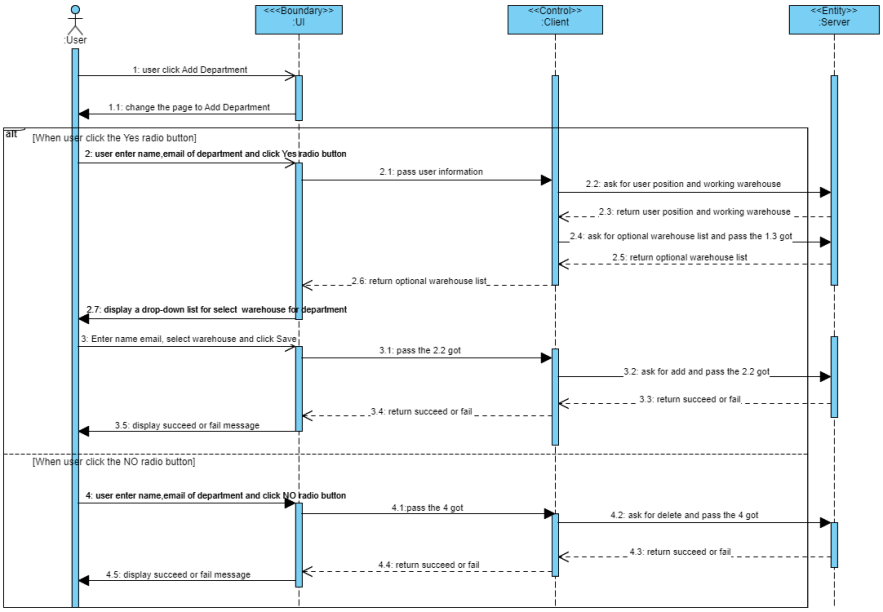
13.11.re-stock



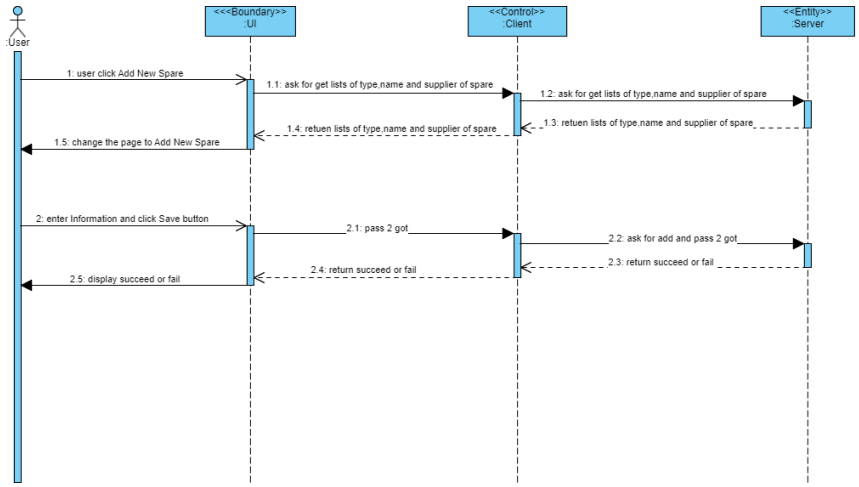
13.12.Timer


:Time
<<subsystem>>
:Application Server
:Timer
1: create timer of a task
1.1: return timer create successful
2: timer start
loop
[until the time passes the set time]
2.1 get time
2.2: return time
{respond not more than 5 sec}
2.3: call task
3: create a timer for this task
3.1: destroy message

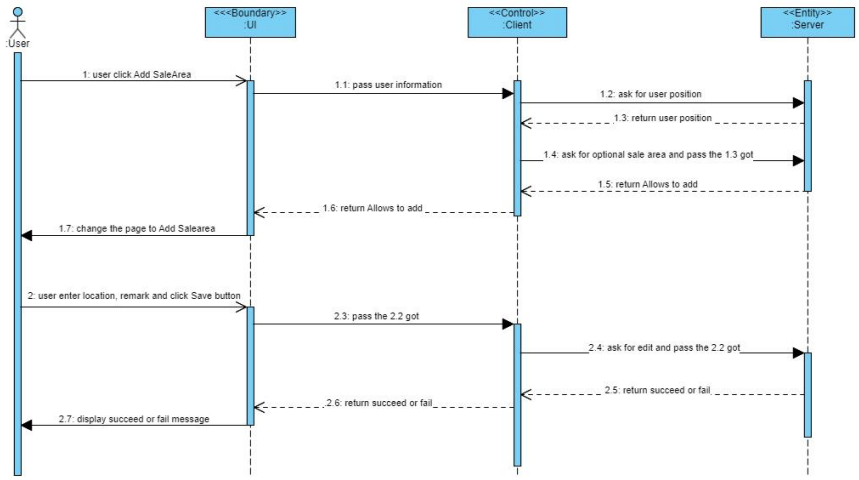
13.13.Add Department



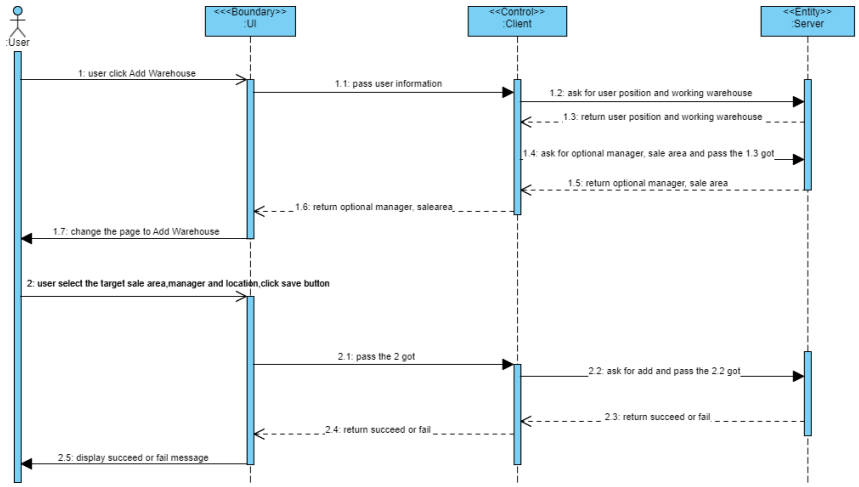
13.14. Add new spare



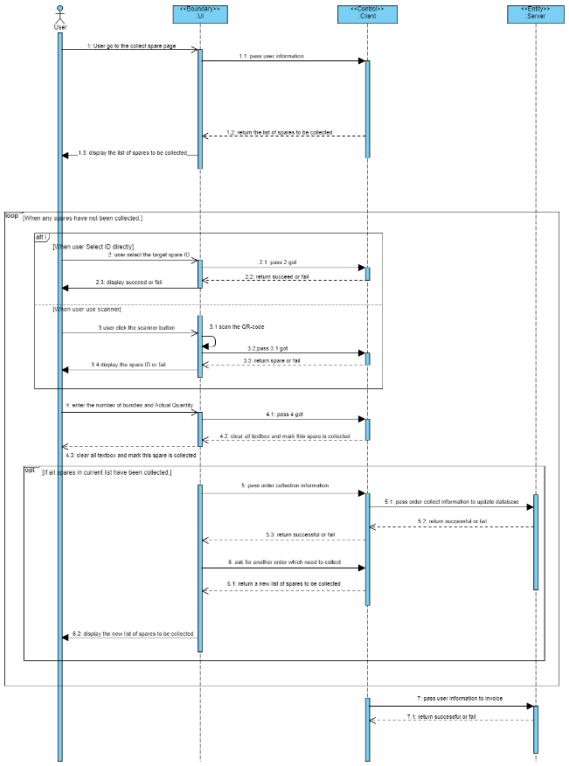
13.15. Add SaleArea



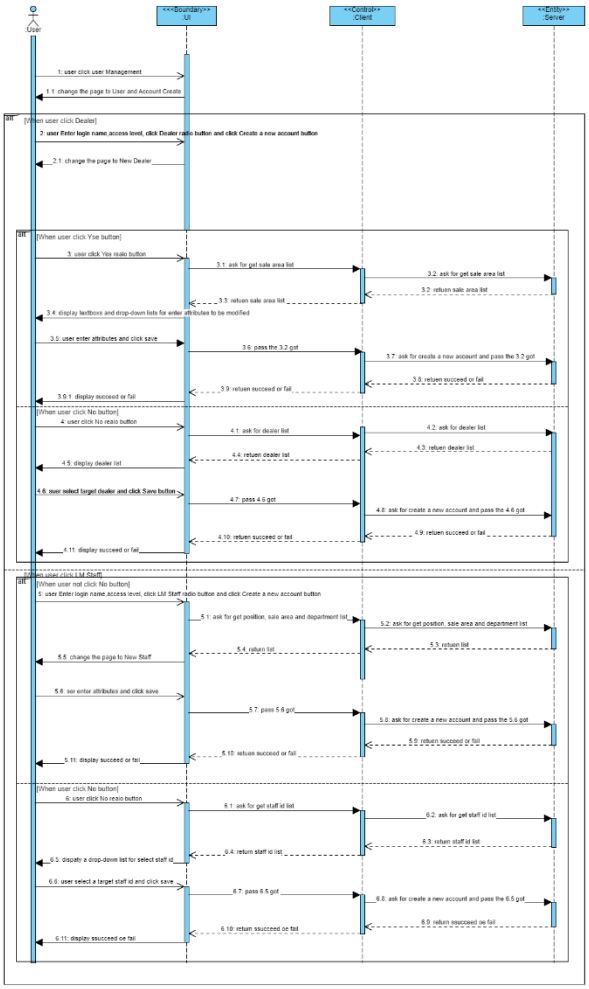
13.16. Add Warehouse



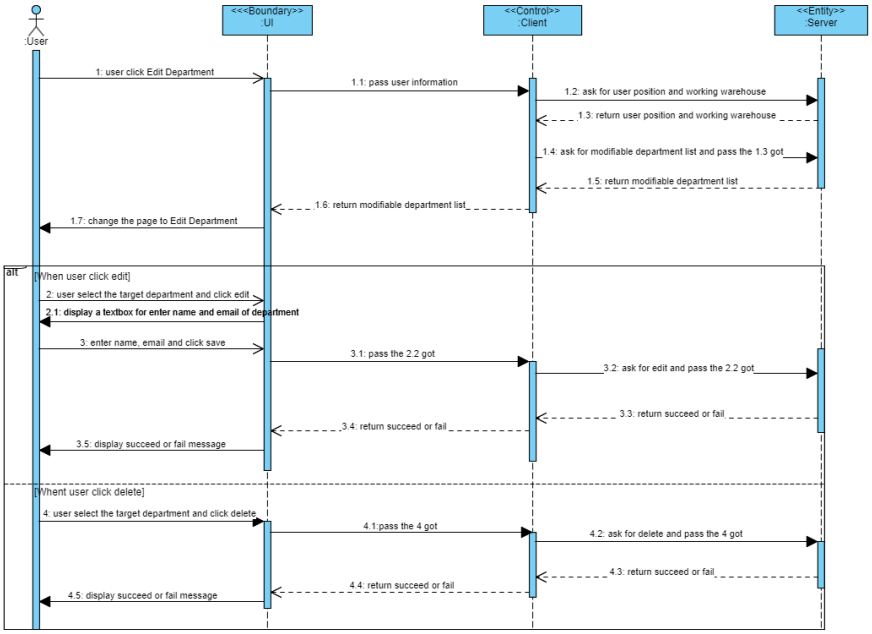
13.17.collect spare



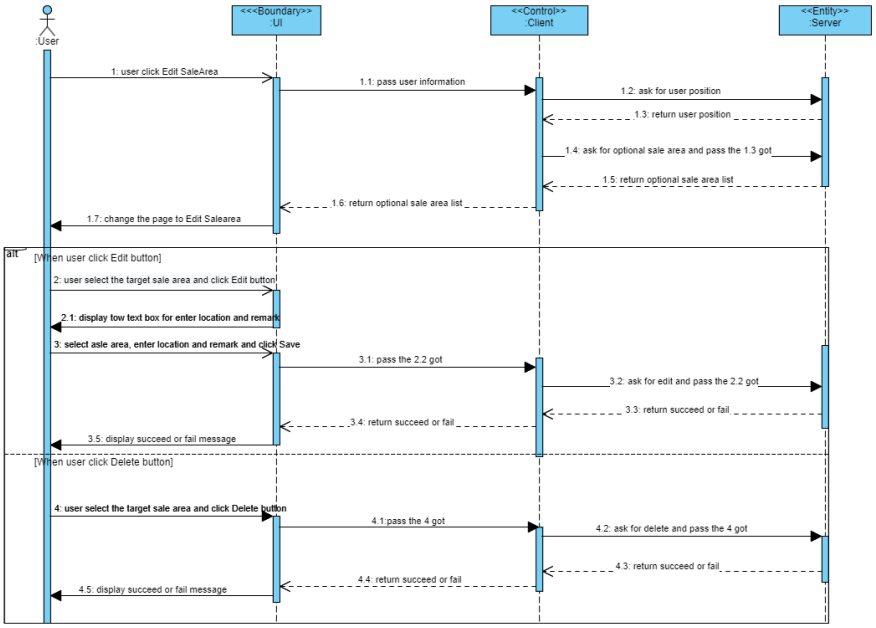
13.18. Create Account



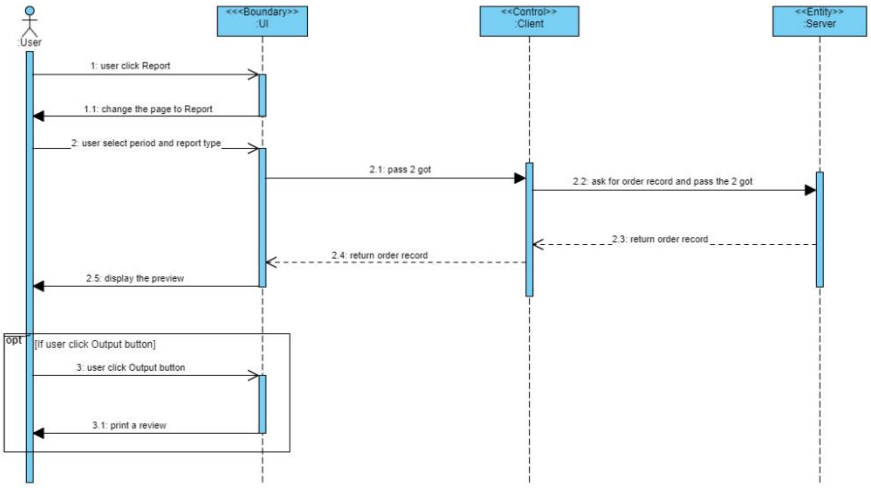
13.19. Edit Department



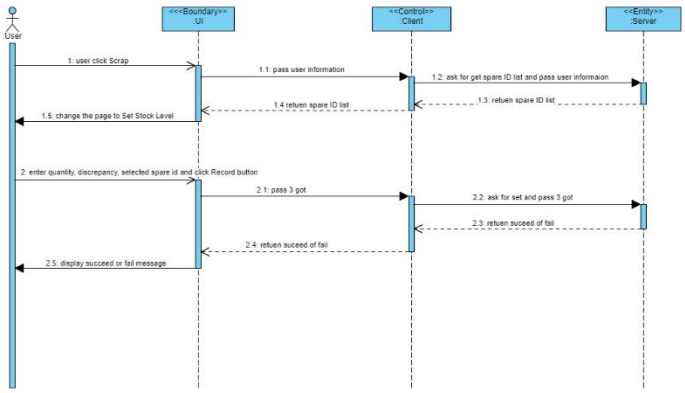
13.20. Edit SaleArea



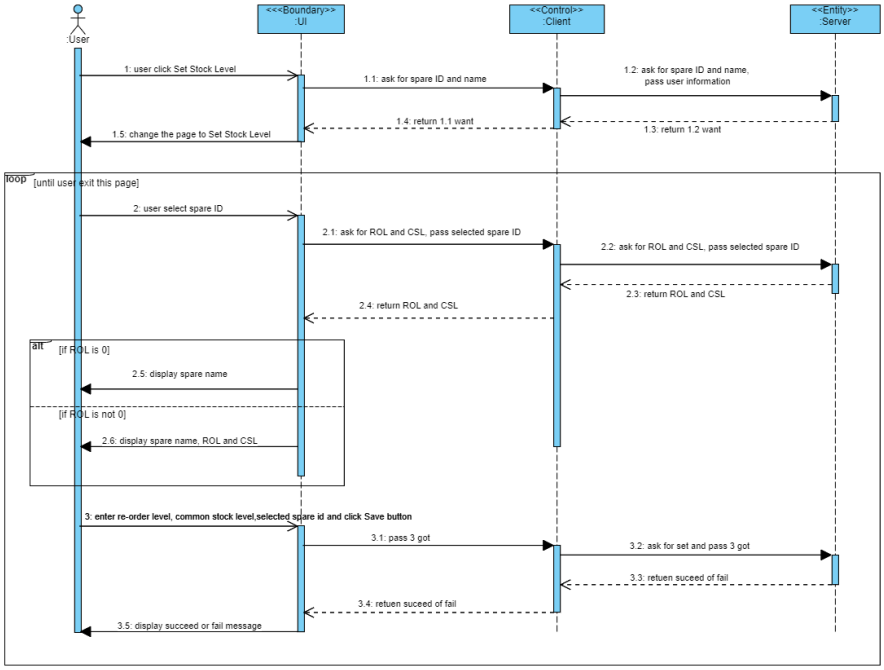
13.21.Generate Annual Report



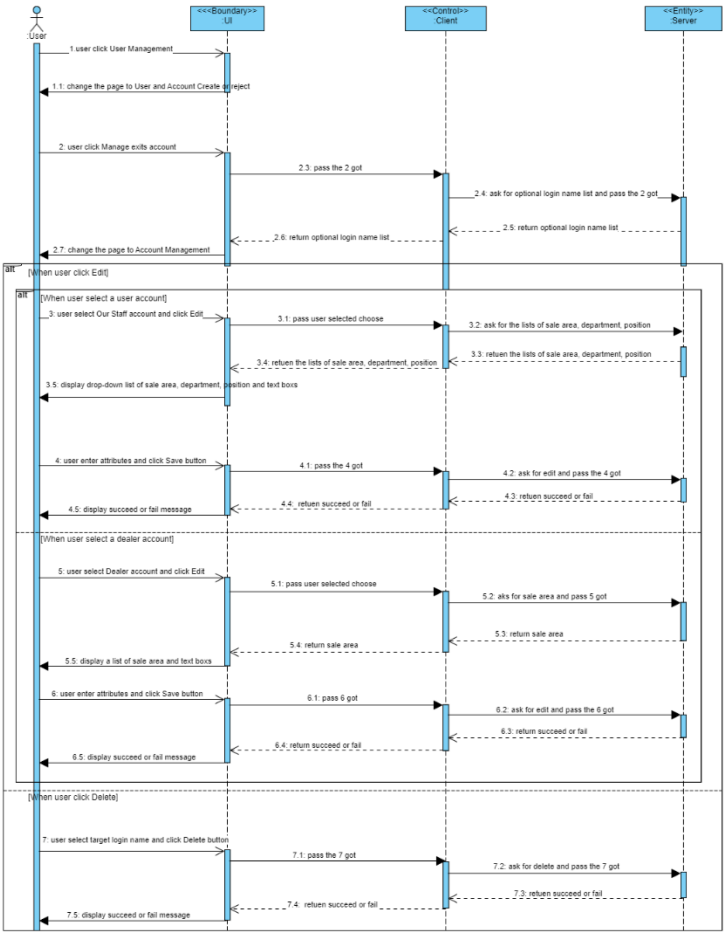
13.22.Scrap



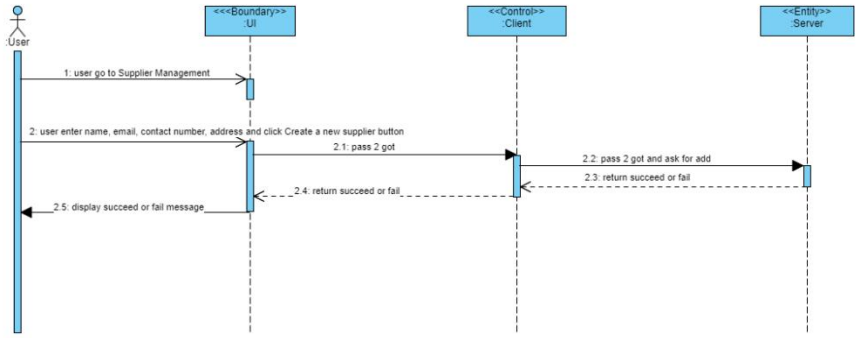
13.23.Set Stock Level



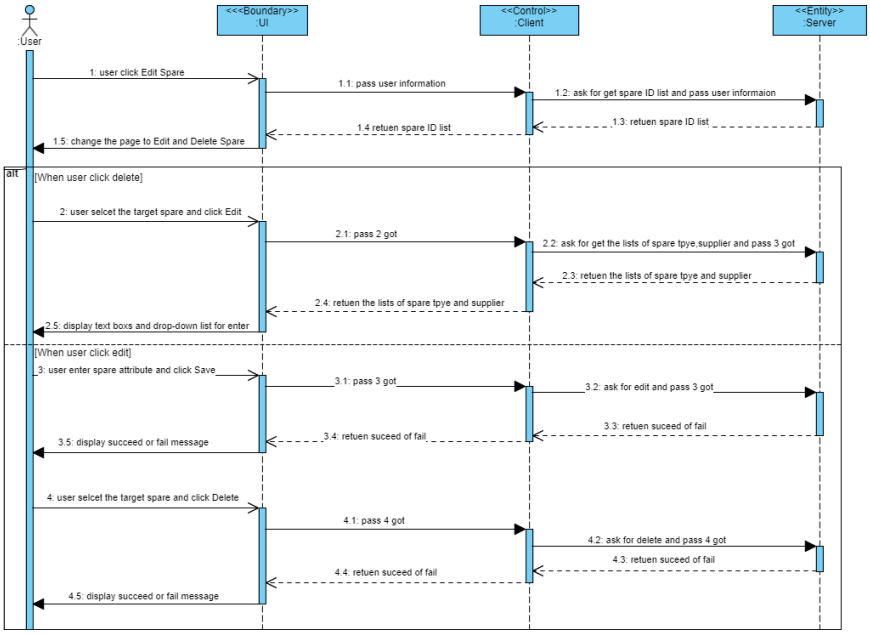
13.24.account management



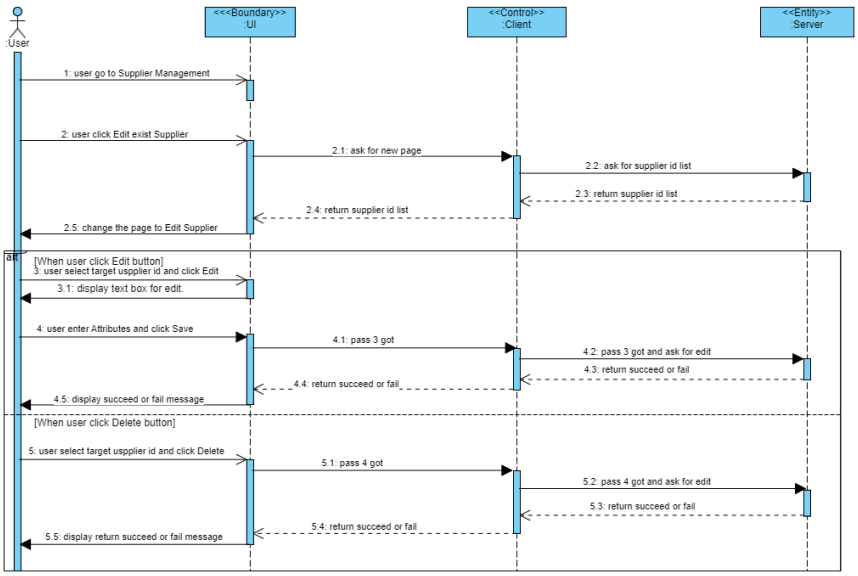
13.25. Add Supplier



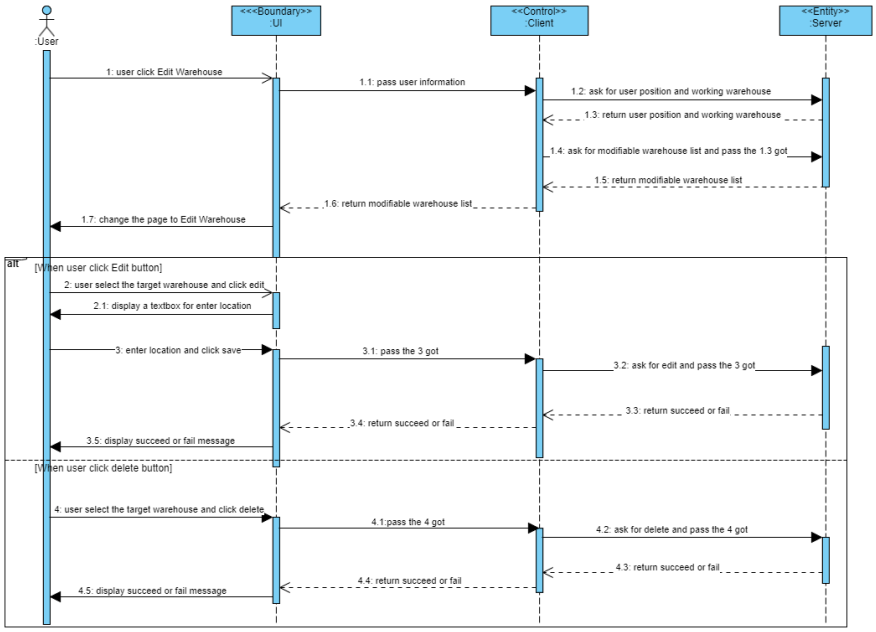
13.26. Edit or Delete Spare



13.27. Edit Supplier



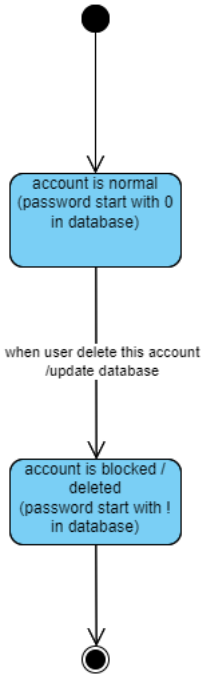
13.28. Edit Warehouse



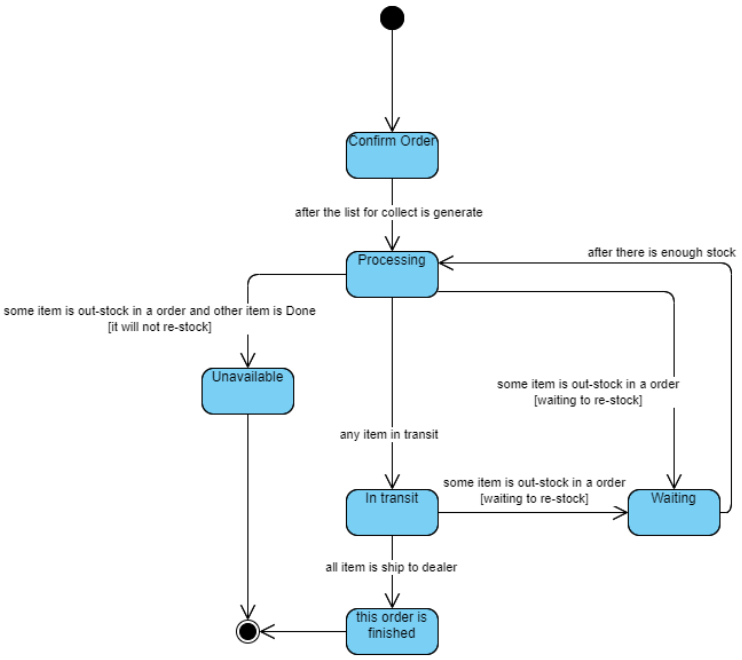
14. State Machine Diagram

(refer to State Machine Diagram.pdf)

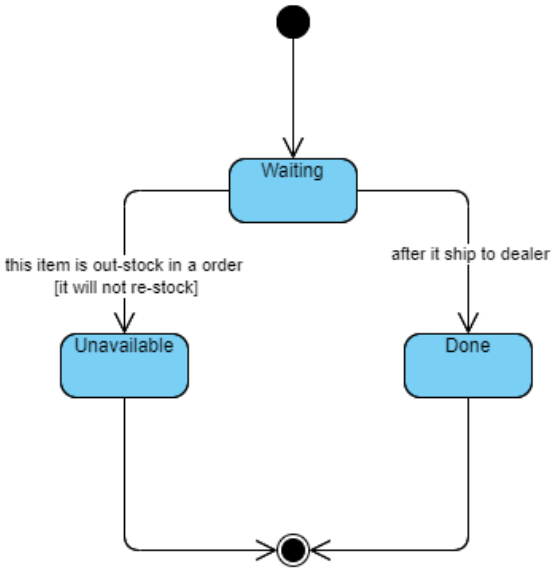
14.1. account state



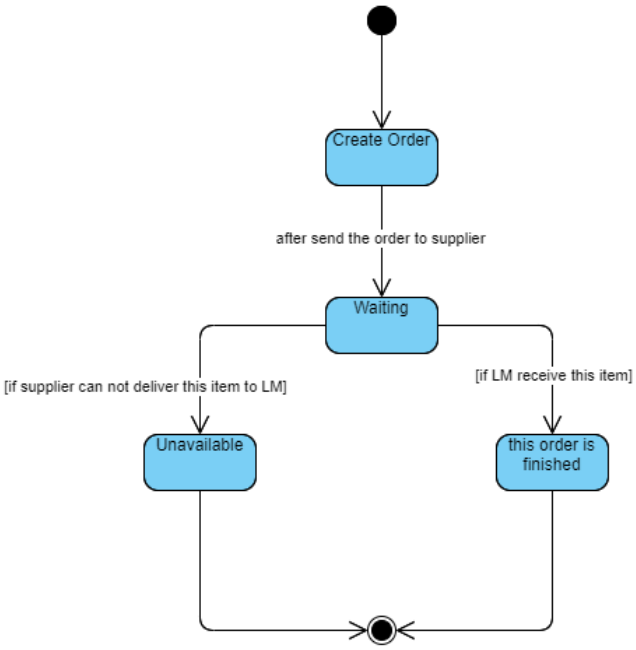
14.2. order state



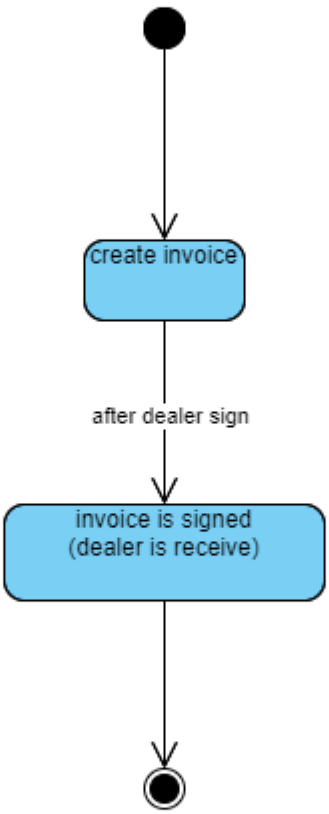
14.3. OrderItemToFollow state



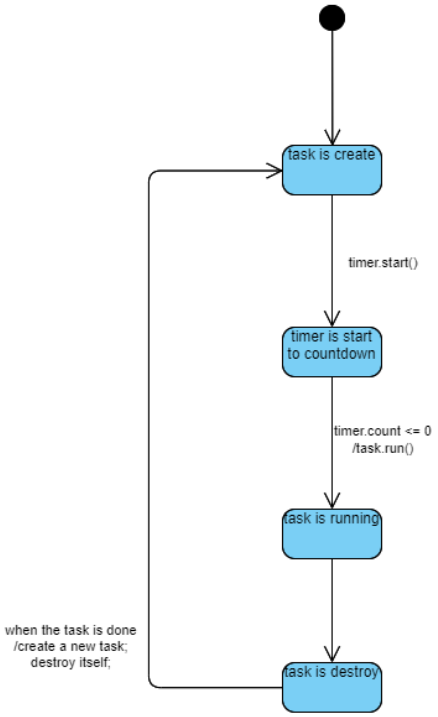
14.4. RestockOrder state



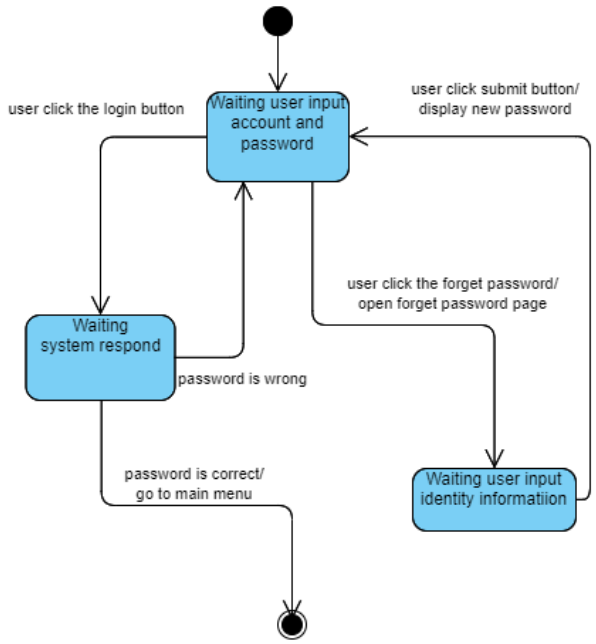
14.5. Invoice state



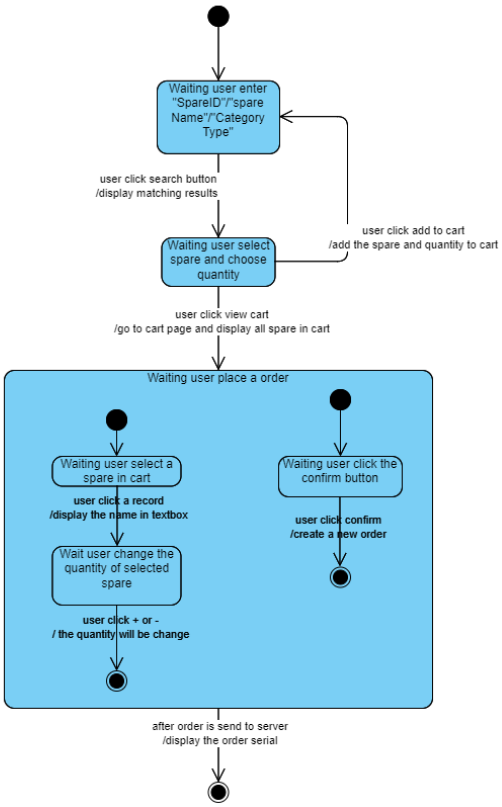
14.6. Timer


task is create
timer.start()
timer is start
to countdown
timer.count <= 0
/task.run()
task is running
when the task is done
/create a new task;
destroy itself;
task is destroy

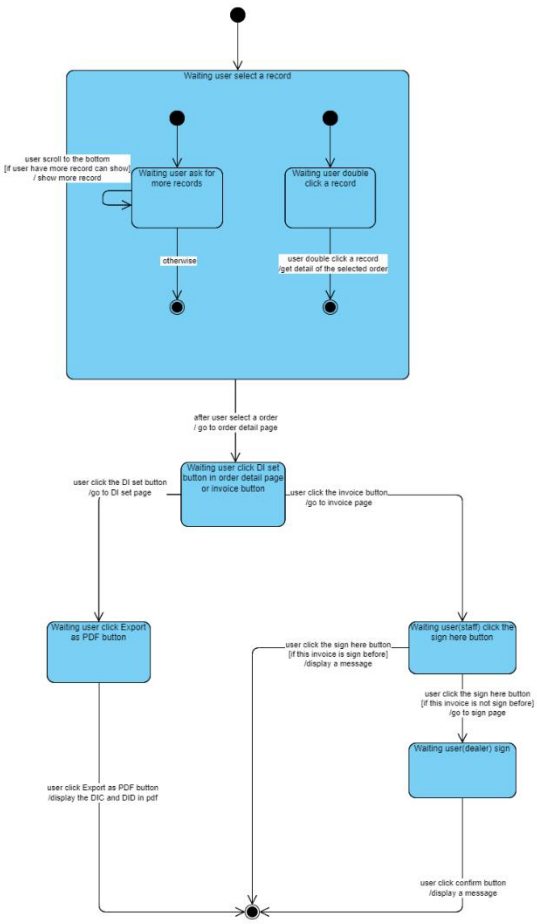
14.7. Login


user click the login button
Waiting user input account and password
user click submit button/ display new password
user click the forget password/ open forget password page
Waiting system respond
password is wrong
password is correct/ go to main menu
Waiting user input identity informatiion

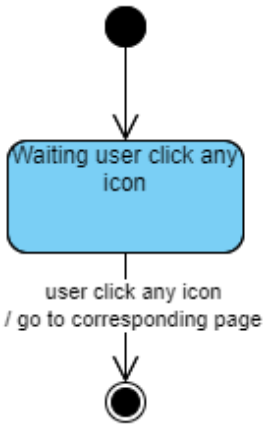
14.8. Browse catalogue and Place An Order



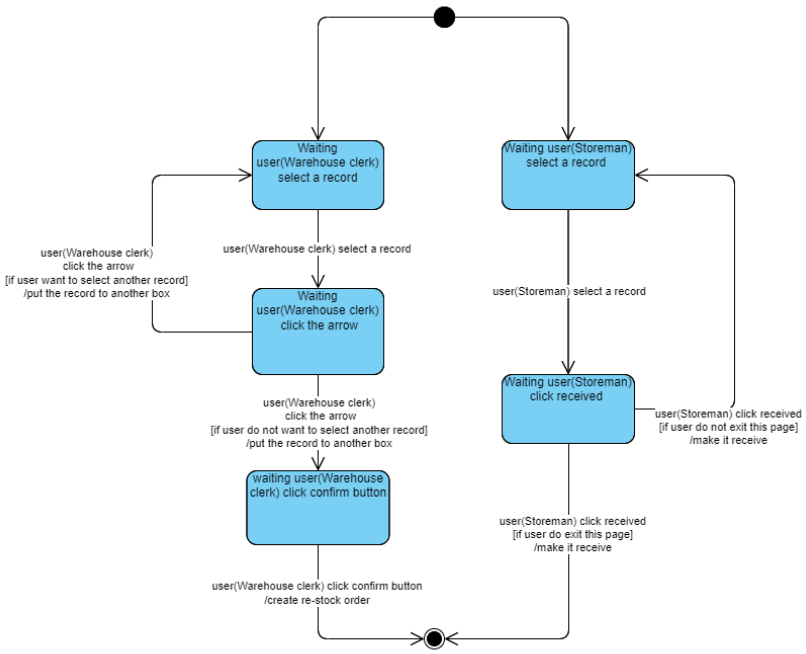
14.9. Check Order Detail and Review Order Status



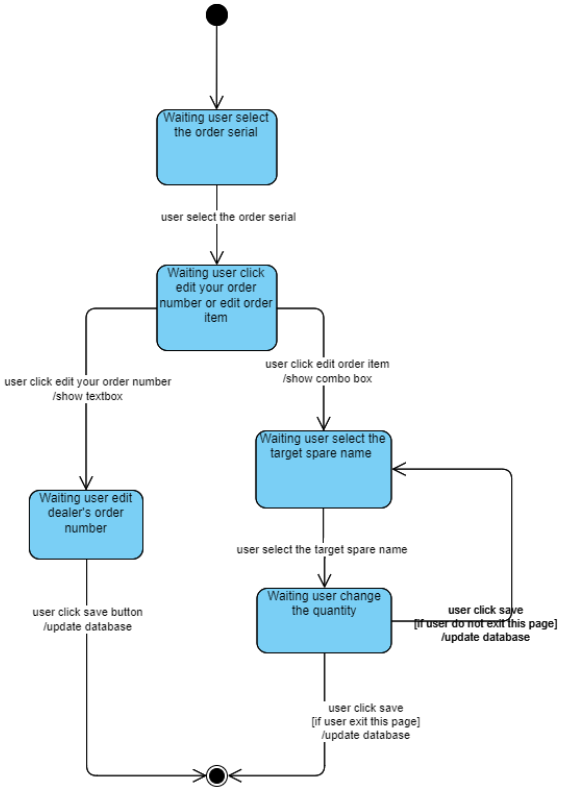
14.10.main menu



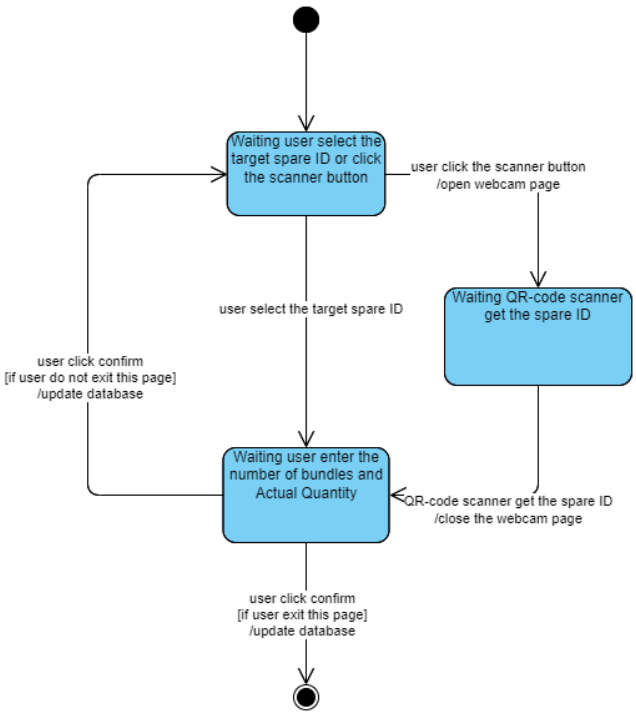
14.11.re-stock



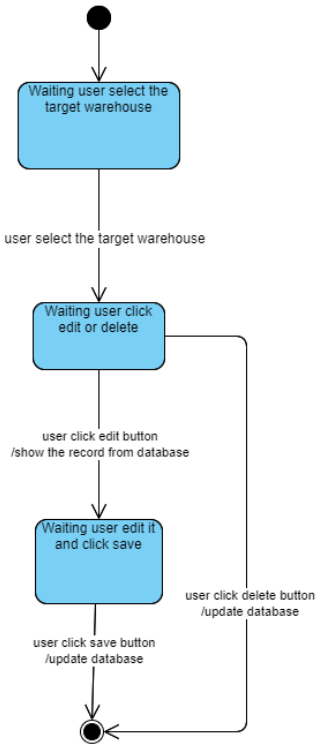
14.12.edit order



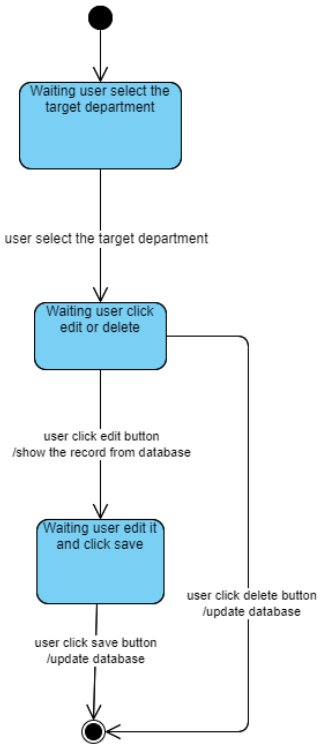
14.13.collect spare



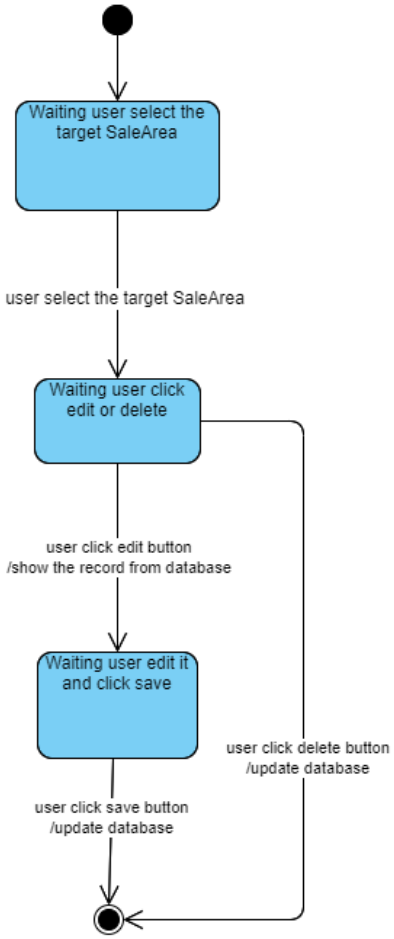
14.14. Edit Warehouse



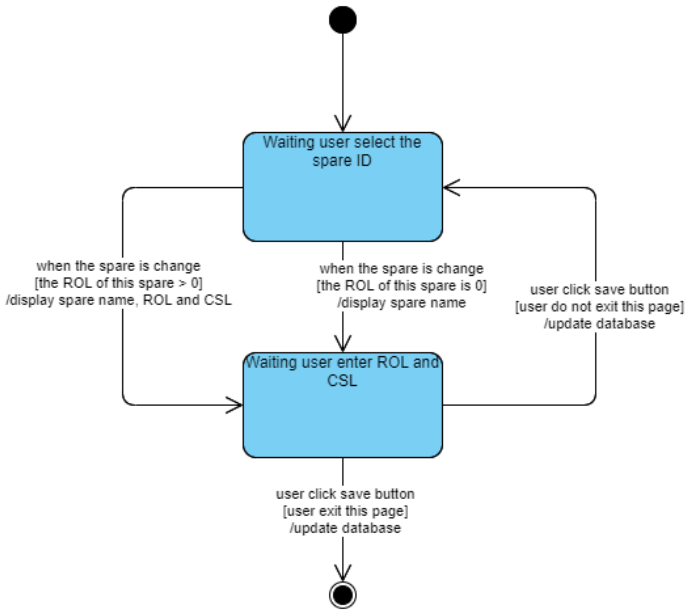
14.15. Edit Department



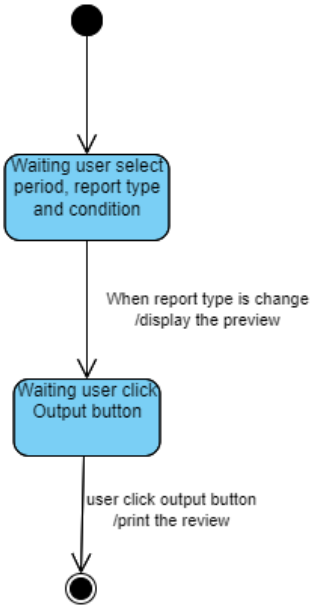
14.16. Edit SaleArea



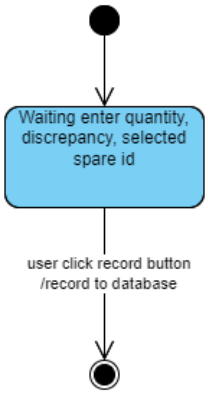
14.17.Set Stock Level



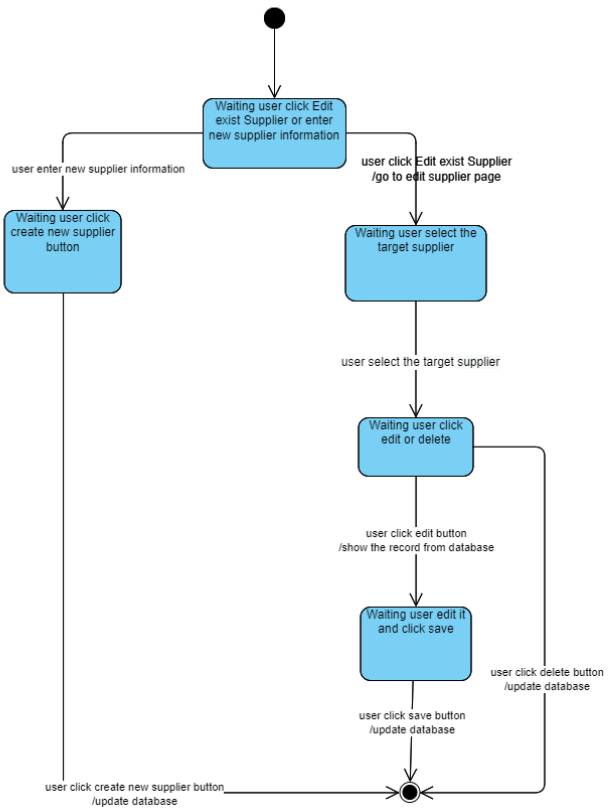
14.18. Generate Annual Report



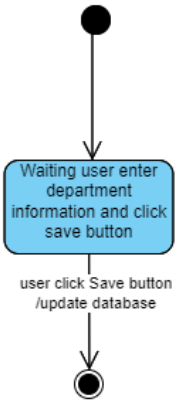
14.19.Scrap


Waiting enter quantity, discrepancy, selected spare id
user click record button /record to database

