Project Final Report
Robot Store Management System
Canvas group: **LiuSCuiX**

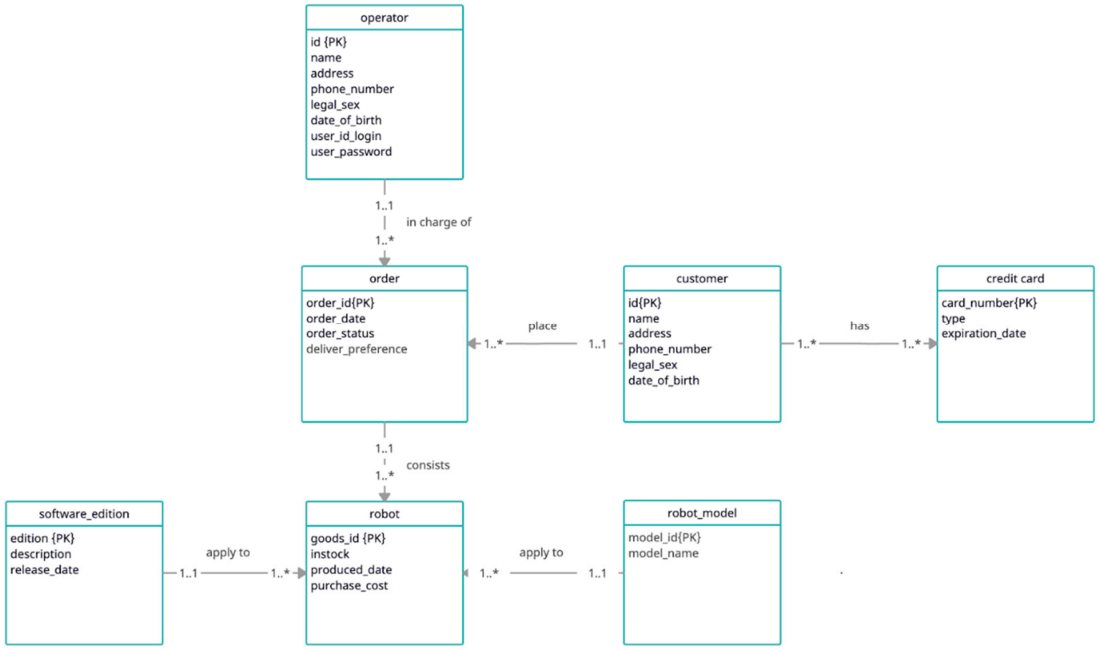
Group member: **Shuhao Liu, Xiangyu Cui**

I. README section:

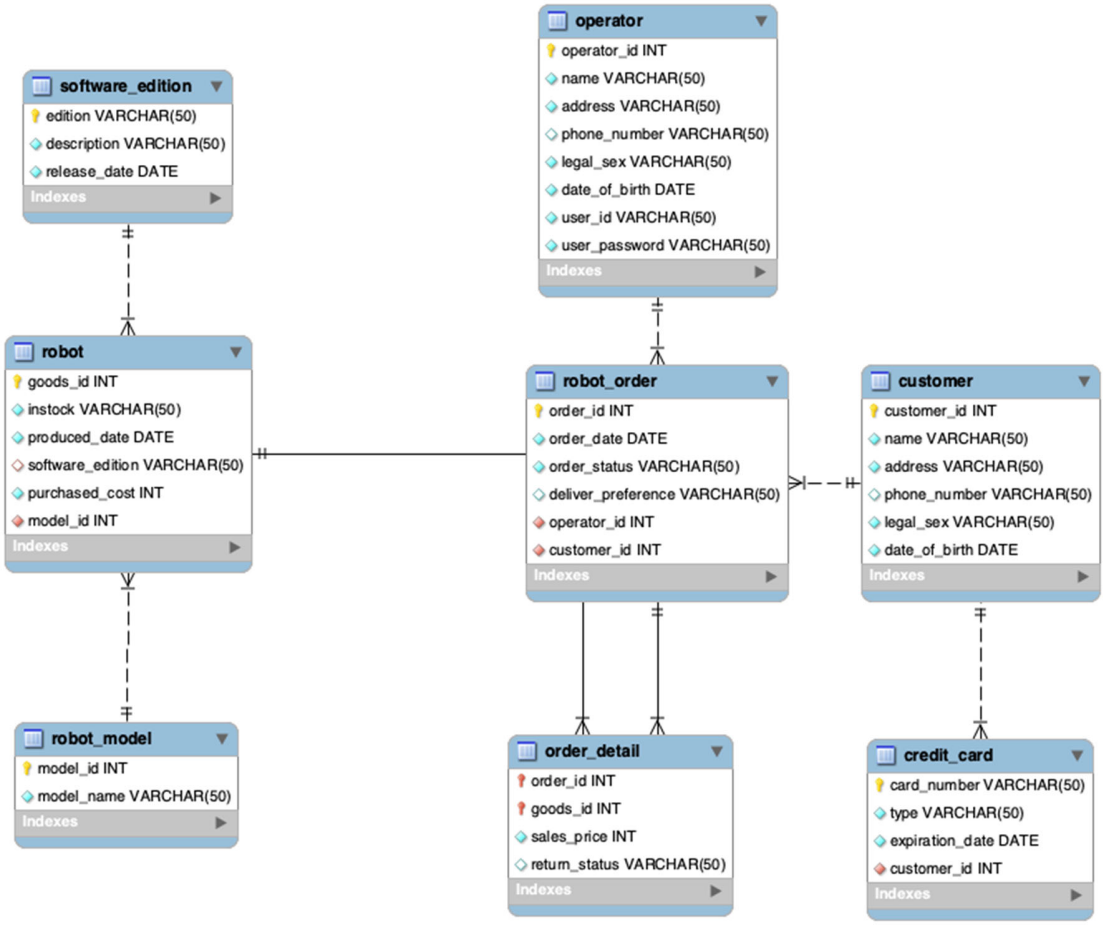
- The project has the project Handbook, which possesses all the program functions and end user test case.
- Github: https://github.com/AndyFCui/OnlineShop-Database
- Programming Language: Python, MySQL
- Python Version: 3.10
- Python Edition Control: Anaconda
- Python Package: pymysql
- MySQL Version: 8.0.33
- IDE: PyCharm, Workbench
- Program User Interface: Terminal
- MySQL default user: root, your_passwd