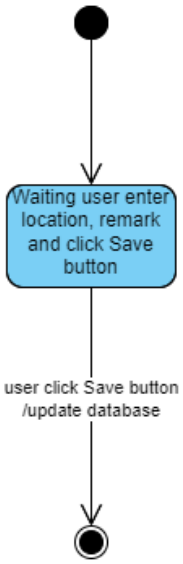
14.20. Supplier management



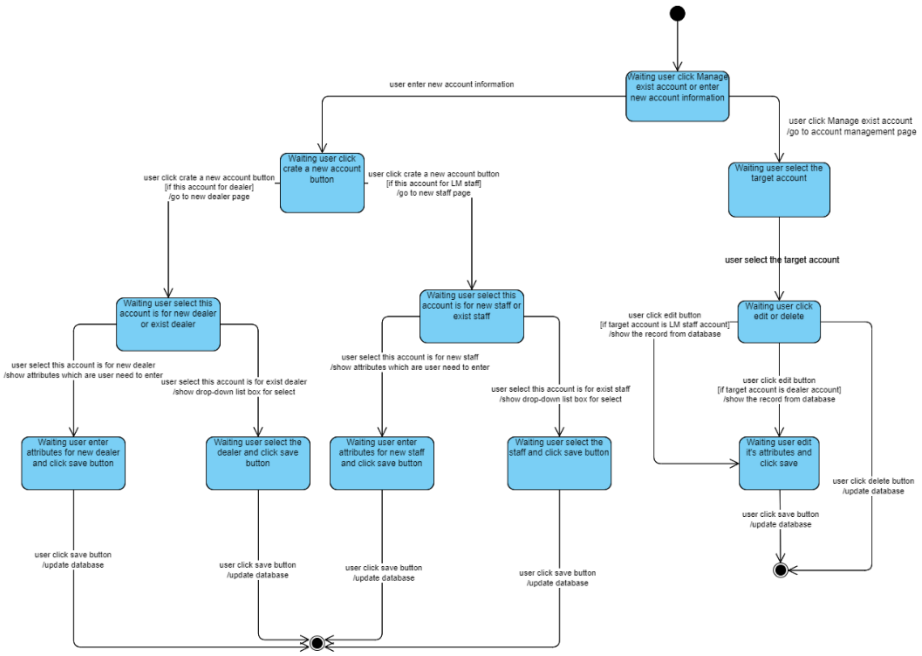
14.21.Add Department



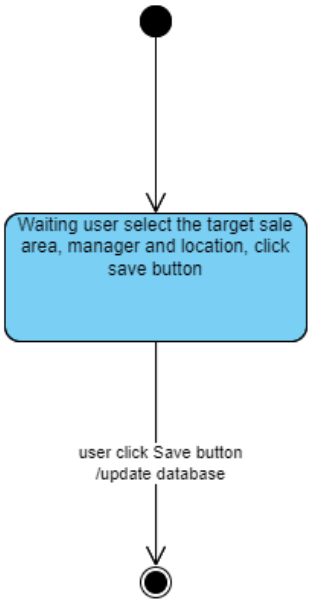
14.22. Add SaleArea



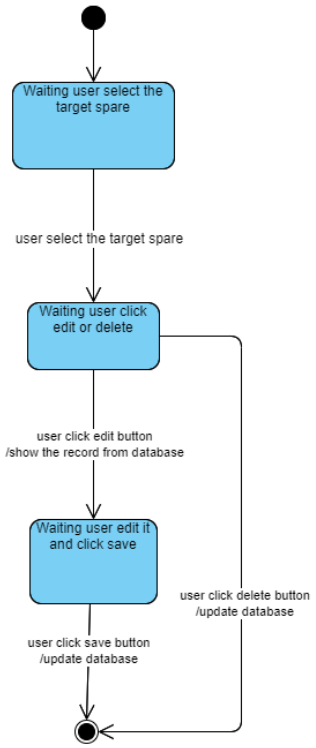
14.23.account management



14.24. Add Warehouse

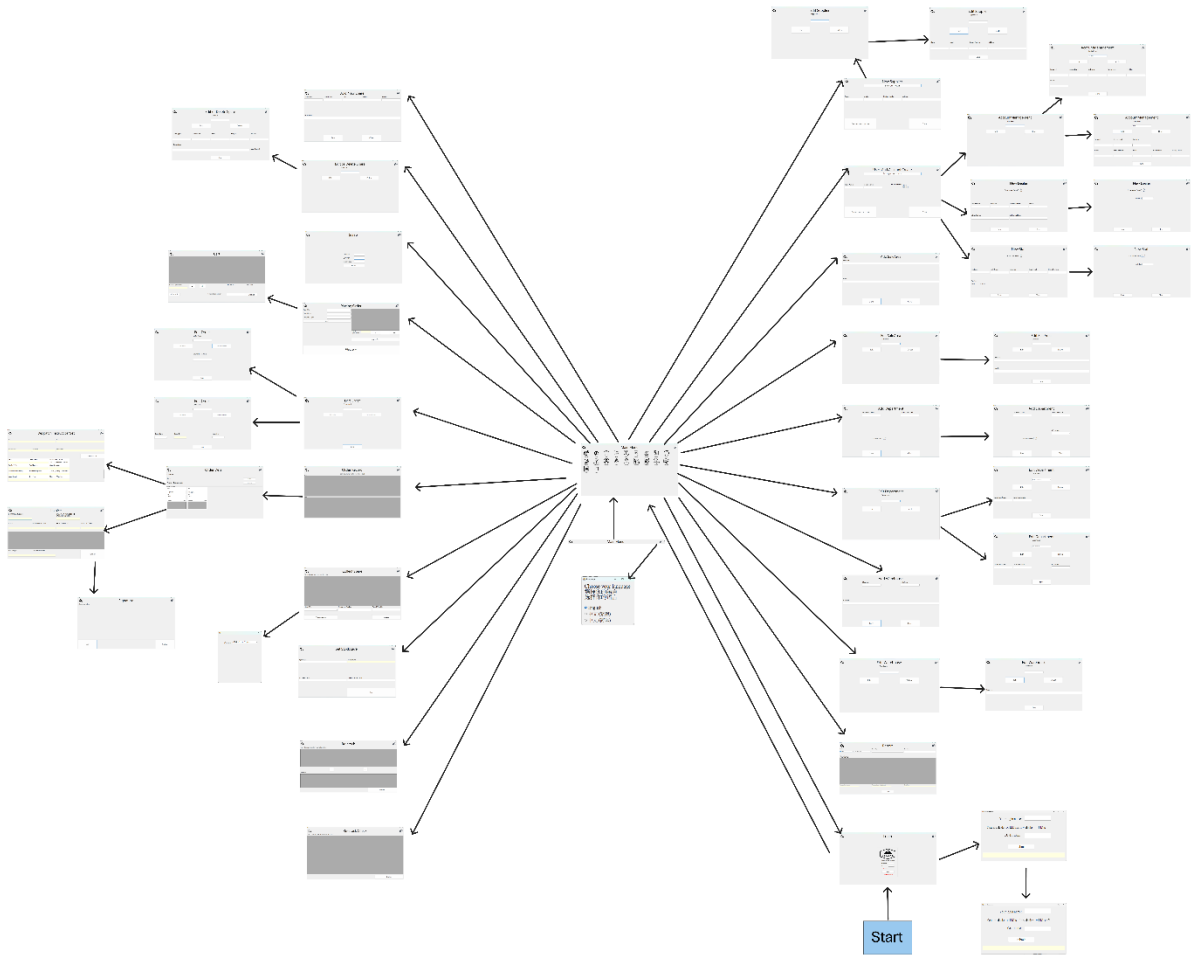


14.25.edit spare



15. User Interface

15.1. Overview



(refer to UI.pdf)

15.2. UI description

I use below color to mean that User may need to input something to the system

I use below color to mean that the system to show some table with attributes and records

I recommend users to run the app at 1280 x 720 to 1920 x 1080 (16:9 resolution)

15.2.1.Main Bar

When user click:

The arrow on the left is used to return to the previous page (not work in login page and menu page)

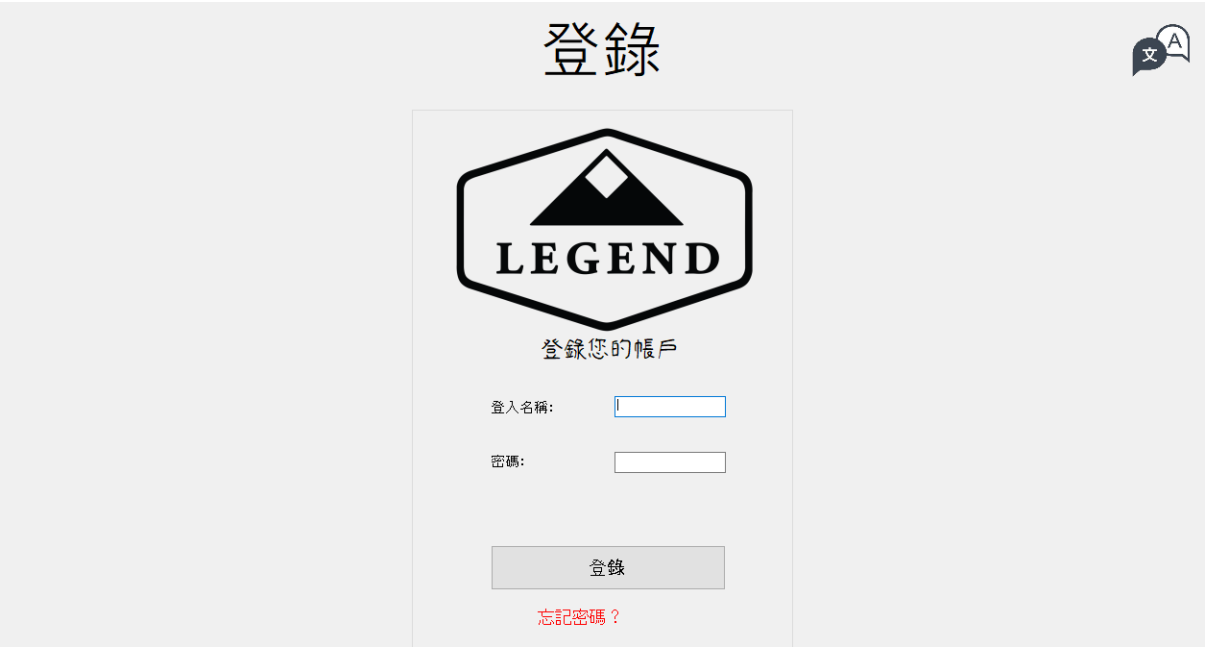
The label in the middle will change the content to the menu page (not work in login page and menu page)

The symbol on the right opens the page (language)

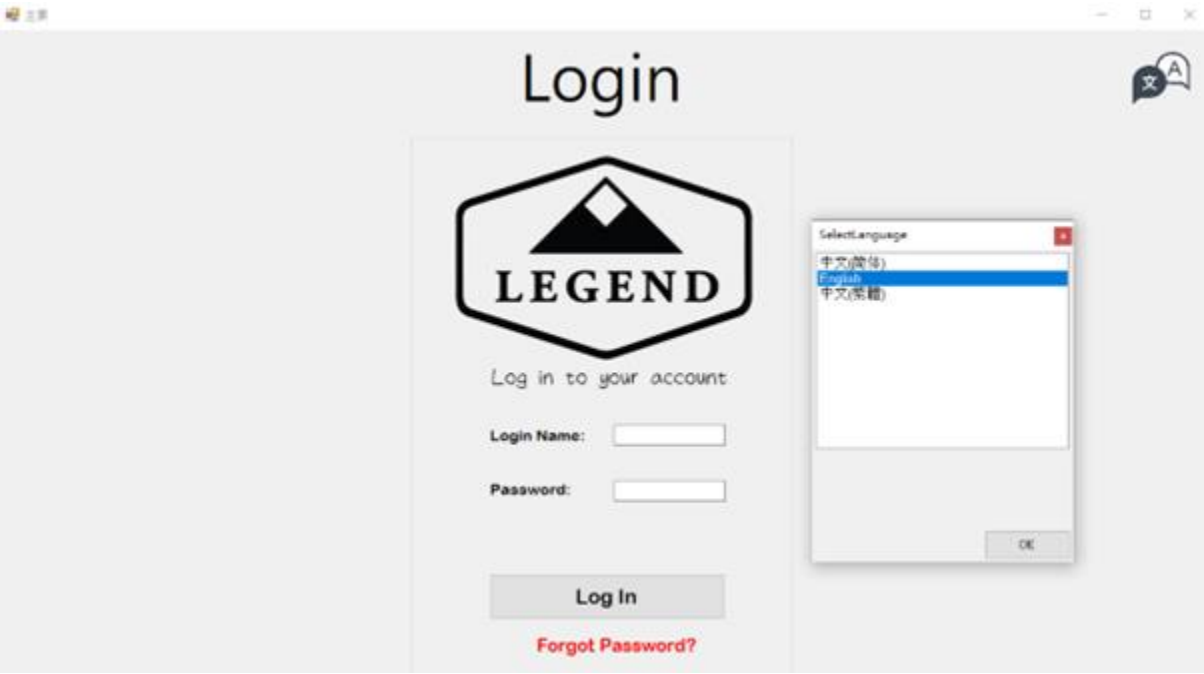
15.2.2.Language

Our system will try to get user's computer language setting

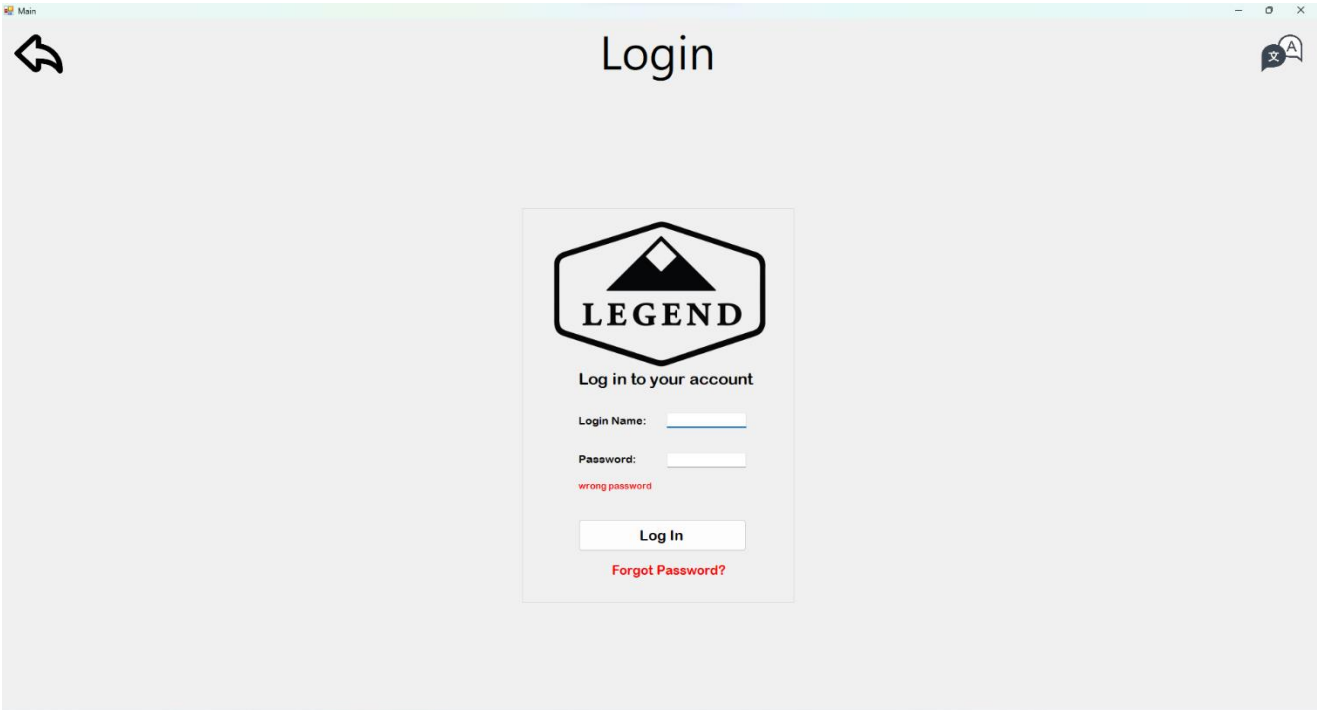
(My PC is chinese traditional)



User can select the language of our system


Login
LEGEND
Log in to your account
Login Name:
Password:
Log In
Forgot Password?
SelectLanguage
English
OK

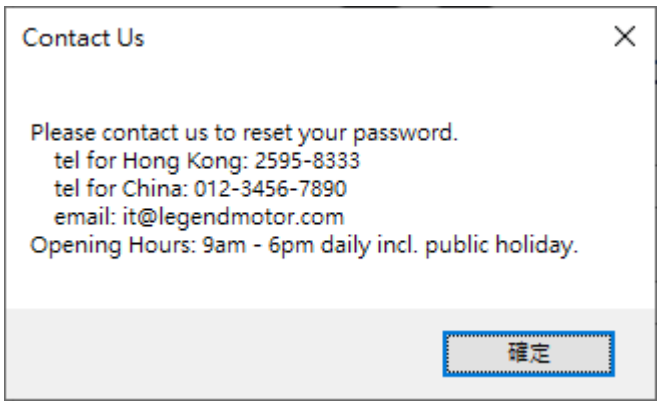
15.2.3.Login



User can login by their own Login Name and Password

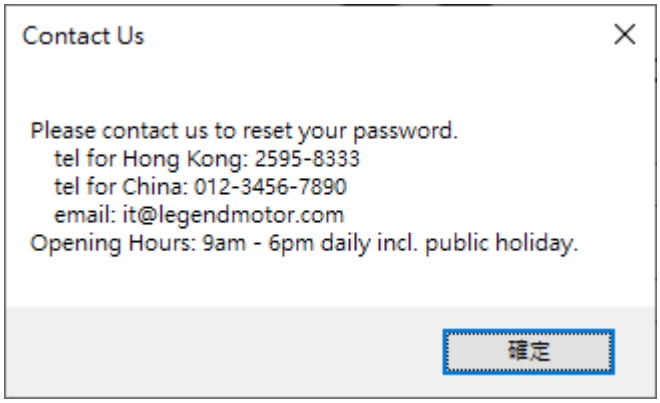
If login fail, form will show the wrong message

User can click “Forgot Password?” to reset the password



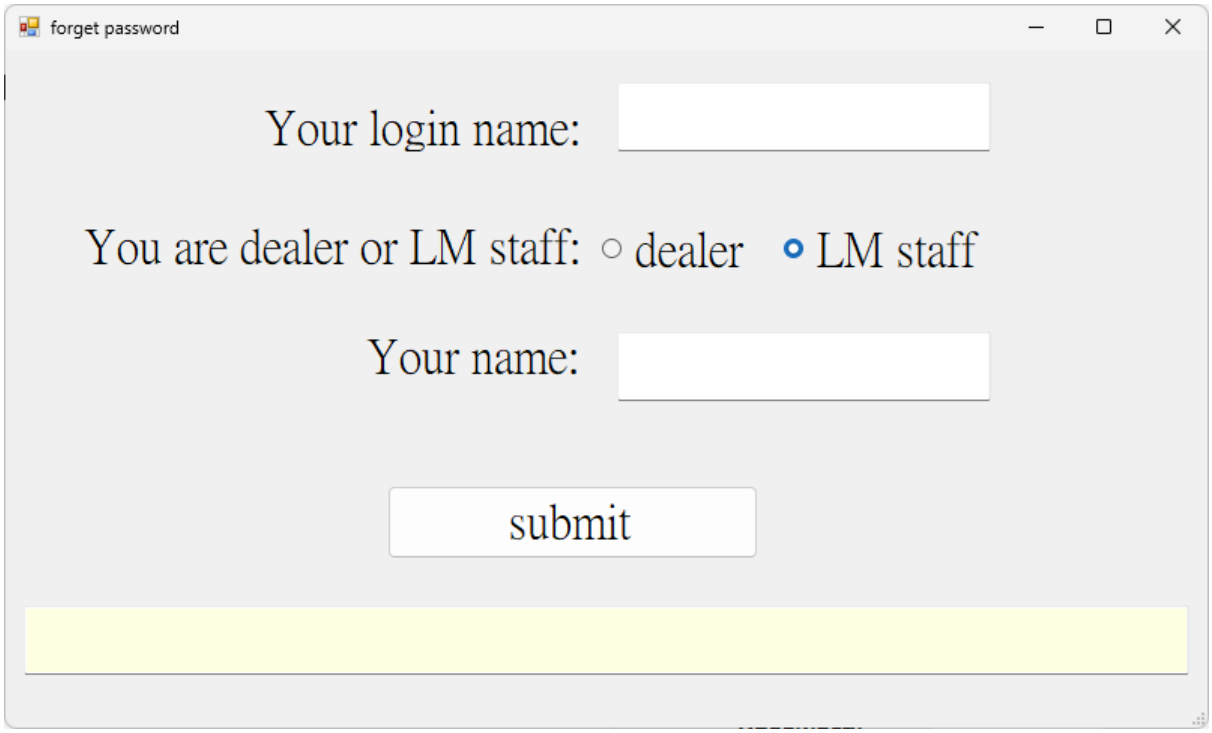
The system will tell the user to contact the company

15.2.3.1.forget password (dealer)



The system will tell the user to contact the company

15.2.3.2.forget password (LM staff)

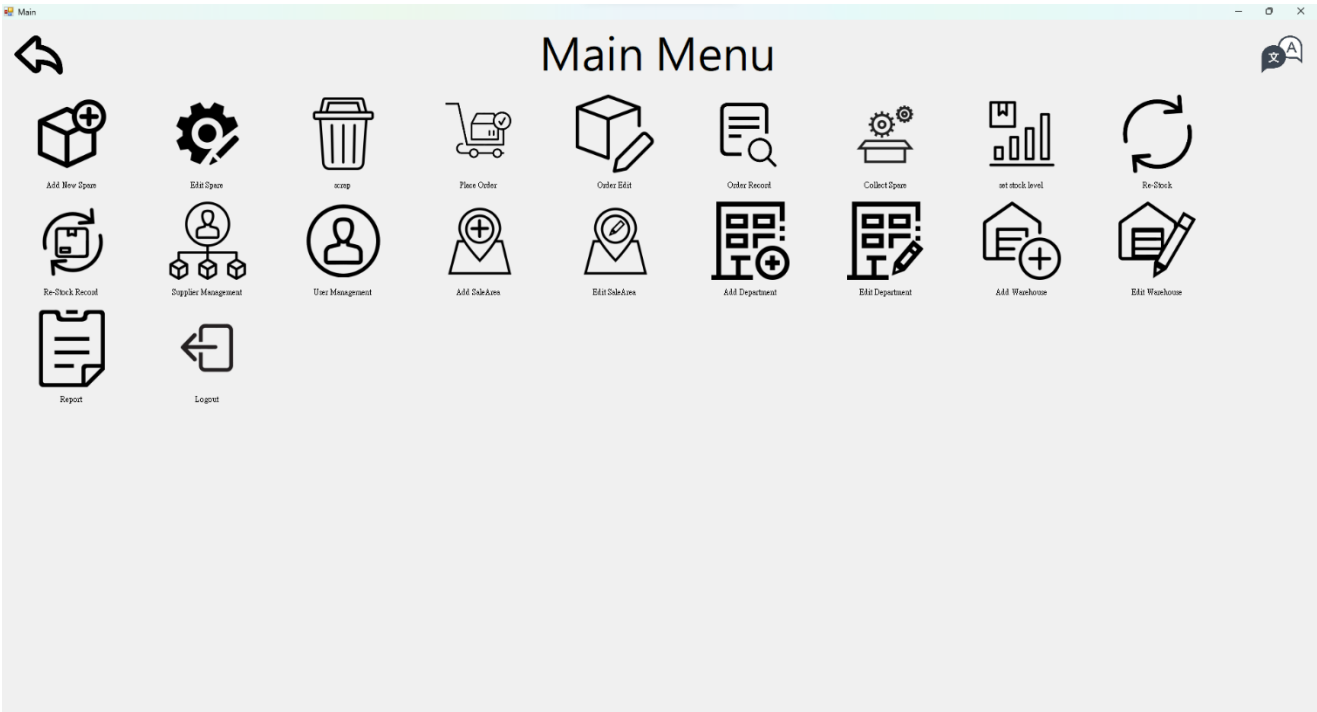


User can enter their Login Name

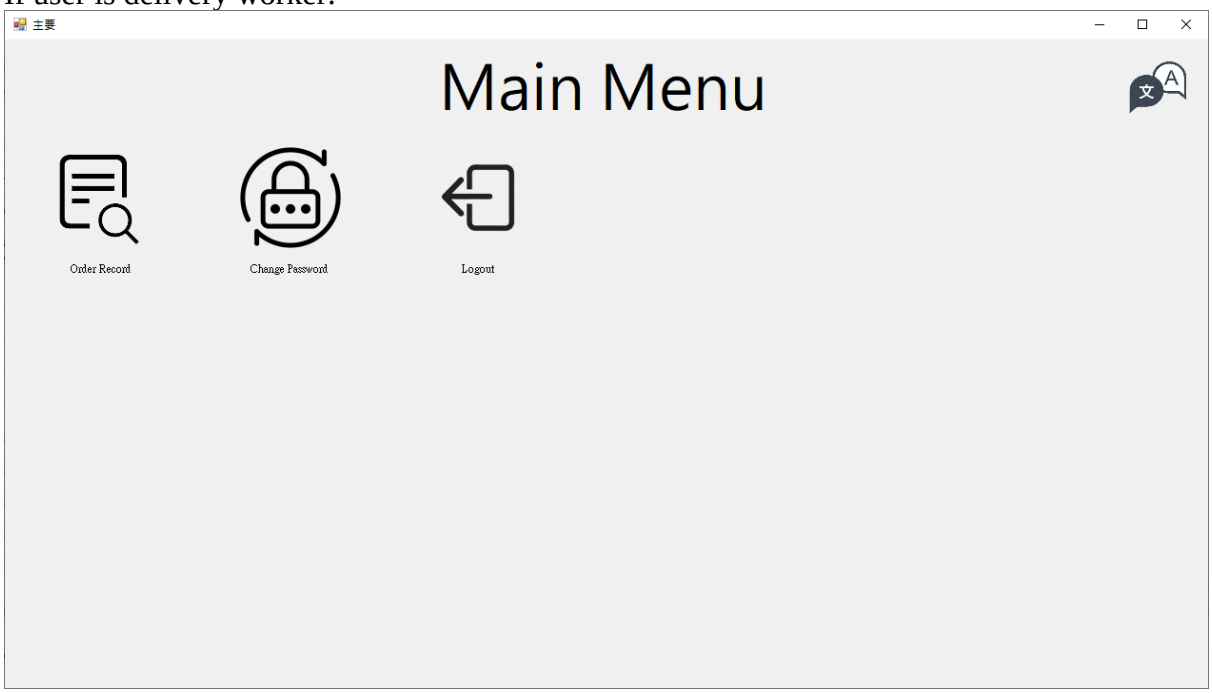
If user is a LM staff, then choose “LM staff” and enter the name

User can click “submit” to wait for allow to change password

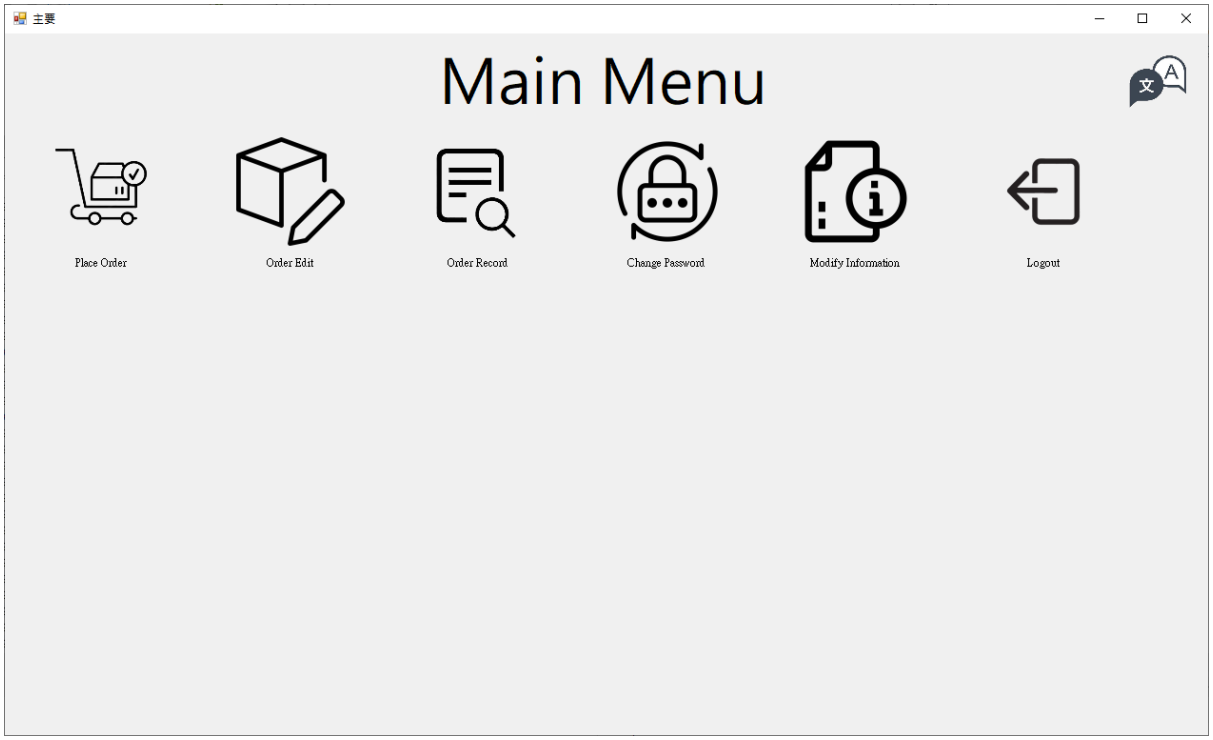
15.2.4.Main Menu



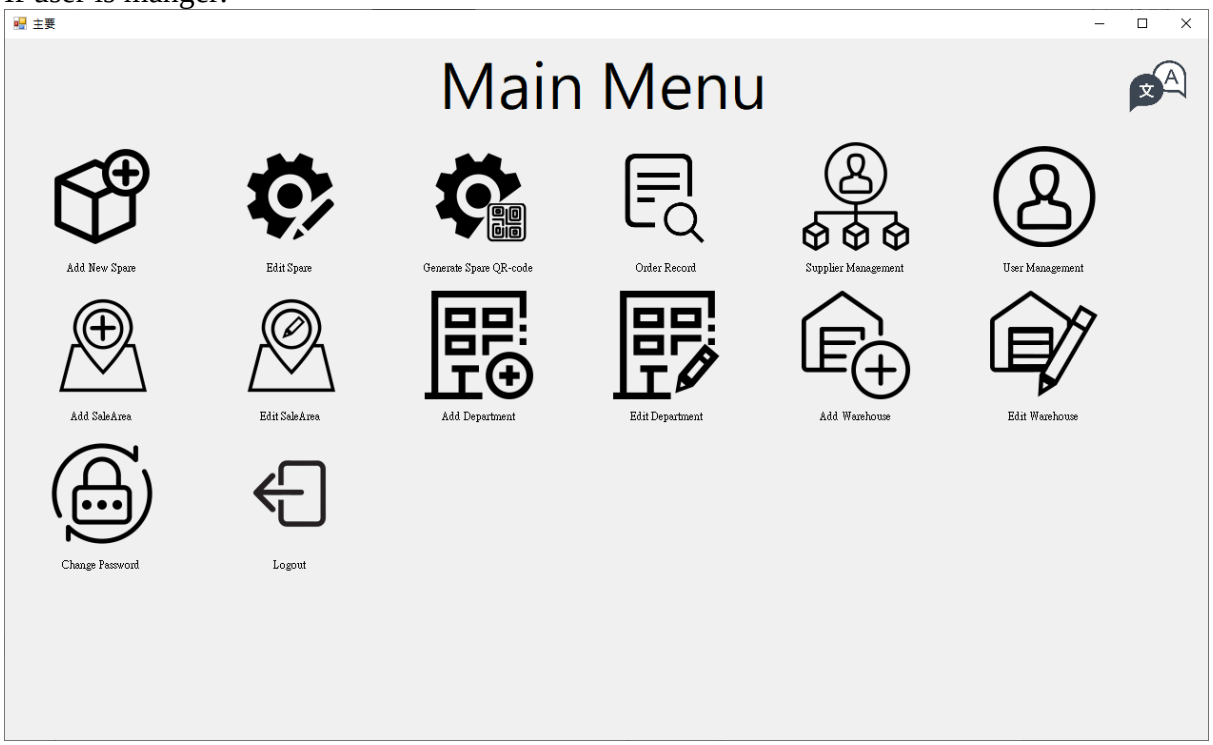
If user is delivery worker.



If user is Dealer.



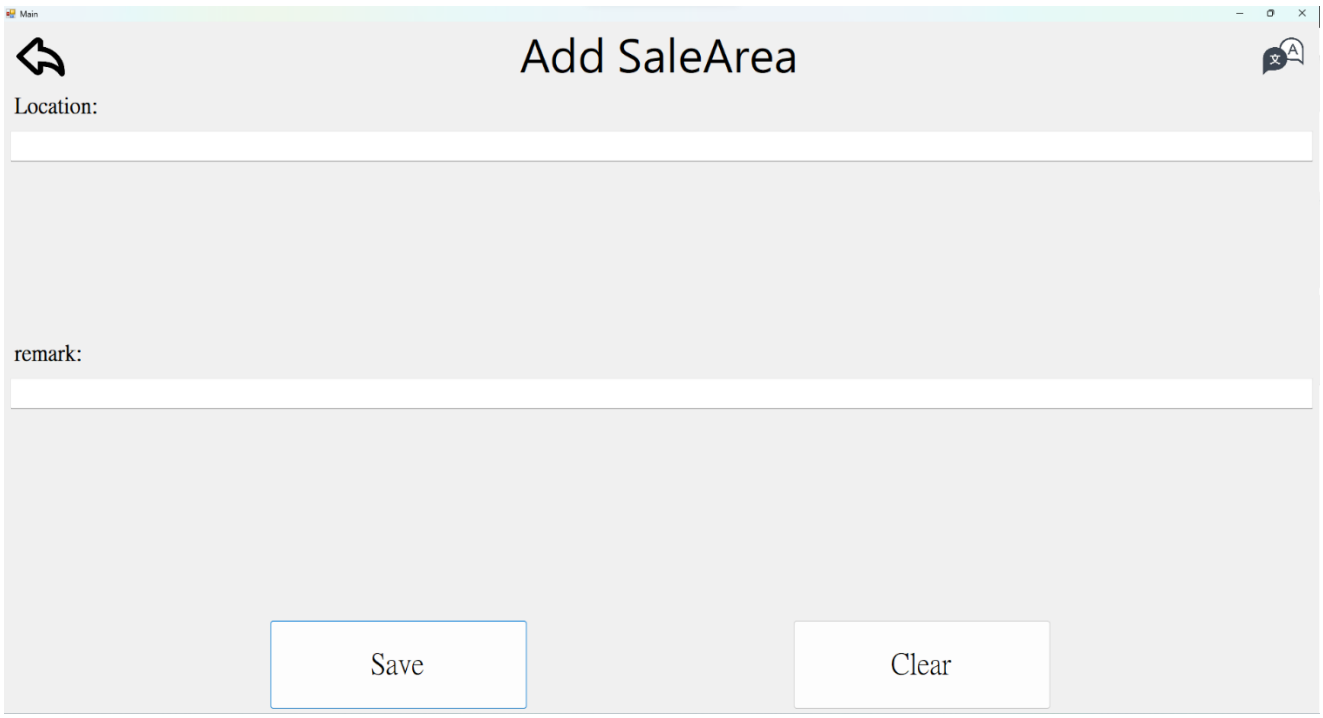
If user is manger.

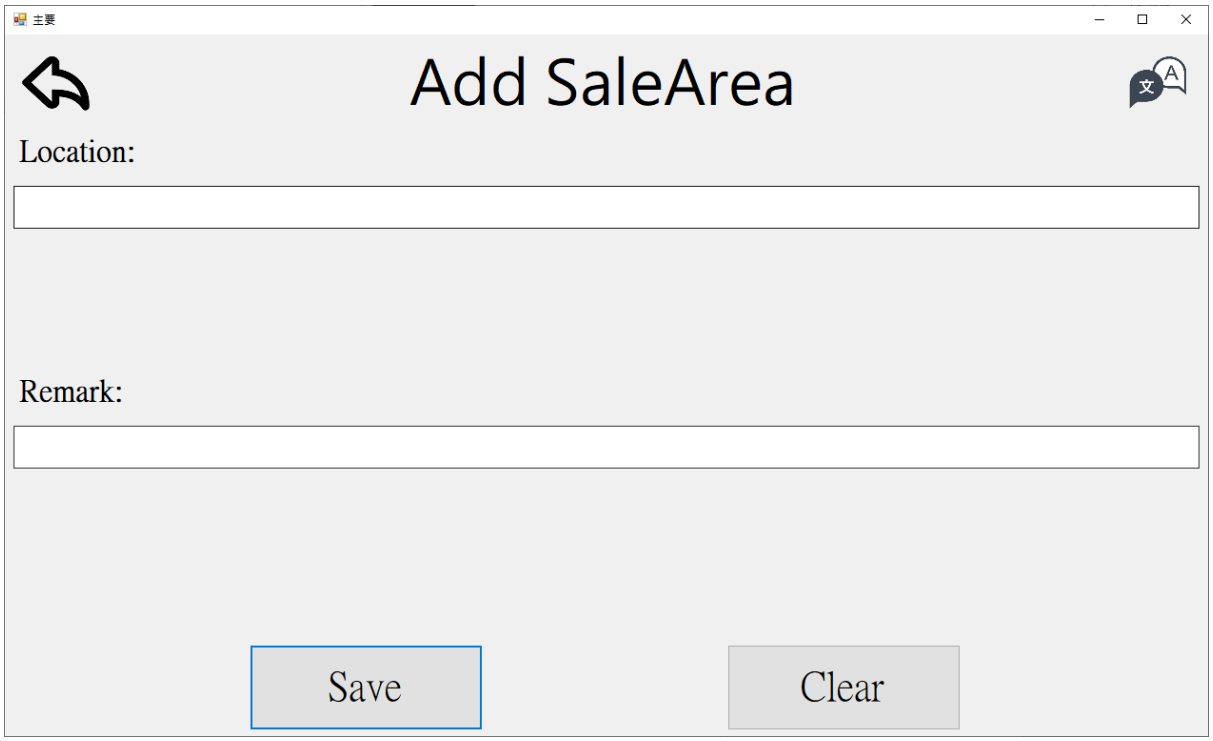


Users can select the corresponding icon to enter the corresponding form

The user can only see the icon displayed by the account permission allow

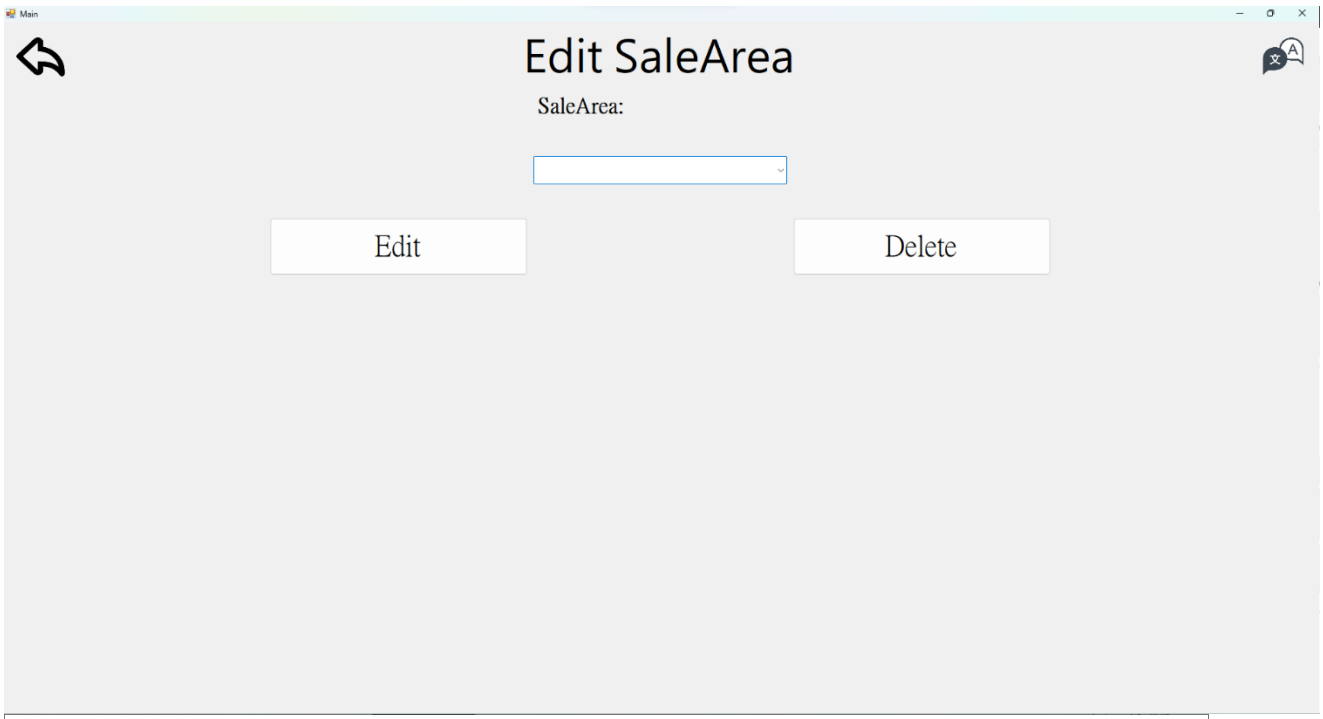
15.2.5.Add Sale Area

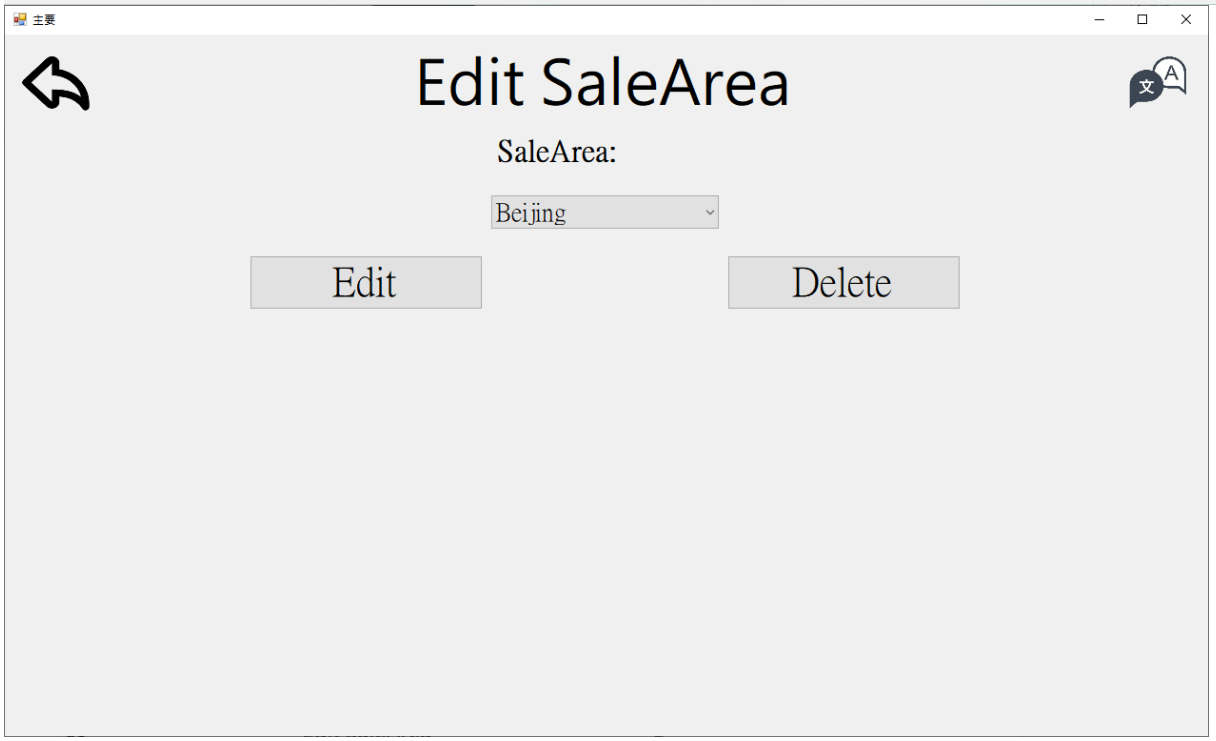




User can add a new Sale Area on this page

15.2.6.Edit Sale Area

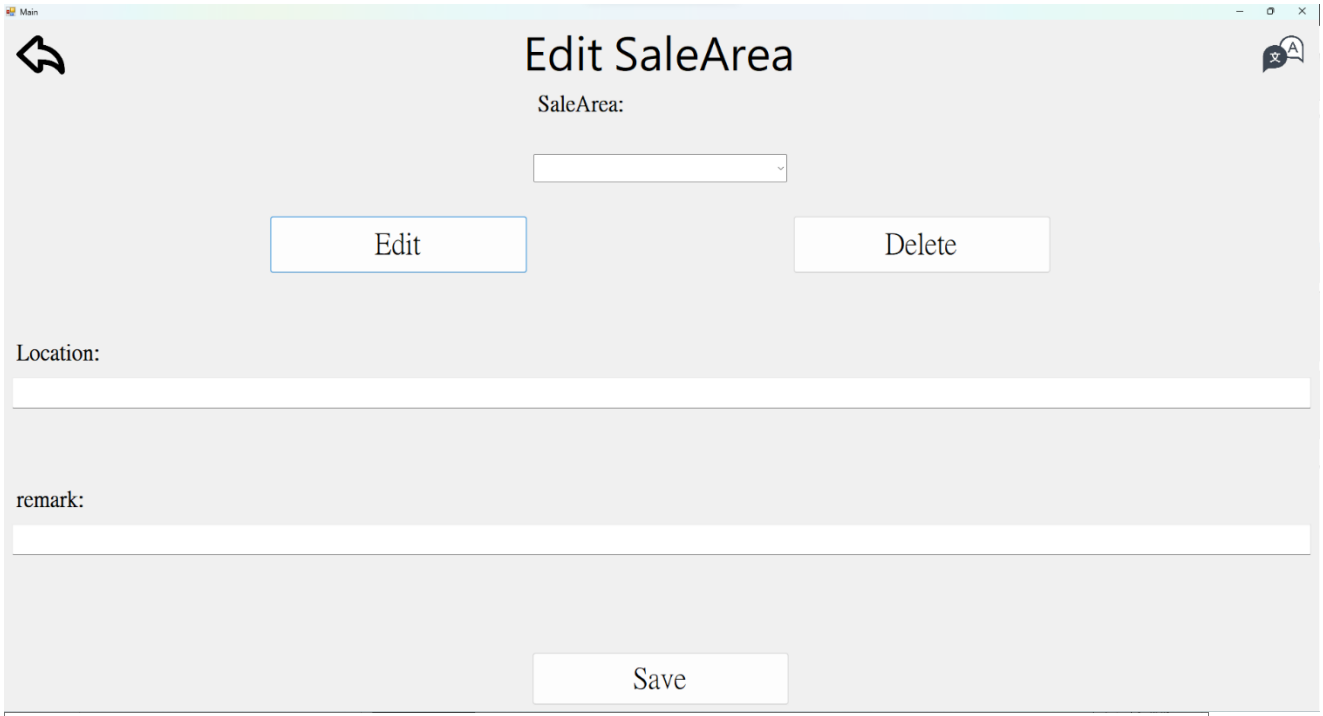


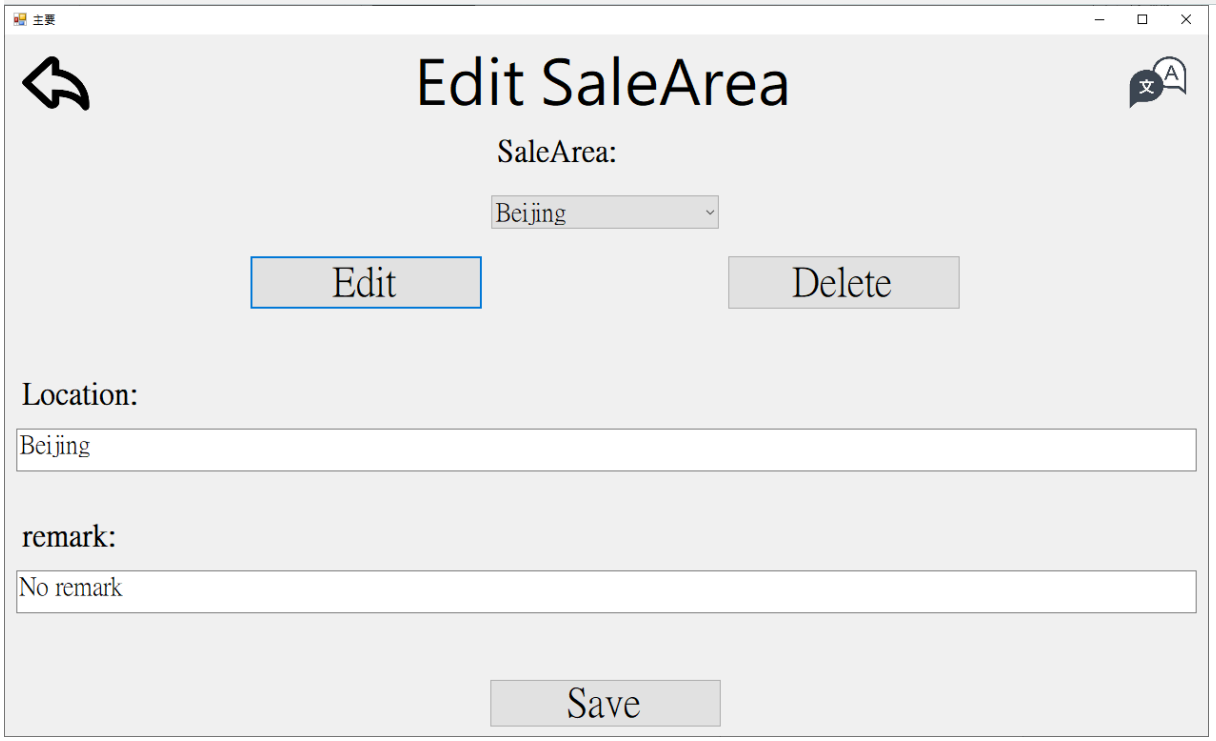


User can select existing Sale Area

And then, User can edit it or delete it

15.2.6.1. When user click “Edit” button



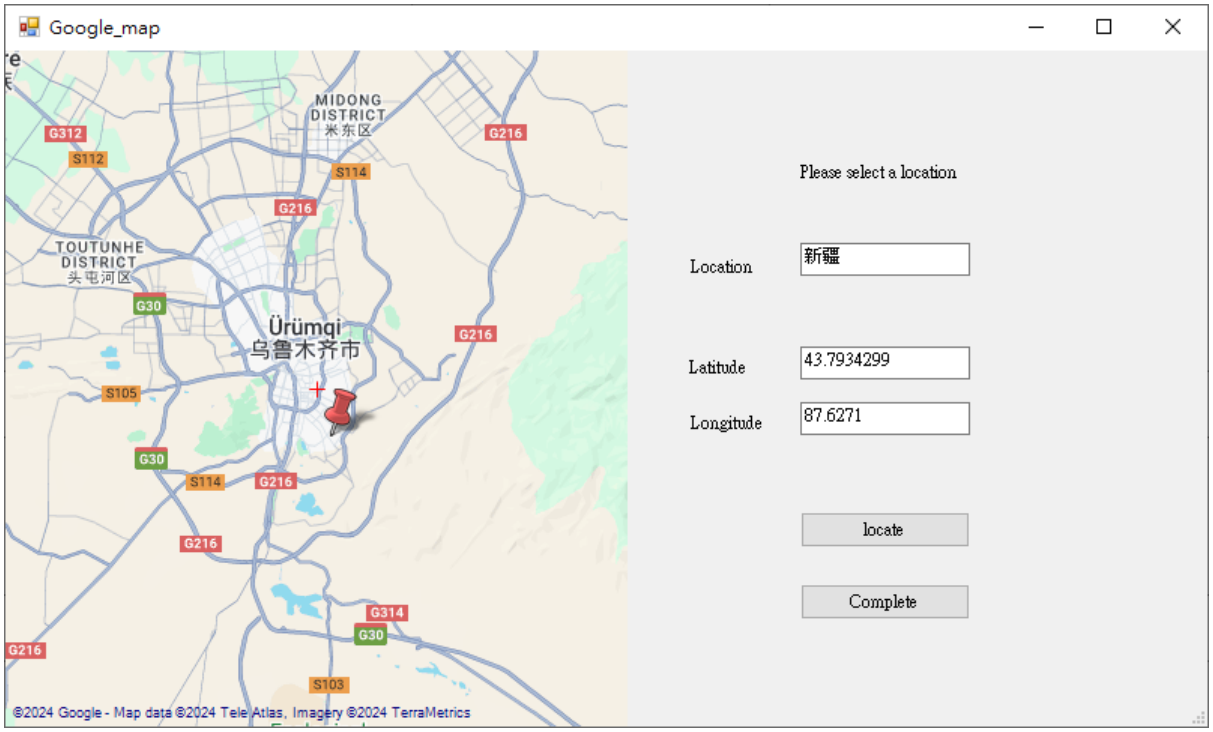


Put the contents of the selected record from the database into the corresponding location

Users can modify data

Users click the “Save” button to edit

15.2.6.2.After click Save button



User clicks the "locate" button to locate the location.

The system will automatically find the latitude and longitude of the location and return it.

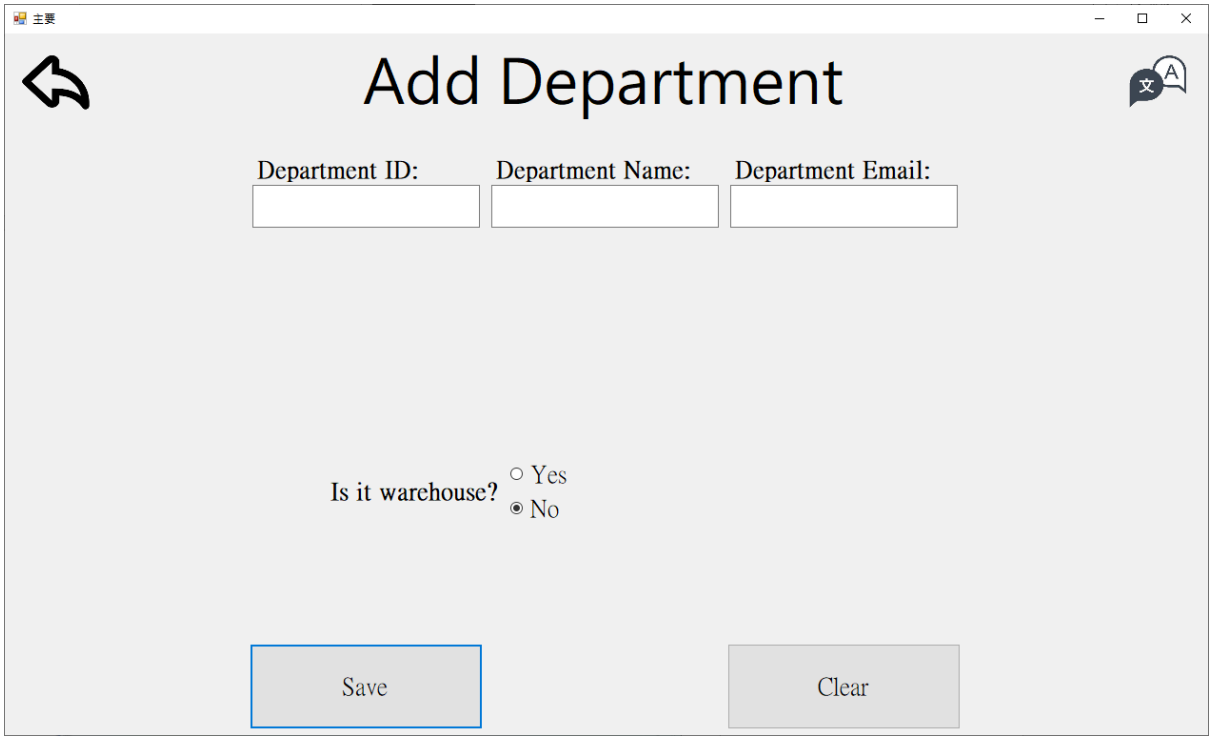
User clicks the “Complete“ button to complete the edit or add.

15.2.7.Add Department

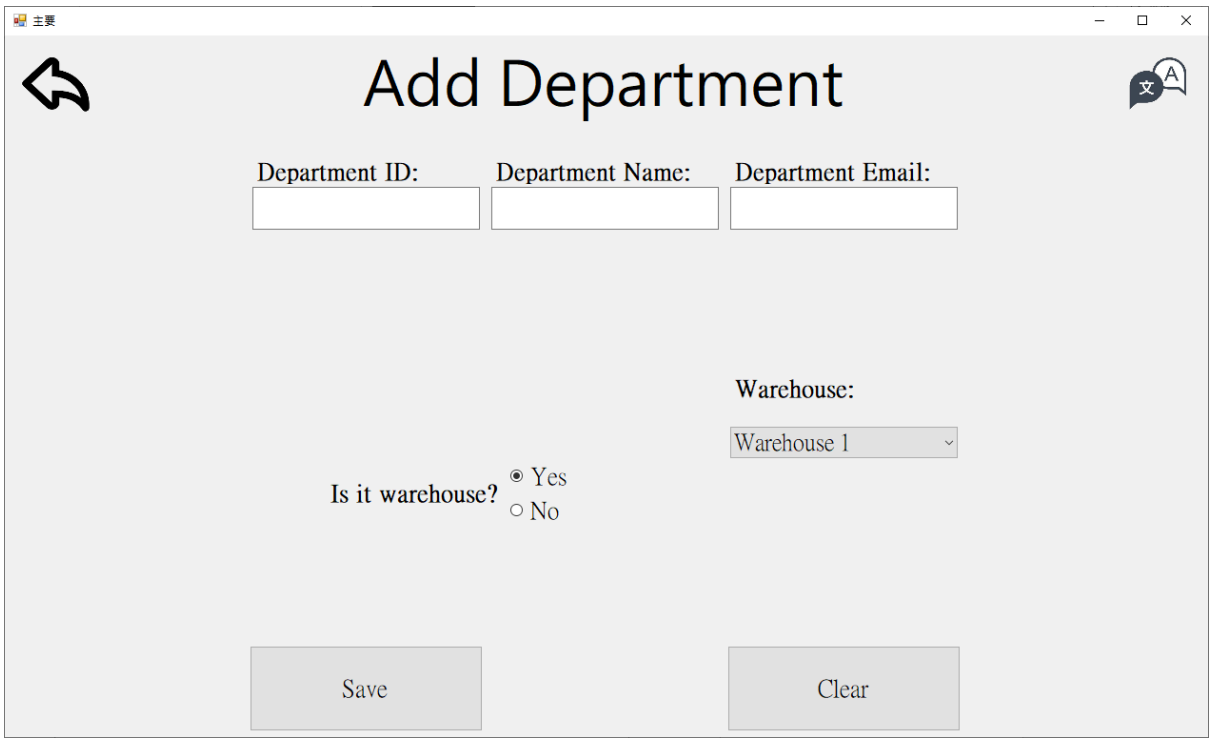
User can add a new department

User should enter Department name and email address

15.2.7.1. Is not warehouse

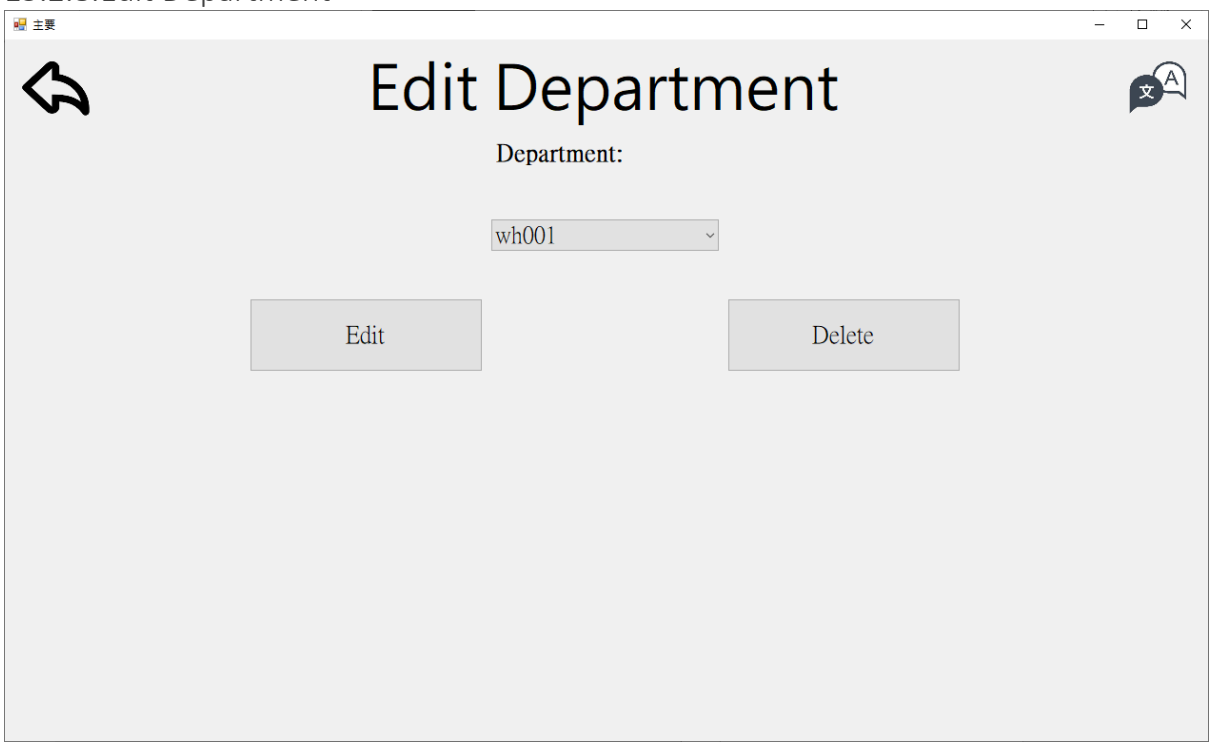


15.2.7.2. Is warehouse



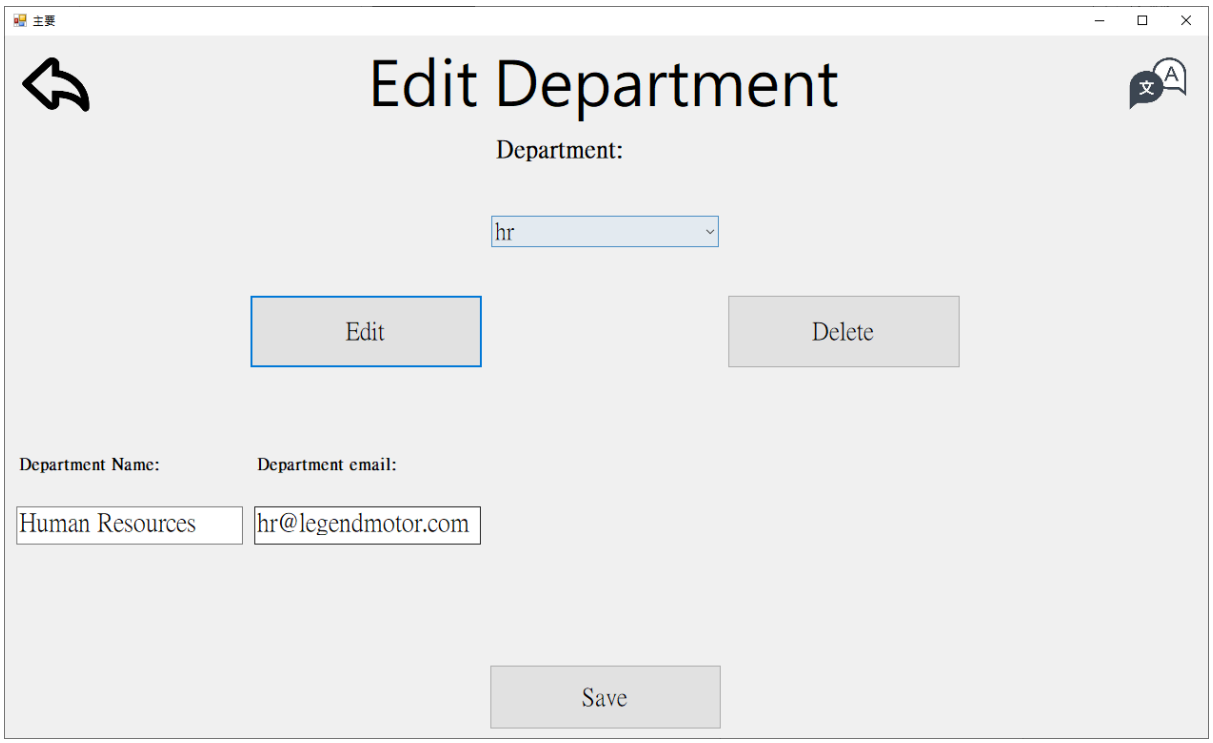
If this department is a warehouse, user should choose the warehouse

15.2.8.Edit Department



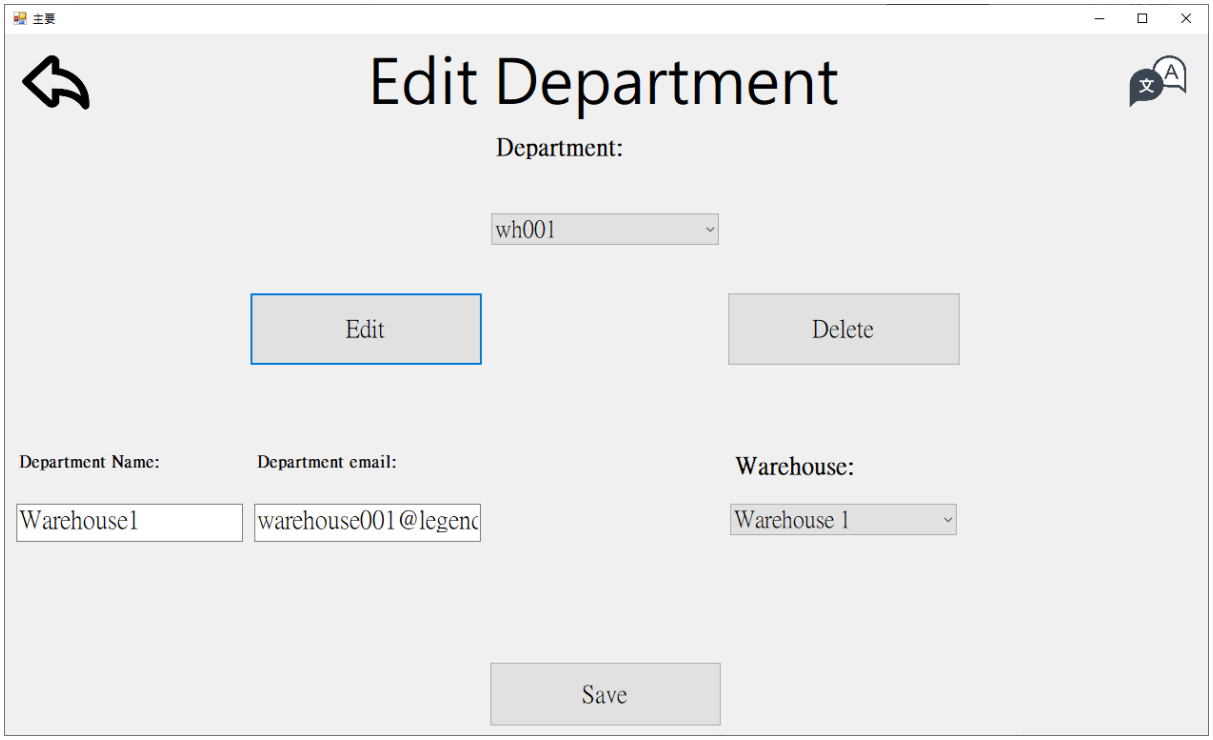
User can edit the information of department

15.2.8.1. When user select a department which is not warehouse



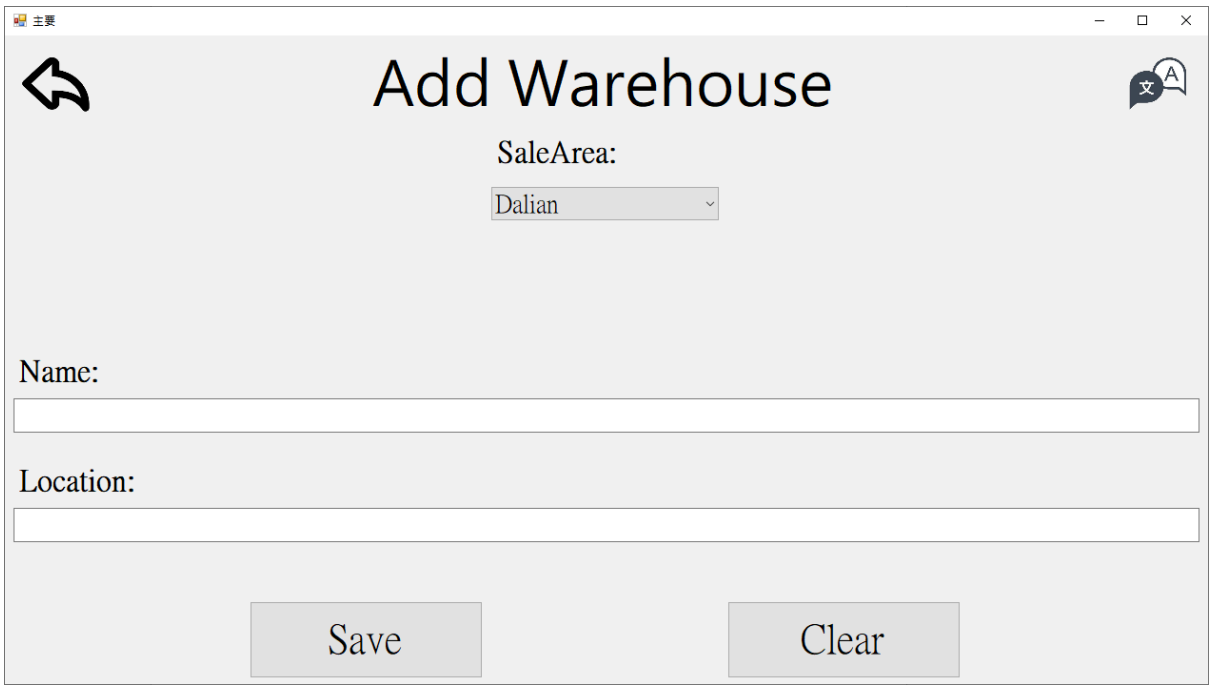
Users can edit Department Name and Department email and click the “Save” button to save the information of Department Name and Department email

15.2.8.2. When user select a department which is warehouse



User can edit Department Name, Department email and Warehouse and click the “Save” button to update the information of Department Name, Department email and warehouse in the system

15.2.9.Add Warehouse

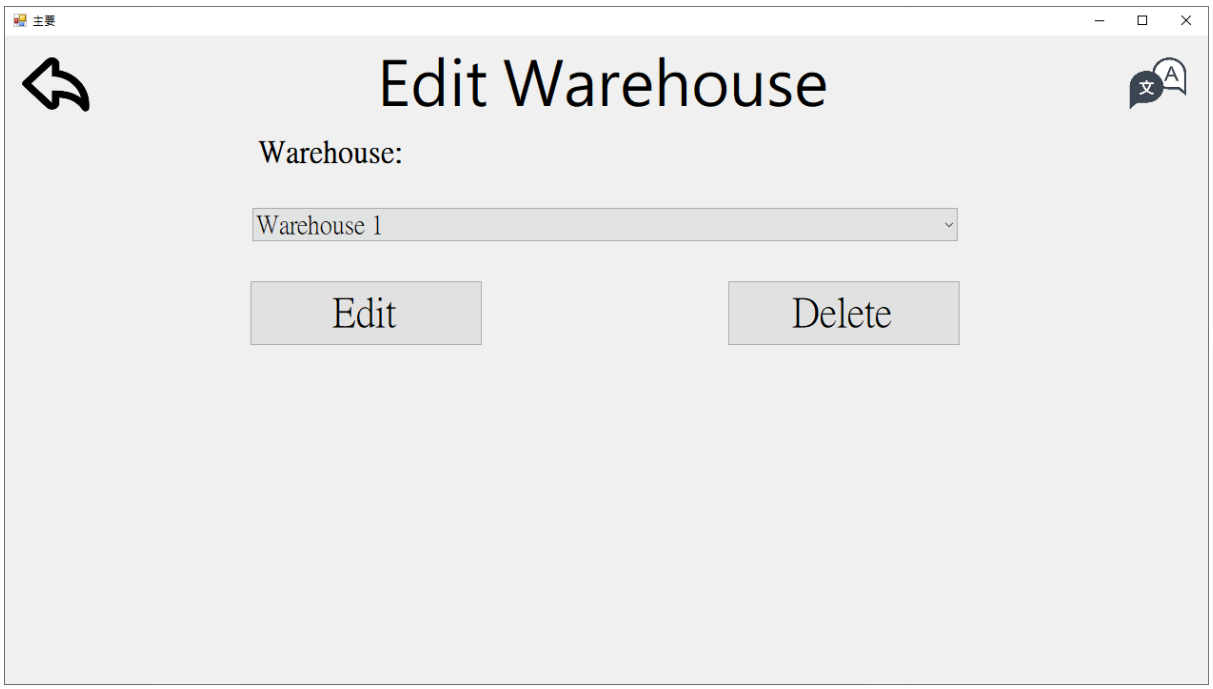


User can add a warehouse and choose the manager and the sale area

User can click the “Save” button to add.

User can click the “Clear” button to clear the input

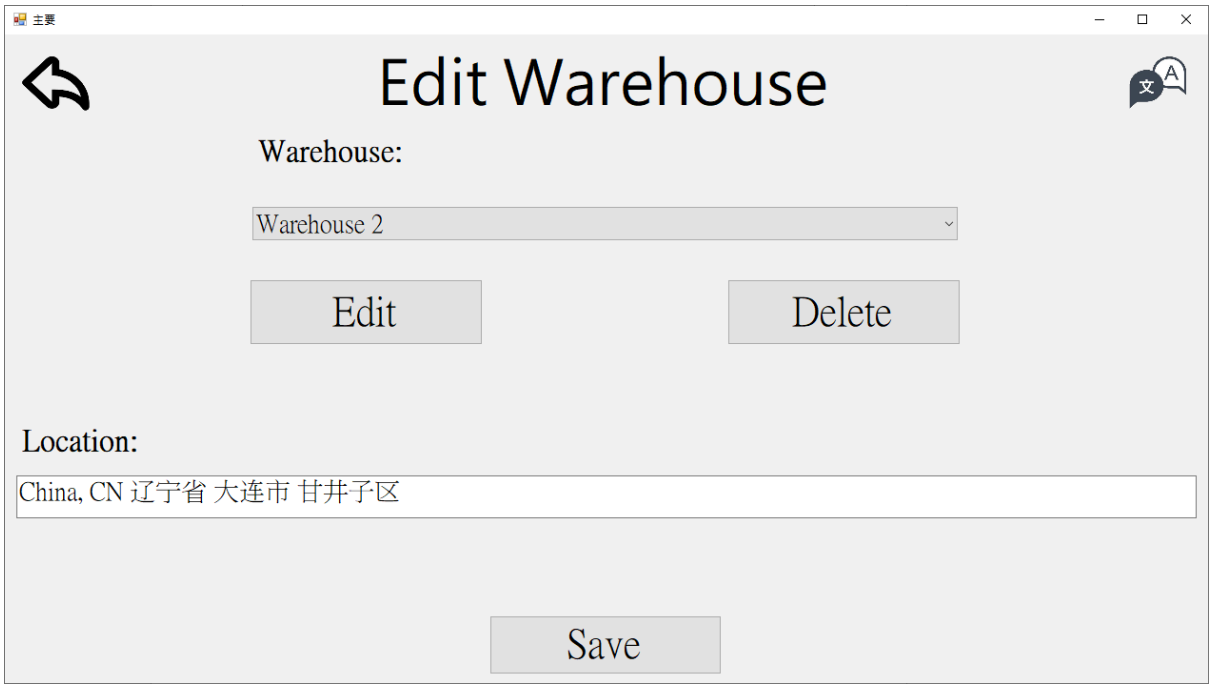
15.2.10. Edit Warehouse



User can choose the warehouse and click “Edit” button to “When user click edit button”

User can click “Delete” button to clear the information of warehouse

15.2.10.1. When user click edit button

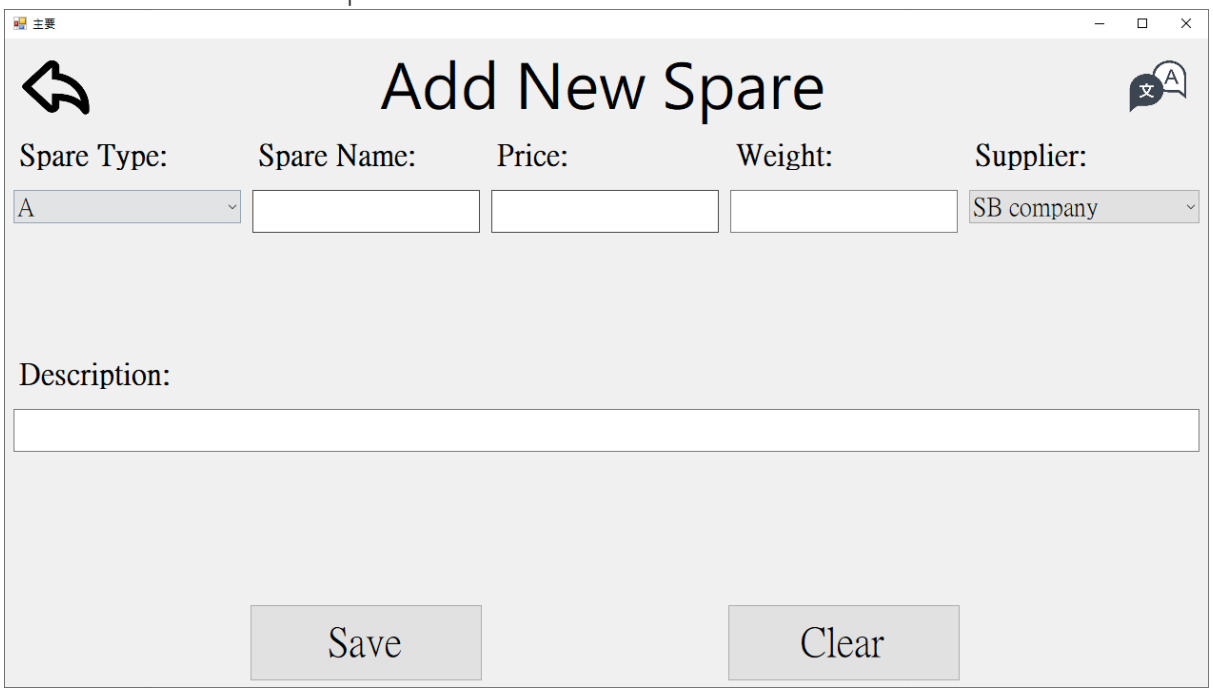


Put the contents of the selected record from the database into the corresponding location

Users can modify data

Users click the “Save” button to save, and then form will change to Edit Warehouse

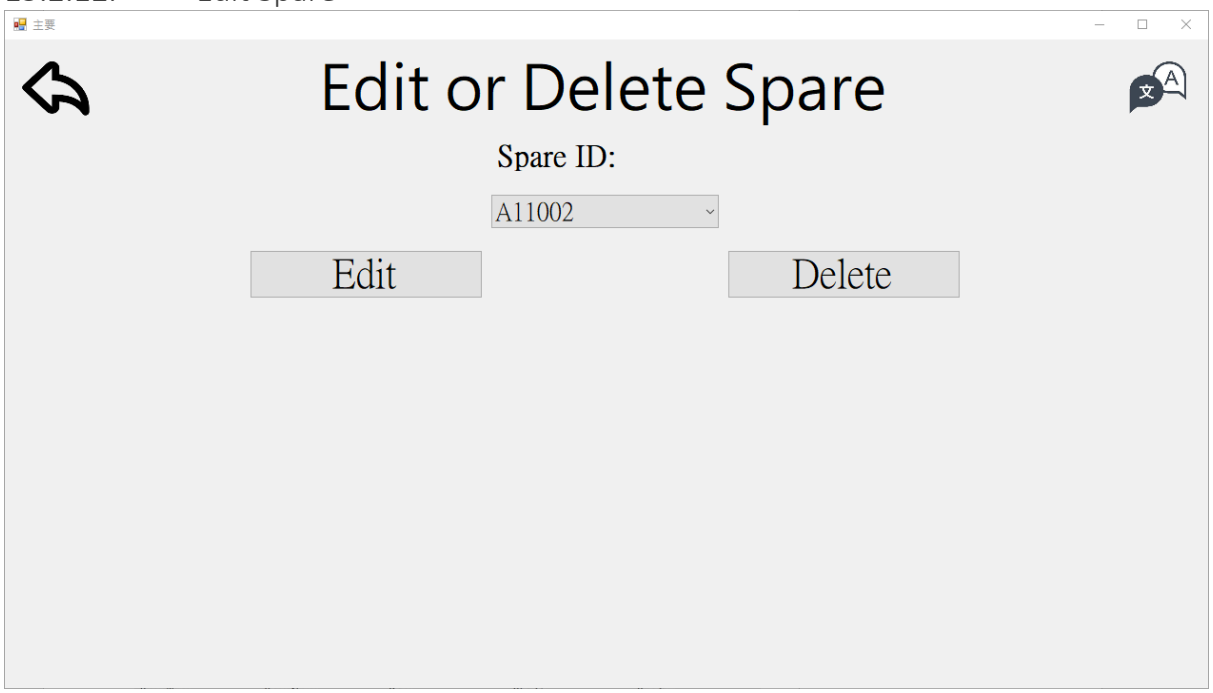
15.2.11. Add New Spare



User can add a new spare and choose the spare type, input the spare name, price, weight, supplier, description

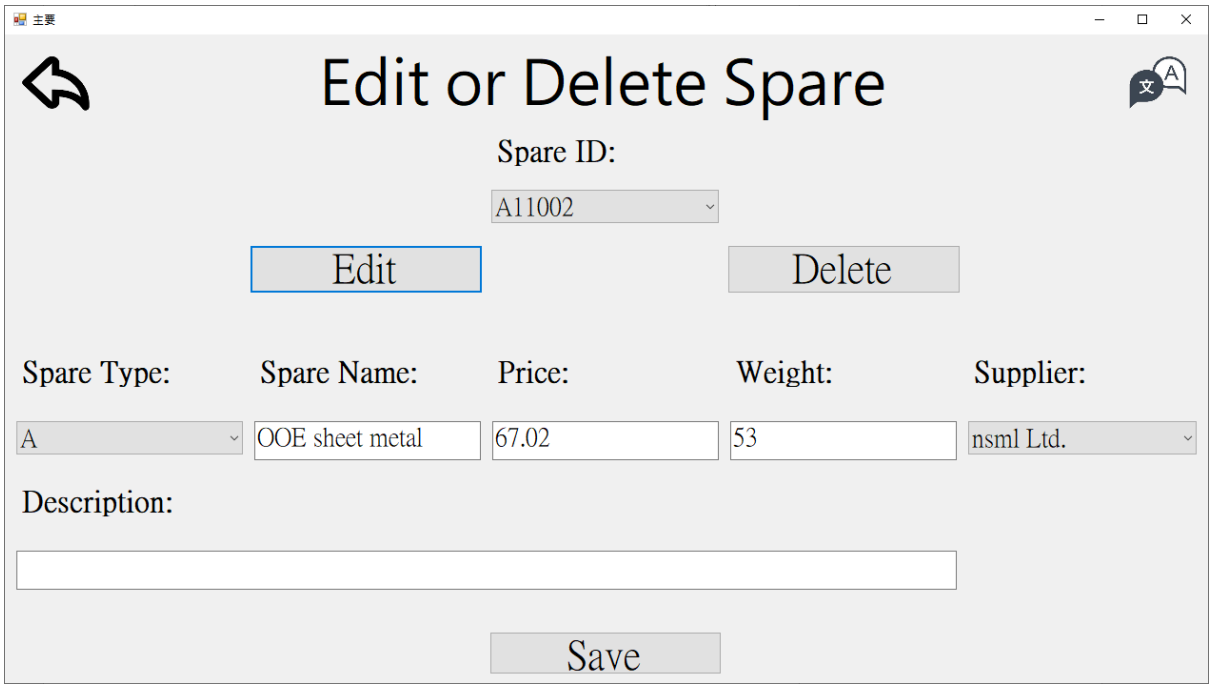
And then, user can click” Save” button to save these in the system