- Download:
- Anaconda: https://www.anaconda.com/
- Pycharm: https://www.jetbrains.com/pycharm/download/#section=windows
- Mysql: https://www.mysql.com/downloads/

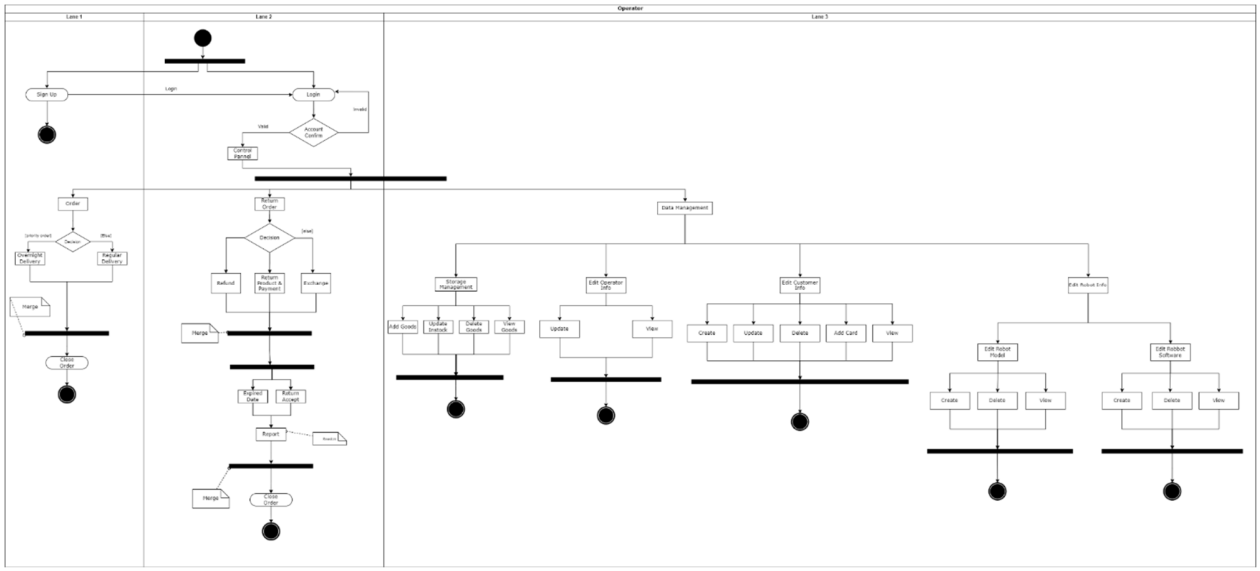
II. UML diagram:



III. Logic Design:



IV. End User Program Activity:



V. Interact With The System (Handbook Section)

Robot Sale System Handbook

System Login View

Account Function

1. Account Part

1.1. Signup

We need to check if the account exists or not, if the account exists already, then we need to create another account.

1.2. Login

We need to enter our account and password, then we login with our existing account. If the account or password doesn't match, then the program will report an error and error. We need to check it and try again.

Operator System Content

System Welcome View

Control Panel

2. Create Order
When the order sheet is created, then the stock goods will show `sold`.

2.1. Overnight Delivery

This is to set the product delivery overnight.

2.2. Regular delivery

This is to set the product delivery regularly.

3. Return Order

3.1. Valid policy return (Date base on local system time zone)

The product is valid to return.

3.1.1. Return Product

Return product directly.

3.1.2. Return Product & Payment

Return product and return the payment to the original card.

3.1.3. Exchange

Exchange the product for another one.