15.2.12. Edit Spare



Users can edit or delete the information of spare

15.2.12.1. When user click edit button

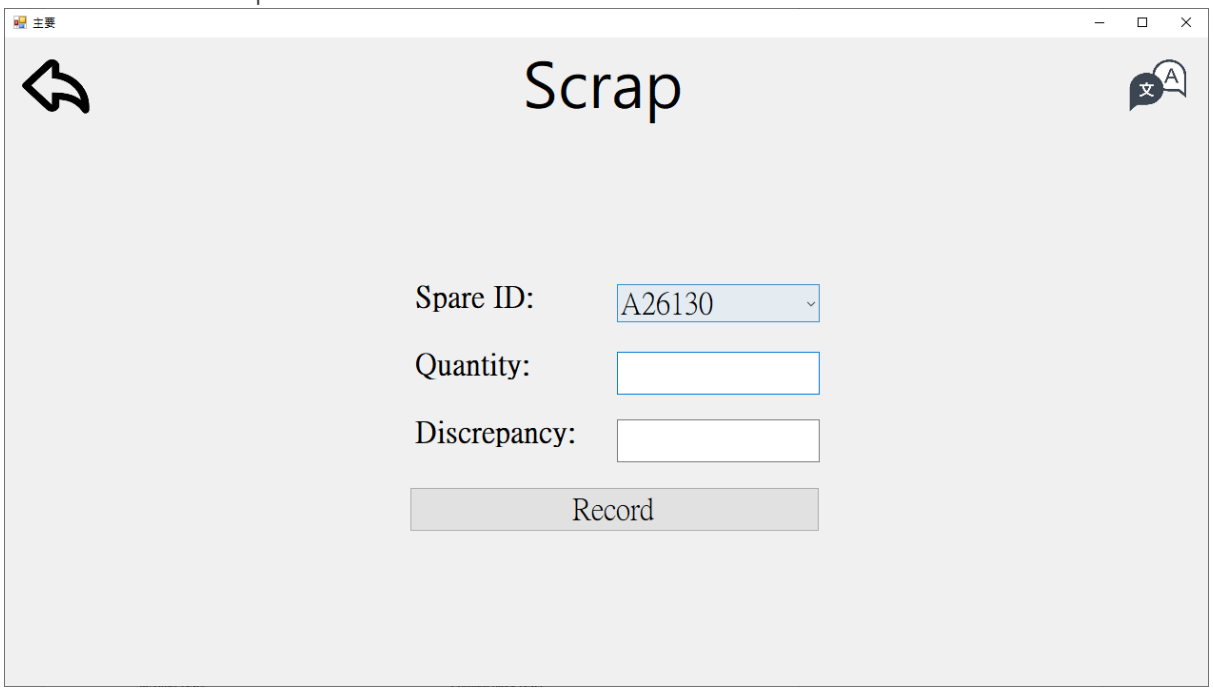


Put the contents of the selected record from the database into the corresponding location

Users can modify data

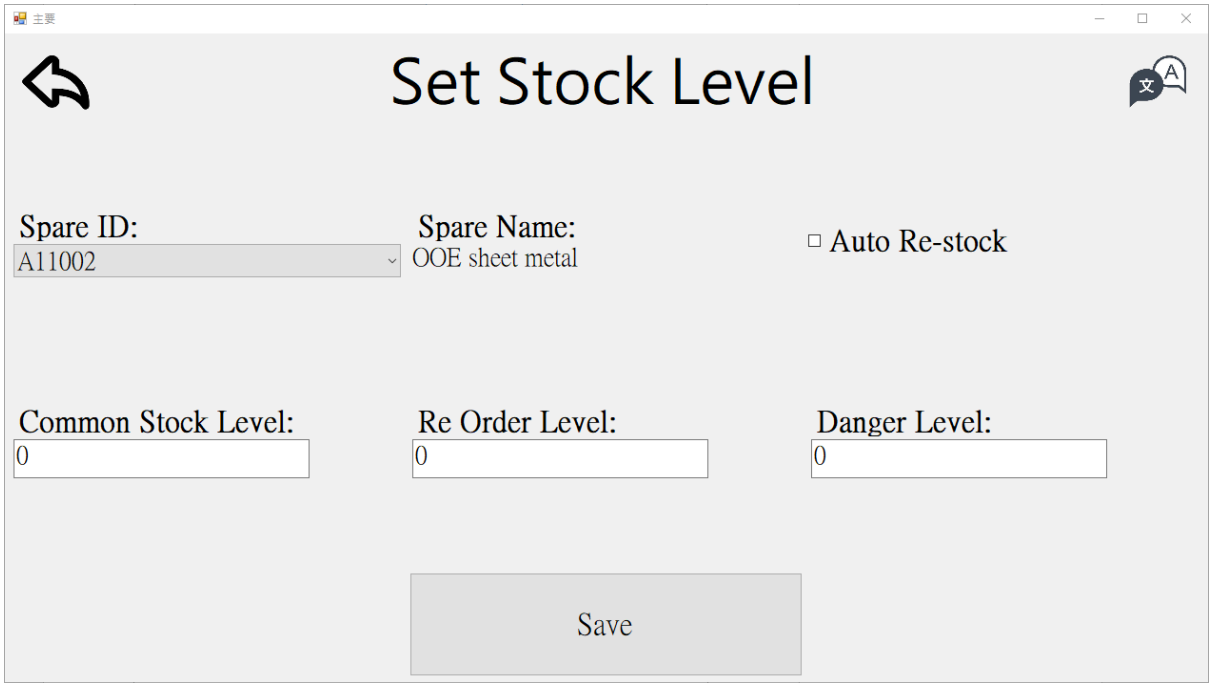
Users click the “Save” button to save, and then form will change to Edit Spare

15.2.13. Scrap



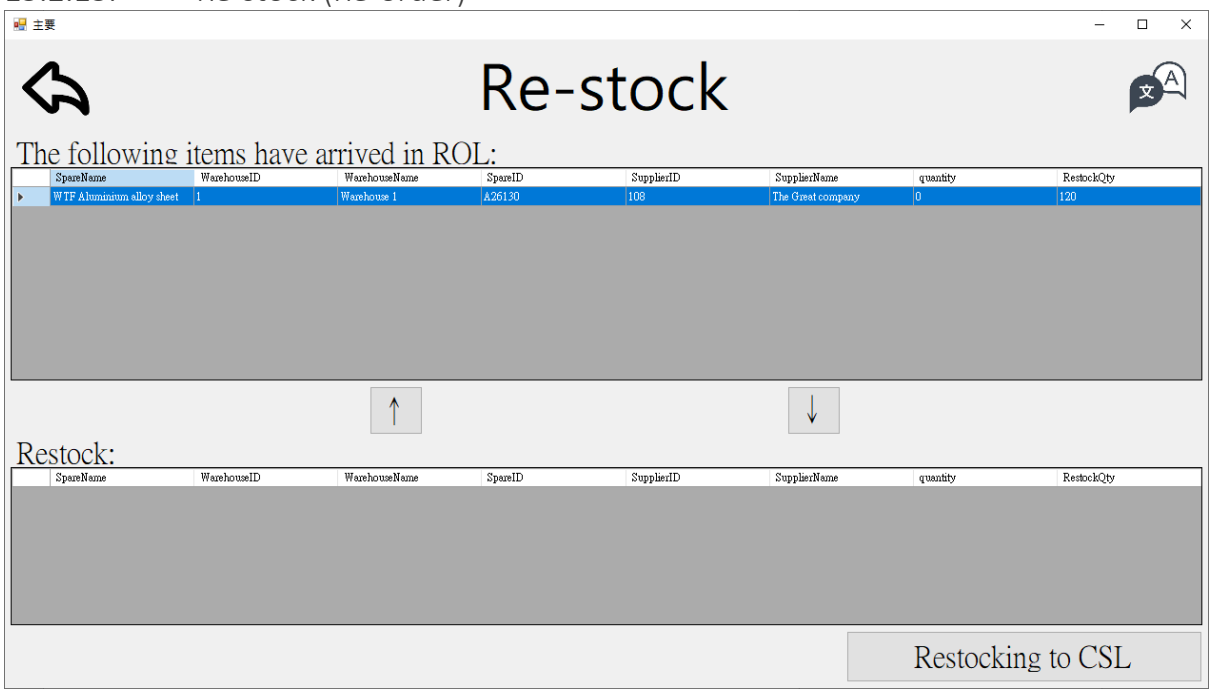
User can choose the spare ID, edit the quantity and discrepancy of spare, and click the “Record” to record the spare

15.2.14. Set Stock Level



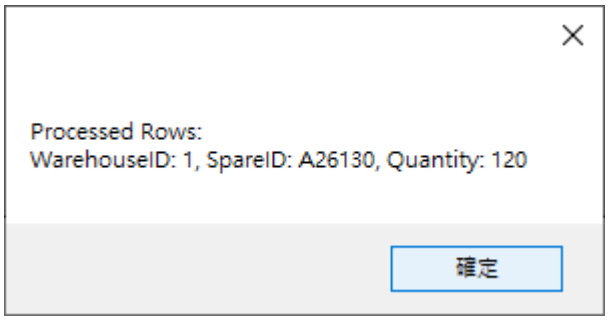
Users can choose the spare ID, set the Re-Order Level, Danger Level and Common Stock Level, and can see the spare name only. Users can click “Save” button to save in the system

15.2.15. Re-stock (Re-order)



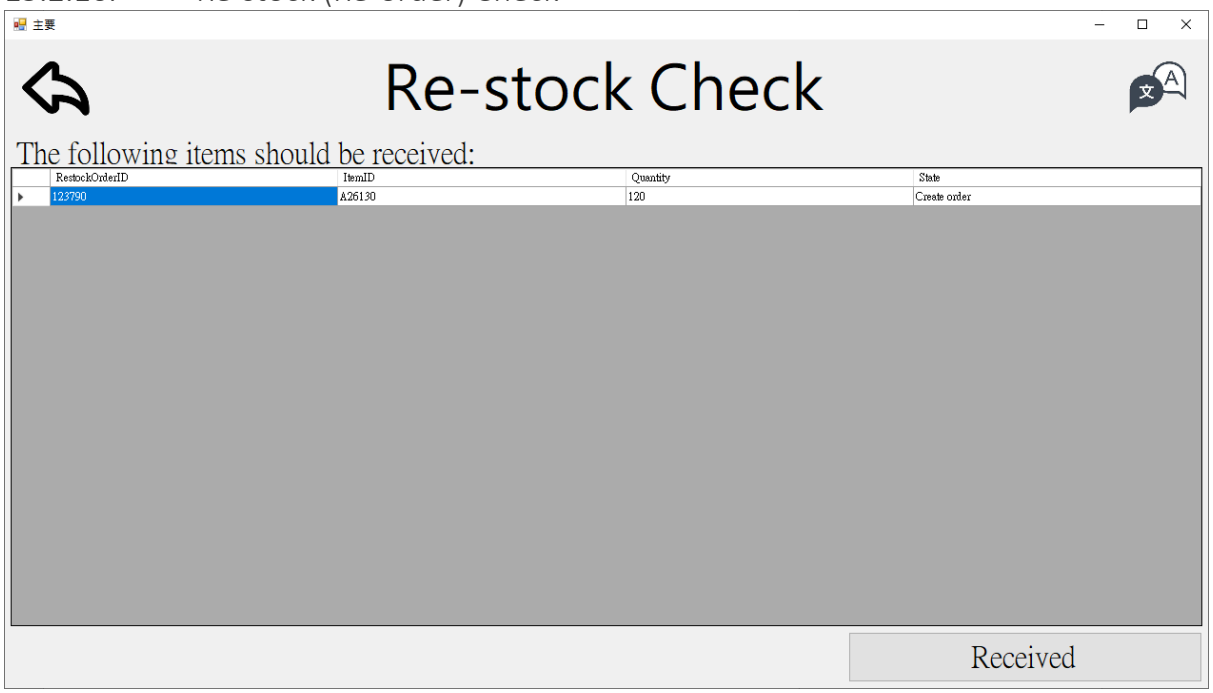
User can see the “The following item have arrived in ROL:” table, click the “Down” button to move item to “Restock” or “Up” button to move item to” The following item have arrived in ROL” table and click the “Restocking to CSL” to create a restock order.

15.2.15.1. After Restocking to CSL



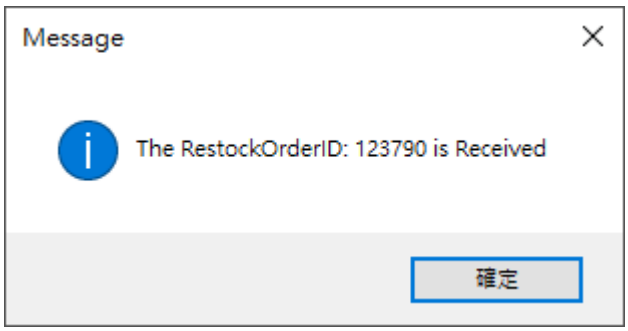
The system will show the message box to show the details of the order.

15.2.16. Re-stock (Re-order) Check



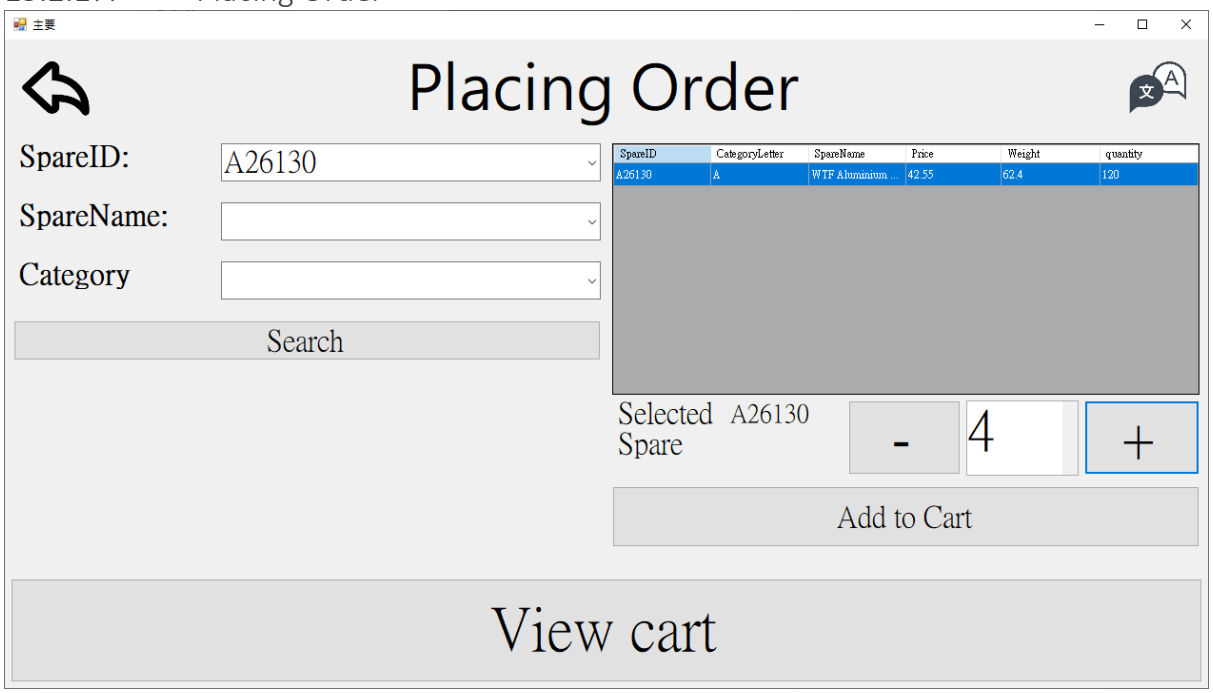
User can see the “Restock” table, and then click “Received” can mark the space is received

15.2.16.1. After click “Received” button



The System the message.A

15.2.17. Placing Order

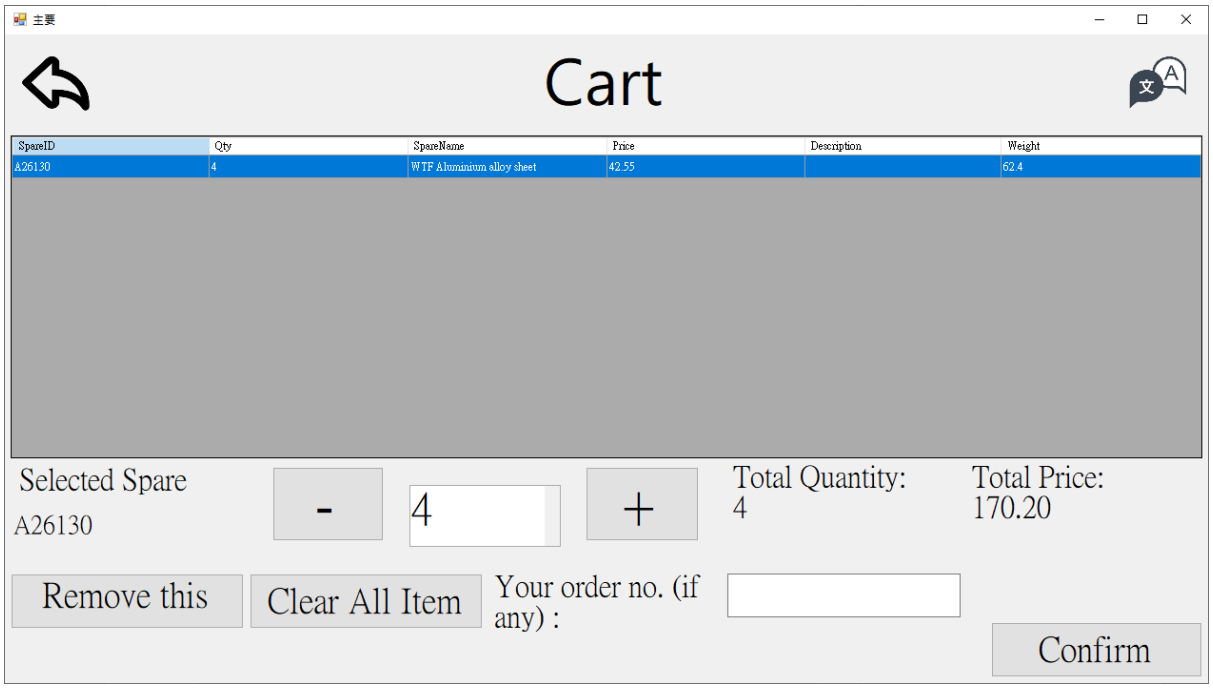


User can choose spare in “SpareID:”, “SpareName:”,” Category Type:” to input the spare, then click “Search” button then will show the selected spare in the table and “Selected spare Name”

User can click “-” or “+” button or input quantity between “-” or “+” to increase or decrease the quantity of the spare

User can click “Add to Cart” button to move the table in the “Cart”

15.2.17.1. Cart



User can see the table with “Placing Order” table (include list “Total Quantity:” and “Total Price:”) and label “Selected Spare Name:” to confirm that the spares are ready for ordering

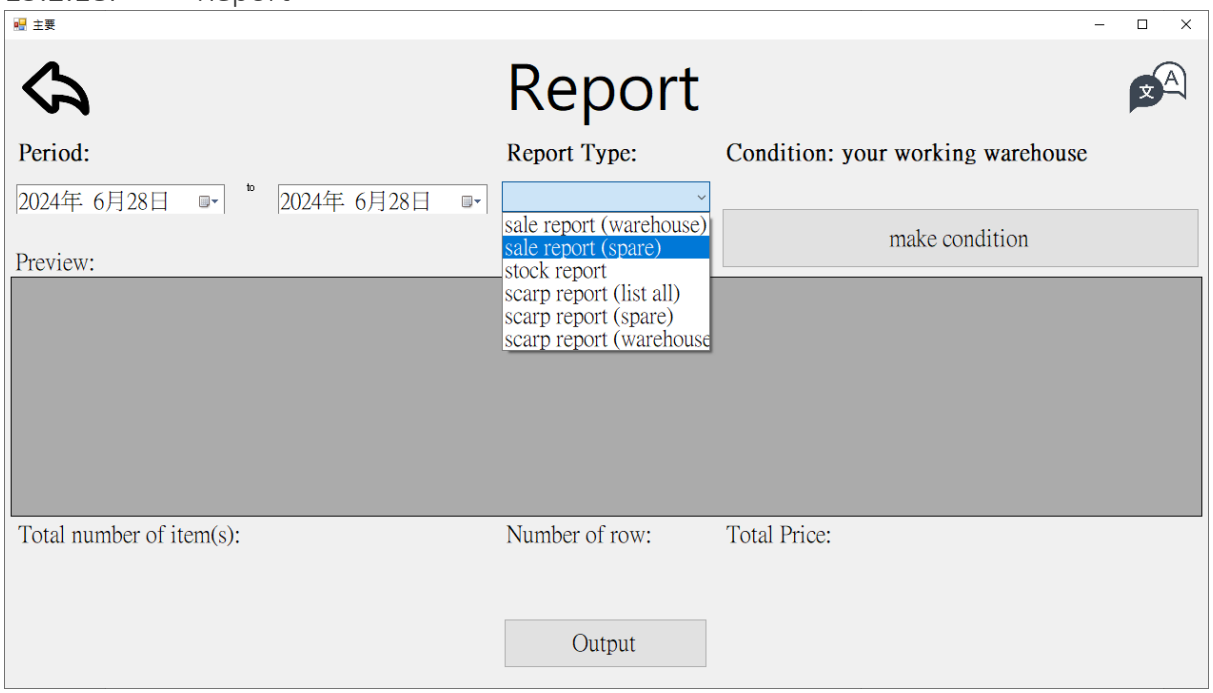
User can click “-” or “+” button to increase or decrease the quantity of the spare

User can click “Clear All” button to clear about spare in the table

User can input the order number in the label “Your order no. (if any)” if they have

Users can click “Confirm” button to send the order to LM

15.2.18. Report



User can set DateTimePicker to search date to date under the label “Period:”

User can search the ComboBox under the label “Report Type:”

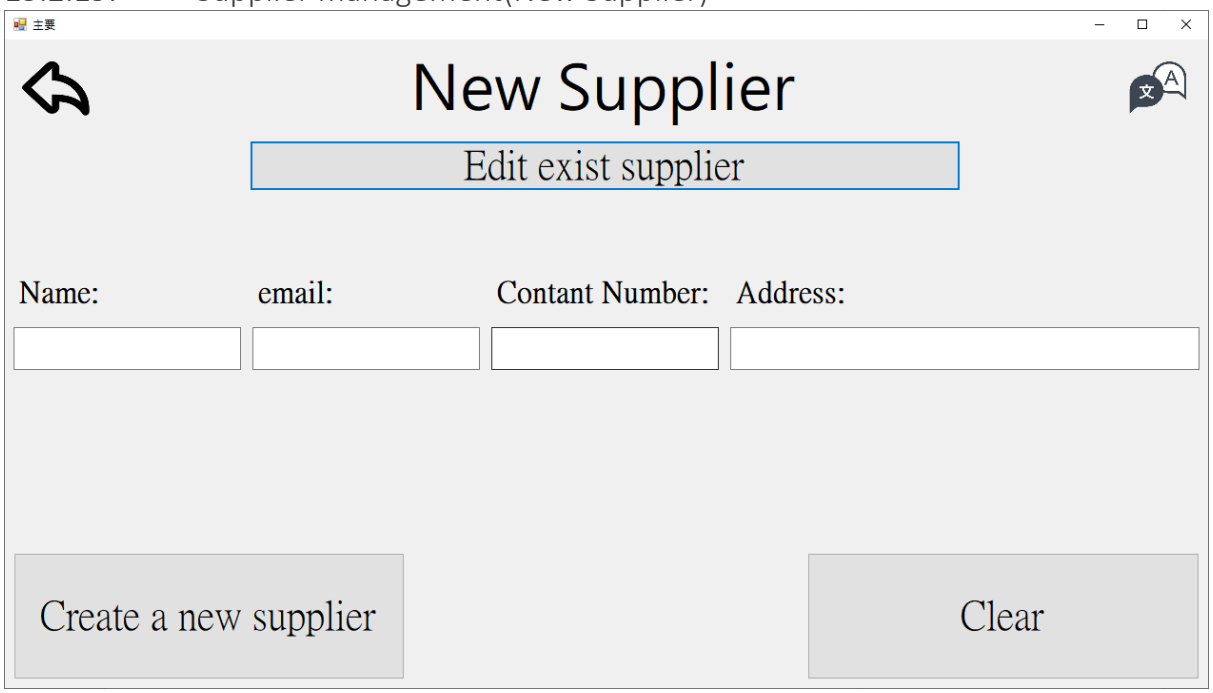
User can input the information under the label “Condition:”

User can see and confirm the table about the preview under the label “Preview:”

User can see the total number of items, total number of distinct items and total price

Users can click “Output” button to generate the report

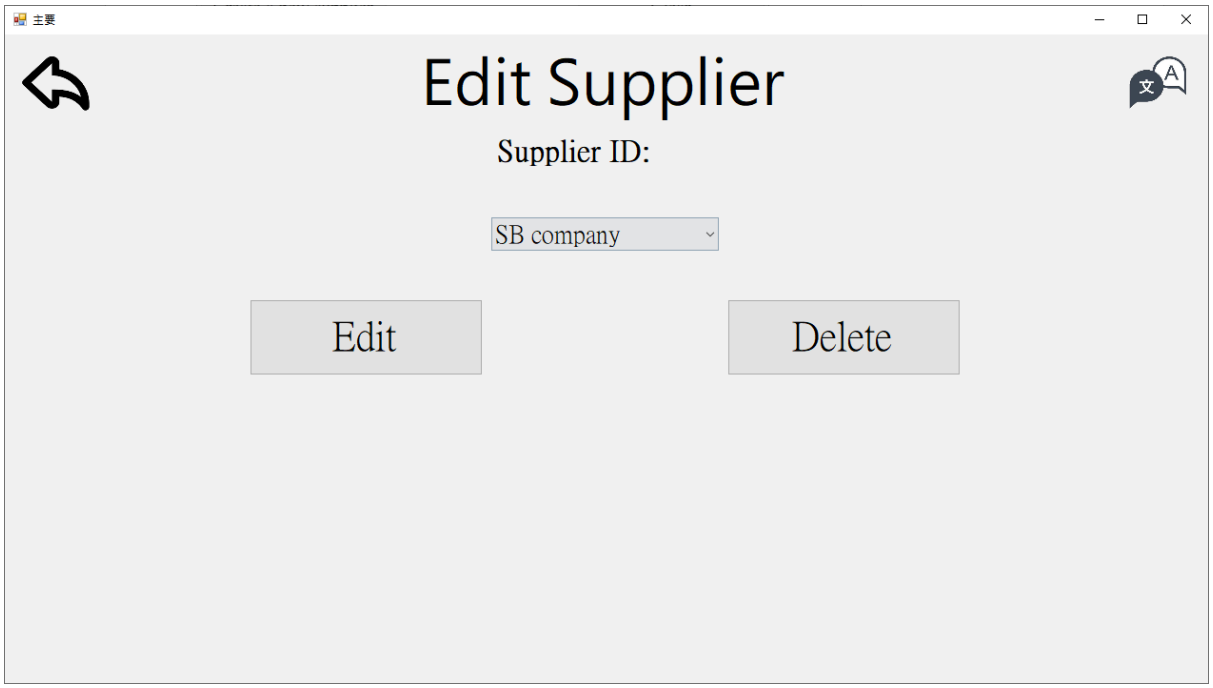
15.2.19. Supplier management(New Supplier)



User can click on "Edit Existing Vendor" to "Edit Vendor".

Users can add a new vendor by clicking "Create New Vendor"

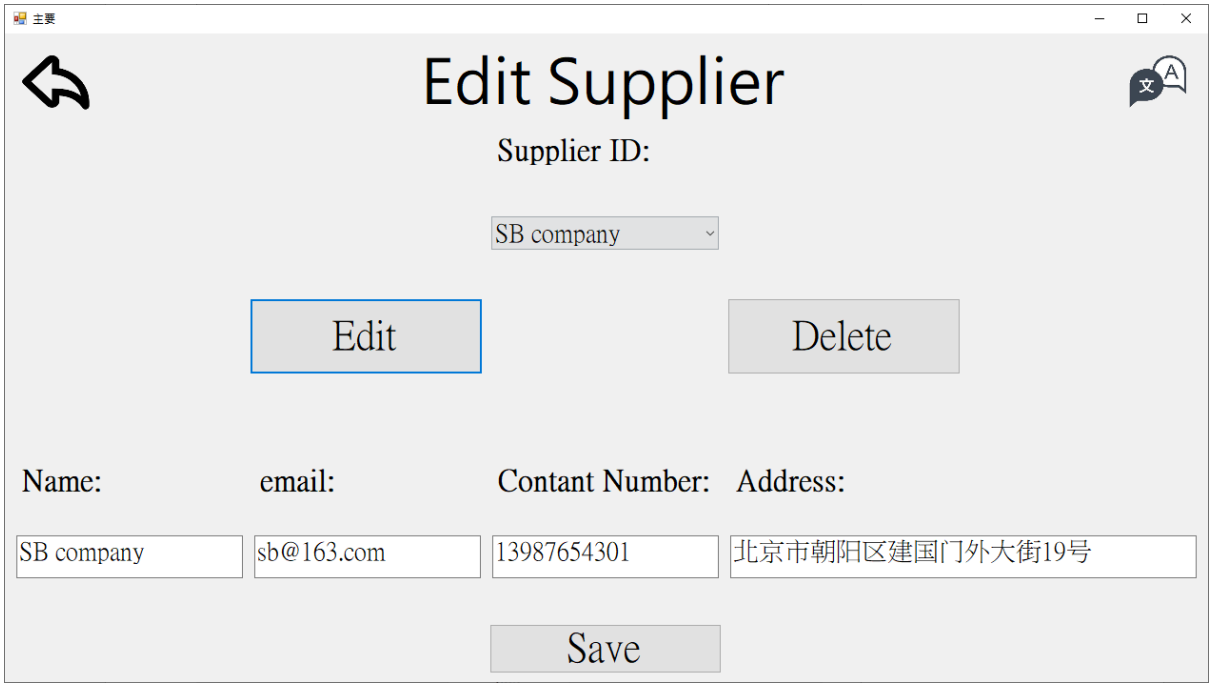
15.2.19.1. Edit supplier



User can choose the Supplier Name

User can click “Edit” or “Delete” the information of the selected Supplier

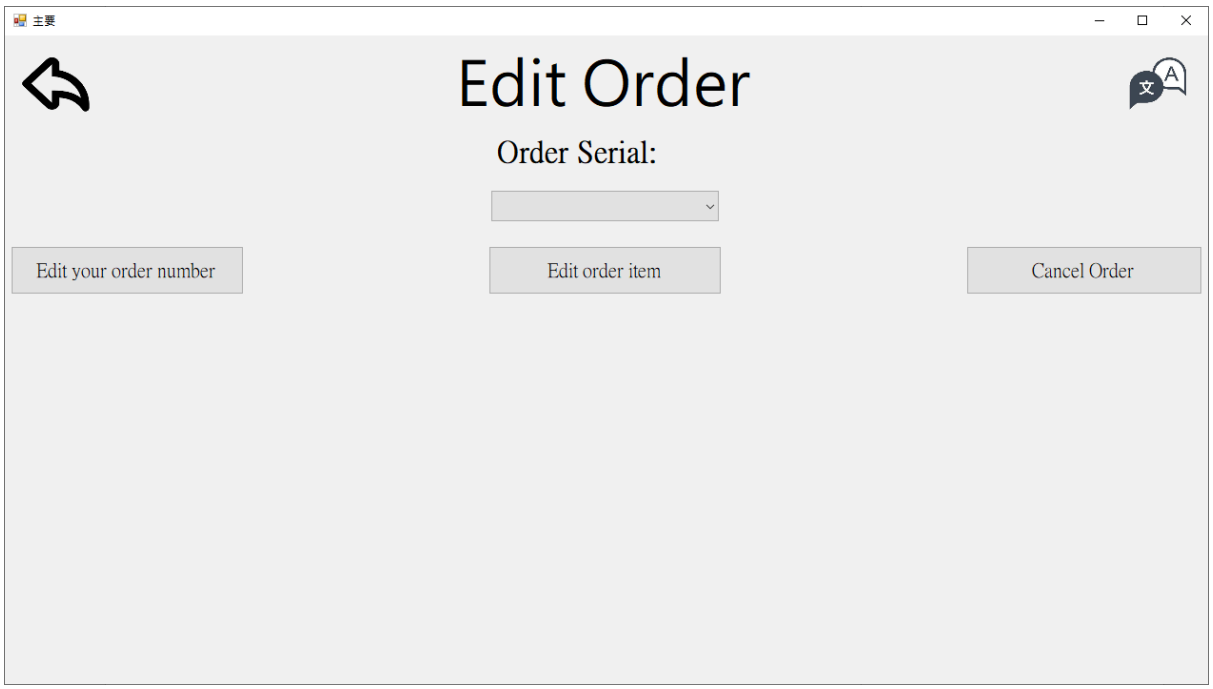
15.2.19.2. When user click “Edit” button



User can input information such as name, email, contact number and address

User can click “Save” to update the data

15.2.20. Edit Order

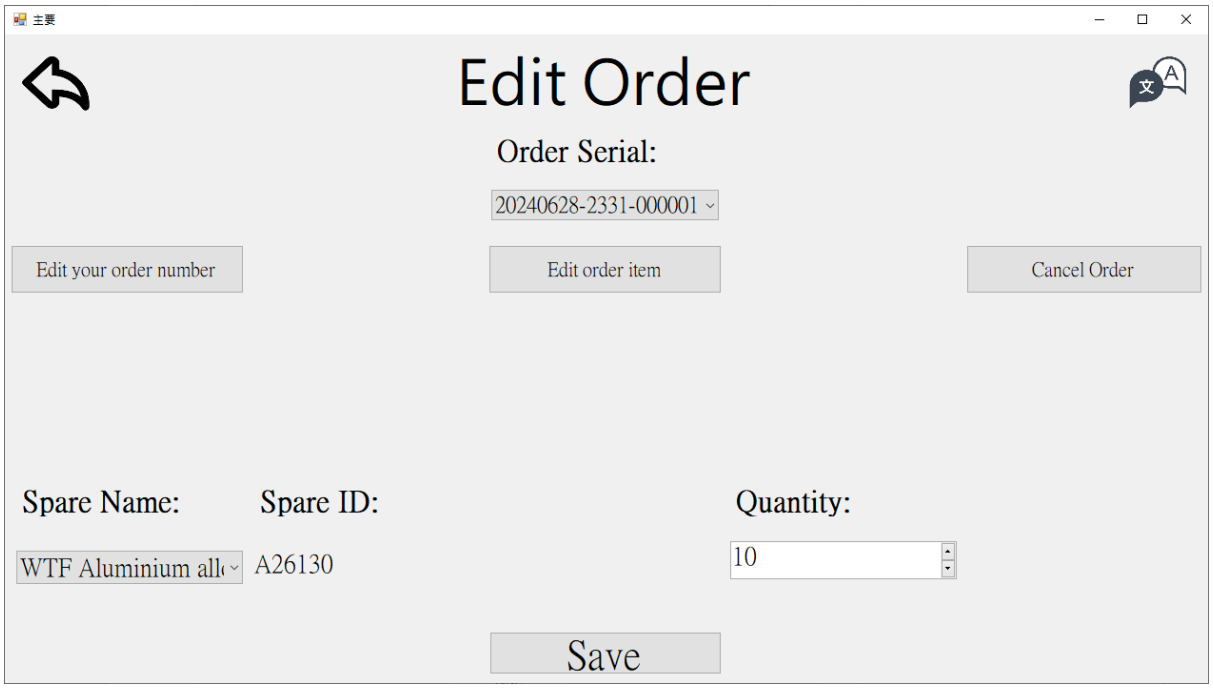


User can choose the order serial

User can click “Edit order item” button to “When user click “Edit order item” button”

User can click “Edit your order number” button to “When user click “Edit your order number” button”

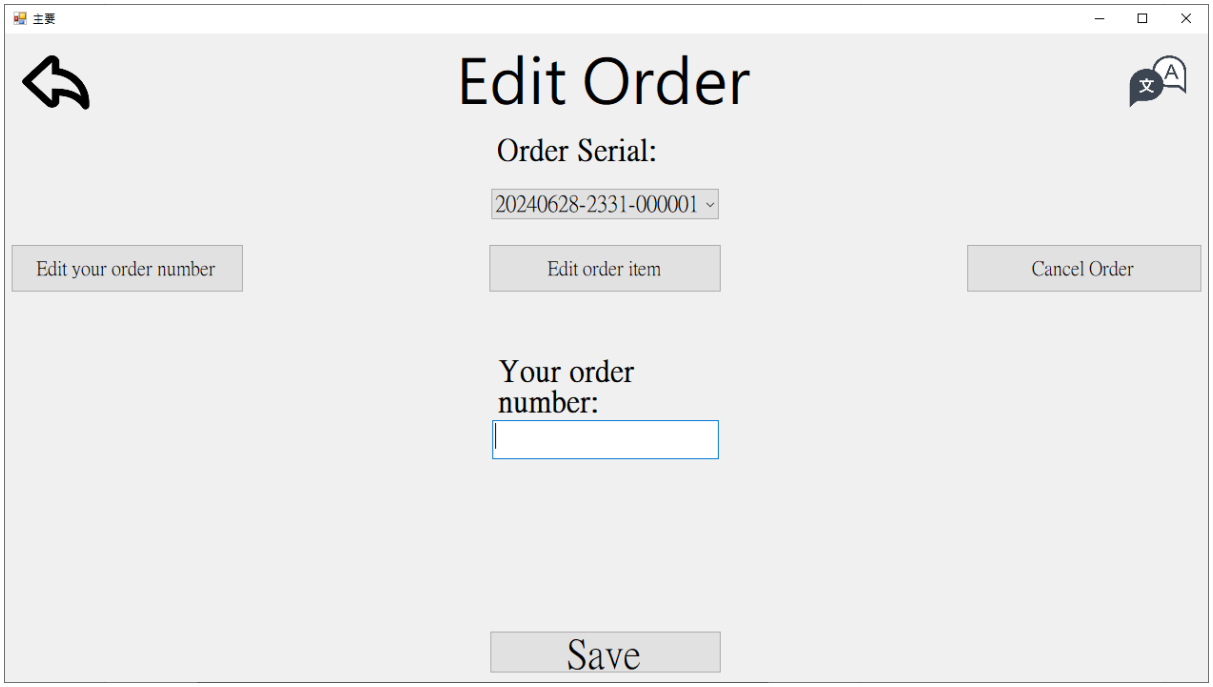
15.2.20.1. When user click “Edit order item” button



User can input the information of spare name, see the spare ID, align the Quantity

User can click “Save” to save the information in the system

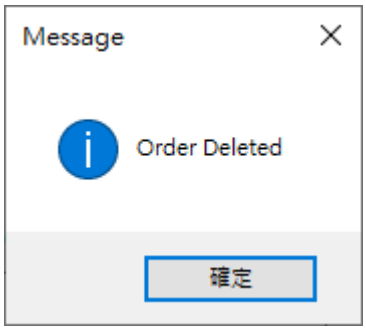
15.2.20.2. When user click “Edit your order number" button

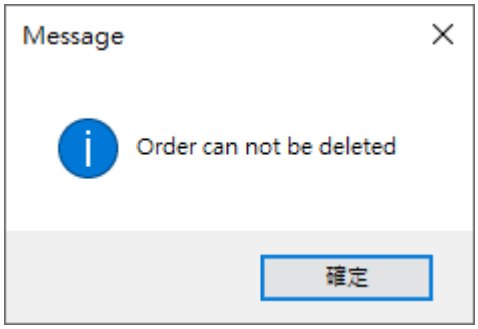


User can input the information the label of “Your order Number:”

User can click “Save” to save the information of the order number in the system

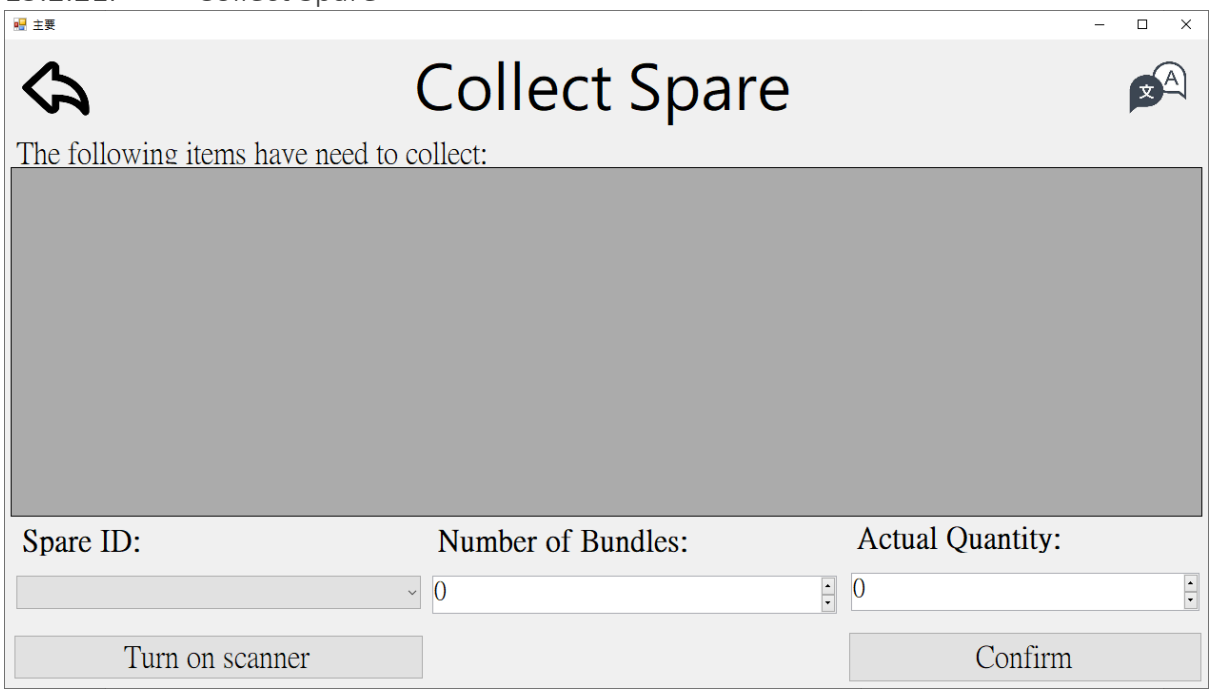
15.2.20.3. When user click “Cancel Order”





The system will show message.

15.2.21. Collect Spare



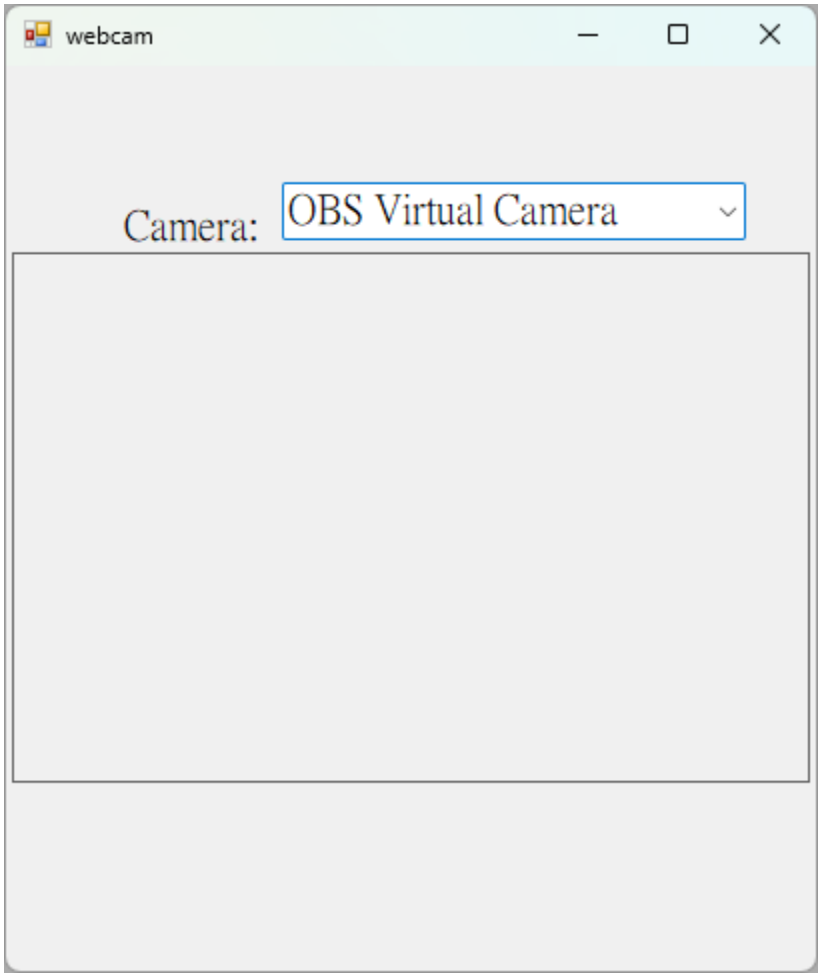
This page displays the items that need to be collected

User can select spares ID, enter the number of bundles and actual quantity

User also can click “Turn on scanner” to “Webacm for QR-code for scanner”

User can click “Confirm” to submit the number of bundles and actual quantity

15.2.21.1. Webcam for QR-code scanner



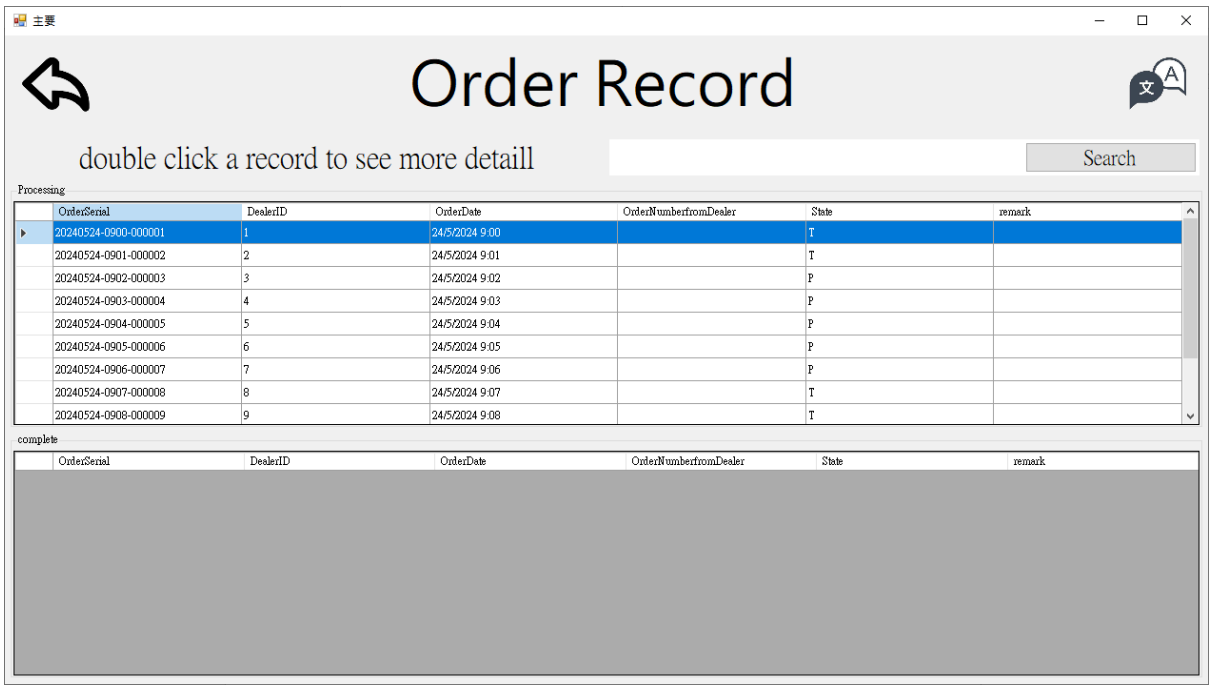
Users can select the camera on the drop-down list

This page will connect to the camera

Users just need to scan the QR code

When scanning is completed will automatically go back to page “Collect Spare”

15.2.22. Order Record



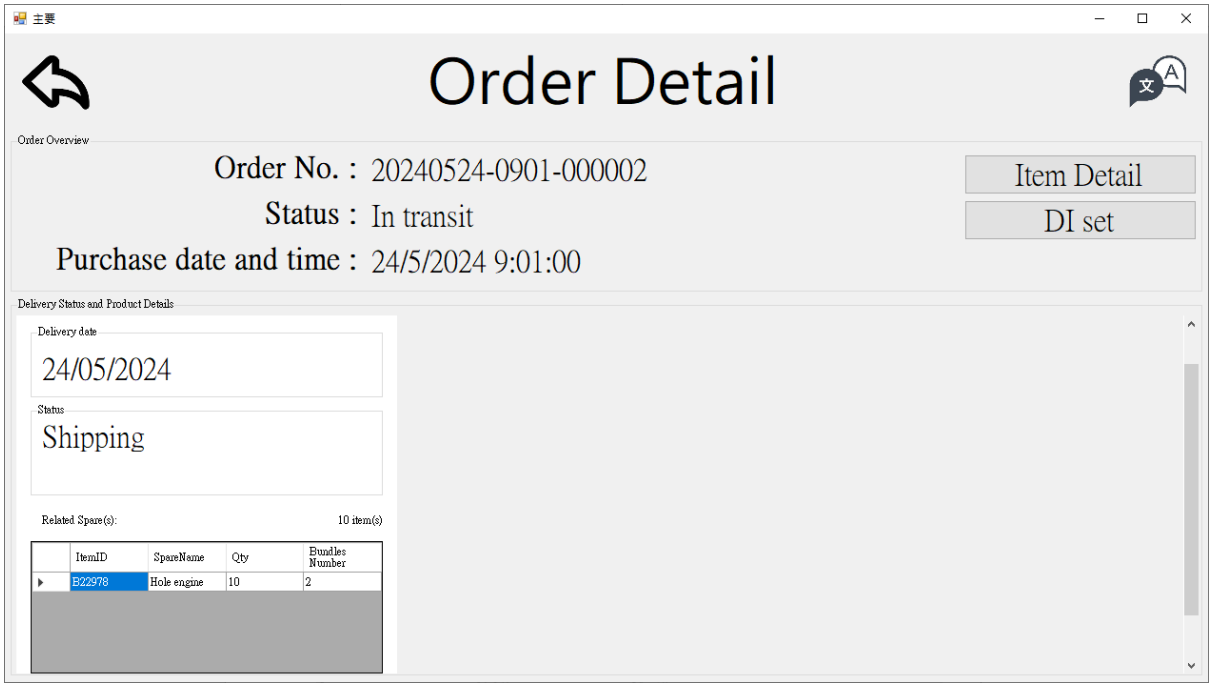
On the top, there can show processing order

On the down, there show complete order

Users can double click the displayed record to see more detail

When users double click the detail, the page will change to “Order Detail”

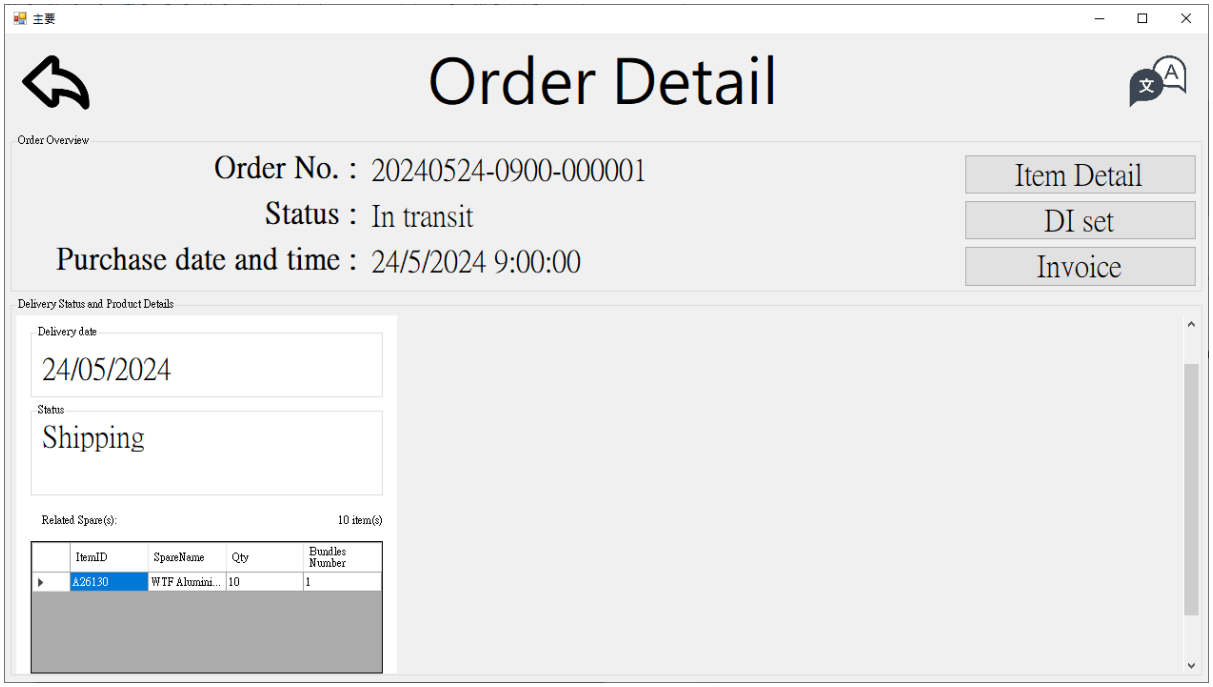
15.2.22.1. Order Detail



This page will display order details such as order number and status.

If user clicks on "DI Setting" "Item Detail" button, it will switch to the corresponding page.

If the user is a delivery_worker, the "Invoice" button will appear, clicking on it will switch to the "Invoice" page

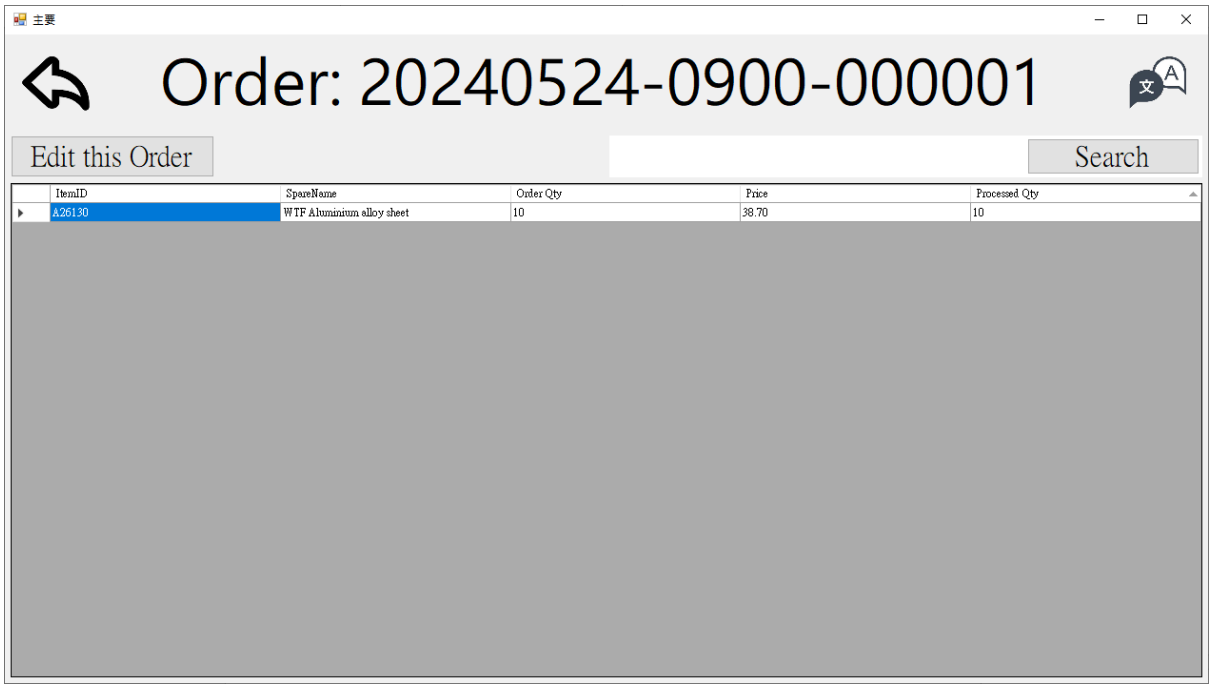


15.2.22.1.1. Despatch Instruction Set

This page shows the data of DI set

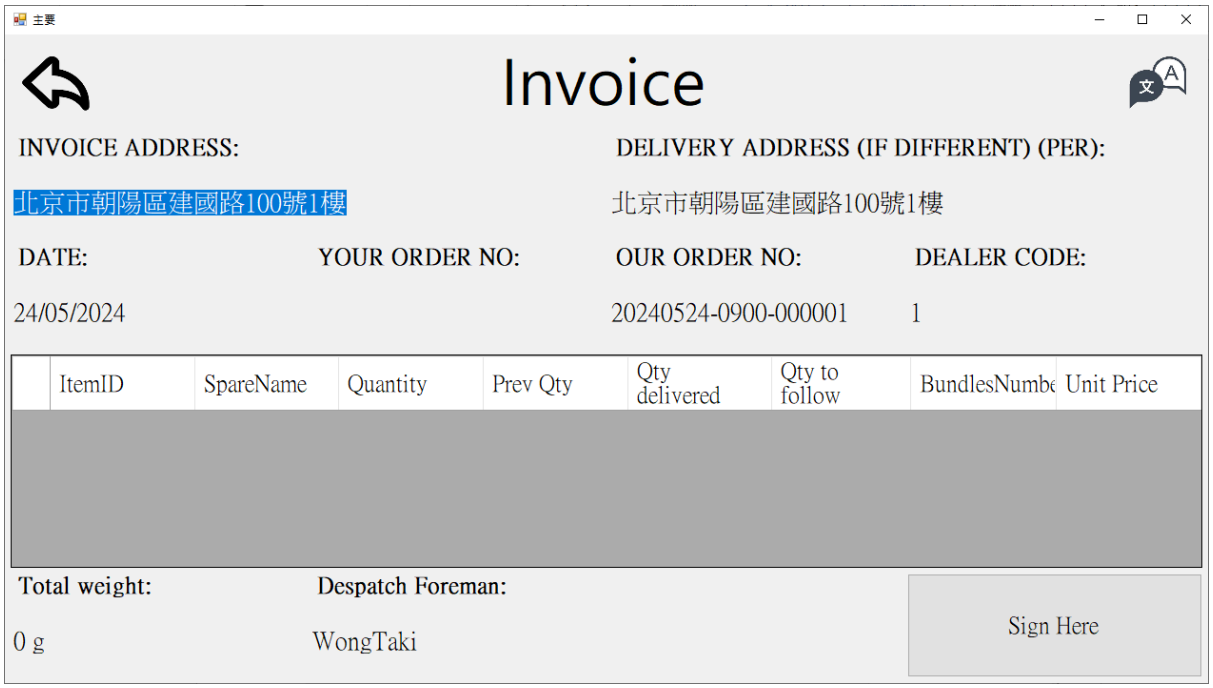
User can click the button “Export as PDF” to print the DI set as PDF

15.2.22.1.2. Item Detail



This page shows the data of Order.

15.2.22.1.3. Invoice

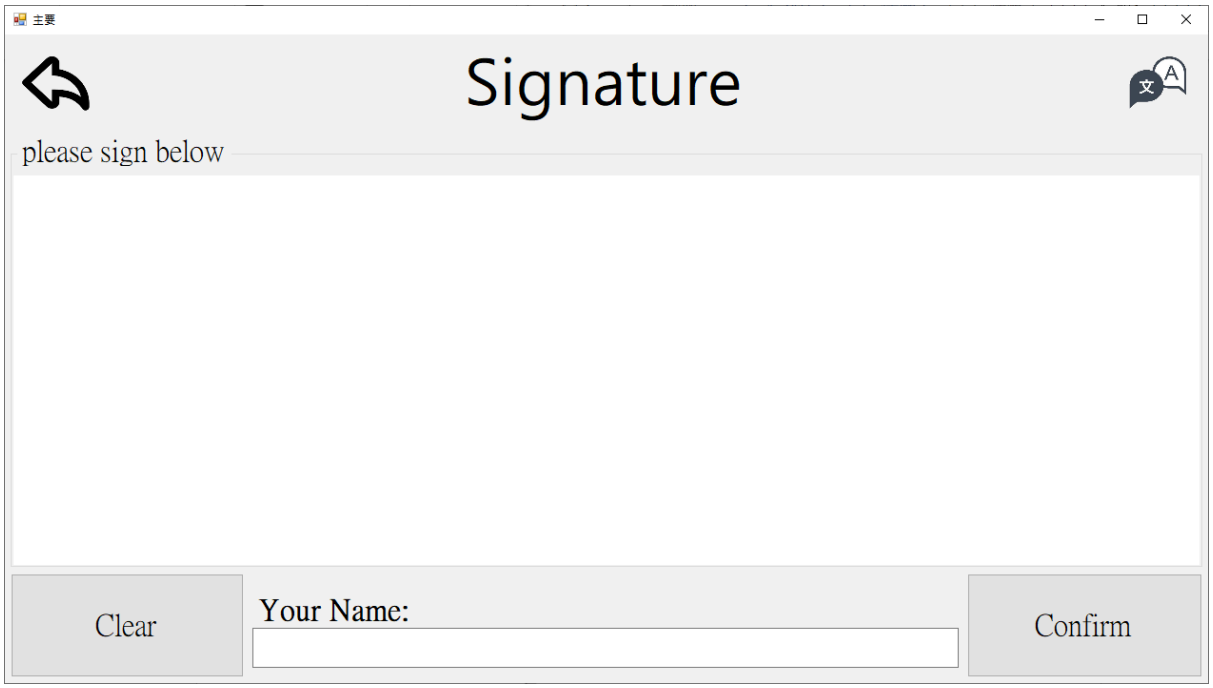


This page shows the data of Invoice

User can click the button “Sign Here” to sign

After user click the button “Sign Here”, the page will change to “Signature”

15.2.22.1.4. Signature

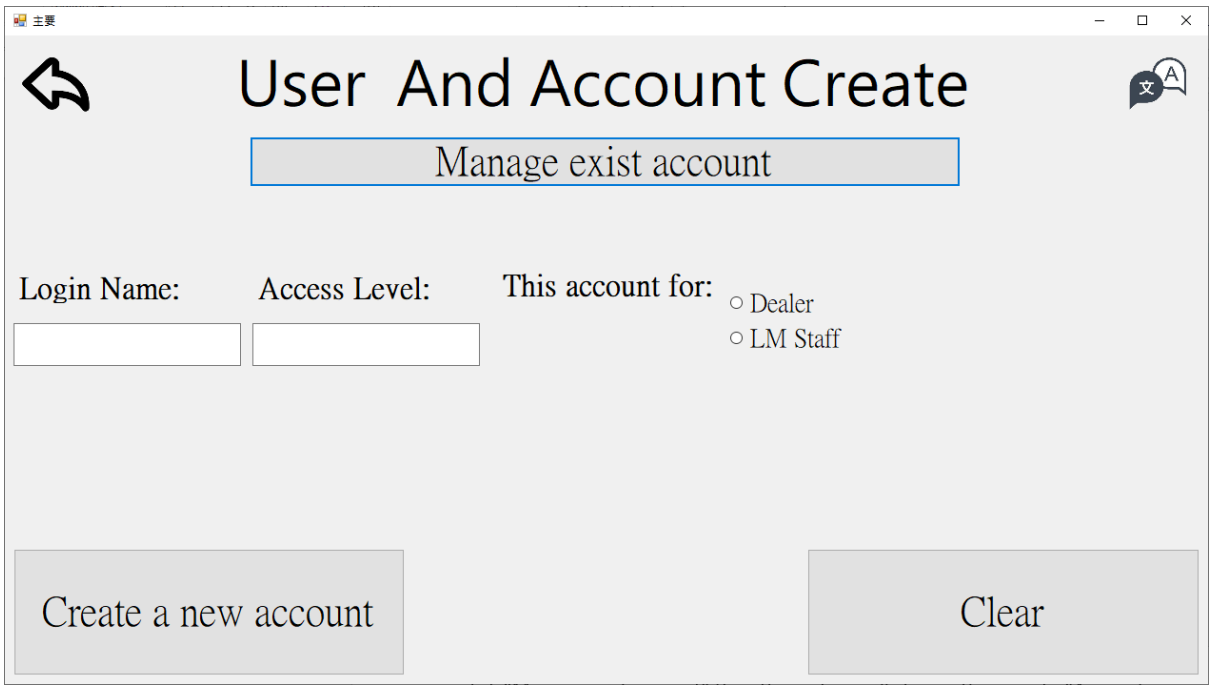


User can sign on this page using the pad

Then click the “Confirm” button to confirm the signature

User can click “Clear” button to clear all input

15.2.23. User and Account Create



If the user wants to manage an existing account on this page, user just need to click “Manage exist account” then the system will change the page to “Account Management”

If a user wants to create an account on this page. Users need to enter login name, access level, and select the account for dealer or LM staff

Then the user can click the button “Create a new account” to create a new account and the page will change to “New Staff” or “New Dealer”.

15.2.23.1. New Staff

After user click the button “Create a new account”, the system will change the page to “New Staff”

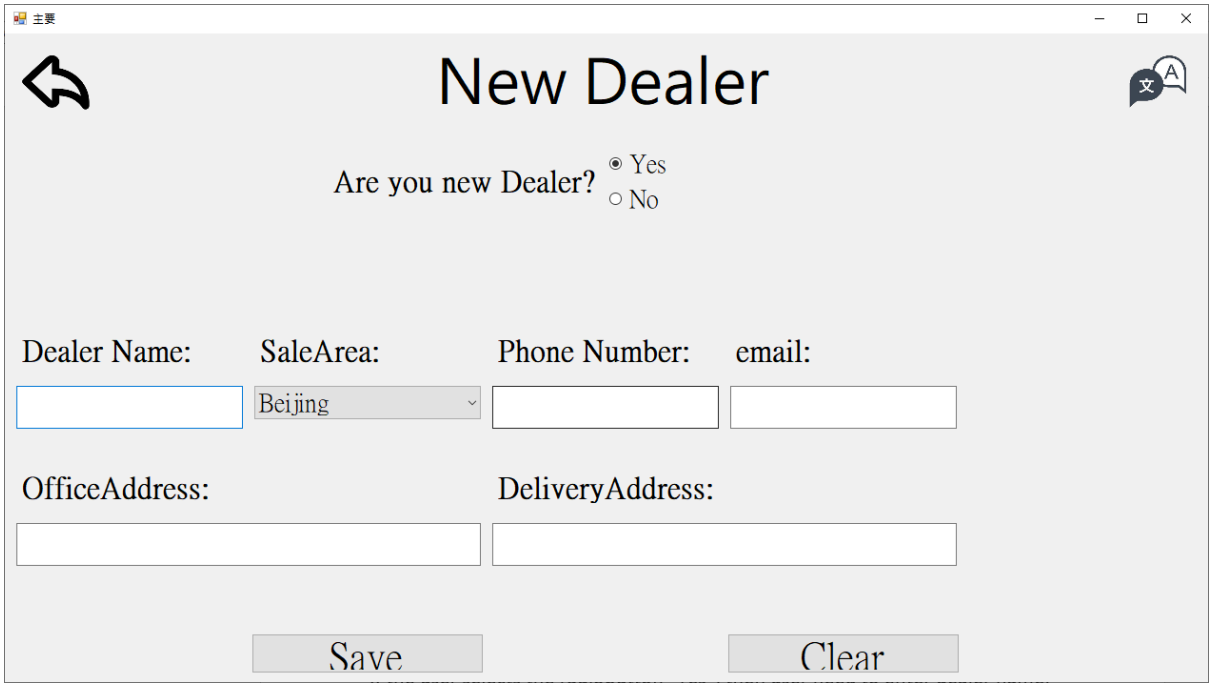
If user select the radio button “Yes”, then user need to enter position, staff name, sale area, department, phone number and select the gender

After entering, the user can click the “Save” button to create a new staff and new staff account

15.2.23.1.1. Is not new staff

If the account is being created for an old staff member, the user just needs to enter staff ID and click the “Save” button to create a new account

15.2.23.2. Is new Dealer.



After user click the button “Create a new account”, the system will change the page to “New Dealer”

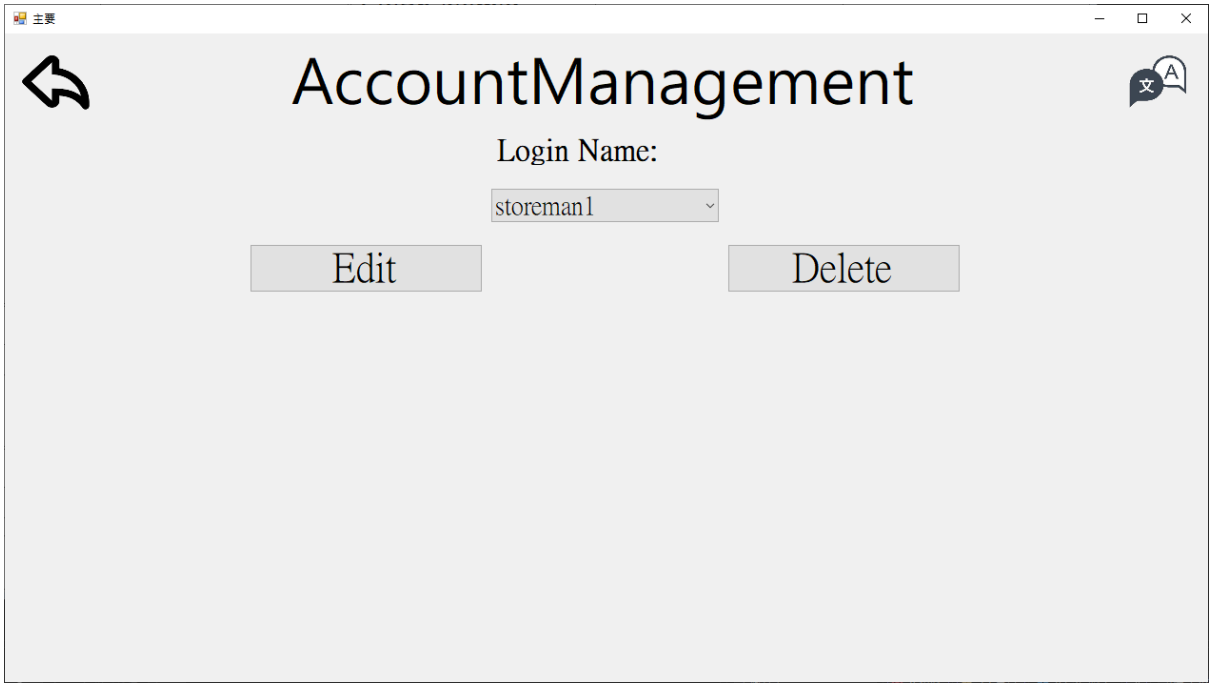
If the user selects the radio button “Yes”, then user need to enter dealer name, sale area, phone number, email, office address and delivery address

After entering, the user can click the “Save” button to create a new dealer and his account

15.2.23.2.1. Is not new Dealer

If the account is being created for an old dealer, the user just needs to enter dealer ID and click the “Save” button to create a new account

15.2.23.3. Account Management



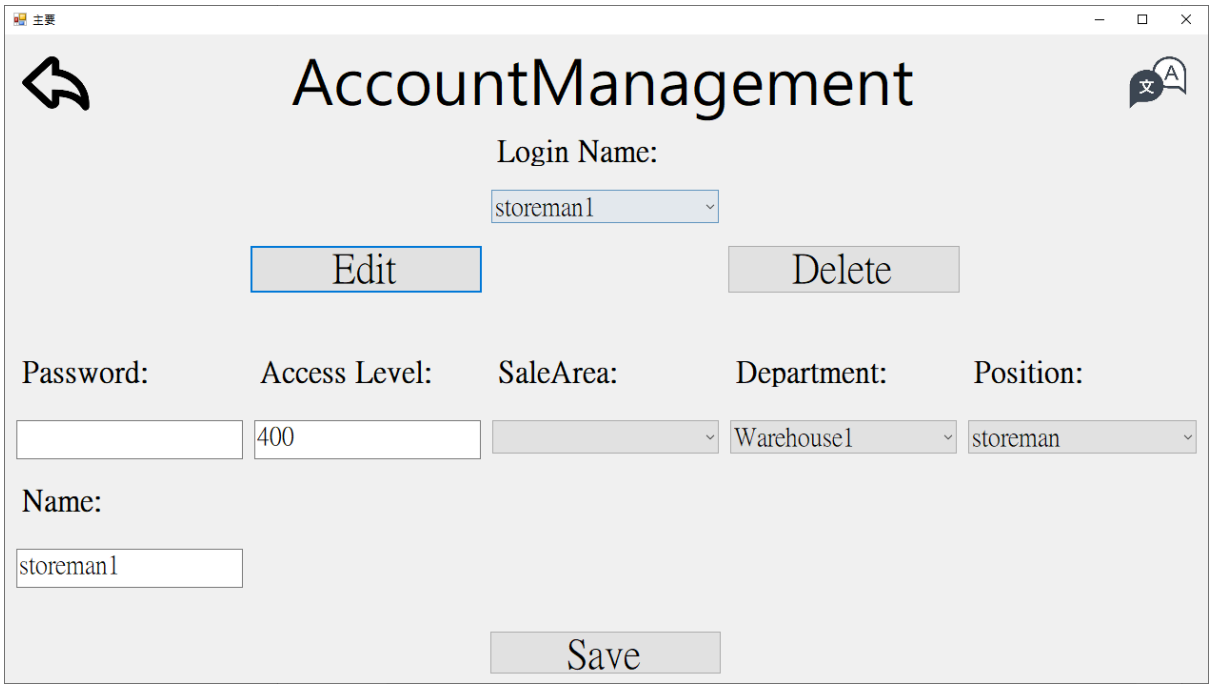
When user click “Manage exist account” then the system will change the page to “Account Management”

Users need to select the target login name on the drop-down menu

Then click the button “Edit” to edit the target account

If the user clicks the "Delete" button, the login name will not appear in the drop-down list.

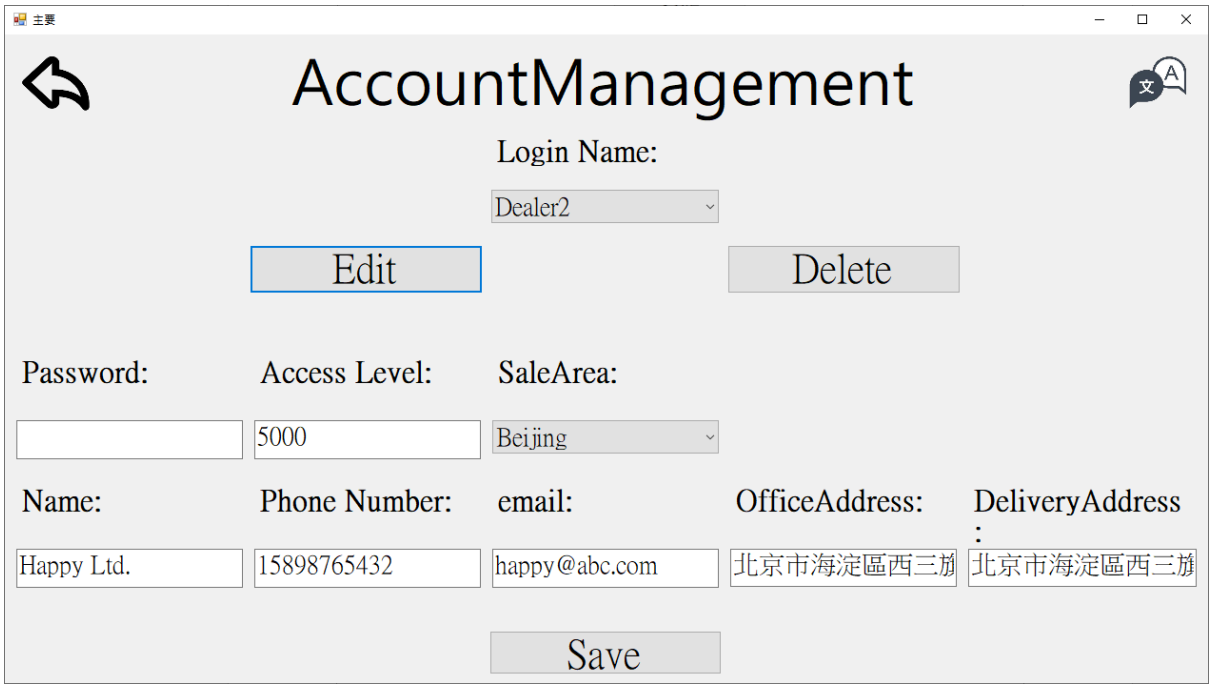
15.2.23.3.1. If the Login Name refer to our staff



When the user selects “Our Staff”, then user needs to enter password, access level, sale area, department, position, and name

Then click “Save” to save the change

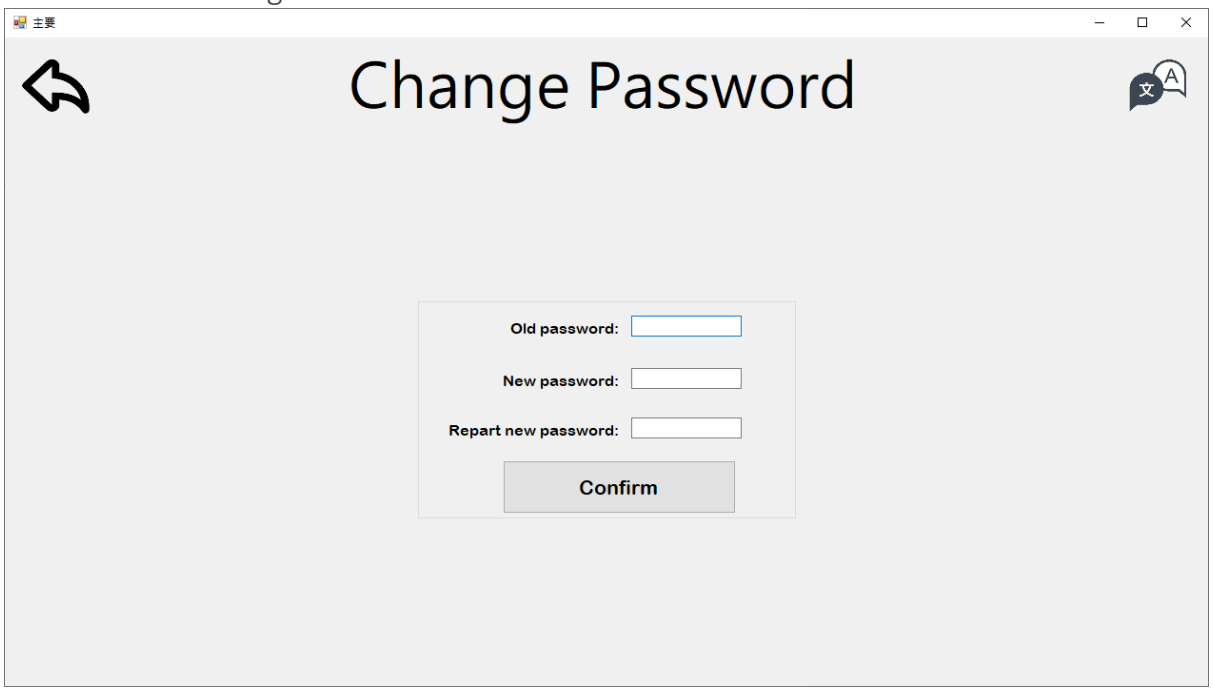
15.2.23.3.2. If the Login Name refer to dealer



When user selects “dealer”, then user need to enter password, access level, sale area, name, phone number, email, office address and delivery address

Then click “Save” to save the change

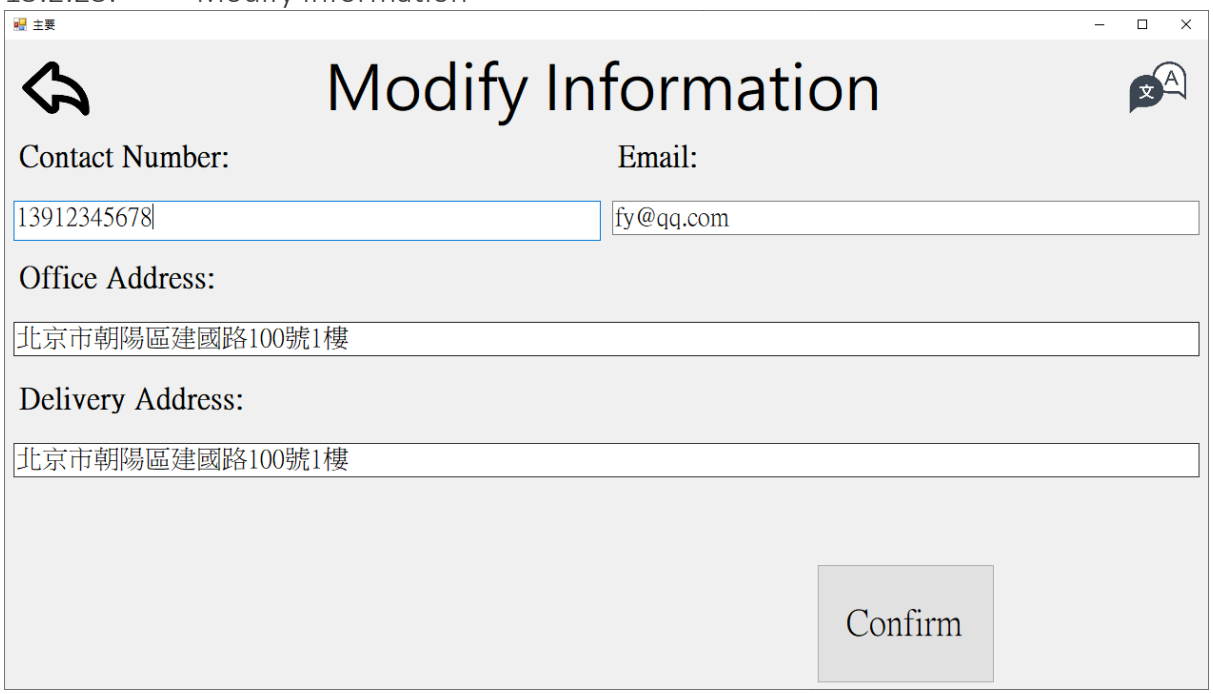
15.2.24. Change Password



Only the dealer can change the password.

Dealer needs to enter the old password and the new password.

15.2.25. Modify Information



Only the dealer can change the own information.

15.3.Report Design

15.3.1.1.Csv type

Users can choose from six types of reports.

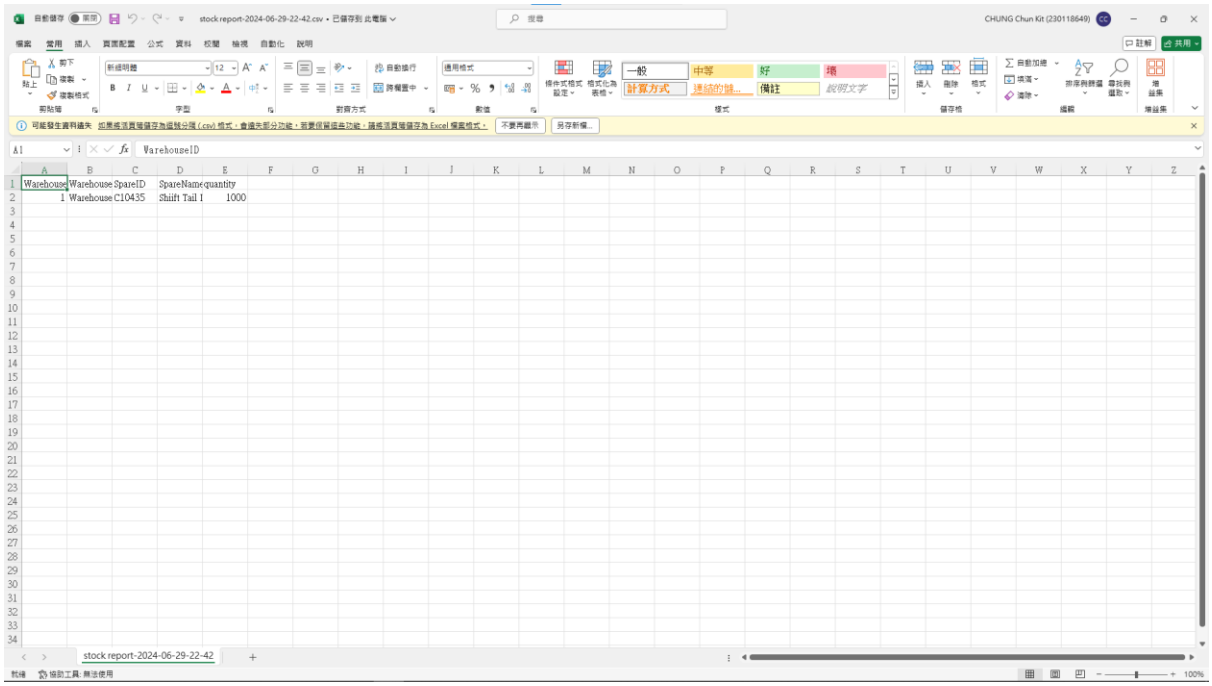
After user click input button,

The user will get a csv file.

There will be a date on the name.

stock report-2024-06-29-22-42.csv	29/6/2024 22:42	Microsoft Excel ...	1 KB

This is the contents of the file.

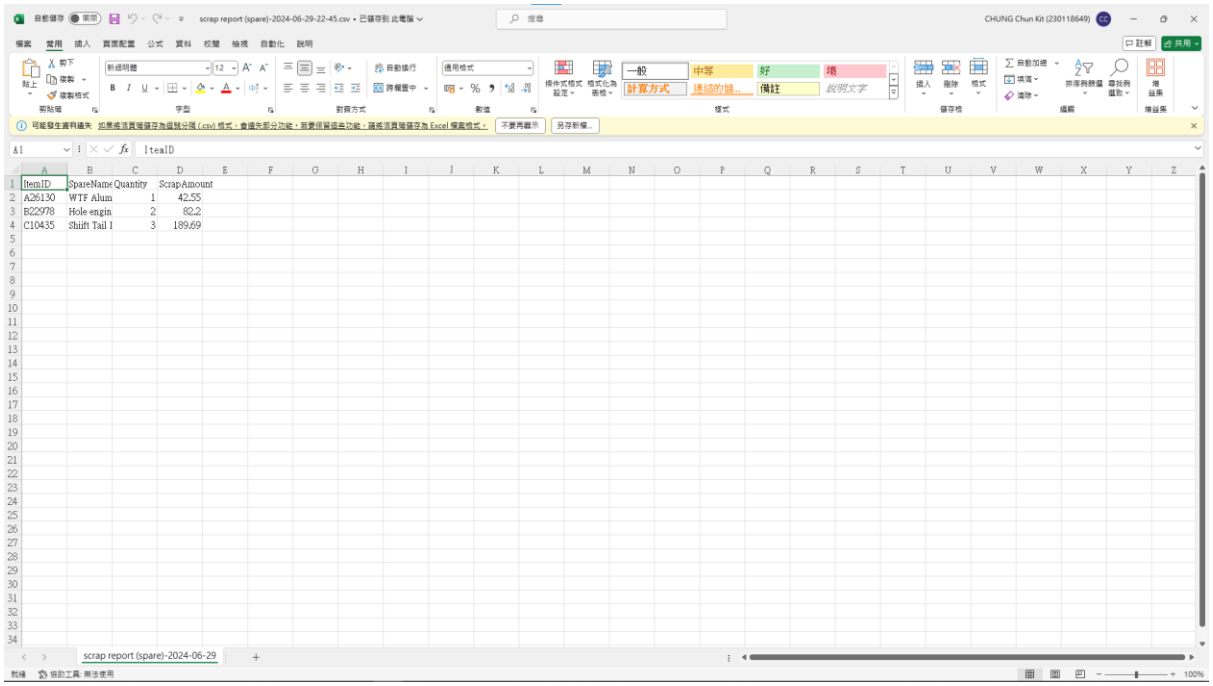


Warehouse	Warehouse	SpareID	SpareName	quantity
1	Warehouse	C10435	Shiift Tail I	1000

There are different types of report.

scrap report (spare)-2024-06-29-22-4...	29/6/2024 22:45	Microsoft Excel ...	1 KB
stock report-2024-06-29-22-42.csv	29/6/2024 22:42	Microsoft Excel ...	1 KB

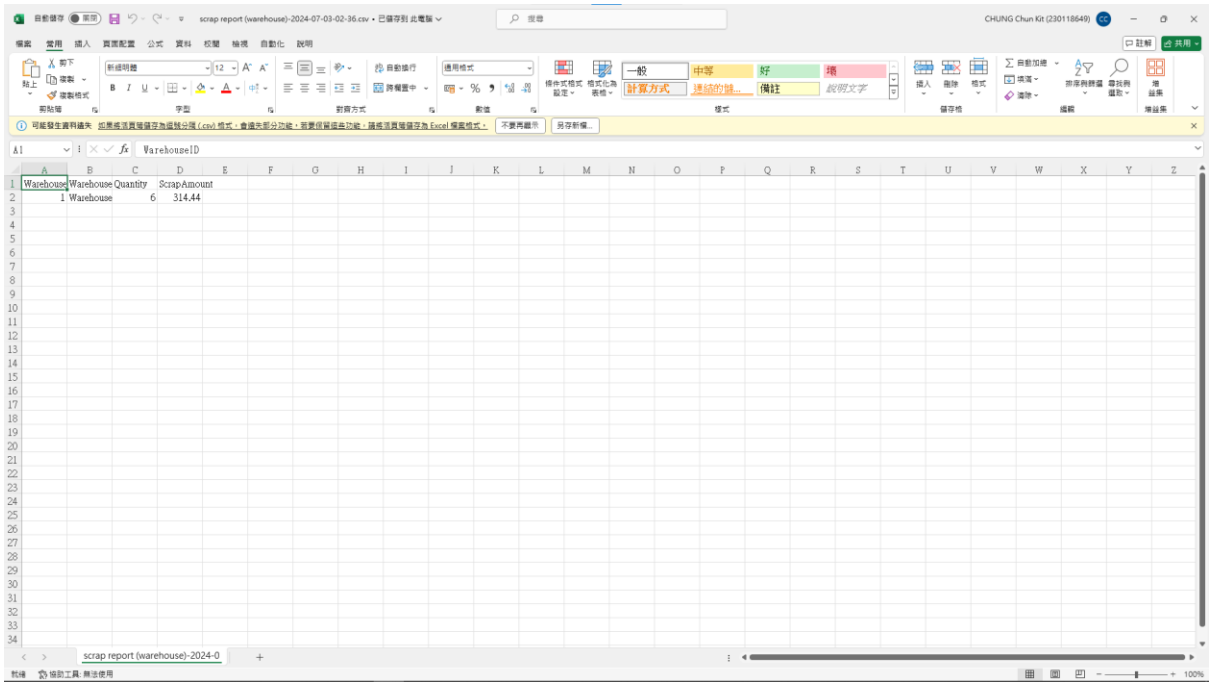
They have different names.