3.2. Invalid policy product.

Products are invalid to return (Base on Local System Time Zone)

3.2.1. Expired date

Product invalid return during expired date, and message a report reason.

3.2.2. Return Invalid

Product invalid return since detailed reason, and message a report reason.

4. Data Management

4.1. Storage Management (Goods stock information)

4.1.1. Add goods

Add new goods to storage.

4.1.2. Update In Stock

Update stock info for goods.

4.1.3. Delete Goods

Delete goods in storage.

4.2. Edit Operator Info

4.2.1. Update (Operator Info)

Update operator info of system.

4.3 Edit Customer Info

4.3.1. Create (New Customer)

Add a new customer in the system.

4.3.2. Update (Customer)

Update customer's information.

4.3.3. Delete (Customer)

Delete the customer's information.

4.3.4. View (Customer)

View the customer's information.

4.4. Edit Robot Info

4.4.1. Edit Robot Model

Edit Robot's Model

4.4.1.1. Create (Robot Model)

Create a new robot model.

4.4.1.2. Delete (Robot Model)

Delete the robot model.

4.4.1.3. View (Robot Model)

View the robot model.

4.4.2. Edit Robot Software

Edit the Robot's software

4.4.2.1. Create (Robot Software)

Create the new robot software.

4.4.2.2. Delete (Robot Software)

Delete the robot's software.

4.4.2.3. View (Robot Software)

View the robot's software.

VI. Procedure Test Case:

Operator:

############################

operator_id int primary key: 1

name varchar(50): tom

address varchar(50): neu

phone_number varchar(50): 123123

legal_sex varchar (50): male

date_of_birth date: 2023-02-16

user_id varchar(50): tom

user_password varchar(50): 123123

############################

Customer:

customer_id int primary key: 100

name varchar(50): Tom

address varchar(50): address

phone_number varchar(50): 123456

legal_sex varchar (50): male

date_of_birth date: 2222-02-22

##############################

Model:

model_id int primary key : 428910

model_name varchar(50): Maru

##############################

Software edition:

edition varchar(50) primary key: Galaxy 0.1

description varchar(50): New software designed for excellent robot

release_date date: 2222-02-22

##############################

goods_id int: 4000

instock varchar(50): instock

produced_date date:2222-02-22

software_edition VARCHAR(50):Galaxy 0.1

purchased_cost int:300

model_id int:428910

##############################

order_id int primary key: 1

order_date date not null: 2222-02-22

order_status varchar(50): Order

deliver_preference varchar(50): regular

operator_id int: 1

customer_id int: 100

##############################

Order detail:

order_id: 1

goods_id int: 4000

sales_price int: 500

return_status varchar(50): None

##############################

order_id int primary key: 2

order_date date not null: 2223-04-15

order_status varchar(50): Order

deliver_preference varchar(50): regular

operator_id int: 1

customer_id int: 100

##############################

Order detail:

order_id: 2

goods_id int: 3000

sales_price int: 500

return_status varchar(50): None

VII. Lessons Learned:

Through this project, we gained technical expertise in several areas. Firstly, we became proficient in using MySQL to create and manage databases, tables, and procedures. We also gained experience in writing SQL queries to extract and manipulate data from the database. Additionally, we learned how to use programming constructs such as loops, conditionals, and variables in MySQL stored procedures.

VIII. Insights & Time Management Insights:

One important insight we gained from this project is the importance of careful planning and organization when working on a database project. It is critical to clearly define the project requirements and design the database schema accordingly. Another insight is the importance of testing and debugging the code thoroughly before deploying it to production. This helps to identify and fix any issues before they cause problems for end users. Additionally, this project made me pay more attention to decomposition issues and process design, which improve the communication between group members and

In terms of time management, we learned the importance of breaking down the project into smaller, manageable tasks and setting realistic deadlines for each task. This helped us to stay on track and complete the project on time.

IX. Alternative Design/Approaches:

During the project, we realized that there were alternative design approaches that could have been used. For example, instead of using a stored procedure to update customer names, we could have used an UPDATE statement directly in the application code. However, we chose to use a stored procedure to encapsulate the update logic and make it easier to maintain in the future.

X. Code Not Working:

During the project, we encountered some issues with the stored procedure that updates customer names. Specifically, we had to modify the UPDATE statement to use a WHERE clause with a key column so that it would work with safe updates enabled. This was a good learning experience, as it helped me to understand the importance of safe updates and how to modify SQL queries to work with this feature enabled.

XI. Bonus Feature:

1. 3 to 5 interesting queries that can be used for analysis or visualization of the data.

   We did many queries, it's a complete project, it's only without website GUI, even though can be used in a business environment. Besides, every PROCEDURE will match one feature catalog. We can use it by menu.

2. Overly complicated translations from user operations to database operations.

   We have user signup and log in to enter the system, we can quit the system and switch different users to login. Each menu can back there pre menu.

3. Visualization of the data. (1-5 points)

   In our menu, each table where records are stored can use `View` to display data. Its list is clear and detailed.