ItemID	SpareName	Quantity	ScrapAmount
A26130	WTF Alum	1	42.55
B22978	Hole engin	2	82.2
C10435	Shift Tail I	3	189.69

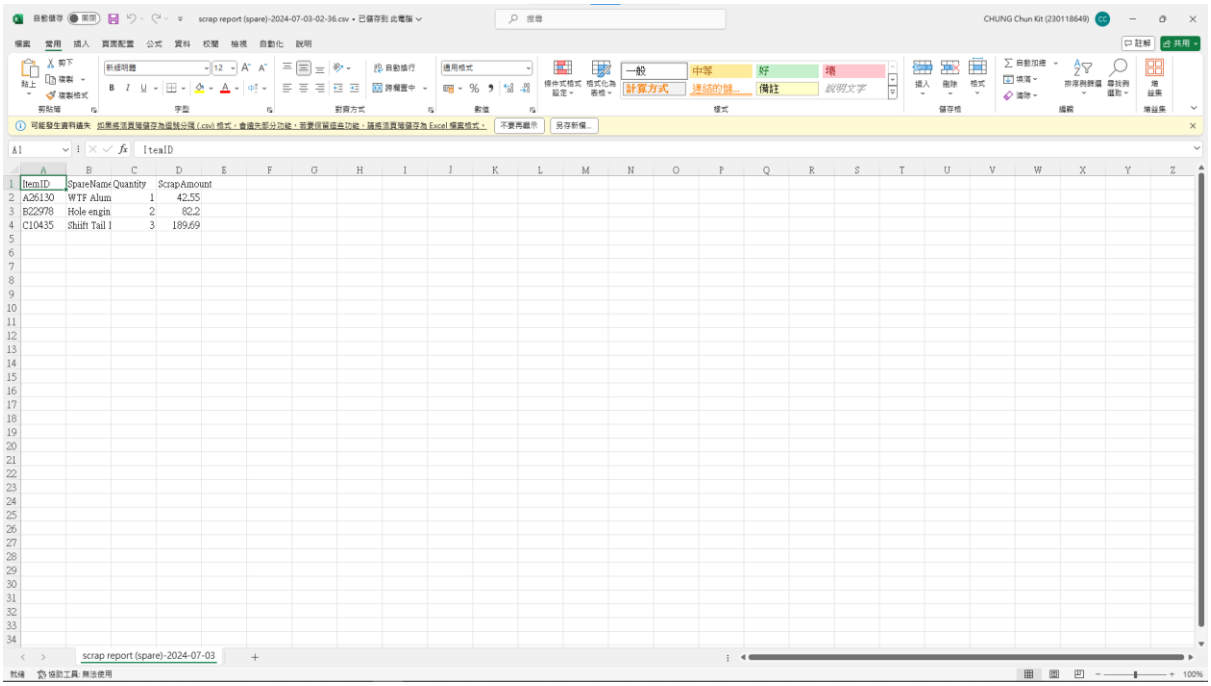
The content and format will change accordingly.

15.3.1.1.1. scrap report (warehouse)



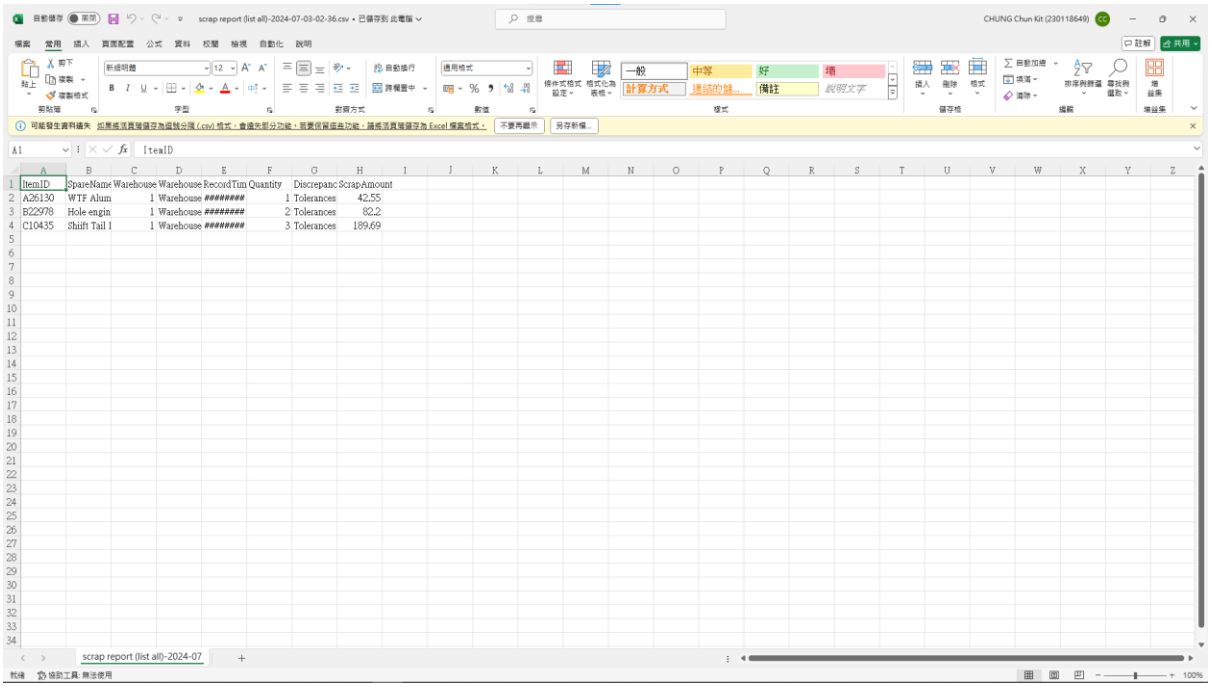
WarehouseID	WarehouseQuantity	ScrapAmount
1	Warehouse 6	314.44

15.3.1.1.2. scrap report (spare)



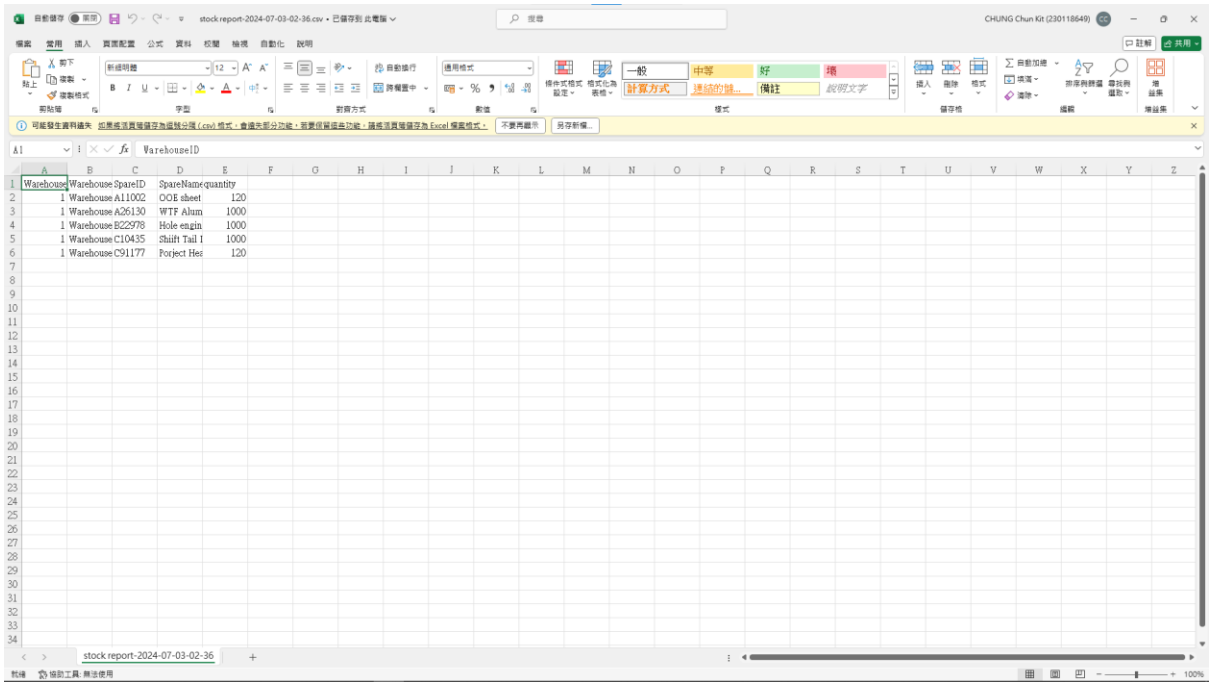
ItemID	SpareName	Quantity	ScrapAmount
A26130	WTF Alum	1	42.55
B22978	Hole engin	2	82.2
C10435	Shift Tail I	3	189.69

15.3.1.1.3. scrap report (list all)



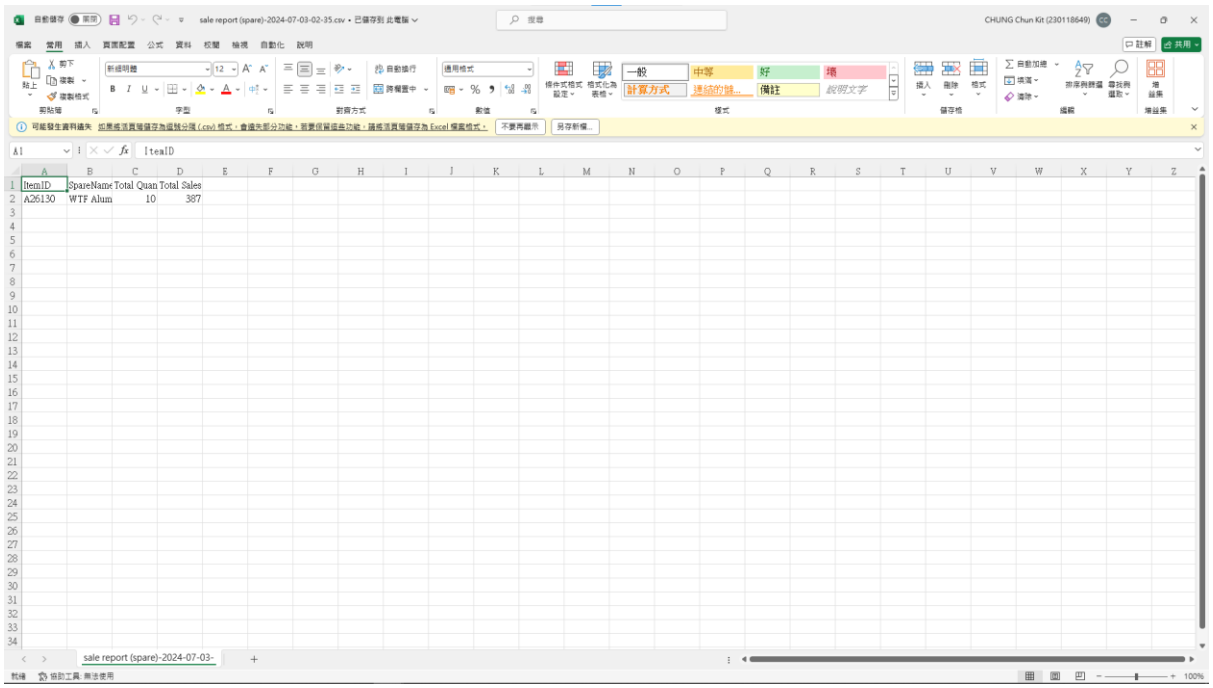
ItemID	SpareName	Warehouse	Warehouse	RecordTim	Quantity	Discrepanc	ScrapAmount
A26130	WTF Alum	1	Warehouse	########	1	Tolerances	42.55
B22978	Hole engin	1	Warehouse	########	2	Tolerances	82.2
C10435	Shift Tail I	1	Warehouse	########	3	Tolerances	189.69

15.3.1.1.4. stock report



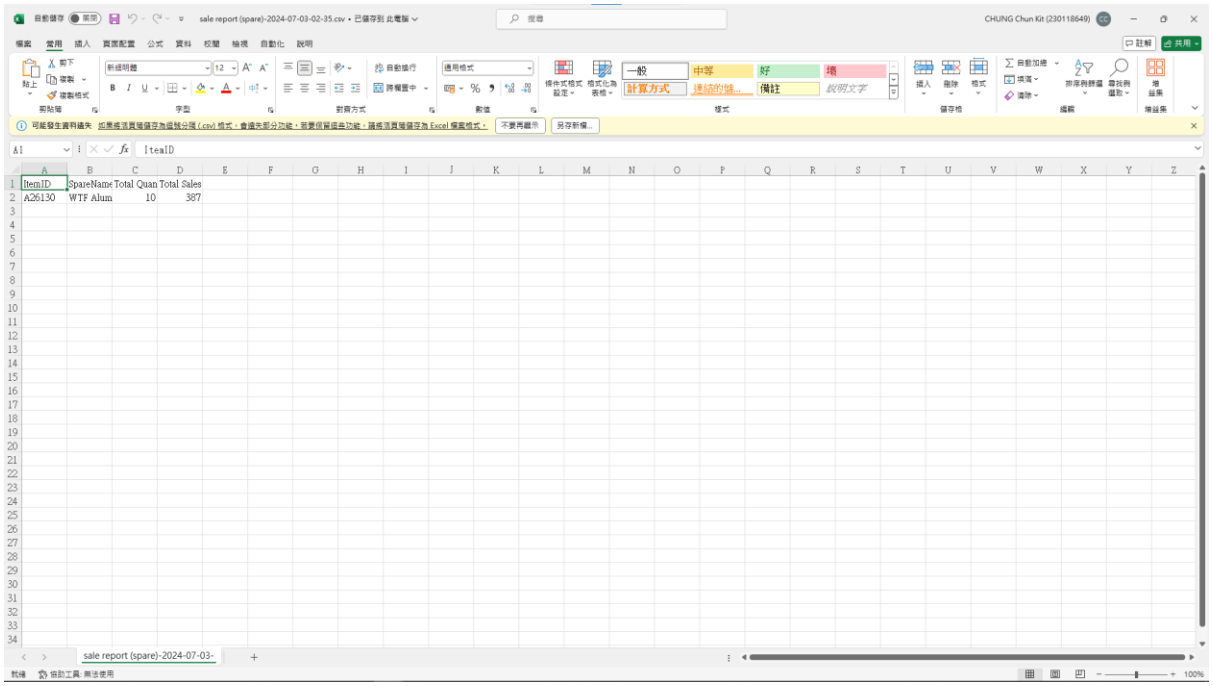
Warehouse	Warehouse	SpareID	SpareName	quantity
1	Warehouse	A11002	OOE sheet	120
1	Warehouse	A26130	WTF Alum	1000
1	Warehouse	B22978	Hole engin	1000
1	Warehouse	C10435	Shiift Tail I	1000
1	Warehouse	C91177	Porject Hea	120

15.3.1.1.5. sale report (spare)



ItemID	SpareName	Total Quan	Total Sales
A26130	WTF Alum	10	387

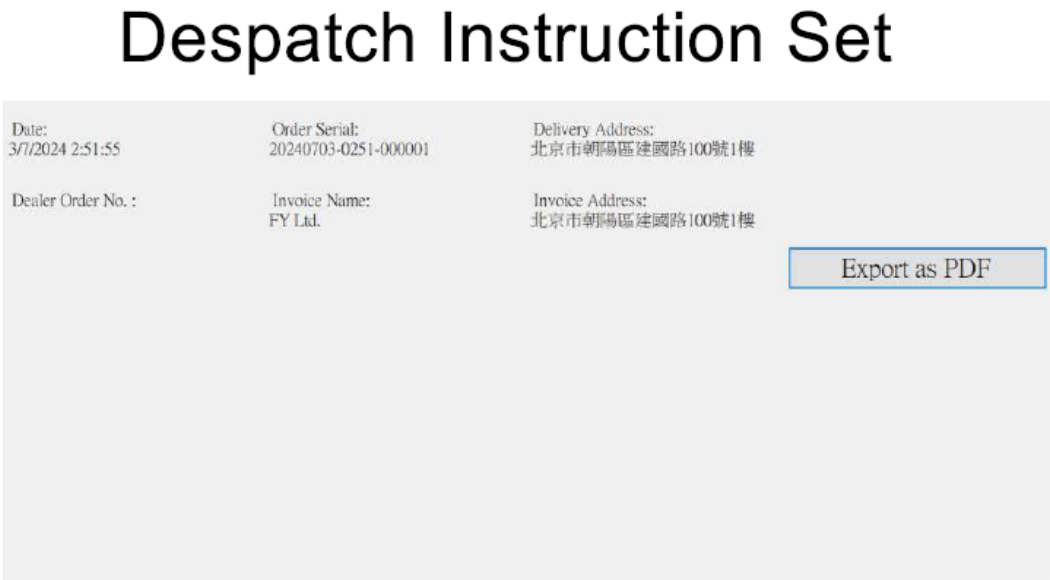
15.3.1.1.6. sale report (warehouse)



ItemID	SpareName	Total Quan	Total Sales
A26130	WTF Alum	10	387

15.3.1.2.Pdf type

15.3.1.2.1. Dlset



Date:	Order Serial:	
3/7/2024 2:51:55	20240703-0251-000001	
Dealer Code:	Part Number:	Order Quantity:
1	A26130	2000
Previous Outstanding:	Number of Bundles:	Actual Quantity
0	1	1000
	To follow:	
	1000	
Weight:	Carrier:	
62.4	storeman1	

Date:	Order Serial:	
3/7/2024 2:51:55	20240703-0251-000001	
Dealer Code:	Part Number:	Order Quantity:
1	B22978	1
Previous Outstanding:	Number of Bundles:	Actual Quantity
0	1	1
	To follow:	
	0	
Weight:	Carrier:	
18.8	storeman1	

Date:	Order Serial:	
3/7/2024 2:51:55	20240703-0251-000001	
Dealer Code:	Part Number:	Order Quantity:
1	C10435	1
Previous Outstanding:	Number of Bundles:	Actual Quantity
0	1	1
	To follow:	
	0	
Weight:	Carrier:	
41.8	storeman1	

This is an example of DIset report

15.3.1.2.2. Invoice

cover page

Legend MOTOR Co. Ltd.
SMLC District, Tienhou, Guangzhou.
Tel: 133 808 12345 Grams: LegendMC

INVOICE ADDRESS
北京市朝陽區建國路100號1樓

DELIVERY ADDRESS (IF DIFFERENT)
北京市朝陽區建國路100號1樓

Date 03/07/2024	DEALER CODE 1
YOUR ORDER NO.	OUR ORDER NO. 20240703-0251-000001
Despatch Foreman storeman1	Total weight 62460.6 g

This invoice create at
2024/07/03 02:55

content page

Part No.	Prev. Qty.	Quantity delivered	Qty to follow	No. of bundles	Unit price ($)
A26130 WTF Aluminium alloy sheet	1000	1000	1000	1	42.55
B22978 Hole engine	1	1	0	1	41.10
C10435 Shiift Tail Light Assembly	1	1	0	1	63.23

end page

Date	Signatory
2/7/2024 21:28:34	sda
Total price ($)	Signature
$56907.00	

This is an example of an Invoice report.

15.3.1.2.3. QR-code PDF

A11002
OOE sheet metal

A26130
WTF Aluminium alloy sheet

15.4.API

We use API for us to provide the map and location searching function to user.

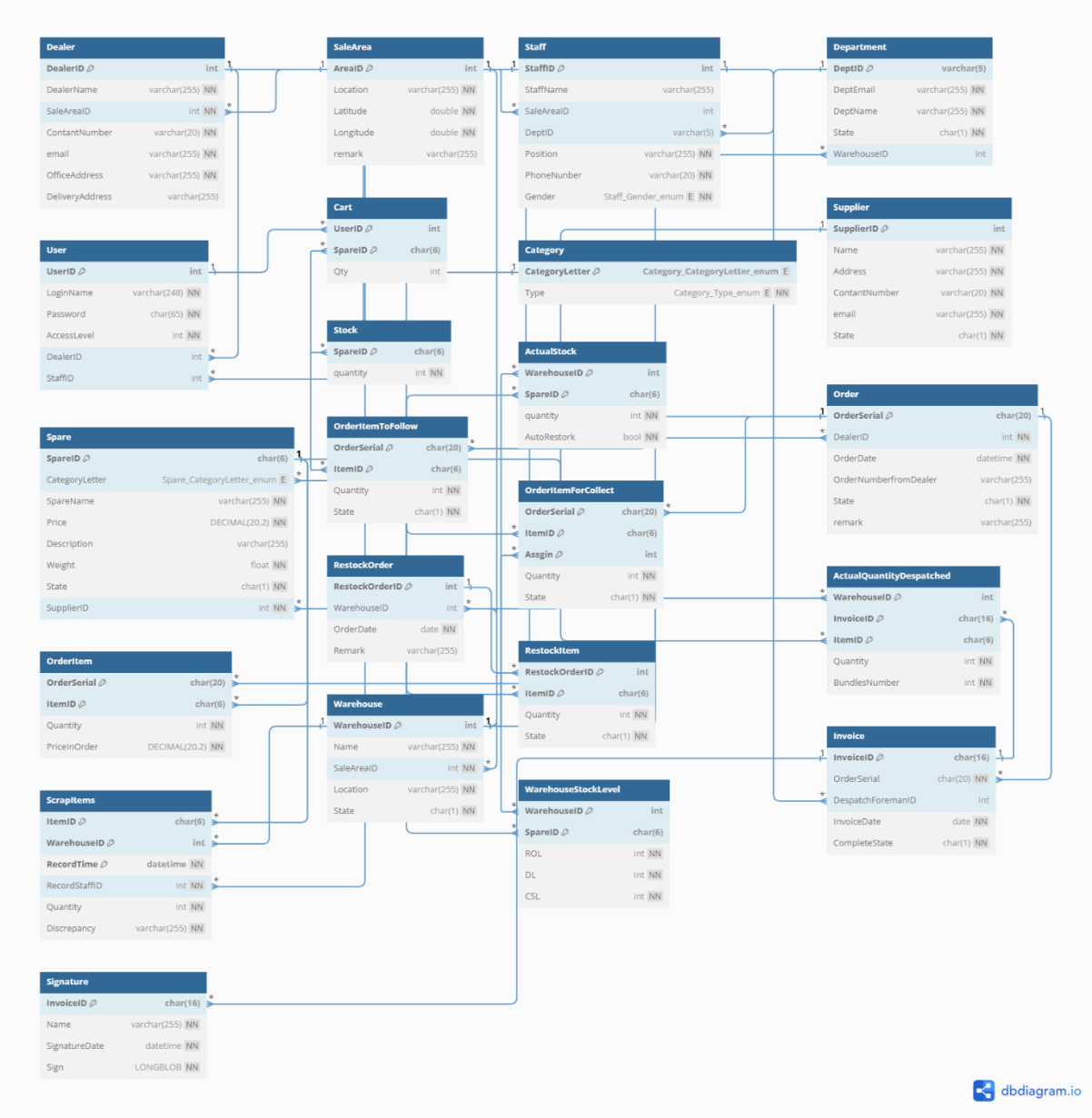
for LM, most of their staff and dealer in China. They may not connect google and use google map, so Baidu map will be suggested.

15.5. Form-actor table

	LM officer	Manager	Accountant	Storemen	Warehouse clerk	Dealer	Delivery worker	Stock record clerk
Login	✓	✓	✓	✓	✓	✓	✓	✓
Main Menu	✓	✓	✓	✓	✓	✓	✓	✓
Add Sale Area	✓	✓						
Edit Sale Area	✓	✓						
Add Department	✓	✓			✓			
Edit Department	✓	✓			✓			
Add Warehouse	✓	✓			✓			
Edit Warehouse	✓	✓			✓			
Add New Spare	✓	✓						
Edit Spare	✓	✓						
Scrap								✓
Set Stock Level								✓
Re-stock (Re-order)								✓
Re-stock (Re-order) Check				✓				
Placing Order						✓		
Cart						✓		
Report			✓					
Supplier management(New Supplier)	✓	✓			✓			
Supplier management(Edit Supplier)	✓	✓			✓			
Edit Order					✓	✓		
Collect Spare				✓				
Order Record and Order Detail	✓	✓	✓	✓	✓	✓	✓	✓
DI set							✓	
Invoice							✓	
User and Account Create		✓						
Account Management		✓				✓		

	LM officer	Manager	Accountant	Storemen	Warehouse clerk	Dealer	Delivery worker	Stock record clerk
New Dealer		✓						
New Staff		✓						

16. Entity Relation Diagram



(refer to dbdiagram.io: https://dbdiagram.io/d/FYP-65de078b5cd0412774f088d3)

17. Detailed Database Design

Table	Column	Data Type	Description
Dealer	DealerID	int	Primary Key AUTO_INCREMENT
	DealerName	varchar	Dealer name, not null
	SaleAreaID	int	Foreign key to SaleArea table, not null
	ContactNumber	varchar(20)	Contact number, unique, not null
	email	varchar	Email, not null
	OfficeAddress	varchar	Office address, not null
	DeliveryAddress	varchar	Delivery address, can be null
SaleArea	AreaID	int	Primary key AUTO_INCREMENT
	Location	varchar	Location, not null
	Latitude	double	NOT NULL
	Longitude	double	NOT NULL
	remark	varchar	Remark, can be null

Table	Column	Data Type	Description
Staff	StaffID	int	Primary key AUTO_INCREMENT
	StaffName	varchar	Staff name
	SaleAreaID	int	Foreign key to SaleArea table, can be null
	DeptID	varchar(5)	Foreign key to Department table
	Position	varchar	Position, not null, dropdown list
	PhoneNunber	varchar(20)	Phone number, unique, not null
	Gender	Gender	Gender only can be M or F, not null
Department	DeptID	varchar(5)	Primary key
	DeptEmail	varchar	Email, not null
	DeptName	varchar	Department name, not null
	State	char(1)	NOT NULL DEFAULT 'N' only N,D

Table	Column	Data Type	Description
	WarehouseID	int	Foreign key to Warehouse table, can be null
User	UserID	int	Primary key AUTO_INCREMENT
	LoginName	varchar	Login name, unique, not null
	Password	char(65)	Password, not null, SHA256 encrypted the first char is the state of the account(such as ! mean block,0 mean ok)the next 64 char is sha256
	AccessLevel	int	Access level, not null, default 10000 0 has the highest permission
	DealerID	char(6)	Foreign key to Dealer table, can be null
	StaffID	int	Foreign key to Staff table

Table	Column	Data Type	Description
Cart	UserID	int	PRIMARY KEY Foreign key to User table
	SpareID	char(6)	PRIMARY KEY Foreign key to Spare table
	Qty	int	not null
Category	CategoryLetter	CategoryLetterData	Primary key, category letter A-D
	Type	CategoryTypeData	Category type, not null only can be Sheet Metal, Major Assemblies, Light Components and Accessories
Supplier	SupplierID	int	Primary key AUTO_INCREMENT
	Name	varchar	Name, not null
	Address	varchar	Address, not null
	ContactNumber	varchar(20)	Contact number, unique, not null

Table	Column	Data Type	Description
	State	char(1)	NOT NULL DEFAULT 'N' only N,D
	email	varchar	Email, not null
Spare	SpareID	char(6)	Primary key, format: CategoryLetter + 5-digit number
	CategoryLetter	char(1)	Foreign key to Category table
	SpareName	varchar	Name, not null
	Price	DECIMAL(20,2)	Price, not null
	Description	varchar	Description, can be null, default "none"
	Weight	float	Weight in grams, not null
	State	char(1)	NOT NULL DEFAULT 'N' only N,D
	SupplierID	int	Foreign key to Supplier table, not null

Table	Column	Data Type	Description
Stock	SpareID	char(6)	Primary key, foreign key to Spare table
	quantity	int	the sum quantity, not null
ActualStock	WarehouseID	int	Primary key, foreign key to Warehouse table
	SpareID	char(6)	Primary key, foreign key to Spare table
	AutoRestork	bool	Auto restock, not null, default true
	quantity	int	Quantity for warehouse in real time, not null
Order	OrderSerial	char(20)	Primary key, format: YYYYMMDD-hhmm-DealerID
	DealerID	char(6)	Foreign key to Dealer table, not null
	OrderDate	datetime	Order date, not null, default now()
	OrderNumberfromDealer	varchar	Order number from dealer, can be null

Table	Column	Data Type	Description
	State	char(1)	State char, not null use a char to record the state C = Confirm Order P = Processing W = Waiting U = Unavailable T = In transit F = this order is finished when order is placing: the state is C when order is collecting(after the list for collect is generate): the state is P when some item is out-stock in a order and waiting to re-stock: the state is W when some item is out-stock in a order and it will not re-stock: the state is U

Table	Column	Data Type	Description
			when any item in transit: the state is T when all item is ship to dealer: the state is F
	remark	varchar	Remark, can be null
OrderItem	OrderSerial	char(20)	Primary key, foreign key to Order table
	ItemID	char(5)	Primary key, foreign key to Spare table
	Quantity	int	Quantity, not null, default 1
	PriceInOrder	DECIMAL(20,2)	The price of this spare at ordering, not null to prevent price change errors
OrderItemToFollow	OrderSerial	char(20)	Primary key, foreign key to Order table
	ItemID	char(5)	Primary key, foreign key to Spare table
	Quantity	int	Quantity, not null, default 1

Table	Column	Data Type	Description
	State	char(1)	State char, not null e.g. D, W etc use a char to record the state D = Done W = Wait U = Unavailable
OrderItemToFollow	OrderSerial	char(20)	Primary key, foreign key to Order table
	ItemID	char(5)	Primary key, foreign key to Spare table
	Assign	int	foreign key to warehouse store the warehouseID which warehouse is assigned
	Quantity	int	Quantity, not null, default 1
	State	char(1)	State char, not null use a char to record the state C = Collecting W = Wait

Table	Column	Data Type	Description
ActualQuantityDespatched	WarehouseID	int	Primary key, foreign key to Warehouse table
	InvoiceID	char(15)	Primary key, foreign key to Invoice table
	ItemID	char(5)	Primary key, foreign key to Spare table
	Quantity	int	Quantity, not null, default 1
	BundlesNumber	int	Bundles number, not null, default 1
ScrapItems	ItemID	char(5)	Primary key, foreign key to Spare table
	WarehouseID	int	foreign key to Warehouse table
	RecordTime	datetime	Primary key, record time, not null
	RecordStaffID	int	Foreign key to Staff table, not null
	Quantity	int	Quantity, not null, default 1
	Discrepancy	varchar	Discrepancy, not null

Table	Column	Data Type	Description
RestockOrder	RestockOrderID	int	Primary key AUTO_INCREMENT
	WarehouseID	int	Primary key, foreign key to Warehouse table
	OrderDate	date	Order date, not null
	Remark	varchar	
RestockItem	RestockOrderID	int	Primary key, foreign key to RestockOrder table
	ItemID	char(5)	Primary key, foreign key to Spare table
	State	char(1)	State char, not null C = Create Order W = Waiting U = Unavailable F = this order is finished
	Quantity	int	Quantity, not null, default 1
Invoice	InvoiceID	char(15)	Primary key, format: YYYYMMDD+DealerID+SerialNo.

Table	Column	Data Type	Description
	OrderSerial	char(20)	Primary key, foreign key to Order table
	DespatchForemanID	int	Foreign key to Staff table, can be null
	InvoiceDate	date	Invoice date, not null
	CompleteState	char(1)	State char, not null, default "C" C = Create S = Signed
Warehouse	WarehouseID	int	Primary key AUTO_INCREMENT
	ManagerID	int	Foreign key to Staff table, not null
	SaleAreaID	int	Foreign key to SaleArea table, not null
	State	char(1)	NOT NULL DEFAULT 'N' only N,D
	Location	varchar	not null

Table	Column	Data Type	Description
WarehouseStock Level	WarehouseID	int	Primary key, foreign key to Warehouse table
	SpareID	char(6)	Primary key, foreign key to Spare table
	ROL	int	Reorder level, not null, default 0
	DL	int	danger level, not null, default 0
	CSL	int	Critical safety level, not null, default ROL * 1.2

18. Test Data

18.1.build database

There SQL is for build database and data table

build.sql
refer to build.sql in appendix

18.2.data insert

This is some of the data that will be used in the test case

data.sql
use OSS; /*SaleArea*/ INSERT INTO `SaleArea` (`AreaID`, `Location`, `Latitude`, `Longitude`, `remark`) VALUES (1, 'Beijing', 39.9042, 116.4074, NULL), (2, 'Shanghai', 31.2304, 121.4737, NULL), (3, 'Guangzhou', 23.1291, 113.2644, NULL), (4, 'Shenzhen', 22.5431, 114.0579, NULL), (5, 'Tianjin', 39.3434, 117.3616, NULL), (6, 'Hong Kong', 22.3193, 114.1694, NULL), (7, 'Macau', 22.1987, 113.5439, NULL), (8, 'Taipei', 25.0330, 121.5654, NULL), (9, 'Nanjing', 32.0603, 118.7969, NULL), (10, 'Qingdao', 36.0671, 120.3826, NULL), (11, 'Hangzhou', 30.2741, 120.1551, NULL), (12, 'Wuhan', 30.5928, 114.3055, NULL), (13, 'Chengdu', 30.5728, 104.0668, NULL),

```
(14, 'Chongqing', 29.5630, 106.5516, NULL),
(15, 'Xiamen', 24.4798, 118.0895, NULL),
(16, 'Xian', 34.3416, 108.9398, NULL),
(17, 'Zhengzhou', 34.7466, 113.6254, NULL),
(18, 'Kunming', 25.0438, 102.7100, NULL),
(19, 'Dalian', 38.9140, 121.6147, NULL),
(20, 'Changsha', 28.2282, 112.9388, NULL),
(21, 'Jinan', 36.6512, 117.1201, NULL),
(22, 'Lanzhou', 36.0611, 103.8343, NULL),
(23, 'Zhuhai', 22.2707, 113.5767, NULL),
(24, 'Nanning', 22.8170, 108.3669, NULL),
(25, 'Harbin', 45.8038, 126.5349, NULL),
(26, 'Haikou', 20.0440, 110.1999, NULL),
(27, 'Shijiazhuang', 38.0428, 114.5149, NULL),
(28, 'Hefei', 31.8206, 117.2290, NULL),
(29, 'Changchun', 43.8171, 125.3235, NULL),
(30, 'Guiyang', 26.6470, 106.6302, NULL);

/*Warehouse*/
INSERT INTO `Warehouse` (`WarehouseID`, `Name`, `SaleAreaID`, `Location`, `State`)
VALUES
  (1, 'Warehouse 1', 18, '昆明市滇池路 1310 號雲南民族村東門', 'N'),
  (2, 'Warehouse 2', 19, 'China, CN 辽宁省 大连市 甘井子区', 'N'),
  (3, 'Warehouse 3', 20, '55WW+CX3, Changsha County, Changsha, Hunan, China, 410141', 'N'),
  (4, 'Warehouse 4', 21, 'P4JJ+P9J, Kaiyuan Rd, Licheng District, Jinan, Jinan, Shandong, China, 250101', 'N');
```

```
/*Department*/
INSERT INTO `Department` (`DeptID`, `DeptEmail`, `DeptName`, `WarehouseID`)
VALUES
('manag', 'manager@legendmotor.com', 'Manager', NULL),
('sale', 'sales@legendmotor.com', 'Sales', NULL),
('makt', 'marketing@legendmotor.com', 'Marketing', NULL),
('hr', 'hr@legendmotor.com', 'Human Resources', NULL),
('fina', 'finance@legendmotor.com', 'Finance', NULL),
('it', 'it@legendmotor.com', 'IT', NULL),
('proc', 'procurement@legendmotor.com', 'Procurement', NULL),
('prod', 'production@legendmotor.com', 'Production', NULL),
('qa', 'quality@legendmotor.com', 'Quality Assurance', NULL),
('logi', 'logistics@legendmotor.com', 'Logistics', NULL),
('cs', 'service@legendmotor.com', 'Customer Service', NULL),
('wh001', 'warehouse001@legendmotor.com', 'Warehouse1', 1),
('wh002', 'warehouse002@legendmotor.com', 'Warehouse2', 2),
('wh003', 'warehouse003@legendmotor.com', 'Warehouse3', 3),
('wh004', 'warehouse004@legendmotor.com', 'Warehouse4', 4),
('Hdt', 'Headquarters@legendmotor.com', 'Headquarters', NULL);


/*Supplier*/
INSERT INTO `Supplier` (`SupplierID`, `Name`, `Address`, `ContantNumber`, `email`)
VALUES
```

```
(100, 'SB company', '北京市朝阳区建国门外大街 19 号', '13987654301', 'sb@163.com'),
(101, 'GG company', '上海市浦东新区世纪大道 8 号', '15876543215', 'gg@qq.com'),
(102, 'OK company', '广东省深圳市罗湖区深南东路 9 号', '13654321098', 'ok@sina.com'),
(103, 'Movement Ltd.', '江苏省南京市鼓楼区汉口路 6 号', '15098765432', 'movement@sohu.com'),
(104, 'FK company', '浙江省杭州市西湖区文二西路 4 号', '13319896445', 'fk@aliyun.com'),
(105, 'OMG company', '四川省成都市锦江区人民中路 20 号', '13765432109', 'omg@foxmail.com'),
(106, 'Killer company', '湖南省长沙市岳麓区麓山南路 19 号', '15987654321', 'killer@qq.com'),
(107, 'nsml Ltd.', '河北省石家庄市裕华区裕华路 7 号', '13876543210', 'nsml@foxmail.com'),
(108, 'The Great company', '山东省青岛市市南区香港中路 2 号', '13543210987', 'thegreat@qq.com'),
(109, 'Translation company', '辽宁省大连市中山区人民广场 1 号', '15120197213', 'translation@qq.com');

/*Category*/
INSERT INTO `Category` (`CategoryLetter`, `Type`)
VALUES
('A', 'Sheet Metal'),
('B', 'Major Assemblies'),
('C', 'Light Components'),
('D', 'Accessories');
```

```
/*Delear*/

INSERT INTO `Dealer` (`DealerID`, `DealerName`, `SaleAreaID`, `ContantNumber`, `email`, `OfficeAddress`, `DeliveryAddress`)

VALUES

(1, 'FY Ltd.', 1, '13912345678', 'fy@qq.com', '北京市朝陽區建國路 100 號 1 樓', '北京市朝陽區建國路 100 號 1 樓'),

(2, 'Happy Ltd.', 1, '15898765432', 'happy@abc.com', '北京市海淀區西三旗街道 1 號', '北京市海淀區西三旗街道 1 號'),

(3, 'IT Ltd.', 1, '13678901234', 'it@qq.com', '北京市昌平區回龍觀東大街 2 號', '北京市昌平區回龍觀東大街 2 號'),

(4, 'SDP Ltd.', 1, '15056789012', 'sdp@qq.com', '北京市西城區西單北大街 3 號', '北京市西城區西單北大街 3 號'),

(5, 'OOT Ltd.', 2, '13123456789', 'oot@qq.com', '上海市浦東新區東方明珠廣場', '香港中環皇后大道中 1 號'),

(6, 'IMAD Ltd.', 4, '13123456759', 'imad@qq.com', '深圳市福田區華強北路 5 號', '廣州市天河區珠江新城華強路 6 號'),

(7, 'ITE Ltd.', 6, '23020831', 'ite@qq.com', '香港中環皇后大道中 1 號', '廣州市天河區珠江新城華強路 6 號'),

(8, 'DP Ltd.', 8, '90747203', 'dp@qq.com', '台北市信義區忠孝東路 5 段 7 號', '南京市鼓樓區中山北路 8 號'),

(9, 'SSD Ltd.', 15, '69400737', 'ssd@qq.com', '廈門市思明區鼓浪屿路 9 號', '香港中環皇后大道中 1 號'),

(10, 'OS Ltd.', 30, '31376030', 'os@qq.com', '貴陽市南明區廣場北路 10 號', '南京市鼓樓區中山北路 8 號');


/*Staff*/

INSERT INTO `Staff` (`StaffID`, `StaffName`, `SaleAreaID`, `DeptID`, `Position`, `PhoneNunber`, `Gender`)

VALUES
```

```
(1,     'Andy', 1, 'wh001', 'storemen',  '13912345678', 'M'),
(2,     '李小明',2, 'wh002', 'storemen', '15898765432', 'M'),
(3,     '张三',3, 'wh003',  'storemen',  '13678901234',     'M'),
(4,     '苹',4, 'wh004', 'storemen', '15056789012',  'M'),
(5,     'ChanTomori', NULL, 'manag', 'manager', '13123456789', 'F'),
(6,     'WongTaki', 1, 'wh001', 'manager', '15816444441', 'F'),
(7,     'CheungSoyo',  NULL, 'Hdt', 'LM officer', '19076550', 'F'),
(8,     '陳大文', NULL, 'HdT', 'Accountant', '58436029', 'F'),
(9,     'ORana',  1, 'wh001', 'Warehouseclerk', '49164576', 'F'),
(10, 'AAA',  NULL, 'Hdt',   'Warehouseclerk', '50552739', 'F'),
(11, 'CCC',  1, 'wh001',     'manager',   '98931962',  'F'),
(12, 'DDD',  2, 'wh002', 'manager',       '23490946',  'F'),
(13, 'Sam',  3, 'wh003', 'manager',       '89774445',  'M'),
(14, 'Apple',  4, 'wh004', 'manager', '14564581', 'M'),
(15, 'ORana',  3, 'wh003', 'Warehouseclerk',  '16043681',  'F'),
(16, '孫培玲',  4, 'wh004',          'Warehouseclerk', '67468369',  'F'),
(17, '望醉冬',  2, 'wh002',          'delivery worker', '25624059','F'),
(18, '皇甫樂成',  3,'wh003', 'delivery worker', '82838532', 'F'),
(19, '山靜白', 4, 'wh004', 'delivery worker', '23023829', 'F'),
(20, '山小明',  1, 'wh001', 'Stock record clerk', '48193089', 'M'),
(100, 'storeman1', 1, 'wh001', 'storeman', '98765432', 'M'),
(107, 'storeman2', 2, 'wh002', 'storeman', '98765437', 'M'),
(108, 'storeman3', 3, 'wh003', 'storeman', '98765438', 'M'),
(109, 'storeman4', 4, 'wh004', 'storeman', '98765439', 'M'),
(110, 'delivery_worke1', 1, 'wh001', 'delivery_worker', '98765428', 'M'),
(111, 'delivery_worker2', 2, 'wh002', 'delivery_worker', '98765438', 'M'),
(112, 'delivery_worker3', 3, 'wh003', 'delivery_worker', '98765448', 'M'),
```

```
(113, 'delivery_worker4', 4, 'wh004', 'delivery_worker', '98765458', 'M'),
(101, 'LMcleck', NULL, 'Hdt', 'LMcleck', '98765431', 'M'),
(102, 'accountant', NULL, 'Hdt', 'accountant', '98765430', 'M'),
(103, 'manager', 1, 'wh001', 'manager', '98765429', 'M'),
(104, 'delivery_worker', 1, 'wh001', 'delivery_worker', '98765428', 'M'),
(105, 'Warehousecleak', 1, 'wh001', 'Warehousecleak', '98765427', 'M'),
(106, 'stockrecord', 1, 'wh001', 'stockrecord', '98765426', 'M');

/*User*/
INSERT INTO `User` (`UserID`, `LoginName`, `Password`, `AccessLevel`,
`DealerID`, `StaffID`)
VALUES
(9, 'ATL',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
0, 1, 1),
(10, 'kirito',
'!a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000, NULL, 2),
(11, 'cusidina',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000, NULL, 3),
(12, 'kemono114',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000, NULL, 4),
(13, 'idkABC',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000, NULL, 5),
(14, 'asuna',
'!a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000, NULL, 6),
```

```
(15, 'HAha123',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000, NULL, 7),

(16, 'happyCat',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000, NULL, 8),

(17, 'waganaai',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000, NULL, 9),

(18, 'ho',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000, NULL, 10),

(19, 'zwYtJrViRPGcruemxwMd',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000,NULL,11),

(20, 'cKCVYAvrPuwhRufZqnnH',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000,NULL,12),

(21, 'ukdeZCKyEGPzanTMuiXC',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000,NULL,13),

(22, 'DmozffipkaEvXUtxZfVV',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000,NULL,14),

(23, 'zRmTpvAqEZhWaAZvWuQq',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000,NULL,15),

(24, 'PHkVKmwcfiTfmGhBaVgg',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000,NULL,16),

(25, 'iKXQbJKpydqVkDDFohbC',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000,NULL,17),
```

```
(26, 'ianWsRYLjsJhgTPdiwgy',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000,NULL,18),

(27, 'HJfXjJbUrEwcnMvnXjos',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000,NULL,19),

(28, 'mHMVNDcyDQVtkXDqRDma',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
10000,NULL,20),

(1, 'storeman1',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
'400', NULL, 100),

(30, 'storeman2',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
'400', NULL, 107),

(31, 'storeman3',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
'400', NULL, 108),

(32, 'storeman4',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
'400', NULL, 109),

(2, 'LMcleck',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
'200', NULL, 101),

(3, 'accountant',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
'300', NULL, 102),

(4, 'manager',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
'100', NULL, 103),

(5, 'delivery_worker',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
'700', NULL, 104),
```

```
(6, 'Warehousecleak',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
'600', NULL, 105),

(7, 'stockrecord',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
'500', NULL, 106),

(8, 'Dealer1',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
'5000', 1, NULL),

(29, 'Dealer2',
'0a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3',
'5000', 2, NULL);


/*Spare*/

INSERT INTO Spare (SpareID, CategoryLetter, SpareName, Price, Description,
Weight, SupplierID)

VALUES

   ('A26130', 'A', 'WTF Aluminium alloy sheet', 42.55, '', 62.4, 108),

   ('B22978', 'B', 'Hole engine', 41.1, '', 18.8, 100),

   ('C10435', 'C', 'Shiift Tail Light Assembly', 63.23, '', 41.8, 102),

   ('D75438', 'D', 'Tis Vehicle Car Recorder', 35.28, '', 22.8, 103),

   ('C91177', 'C', 'Porject Headlight Assembly', 22.76, '', 34.5, 100),

   ('D52143', 'D', 'IS In-vehicle mobile phone holder', 39.08, '', 61.1, 105),

   ('A48880', 'A', 'fire Stainless Steel Sheet', 49.09, '', 61, 107),

   ('D31172', 'D', 'DDD Vehicle Air Cleaner', 16.03, '', 17.9, 106),

   ('D62438', 'D', 'GGG Car USB Charger', 53.69, '', 40, 107),

   ('B25912', 'B', 'www cooling system', 39.69, '', 28.9, 104),

   ('A11002', 'A', 'OOE sheet metal', 67.02, '', 53, 107),

   ('B11003', 'B', 'ASD ddmetal', 12.51, '', 47.6, 109),

   ('C11004', 'C', 'HHH SS engine', 18.44, '', 30.9, 101),
```

```
    ('D11005', 'D', 'WEE Cleaner', 59.51, '', 47.7, 107),
    ('A11006', 'A', 'UUV erMetal', 53.76, '', 17.8, 105),
    ('B11007', 'B', 'BB ssEngine', 45.64, '', 13.9, 101),
    ('C11008', 'C', 'CEE Assembly', 64.94, '', 25, 108),
    ('D11009', 'D', 'RRE Charger', 15.42, '', 21.1, 102),
    ('A11010', 'A', 'AV Metal', 45.98, '', 24.5, 106),
    ('B11011', 'B', 'Bill Engine', 63.69, '', 14.9, 109),
    ('C11012', 'C', 'CCC Assembly', 24.49, '', 60.1, 109),
    ('D11013', 'D', 'DOV mobile phone holder', 43.68, '', 30.4, 103),
    ('A11014', 'A', 'MYGO alloy sheet', 51.13, '', 40, 100),
    ('B11015', 'B', 'BS cooling system', 34.73, '', 69.2, 105),
    ('C11016', 'C', 'CAD Headlight Assembly', 57.83, '', 28.8, 106),
    ('D11017', 'D', 'DEE Vehicle Air Cleaner', 40.26, '', 27.2, 104);

/*Stock*/
INSERT INTO Stock (SpareID, quantity)
VALUES
('A26130', 2000),
('B22978', 1000),
('C10435', 1000),
('D75438', 1000),
('C91177', 1000),
('D52143', 1000),
('A48880', 1000),
('D31172', 90),
('D62438', 90),
('B25912', 90);
```

```
INSERT INTO ActualStock (WarehouseID, SpareID, AutoRestork, quantity)
VALUES
(1, 'A26130', TRUE, 1000),
(1, 'B22978', TRUE, 1000),
(1, 'C10435', FALSE, 1000),
(2, 'A26130', FALSE, 1000),
(2, 'D75438', FALSE, 1000),
(2, 'C91177', TRUE, 1000),
(2, 'D52143', TRUE, 1000),
(3, 'A48880', TRUE, 1000),
(3, 'D31172', TRUE, 100),
(4, 'D62438', TRUE, 100),
(4, 'B25912', TRUE, 100);



/*Order*/
INSERT INTO `Order` (`OrderSerial`, `DealerID`, `OrderDate`,
`OrderNumberfromDealer`, `State`, `remark`)
VALUES
  ('20240524-0900-000001', 1, '2024-05-24 09:00:00', '', 'T', null),
  ('20240524-0901-000002', 2, '2024-05-24 09:01:00', '', 'T', null),
  ('20240524-0902-000003', 3, '2024-05-24 09:02:00', '', 'P', null),
  ('20240524-0903-000004', 4, '2024-05-24 09:03:00', '', 'P', null),
  ('20240524-0904-000005', 5, '2024-05-24 09:04:00', '', 'P', null),
  ('20240524-0905-000006', 6, '2024-05-24 09:05:00', '', 'P', null),
  ('20240524-0906-000007', 7, '2024-05-24 09:06:00', '', 'P', null),
  ('20240524-0907-000008', 8, '2024-05-24 09:07:00', '', 'W', null),
  ('20240524-0908-000009', 9, '2024-05-24 09:08:00', '', 'W', null),
```

```
  ('20240524-0909-000010', 10, '2024-05-24 09:09:00', '', 'W', null);

/*OrderItem*/
INSERT INTO `OrderItem` (`OrderSerial`, `ItemID`, `Quantity`, `PriceInOrder`)
VALUES
  ('20240524-0900-000001', 'A26130', 10, 38.7),
  ('20240524-0901-000002', 'B22978', 10, 58.54),
  ('20240524-0902-000003', 'C10435', 10, 36.26),
  ('20240524-0902-000003', 'D75438', 20, 70.64),
  ('20240524-0903-000004', 'D75438', 10, 51.07),
  ('20240524-0904-000005', 'C91177', 10, 27.13),
  ('20240524-0905-000006', 'D52143', 10, 30.42),
  ('20240524-0906-000007', 'A48880', 10, 54.21),
  ('20240524-0907-000008', 'D31172', 100, 13.23),
  ('20240524-0908-000009', 'D62438', 100, 57.28),
  ('20240524-0909-000010', 'B25912', 100, 24.46);

/*OrderItemToFollow*/
INSERT INTO `OrderItemToFollow` (`OrderSerial`, `ItemID`, `Quantity`,
`State`)
VALUES
  ('20240524-0907-000008', 'D31172', 10, 'W'),
  ('20240524-0908-000009', 'D62438', 10, 'W'),
  ('20240524-0909-000010', 'B25912', 10, 'W');

/*Invoice*/
INSERT INTO `Invoice` (`InvoiceID`, `OrderSerial`, `DespatchForemanID`,
`InvoiceDate`, `CompleteState`)
```

```
VALUES
  ('2024052400000101', '20240524-0900-000001', 6, '2024-05-24', 'C'),
  ('2024052400000201', '20240524-0901-000002', 6, '2024-05-24', 'C');
/*ActualQuantityDespatched*/

INSERT INTO `ActualQuantityDespatched` (`WarehouseID`, `InvoiceID`, `ItemID`, `Quantity`, `BundlesNumber`)
VALUES
  (1, '2024052400000101', 'A26130', 10, 1),
  (1, '2024052400000201', 'B22978', 10, 2);

/*ScrapItems*/
INSERT INTO `ScrapItems` (`ItemID`, `WarehouseID`,`RecordTime`, `RecordStaffID`, `Quantity`, `Discrepancy`)
VALUES
  ('A26130', 1,'2024-05-24 21:05', 20, 1, 'Tolerances for sheet metal can range from a few millimetres to tens of millimetres.'),
  ('B22978', 1,'2024-05-24 21:05', 20, 2, 'Tolerances of components may be a few millimetres to tens of millimetres'),
  ('C10435', 1,'2024-05-24 21:05', 20, 3, 'Tolerances for lighting fittings may be a few millimetres to a fraction of a millimetre');

/*RestockOrder*/
INSERT INTO `RestockOrder` (`RestockOrderID`, `WarehouseID`, `OrderDate`, `Remark`)
VALUES
  (123123, 3, '2024-05-24', null),
  (123789, 4, '2024-05-24', null),
  (123456, 4, '2024-05-24', null);
```

```
/*RestockItem*/
INSERT INTO `RestockItem` (`RestockOrderID`, `ItemID`, `Quantity`, `State`)
VALUES
  (123123, 'D31172', 1000, 'C'),
  (123789, 'D62438', 1000, 'C'),
  (123456, 'B25912', 1000, 'C');

/*WarehouseStockLevel*/
INSERT INTO `WarehouseStockLevel` (`WarehouseID`, `SpareID`,`DL` ,`ROL`, `CSL`)
VALUES
  (1, 'A26130', 50, 100, 120),
  (1, 'B22978', 50, 100, 120),
  (1, 'C10435', 50, 100, 120),
  (2, 'D75438', 50, 100, 120),
  (2, 'C91177', 50, 100, 120),
  (2, 'D52143', 50, 100, 120),
  (3, 'A48880', 50, 100, 120),
  (3, 'D31172', 50, 100, 120),
  (4, 'D62438', 50, 100, 120),
  (4, 'B25912', 50, 100, 120);
```

18.3.drop table

Here is how to reset the data table.

Users need to run the following code and then run build.sql.

Drop_all.sql
refer to drop_all.sql in appendix

19. Test Cases

19.1.Login

19.1.1.Test Procedure 1

User trying to login by “idkABC” with the password “123”

19.1.1.1.Expected Result for Test Procedure 1

Login successful.

Username “idkABC” is exist, and this account do not get block (password start with “0”) and the sha256 of password is match

19.1.2.Test Procedure 2

User trying to login by “asuna” with the password “123”

19.1.2.1.Expected Result for Test Procedure 2

Login fail.

Username “asuna” is exist, but this account get block(password start with “!”)and the sha256 of password is match

19.2.Add new Spare

Login as “manager” with password “123”

19.2.1.Test Procedure 1

Enter new spare.

With Spare Name is “RG118”, price is 50.00, Weight is 5.

19.2.1.1.Expected Result for Test Procedure 1

Add successful, all fields except description have data entered and the data is within the limits.

19.2.2.Test Procedure 2

Enter new spare.

Without Spare Name and Weight.

19.2.2.1.Expected Result for Test Procedure 2

Add fail

Adding a new spare part requires all the necessary information except the description

19.3.Edit Spare

Login as “manager” with password “123”

19.3.1.Test Procedure 1

Select Spare ID “A11002”, change the weight to 60 and click save.

19.3.1.1.Expected Result for Test Procedure 1

Edit successfully.

All fields except description have data entered and the data is within the limits.

19.3.2.Test Procedure 2

Select Spare ID “A11002”, delete the name and click save.

19.3.2.1.Expected Result for Test Procedure 2

Edit fail.

Editing a new spare part requires all the necessary information except the description.

19.4.Generate Spare QE-code

Login as “manager” with password “123”

19.4.1.Test Procedure 1

Select Spare Name “DOV mobile phone holder”.

19.4.1.1.Expected Result for Test Procedure 1

Successful, “DOV mobile phone holder” has a corresponding SpareID.

19.4.2.Test Procedure 2

Enter Spare Name “DOV mobile phone holder123”.

19.4.2.1.Expected Result for Test Procedure 2

Fail, “DOV mobile phone holder123” has not corresponding SpareID.

19.4.3.Test Procedure 3

Enter Spare ID “D11451”.

19.4.3.1.Expected Result for Test Procedure 3

Fail, “D11451” has not corresponding Spare Name.

19.5.Scrap

Login with Login Name “stockrecord” and password 123.

19.5.1.Test Procedure 1

Select Spare ID “A26130”.Enter Quantity 10 and Discrepancy “I want to”

19.5.1.1.Expected Result for Test Procedure 1

Scrap Successful, all required information has been entered.

19.5.2.Test Procedure 2

Select Spare ID “A26130”.Does enter Quantity and Discrepancy.

19.5.2.1.Expected Result for Test Procedure 2

Fail, scrap spare must enter quantity and discrepancy.

19.6.Place Order

Login” Dealer1” with password 123

19.6.1.Test Procedure 1

Search with Spare ID ‘122121445‘, Spare Name ’122121445‘, Category ’ 122121445 ‘.

19.6.1.1.Expected Result for Test Procedure1

No data found. Data does not exist.

19.6.2.Test Procedure 2

Search with Spare Name “C”.

19.6.2.1.Expected Result for Test Procedure2

Successful, the system allows users to search Spare Name by keywords.

19.6.3.Test Procedure 3

Search with Spare ID “C”

19.6.3.1.Expected Result for Test Procedure3

Failed, the system does not allow users to search for alternate IDs by keyword.

19.6.4.Test Procedure 4

Search with Spare ID “D52143” and click ‘add the cart‘ two time.

19.6.4.1.Expected Result for Test Procedure 4

Failed, the system does not allow the user to add the same spare part to the cart twice in the same order.

19.6.5.Test Procedure 5

Click “View chart” and click confirm to place an order. After you search with Spare ID “D52143” and click ‘add the cart‘.

19.6.5.1.Expected Result for Test Procedure 5

Successful, system will show the succeed message and the order number.

19.7.Order Edit

Login” Dealer1” with password 123

19.7.1.Test Procedure 1

Choose order serial “20240524-0990-000001” and click Edit your order number.

19.7.1.1.Expected Result for Test Procedure 1

Successful, order number is saved.

19.7.2.Test Procedure 2

Choose order serial “20240524-0900-000001” and click Cancel.

19.7.2.1.Expected Result for Test Procedure 2

Fail, this order cannot be cancel.

19.7.3.Test Procedure 3

When placing an order select your most recent order sequence (you can try to place the order immediately if you need to) and click "Cancel Order".

19.7.3.1. Expected Result for Test 3

Successful, order will be delete.

19.7.4. Test Procedure 4

Choose order serial “20240524-0990-000001” and “WTF Aluminium alloy” and click “Edit order item”

19.7.4.1.Expected Result for Test 4

Failed because the status of order "20240524-0990-000001" is not C or W. Editing is only possible if the status of the order is C or W.

19.8.Order Record

Login as “manager” with password “123”

19.8.1.Test Procedure 1

Double clicks order serial “20240524-0900-000001” click “DI set” and click “Export as PDF”

19.8.1.1.Expected Result for Test 1

Successful.

19.8.2.Test Procedure 2

Search “0000101”

19.8.2.1.Expected Result for Test 2

No result, no order serial like “0000101”.

19.9.Collect Spare

Login” storeman1” with password 123

19.9.1.Test Procedure 1

Enter Collect Spare when there is no order to be collected

19.9.1.1. Expected Result for Test 1

Show no order to collect

19.9.2. Test Procedure 2

Login as “Dealer1”

Place order with Spare “C10435”.

Login as “ATL” go to “Simulate Server” and click “Simulate server on time”.

Login as “storeman” and go to “Collect Spare”.

19.9.2.1. Expected Result for Test 2

“C10435” need to be collect

19.9.3. Test Procedure 3

After Test Procedure 2, click the row of item or click “Turn on Scanner”(use Scanner the QR code of Spare “C10435”) to select a item

19.9.3.1. Expected Result for Test 3

“C10435” selected

19.9.4. Test Procedure 4

After Test Procedure 3, input “Number of Bundles” as 10 and click “confirm”

19.9.4.1. Expected Result for Test 4

item is collected

19.10. Set Stock Level

Login” stockrecord” with password 123

19.10.1. Test Procedure 1

Select A11002 and set stock level, Common Stock Level = 500, Re stock level = 10, Danger Level = 20.

19.10.1.1. Expected Result for Test 1

Fail, set Stock level Need to follow DL < ROL < CSL rule.

19.10.2. Test Procedure 2

Select A11002 and set stock level, Common Stock Level = 120, Re stock level = 100, Danger Level = 50 and click Auto Re-stock.

19.10.2.1. Expected Result for Test Procedure 2

Successful, system will set the stock level.

19.10.3. Test Procedure 3

Select A26130 and set stock level, Common Stock Level = 130.

19.10.3.1. Expected Result for Test Procedure 3

Successful, system will set the stock level and but will not create a restock order.

19.11. Re-Stock

After you done 18.2.10.3.

Login" stockrecord" with password 123

19.11.1. Test Procedure 1

Select row with Spare ID A11002 and Click on the down arrow. Then click the "Restocking to CSL button".

19.11.1.1. Expected Result for Test Procedure 1

Successful, a restock order is placed.

19.12. Re-stock Record

After you done 18.2.11.1

Login" stockrecord" with password 123

19.12.1. Test Procedure 1

Select row with Spare ID A11002 and Click on the down arrow. Then click the "Received".

19.12.1.1. Expected Result for Test Procedure 1

Successful, the restock order is received.

19.12.2. Test Procedure 2

Select row with State "this order is finished" and Click on the down arrow. Then click the "Received".

19.12.2.1. Expected Result for Test Procedure 2

Fail, the system will show the message and tell you the order is already received.

19.13. Supplier Management

Login as "manager" with password "123"

19.13.1. Test Procedure 1

Click "Create a new supplier"

19.13.1.1. Expected Result for Test Procedure 1

Fail, create a new supplier need user to enter all information.

19.13.2. Test Procedure 2

Enter Name "Tesla"

email "onlineordersHKMO@tesla.com"

Contant number "01628450660"

Address "Corporate Secretary Tesla, Inc. 1 Tesla Road Austin, TX 78725"

Click "Create a new supplier"

19.13.2.1. Expected Result for Test Procedure 2

Successful, supplier create successful.

19.13.3. Test Procedure 3

Click "Edit exist supplier" and select "SB company". Delete the email and Click Edit.

19.13.3.1. Expected Result for Test Procedure 3

Successful, the system allows individual modifications, i.e. individual data can be modified without entering the full data, but data with empty fields will not be modified.

19.13.4. Test Procedure 4

Click “Edit exist supplier” and select “Tesla”. Click Delete button.

19.13.4.1. Expected Result for Test Procedure 4

It worked; "Tesla" was able to be removed.

19.14. User Management

Login as “manager” with password “123”

19.14.1. Test Procedure 1

Enter Login Name “ATL” and click “Create a new account”.

19.14.1.1. Expected Result for Test Procedure 1

Fail, Login name "ATL" already exists.

19.14.2. Test Procedure 2

Enter Login Name “CCK” and Access level “1” and click “Create a new account”.

Enter Staff Name “CCK” Phone Number 12345678.Gender Male. Other parameters according to preset values and click Save.

19.14.2.1. Expected Result for Test Procedure 2

Successful create a staff and a new account, system will show an initial password.

19.15. Add Sale Area

Login as “manager” with password “123”

19.15.1. Test Procedure 1

Enter Beijing and no remark and Click Save.
Click location and complete.

19.15.1.1. Expected Result for Test Procedure 1

Successful. All required information has been filled out

19.15.2. Test Procedure 2

Enter nothing and click save and click complete.

19.15.2.1. Expected Result for Test Procedure 2

Fail, No latitude and longitude.

19.15.3. Test Procedure 3

Enter nothing and click save. Enter “Hong Kong” click location and complete.

19.15.3.1. Expected Result for Test Procedure 3

Fail, the position must be filled in on the page “Add Sale Area”.

19.16. Edit Sale Area

Login as “manager” with password “123”

19.16.1. Test Procedure 1

Select “Beijing” and click edit, enter “Beijing” and click Save.

19.16.1.1. Expected Result for Test Procedure 1

Fail, Location is repeat.

19.16.2. Test Procedure 2

Select “Guiyang” and click edit, enter “Tibet” and click Save. Click locate and complete.

19.16.2.1. Expected Result for Test Procedure 2

Successful, all data has been entered and the system allows the sale area to be changed.

19.17. Add Department

Login as “manager” with password “123”

19.17.1. Test Procedure 1

Enter Department ID “wh001” and Name “Warehouse5”, email 123@123.com. No warehouse.

19.17.1.1. Expected Result for Test Procedure 1

Fail, DeptID “wh001” already exists.

19.17.2. Test Procedure 2

Enter Department ID “wh005” and Name “Warehouse5”, email 123@123.com. Warehouse 1

19.17.2.1. Expected Result for Test Procedure 2

Successful, DeptID “wh005” does not exist.

19.18. Edit Department

Login as “manager” with password “123”

19.18.1. Test Procedure 1

Select “wh004“ and change the Name to “Warehouse 1” ,then click save.

19.18.1.1. Expected Result for Test Procedure 1

Fail, Name “Warehouse 1” already exists.

19.18.2. Test Procedure 2

Select “wh004“ and click delete.

19.18.2.1. Expected Result for Test Procedure 2

Successful, the department can no longer be edited.

19.18.3. Test Procedure 3

Select “manag” and click Edit

19.18.3.1. Expected Result for Test Procedure 3

Can not see the combo box "warehouse".

19.19. Add Warehouse

Login as “manager” with password “123”

19.19.1. Test Procedure 1

Select “Kunming” enter name Warehouse 1 and location “昆明市滇池路 1310 號雲南民族村東門”

19.19.1.1. Expected Result for Test Procedure 3

Failure, the system does not allow warehouses with the same name or location.

19.20. Edit Warehouse

Login as “manager” with password “123”

19.20.1. Test Procedure 1

Select “Warehouse 1” and Click Delete button.

Select “Warehouse 1” in the dialogue.

19.20.1.1. Expected Result for Test Procedure 1

Fail, cannot select locations for deletion.

19.20.2. Test Procedure 2

Select “Warehouse 1” and Click Delete button.

Select “Warehouse 3” in the dialogue.

19.20.2.1. Expected Result for Test Procedure 2

Successful, all spares will be moved to the target warehouse.

19.20.3. Test Procedure 3

Select “Warehouse 4” and Click Delete button.

19.20.3.1. Expected Result for Test Procedure 3

Fail, Warehouse 4 are still unfinished restock orders.

19.21. Report

19.21.1. Test Procedure 1

Select sale report(warehouse).

19.21.1.1. Expected Result for Test Procedure 1

	WarehouseID	Name	Total Quantity	Total Sales
▸	1	Warehouse 1	1002	42654.33

Total number of item(s):	Number of row:	Total Price:
1002	1	42654.33

19.22. Change Password

19.22.1. Test Procedure 1

Login “storeman1”. Enter old password 456 new password 147, repeat password 147

19.22.1.1. Expected Result for Test Procedure 1

Fail, the old password of “storeman1” is 123, It is not match.

19.22.2. Test Procedure 2

Login “storeman1”. Enter old password 123 new password 147, repeat password 145

19.22.2.1. Expected Result for Test Procedure 2

Fail, the new password and repeat password is not match.

19.22.3. Test Procedure 3”

Login “storeman1”. Enter old password 123 new password 147, repeat password 147

19.22.3.1. Expected Result for Test Procedure 3

Successful, everything is match.

19.23. Modify Information

19.23.1. Test Procedure 1

Login “LMcleck”.

19.23.1.1. Expected Result for Test Procedure 1

User can't find this option. All LM staff cannot change the data on their own initiative

19.23.2. Test Procedure 2

Login “Dealer1” and delete everything. Click Confirm.

19.23.2.1. Expected Result for Test Procedure 2

Fail, there must be at least one column that is not empty.

19.23.3. Test Procedure 3

Login “Dealer1” and delete everything. Enter contact Number 114514 and Click Confirm.

19.23.3.1. Expected Result for Test Procedure 3

Successful, the system allows only one piece of data to be changed, and data in empty fields will not be changed.

20. User Guide

20.1. Installation and set up

20.1.1. Server and MYSQL

1. install server OS (window server 2022 datacenter)
2. Choosing an Installation Package of MYSQL (Recommended MSI installation method and the latest version). Download the MSI from https://dev.mysql.com/downloads/ and execute it.
   - MySQL 8.4 Server requires the Microsoft Visual C++ 2019 Redistributable Package to run on Windows platforms.
3. The MySQL server MSI prompts to execute MySQL Configurator immediately after it installs the MySQL server. if you miss this, you can find it from the Start Menu: the MSI creates a MySQL Configurator start menu item.
   - you can set the root password on this time, please remember it.
4. install MySQL Workbench

5. make sure you turn on the MySQL server, and connect it in MySQL Workbench (username: root, password: <your root password>, ip: 127.0.0.1, port: 3306)
   - port default is 3306
6. use the following SQL string to create a MySQLuser:

   Mysql username: mysqluser

   password: LMP@$$w0rd

   database name: OSS

```
CREATE USER 'mysqluser'@'localhost' IDENTIFIED BY 'LMP@$$w0rd';

CREATE DATABASE OSS;

GRANT ALL PRIVILEGES ON OSS.* TO ' mysqluser '@'localhost' WITH GRANT OPTION;
```

7. use the drop_all.sql in appendix to create data table
8. use the build.sql in appendix to create data table
9. use the following SQL string to create an initialization root user for login to application

   Login name: init

   password: LMP@$$w0rd

```
INSERT INTO `SaleArea` (`AreaID`, `Location`, `Latitude`, `Longitude`, `remark`)

VALUES

(1, 'Beijing', 39.9042, 116.4074, NULL);


INSERT INTO `Department` (`DeptID`, `DeptEmail`, `DeptName`, `WarehouseID`)

VALUES

('manag', 'manager@legendmotor.com', 'Manager', NULL);
```

```
INSERT INTO `Staff` (`StaffID`, `StaffName`, `SaleAreaID`, `DeptID`, `Position`, `PhoneNunber`, `Gender`)
VALUES
(1,     'init', 1, 'manag',    'manager',    '13912345678', 'M');


INSERT INTO `User` (`UserID`, `LoginName`, `Password`, `AccessLevel`, `DealerID`, `StaffID`)
VALUES
(1, 'init',
'0853fd1ee2cd0118029ff34c2aec4199883fb7cb213f28af7e1e2ba88a2e10471', 0, Null, 1);
```

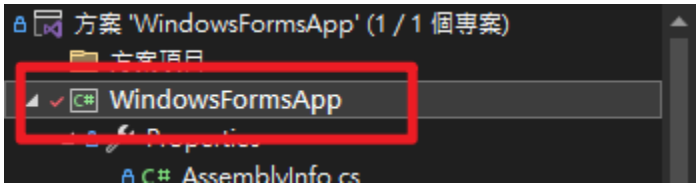
10. make sure the port 22 is open and turn on ssh service on window server.
11. fill the server and MYSQL information to sshdatabase.cs Line 22-23.
12. Test the application using Test Cases and Test Data
13. use “init” account to create a new account and delete it form Table `staff` and `user`

20.1.2.client application

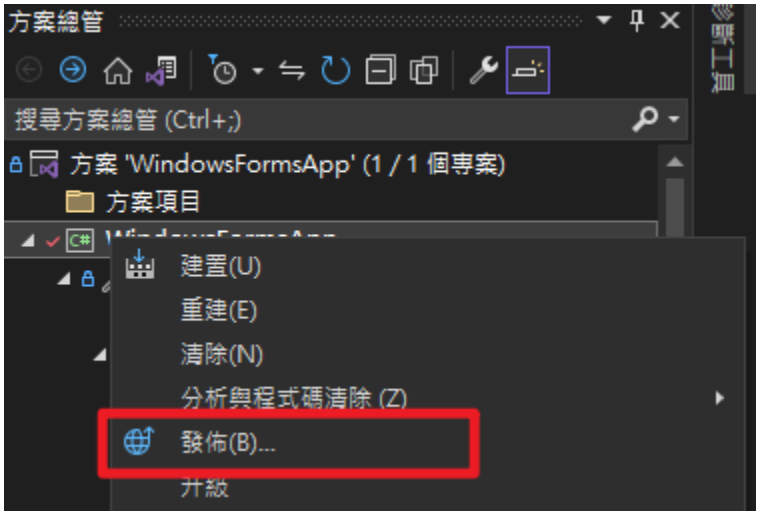
20.1.2.1.build the application

- Use visual studio 2022, WinForms.NET

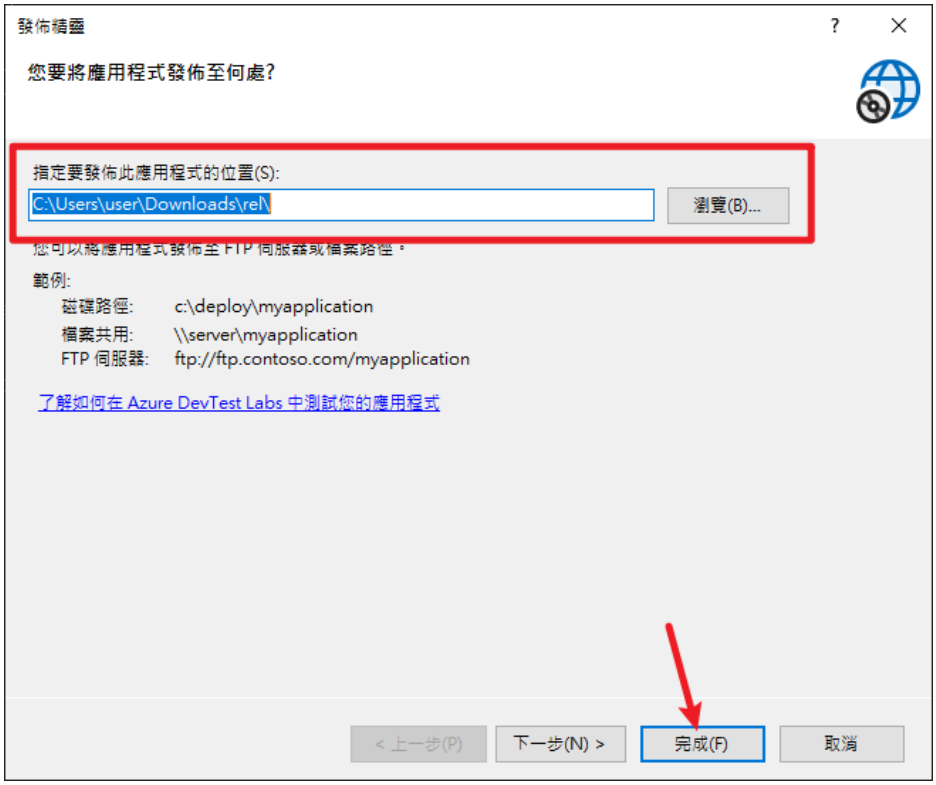
1. fill the server and MYSQL information to sshdatabase.cs Line 22-23.
2. right click there



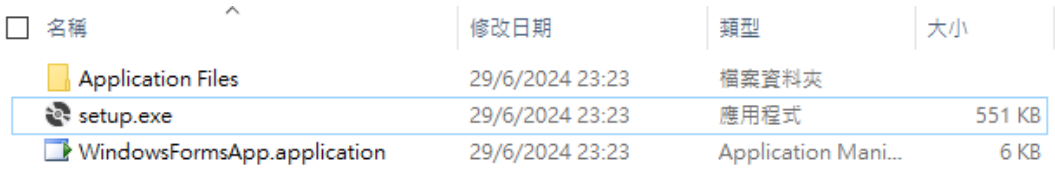
3. click publish



4. select the path for store the application exe file, and click finish



5. find it in file explorer

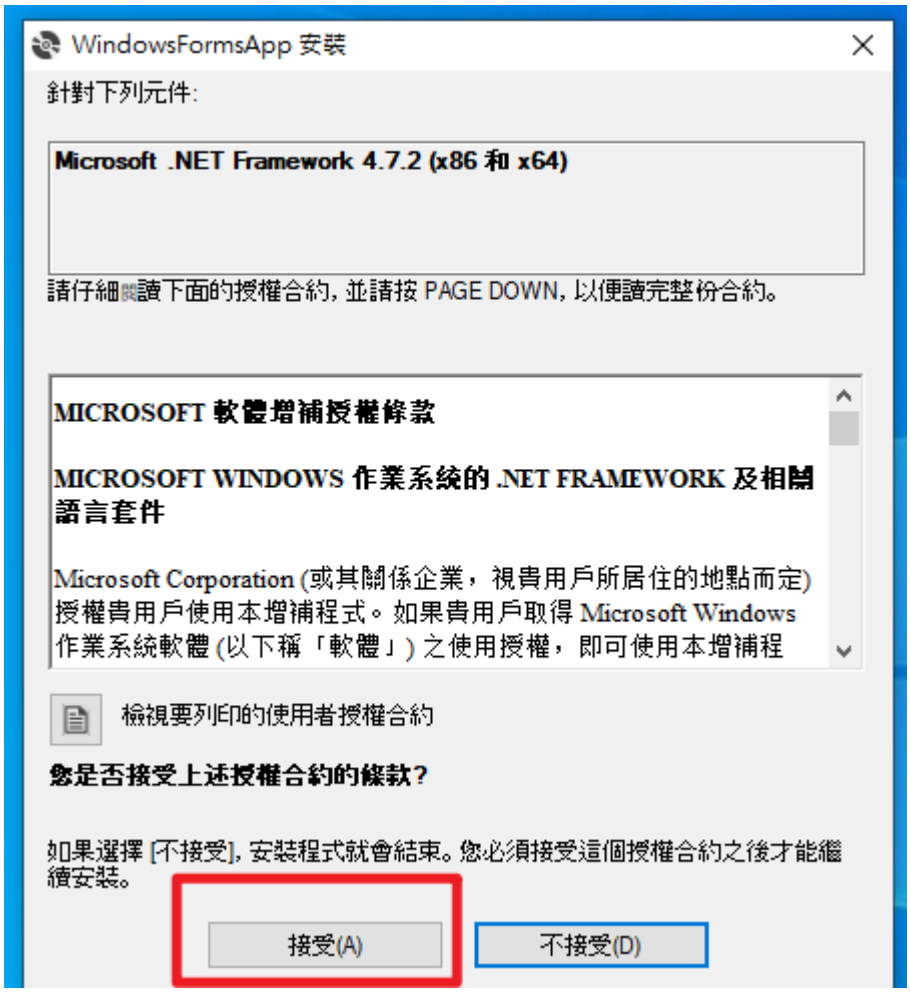


名稱	修改日期	類型	大小
Application Files	29/6/2024 23:23	檔案資料夾	
setup.exe	29/6/2024 23:23	應用程式	551 KB
WindowsFormsApp.application	29/6/2024 23:23	Application Mani...	6 KB

6. now, user can click setup.exe to install the application.

20.1.2.2.install application

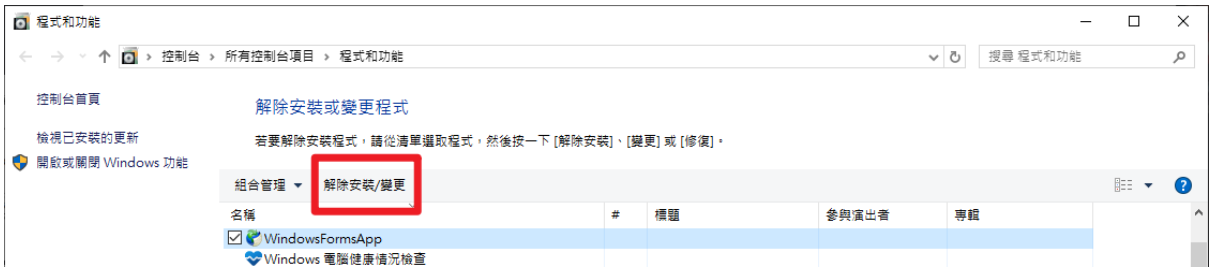
1. double click the setup.exe, make sure your computer has connected internet
2. click install (this may not show to anyone)



3. waiting the installation finish, and you can turn on the application in desktop

20.1.2.3.uninstall application

1. open the “control panel” and go to “Programs and Features”
2. find and select the application “WindowsFormApp” and click uninstall



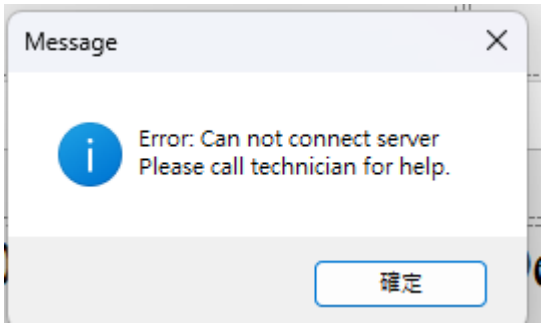
3. select the choose of uninstall and click confirm

20.2.FAQ

Q: How to have an account?

A: You should ask LM's staff to get an account

Q: Why cannot I connect to server?



A: make sure your compute has connected to network or ask LM technician for help

Q: How to execute the application in mac / iPhone / iPad / android phone / android tablet / Linux?

A: Sorry, this application only can execute in Windows.

21. Implement Plan

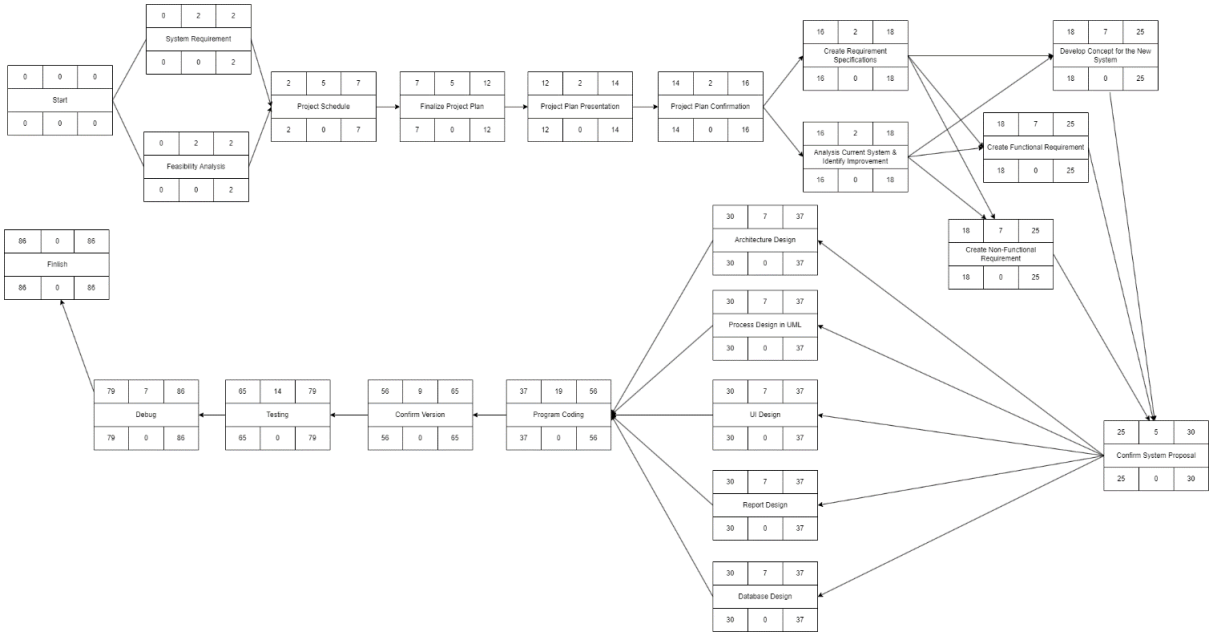
21.1.Project Schedule

21.1.1.Planning Table

Task	Duration Day(s)	Predecessor
Requirements Specification	28(total)	
System Requirement	2	
Feasibility Analysis	2	
Project Schedule	5	System Requirement Feasibility Analysis
Finalize Project Plan	5	Project Schedule
Project Plan Presentation	2	Finalize Project Plan
Project Plan Confirmation	2	Project Plan Presentation
Create Requirement Specifications	2	Project Plan Confirmation
Analysis Current System & Identify Improvement	2	Project Plan Confirmation
Develop Concept for the New System	7	Create Requirement Specifications Analysis Current System & Identify improvement
Create Functional Requirement	7	Create Requirement Specifications Analysis Current System & Identify improvement
Create Non-Functional Requirement	7	Create Requirement Specifications Analysis Current System & Identify improvement
Confirm System Proposal	5	Develop Concept for the New System Create Functional Requirement Create Non-Functional Requirement
Design Specification	35(Total)	
Architecture Design	7	Confirm System Proposal
Process Design in UML	7	Confirm System Proposal
UI Design	7	Confirm System Proposal

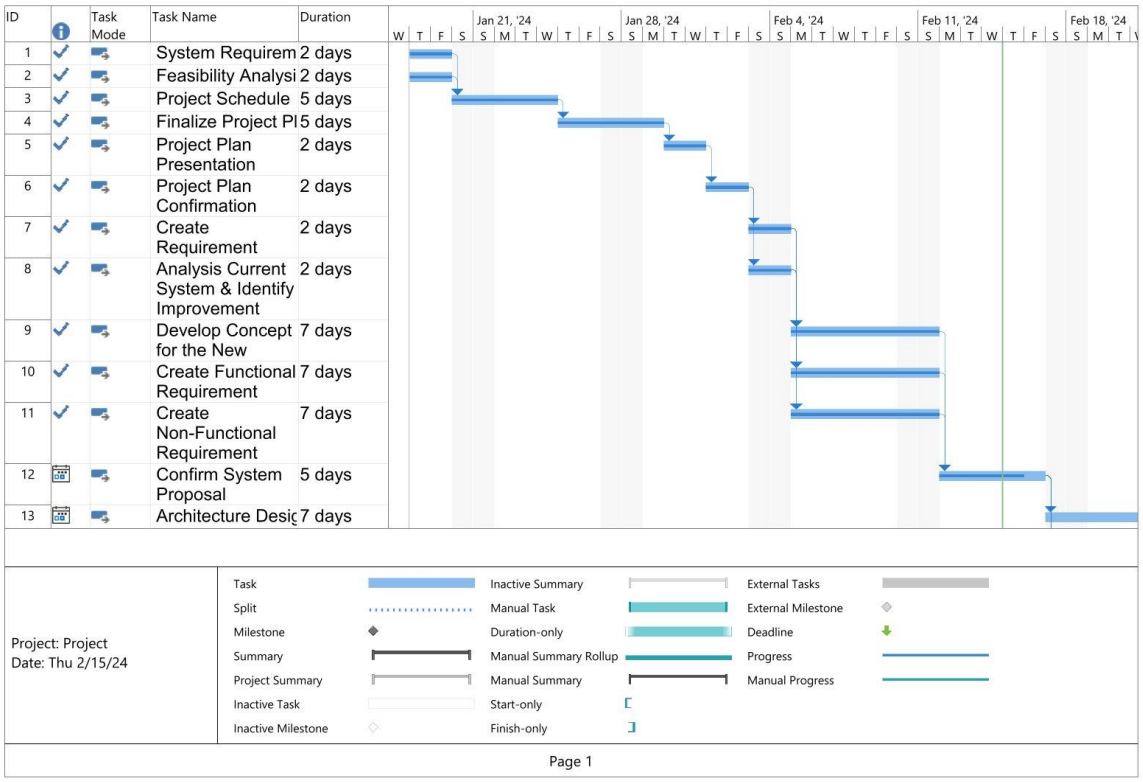
Report Design	7	Confirm System Proposal
Database Design	7	Confirm System Proposal
Program Coding	19	Architecture design Process design in UML UI design Report design Database design
Confirm Version	9	Program coding
Test Plan	21(Total)	
Testing	14	Confirm version
Debug	7	Testing

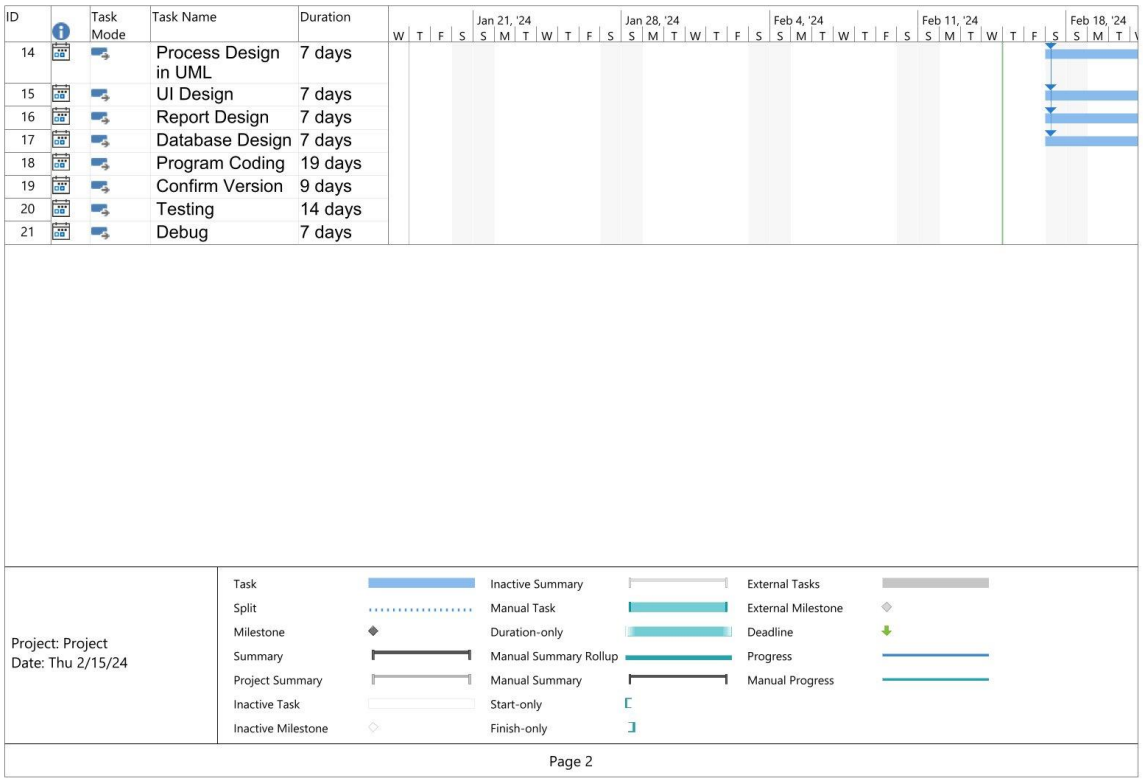
21.1.2.PERT Chart

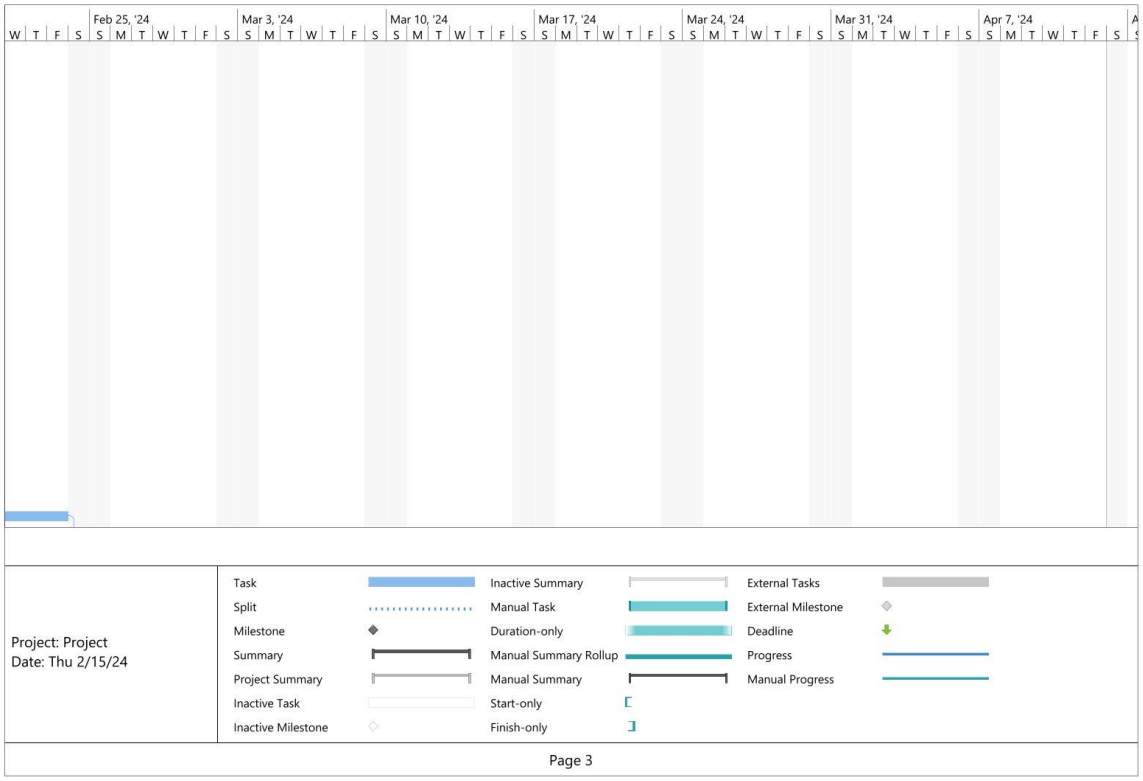


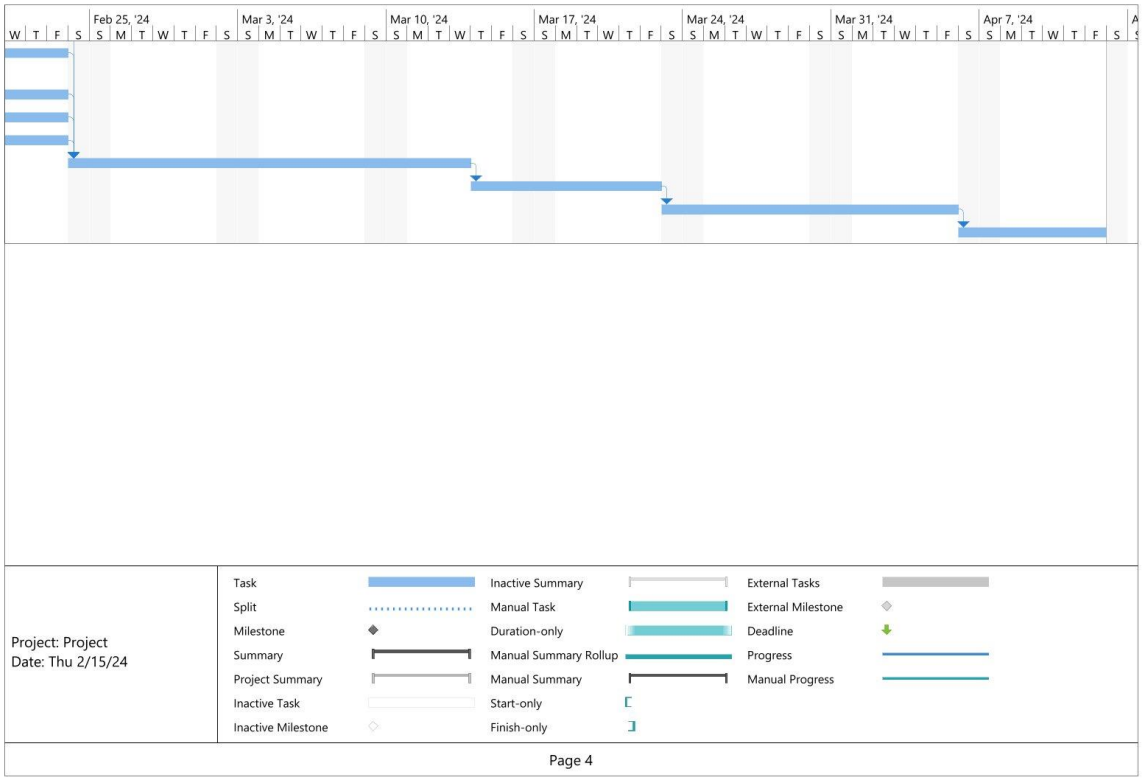
(Read more on the appendix page.)

21.1.3.Gantt Chart






Feb 25, '24
Mar 3, '24
Mar 10, '24
Mar 17, '24
Mar 24, '24
Mar 31, '24
Apr 7, '24
Project: Project
Date: Thu 2/15/24
Task
Split
Milestone
Summary
Project Summary
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
External Tasks
External Milestone
Deadline
Progress
Manual Progress
Page 3


Feb 25, '24
Mar 3, '24
Mar 10, '24
Mar 17, '24
Mar 24, '24
Mar 31, '24
Apr 7, '24
Project: Project
Date: Thu 2/15/24
Task
Split
Milestone
Summary
Project Summary
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
External Tasks
External Milestone
Deadline
Progress
Manual Progress
Page 4

21.2.Cost Estimate

21.2.1.Server Requirements:

To meet the operational needs of Legend Motor Limited, considering the large number of dealers located in different provinces and cities in China, it is recommended to purchase 100 servers. This is necessary to handle the significant amount of data and high throughput.

21.2.2.Operating System Requirements:

For the sake of maintainability and technical support, it is suggested to use Windows Server 2022 Datacenter as the operating system for the central computerized management system.

21.2.3.Client Device Requirements:

Since the development framework WinForm (.Net) can only operate on Windows systems and the users require functionality such as camera and signature, it is proposed to purchase 1500 units of Surface Go 2 as the client devices for store man and delivery worker.

	One Time	Every Year
Development Costs		
100 Server @HK$125000	HK$12,500,000	HK$0
Server OS @US$6155 (window server 2022 datacenter)	HK$48,141	HK$0
Development labor	HK$12,345,678	HK$0
Total Development Costs	**HK$24,893,819**	**HK$0**
Operational Costs		
Hardware		HK$70,000
Software		HK$8,366
Operational labor		HK$1,200,000
Total Operational Costs		**HK$1,278,366**
Implement Costs		
Windows tablet @HK$2300 (Microsoft Surface Go 2)	**HK$3,450,000**	
Total Implement Costs	**HK$3,450,000**	
Total Costs	**HK$28,343,819**	**HK$1,278,366**

22. Conclusion

In conclusion, the development of a central computerized management system for Legend Motor Limited is crucial to overcome the existing operational challenges and improve overall efficiency. By implementing standardized communication, automating order processing and stock recording procedures, and introducing real-time stock management, the company can achieve better integration, reduce manual errors, and enhance decision-making processes. The proposed system offers numerous benefits, including improved operational efficiency, enhanced information management, optimized processes, reduced delays and errors, decreased write-offs, enhanced collaboration and communication, improved risk management and security, better customer service, and a competitive advantage. With the successful implementation of the system, Legend Motor Limited can optimize its resources, improve stock management, and ultimately provide a better experience for its customers.

23. Project Log

23.1.meeting

	duration (mins)	Tse Man Fung	CHUNG Chun Kit	SZE Wang Lik	WONG Wing Wong	remark
22/1/2024	90	90	90	90	90	Discuss the case
29/1/2024	120	120	120	120	120	Discuss the system and jobs
5/2/2024	90	90	90	90	75	Review work
26/2/2024	90	90	90	90	/	Talk about the schedule of Design spec
4/3/2024	90	90	90	90	/	Discuss about UI and database
11/3/2024	60	60	60	60	/	Review work
15/3/2024	60	60	60	60	/	talk about sequence diagram
22/3/2024	60	60	60	60	/	talk about state machine diagram
4/4/2024	60	60	60	60	/	Review work
15/4/2024	60	60	60	60	/	Discuss division of work
29/4/2024	60	60	60	60	/	report progress to other
13/5/2024	60	60	60	60	/	report progress to other
27/5/2024	60	60	60	60	/	review for Prototype I
17/6/2024	60	60	60	60	/	report progress to other
2/7/2024	60	60	60	60	/	code and report review

23.2.Job Distribution

Task		Resonsible Student	contribution	Resonsible Student	contribution	Resonsible Student	contribution
• Menu Program / Login							
	Login	Tse Man Fung	100%				
	Menu	Sze Wang Lik	50%	Tse Man Fung	50%		
• Sales Order Processing Management							
	o Create and Edit Sales Order	Tse Man Fung	100%				
	o View Sales Order	Tse Man Fung	100%				
	o Cancel Sales Order						
	o Search Sales Order						
• Dispatch Processing							
	o Generate Dispatch Notes	Tse Man Fung	100%				
	o Handling Goods received	Tse Man Fung	100%				
	o Handling the picking process						
• Stock Recording/ Inventory Control							
	o Setting of record/danger levels	Chung Chun Kit	100%				
	o Record of inward goods	Chung Chun Kit	100%				
	o Handling the reordering process						
• Master Data Maintenance							
	supplier, staff, restaurant, etc	Chung Chun Kit	100%				
• System Security & Control							
	password, data access control	Tse Man Fung	100%				
• Database Implementation							
	build database and data table	Tse Man Fung	100%				
	madeup data	Chung Chun Kit	65%	Sze Wang Lik	35%		
• Other (Extra feature)							
	Preparation work before coding	Sze Wang Lik	100%				

	Form design	Tse Man Fung	70%	Sze Wang Lik	15%	Chung Chun Kit	15%
	Multi-Language	Sze Wang Lik	50%	Tse Man Fung	25%	Chung Chun Kit	25%
	auto size	Tse Man Fung	100%				
	Generate QR code	Tse Man Fung	100%				
	Scan QR code	Tse Man Fung	100%				
	Generate Document	Tse Man Fung	100%				
	Print in Printer	Tse Man Fung	100%				
	Connect to server	Tse Man Fung	100%				
	signature	Tse Man Fung	100%				
	map	Chung Chun Kit	100%				
	assign order to closed warehouse	Tse Man Fung	100%				
	assign order item to different warehouse	Tse Man Fung	100%				

24. Reference

https://scaledagileframework.com/nonfunctional-requirements/

https://www.geeksforgeeks.org/software-requirement-specification-srs-format/

https://www.jamasoftware.com/requirements-management-guide/writing-requirements/how-to-write-system-requirement-specification-srs-documents/

https://www.oracle.com/hk/cloud/costestimator.html

https://en.wikipedia.org/wiki/Non-functional_requirement

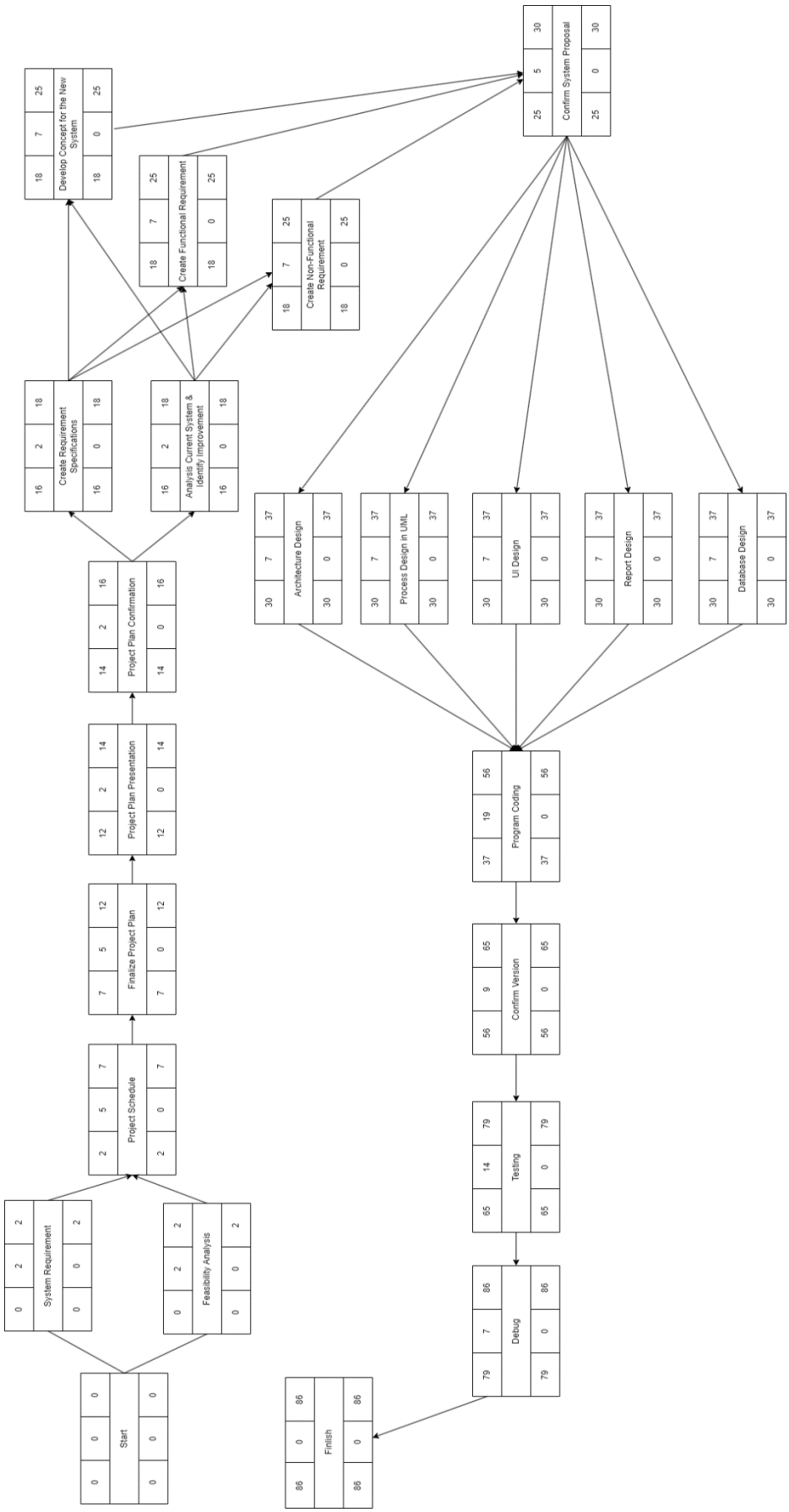
https://www.osp.de/tw/resources/articles/how-to-improve-stocktaking

25. Appendix

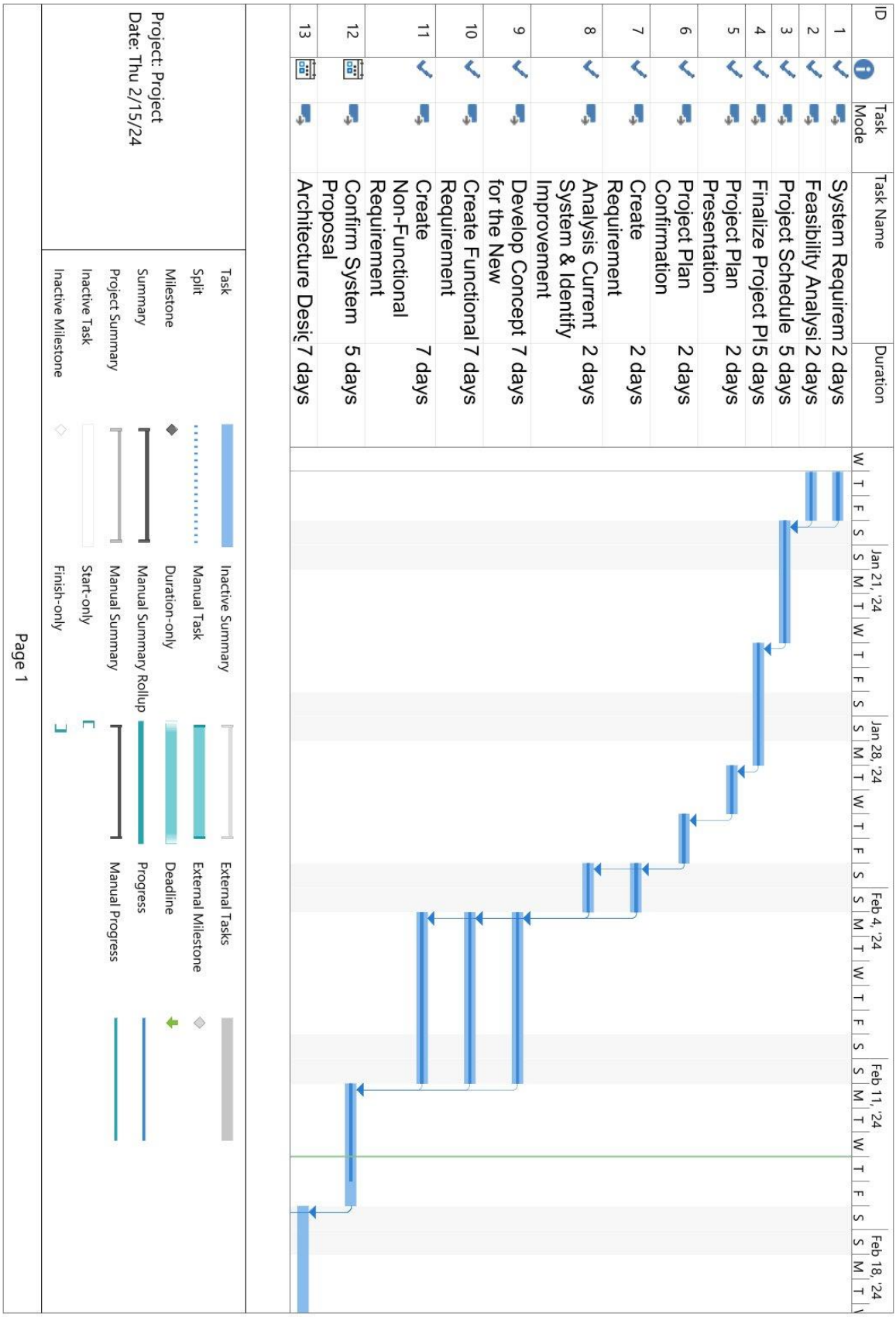
25.1. Glossary

Abbreviation	Full form	Description
API	Application Programming Interface	a way for two or more computer programs or components to communicate with each other.
CSL	Common Stock Level	the quantity of stock that a company maintains to meet its regular operational needs
DI set	Despatch Instruction set	DIC together with DID
DIC	Despatch Instruction Cover	Cover of Despatch Instruction
DID	Despatch Instruction Detail Sheet	Detail of each item in a despathc
EIV	Electronic version of Invoice	/
GRN	Good Received Notes	For settlement of invoices
GUI	Graphical User Interface	a form of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation
LM	Legend Motor Limited Company	Subsidiaries of LMC
LMC	Legend Motor Company	/
ROI	Return on investment	a ratio between net income (over a period) and investment (costs resulting from an investment of some resources at a point in time)

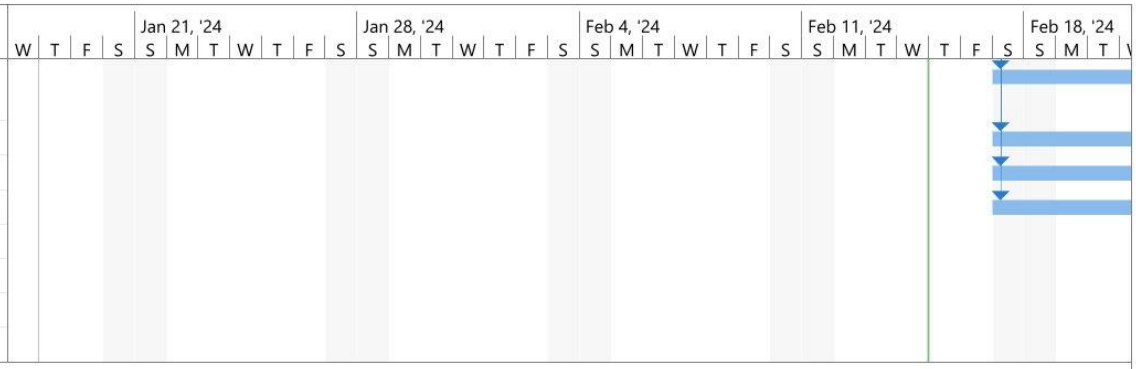
25.2. PERT Chart



25.3. Gantt Chart



ID	Task Mode	Task Name	Duration
14		Process Design in UML	7 days
15		UI Design	7 days
16		Report Design	7 days
17		Database Design	7 days
18		Program Coding	19 days
19		Confirm Version	9 days
20		Testing	14 days
21		Debug	7 days

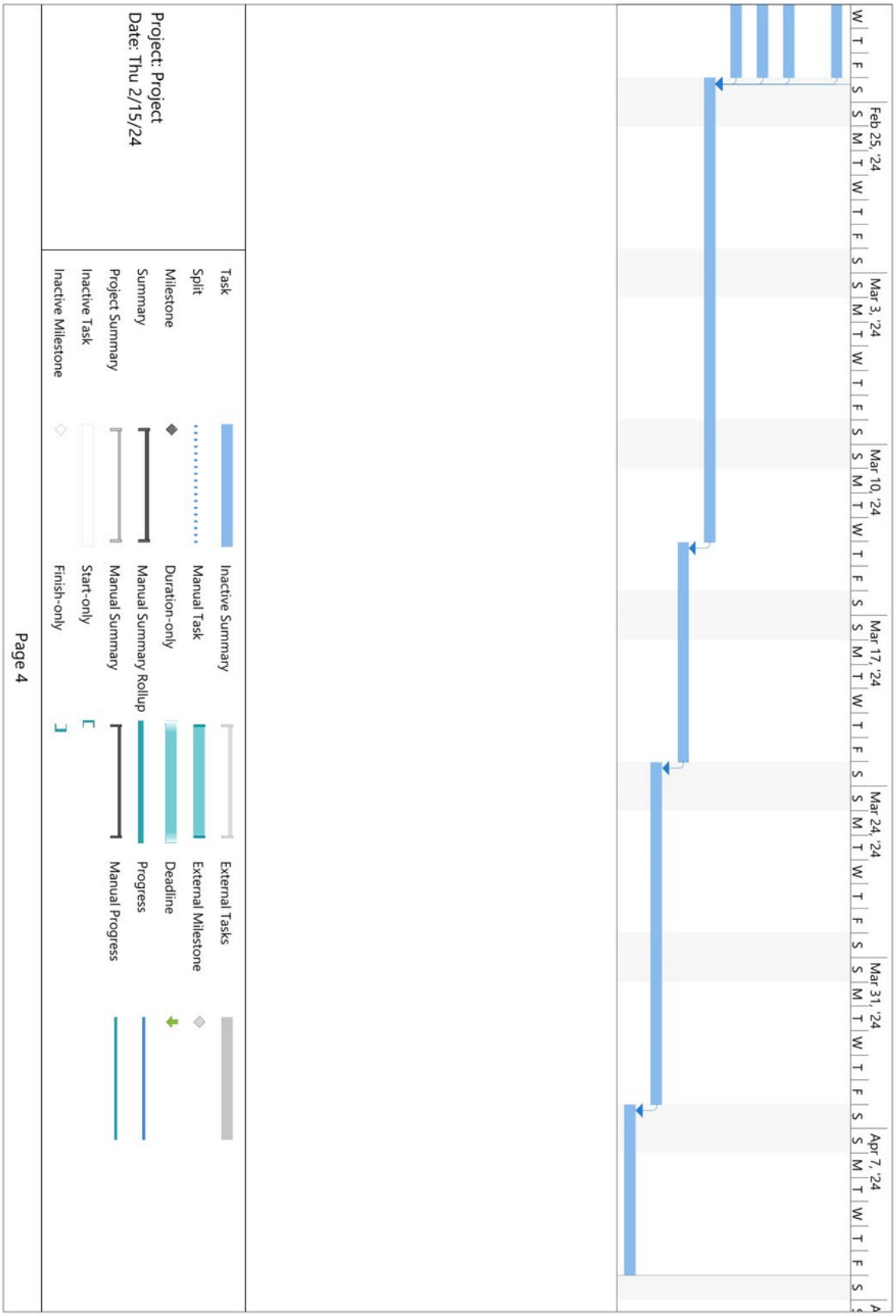


Project: Project
Date: Thu 2/15/24

Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	
Milestone		Duration-only		Deadline	
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only			
Inactive Milestone		Finish-only			

Page 2

Feb 25, '24
Mar 3, '24
Mar 10, '24
Mar 17, '24
Mar 24, '24
Mar 31, '24
Apr 7, '24
W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S
Project: Project
Date: Thu 2/15/24
Task
Split
Milestone
Summary
Project Summary
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
External Tasks
External Milestone
Deadline
Progress
Manual Progress
Page 3


Feb 25, '24
Mar 3, '24
Mar 10, '24
Mar 17, '24
Mar 24, '24
Mar 31, '24
Apr 7, '24
W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S
Project: Project
Date: Thu 2/15/24
Task
Split
Milestone
Summary
Project Summary
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
External Tasks
External Milestone
Deadline
Progress
Manual Progress
Page 4

25.4.drop_all.sql

Drop_all.sql

```
use OSS;
SET FOREIGN_KEY_CHECKS = 0;
DROP TABLE IF EXISTS ActualQuantityDespatched;
DROP TABLE IF EXISTS Category;
DROP TABLE IF EXISTS Dealer;
DROP TABLE IF EXISTS Department;
DROP TABLE IF EXISTS Invoice;
DROP TABLE IF EXISTS `Order`;
DROP TABLE IF EXISTS OrderItem;
DROP TABLE IF EXISTS OrderItemToFollow;
DROP TABLE IF EXISTS RestockItem;
DROP TABLE IF EXISTS RestockOrder;
DROP TABLE IF EXISTS SaleArea;
DROP TABLE IF EXISTS ScrapItems;
DROP TABLE IF EXISTS Spare;
DROP TABLE IF EXISTS Staff;
DROP TABLE IF EXISTS Stock;
DROP TABLE IF EXISTS Supplier;
DROP TABLE IF EXISTS `User`;
DROP TABLE IF EXISTS Warehouse;
DROP TABLE IF EXISTS WarehouseStockLevel;
DROP TABLE IF EXISTS Cart;
DROP TABLE IF EXISTS Signature;
DROP TABLE IF EXISTS ActualStock;
DROP TABLE IF EXISTS OrderItemForCollect;
SET FOREIGN_KEY_CHECKS = 1;
```

25.5.build.sql

build.sql

```
use OSS;
CREATE TABLE `Dealer` (
  `DealerID` int PRIMARY KEY AUTO_INCREMENT,
  `DealerName` varchar(255) NOT NULL,
  `SaleAreaID` int NOT NULL,
  `ContantNumber` varchar(20) UNIQUE NOT NULL,
  `email` varchar(255) NOT NULL,
  `OfficeAddress` varchar(255) NOT NULL,
  `DeliveryAddress` varchar(255)
);

CREATE TABLE `SaleArea` (
  `AreaID` int PRIMARY KEY AUTO_INCREMENT,
  `Location` varchar(255) NOT NULL,
  `Latitude` double NOT NULL,
  `Longitude` double NOT NULL,
  `remark` varchar(255)
);

CREATE TABLE `Staff` (
  `StaffID` int PRIMARY KEY AUTO_INCREMENT,
  `StaffName` varchar(255),
  `SaleAreaID` int,
  `DeptID` varchar(5),
  `Position` varchar(255) NOT NULL COMMENT '下拉表單',
  `PhoneNunber` varchar(20) NOT NULL,
```

```
  `Gender` ENUM ('F', 'M') NOT NULL
);

CREATE TABLE `Department` (
  `DeptID` varchar(5) PRIMARY KEY,
  `DeptEmail` varchar(255) NOT NULL,
  `DeptName` varchar(255) NOT NULL,
  `State` char(1) NOT NULL DEFAULT 'N' COMMENT 'only N,D',
  `WarehouseID` int
);

CREATE TABLE `User` (
  `UserID` int PRIMARY KEY AUTO_INCREMENT,
  `LoginName` varchar(248) UNIQUE NOT NULL,
  `Password` char(65) NOT NULL COMMENT 'the first char is the state of the account(such as ! mean bock)the next 64 char is sha256',
  `AccessLevel` int NOT NULL DEFAULT 10000 COMMENT '0 has the highest permission',
  `DealerID` int COMMENT 'if it is null, mean this user is a staff of Dealer',
  `StaffID` int
);

CREATE TABLE `Cart` (
  `UserID` int,
  `SpareID` char(6),
  `Qty` int,
  PRIMARY KEY (`UserID`, `SpareID`)
);
```

```
CREATE TABLE `Category` (
  `CategoryLetter` ENUM ('A', 'B', 'C', 'D') PRIMARY KEY COMMENT 'only can be A-D',
  `Type` ENUM ('Sheet Metal', 'Major Assemblies', 'Light Components', 'Accessories') NOT NULL COMMENT 'only can be Sheet Metal, Major Assemblies, Light Components and Accessories'
);

CREATE TABLE `Supplier` (
  `SupplierID` int PRIMARY KEY AUTO_INCREMENT,
  `Name` varchar(255) NOT NULL,
  `Address` varchar(255) NOT NULL,
  `ContantNumber` varchar(20) UNIQUE NOT NULL,
  `email` varchar(255) NOT NULL,
  `State` char(1) NOT NULL DEFAULT 'N' COMMENT 'only N,D'
);

CREATE TABLE `Spare` (
  `SpareID` char(6) PRIMARY KEY COMMENT 'CategoryLetter+5-digit number
Sometime it name part no.',
  `CategoryLetter` ENUM ('A', 'B', 'C', 'D'),
  `SpareName` varchar(255) NOT NULL,
  `Price` DECIMAL(20,2) NOT NULL,
  `Description` varchar(255) DEFAULT 'none',
  `Weight` float NOT NULL COMMENT 'unit is g',
  `State` char(1) NOT NULL DEFAULT 'N' COMMENT 'only N,D',
  `SupplierID` int NOT NULL
```

```
);

CREATE TABLE `Stock` (
  `SpareID` char(6) PRIMARY KEY ,
  `quantity` int NOT NULL COMMENT 'Total Stock of all warehouse'
);

CREATE TABLE `ActualStock` (
  `WarehouseID` int,
  `SpareID` char(6),
  `quantity` int NOT NULL,
  `AutoRestork` bool NOT NULL DEFAULT true,
  PRIMARY KEY (`WarehouseID`, `SpareID`)
);

CREATE TABLE `Order` (
  `OrderSerial` char(20) PRIMARY KEY COMMENT 'Format: YYYYMMDD-hhmm-Dealer.DealerID',
  `DealerID` int NOT NULL,
  `OrderDate` datetime NOT NULL DEFAULT (now()) COMMENT 'it can not use timestamp',
  `OrderNumberfromDealer` varchar(255),
  `State` char(1) NOT NULL COMMENT 'use a char to record the state(F = this order is finished, W = Waiting to finished, etc...)',
  `remark` varchar(255)
);

CREATE TABLE `OrderItem` (
```

```
  `OrderSerial` char(20),
  `ItemID` char(6),
  `Quantity` int NOT NULL DEFAULT 1,
  `PriceInOrder` DECIMAL(20,2) NOT NULL COMMENT '防止改價，令報表出錯',
  PRIMARY KEY (`OrderSerial`, `ItemID`)
);

CREATE TABLE `OrderItemToFollow` (
  `OrderSerial` char(20),
  `ItemID` char(6),
  `Quantity` int NOT NULL DEFAULT 1,
  `State` char(1) NOT NULL COMMENT 'use a char to record the state(D = Dnoe, W = Wait, etc...)',
  PRIMARY KEY (`OrderSerial`, `ItemID`)
);

CREATE TABLE `OrderItemForCollect` (
  `OrderSerial` char(20),
  `ItemID` char(6),
  `Assgin` int,
  `Quantity` int NOT NULL DEFAULT 1,
  `State` char(1) NOT Null DEFAULT "W" COMMENT 'use a char to record the state(C = Collecting, W = Wait)',
  PRIMARY KEY (`OrderSerial`, `ItemID`,`Assgin`)
);

CREATE TABLE `ActualQuantityDespatched` (
```

```
  `WarehouseID` int,
  `InvoiceID` char(16),
  `ItemID` char(6),
  `Quantity` int NOT NULL DEFAULT 1,
  `BundlesNumber` int NOT NULL DEFAULT 0,
  PRIMARY KEY (`WarehouseID`, `InvoiceID`, `ItemID`)
);

CREATE TABLE `ScrapItems` (
  `ItemID` char(6),
  `WarehouseID` int,
  `RecordTime` datetime NOT NULL,
  `RecordStaffID` int NOT NULL,
  `Quantity` int NOT NULL DEFAULT 1,
  `Discrepancy` varchar(255) NOT NULL,
  PRIMARY KEY (`ItemID`, `RecordTime`,`WarehouseID`)
);

CREATE TABLE `RestockOrder` (
  `RestockOrderID` int PRIMARY KEY AUTO_INCREMENT,
  `WarehouseID` int,
  `OrderDate` date NOT NULL,
  `Remark` varchar(255)
);

CREATE TABLE `RestockItem` (
  `RestockOrderID` int,
  `ItemID` char(6),
```

```
  `Quantity` int NOT NULL DEFAULT 1,
  `State` char(1) NOT NULL,
  PRIMARY KEY (`RestockOrderID`, `ItemID`)
);

CREATE TABLE `Invoice` (
  `InvoiceID` char(16) PRIMARY KEY COMMENT
'YYYYMMDD+DealerID+SerialNo.(01-99)',
  `OrderSerial` char(20) NOT NULL,
  `DespatchForemanID` int,
  `InvoiceDate` date NOT NULL,
  `CompleteState` char(1) NOT NULL DEFAULT 'C'
);

CREATE TABLE `Signature` (
  `InvoiceID` char(16) PRIMARY KEY,
  `Name` varchar(255) NOT NULL,
  `SignatureDate` datetime NOT NULL,
  `Sign` LONGBLOB NOT NULL
);

CREATE TABLE `Warehouse` (
  `WarehouseID` int PRIMARY KEY AUTO_INCREMENT,
  `Name` varchar(255) NOT NULL,
  `SaleAreaID` int NOT NULL,
  `Location` varchar(255) NOT NULL,
  `State` char(1) NOT NULL DEFAULT 'N' COMMENT 'only N,D'
);
```

```
CREATE TABLE `WarehouseStockLevel` (
  `WarehouseID` int,
  `SpareID` char(6),
  `ROL` int NOT NULL DEFAULT 0,
  `DL` int NOT NULL DEFAULT 0,
  `CSL` int NOT NULL DEFAULT (ROL*1.2) COMMENT '如果短期內多次到達ROL, 則 CSL 會變大',
  PRIMARY KEY (`WarehouseID`, `SpareID`)
);

ALTER TABLE `ActualStock` ADD FOREIGN KEY (`WarehouseID`) REFERENCES `Warehouse` (`WarehouseID`);

ALTER TABLE `ActualStock` ADD FOREIGN KEY (`SpareID`) REFERENCES `Spare` (`SpareID`);

ALTER TABLE `Signature` ADD FOREIGN KEY (`InvoiceID`) REFERENCES `Invoice` (`InvoiceID`);

ALTER TABLE `Cart` ADD FOREIGN KEY (`UserID`) REFERENCES `User` (`UserID`);

ALTER TABLE `Cart` ADD FOREIGN KEY (`SpareID`) REFERENCES `Spare` (`SpareID`);

ALTER TABLE `Dealer` ADD FOREIGN KEY (`SaleAreaID`) REFERENCES `SaleArea` (`AreaID`);
```

```
ALTER TABLE `Staff` ADD FOREIGN KEY (`SaleAreaID`) REFERENCES `SaleArea` (`AreaID`);

ALTER TABLE `Staff` ADD FOREIGN KEY (`DeptID`) REFERENCES `Department` (`DeptID`);

ALTER TABLE `Department` ADD FOREIGN KEY (`WarehouseID`) REFERENCES `Warehouse` (`WarehouseID`);

ALTER TABLE `User` ADD FOREIGN KEY (`DealerID`) REFERENCES `Dealer` (`DealerID`);

ALTER TABLE `User` ADD FOREIGN KEY (`StaffID`) REFERENCES `Staff` (`StaffID`);

ALTER TABLE `Spare` ADD FOREIGN KEY (`CategoryLetter`) REFERENCES `Category` (`CategoryLetter`);

ALTER TABLE `Spare` ADD FOREIGN KEY (`SupplierID`) REFERENCES `Supplier` (`SupplierID`);

ALTER TABLE `Stock` ADD FOREIGN KEY (`SpareID`) REFERENCES `Spare` (`SpareID`);

ALTER TABLE `Order` ADD FOREIGN KEY (`DealerID`) REFERENCES `Dealer` (`DealerID`);

ALTER TABLE `OrderItem` ADD FOREIGN KEY (`OrderSerial`) REFERENCES `Order` (`OrderSerial`);
```

```
ALTER TABLE `OrderItem` ADD FOREIGN KEY (`ItemID`) REFERENCES `Spare` (`SpareID`);

ALTER TABLE `OrderItemToFollow` ADD FOREIGN KEY (`OrderSerial`) REFERENCES `Order` (`OrderSerial`);

ALTER TABLE `OrderItemToFollow` ADD FOREIGN KEY (`ItemID`) REFERENCES `Spare` (`SpareID`);

ALTER TABLE `ActualQuantityDespatched` ADD FOREIGN KEY (`WarehouseID`) REFERENCES `Warehouse` (`WarehouseID`);

ALTER TABLE `ActualQuantityDespatched` ADD FOREIGN KEY (`InvoiceID`) REFERENCES `Invoice` (`InvoiceID`);

ALTER TABLE `ActualQuantityDespatched` ADD FOREIGN KEY (`ItemID`) REFERENCES `Spare` (`SpareID`);

ALTER TABLE `ScrapItems` ADD FOREIGN KEY (`ItemID`) REFERENCES `Spare` (`SpareID`);

ALTER TABLE `ScrapItems` ADD FOREIGN KEY (`RecordStaffID`) REFERENCES `Staff` (`StaffID`);

ALTER TABLE `RestockOrder` ADD FOREIGN KEY (`WarehouseID`) REFERENCES `Warehouse` (`WarehouseID`);

ALTER TABLE `RestockItem` ADD FOREIGN KEY (`RestockOrderID`) REFERENCES `RestockOrder` (`RestockOrderID`);
```

```
ALTER TABLE `RestockItem` ADD FOREIGN KEY (`ItemID`) REFERENCES `Spare` (`SpareID`);

ALTER TABLE `Invoice` ADD FOREIGN KEY (`OrderSerial`) REFERENCES `Order` (`OrderSerial`);

ALTER TABLE `Invoice` ADD FOREIGN KEY (`DespatchForemanID`) REFERENCES `Staff` (`StaffID`);

ALTER TABLE `Warehouse` ADD FOREIGN KEY (`SaleAreaID`) REFERENCES `SaleArea` (`AreaID`);

ALTER TABLE `WarehouseStockLevel` ADD FOREIGN KEY (`WarehouseID`) REFERENCES `Warehouse` (`WarehouseID`);

ALTER TABLE `WarehouseStockLevel` ADD FOREIGN KEY (`SpareID`) REFERENCES `Spare` (`SpareID`);

ALTER TABLE `OrderItemForCollect` ADD FOREIGN KEY (`Assgin`) REFERENCES `Warehouse` (`WarehouseID`);

ALTER TABLE `OrderItemForCollect` ADD FOREIGN KEY (`OrderSerial`) REFERENCES `Order` (`OrderSerial`);

ALTER TABLE `OrderItemForCollect` ADD FOREIGN KEY (`ItemID`) REFERENCES `Spare` (`SpareID`);

ALTER TABLE `ScrapItems` ADD FOREIGN KEY (`WarehouseID`) REFERENCES `Warehouse` (`WarehouseID`);
```

End of Report