1. **The Intro**

   First of all, I tried to build an Airbnb database management system. However, I can't import the Airbnb data csv into MySQL through workbench. I have to change my project into another one.

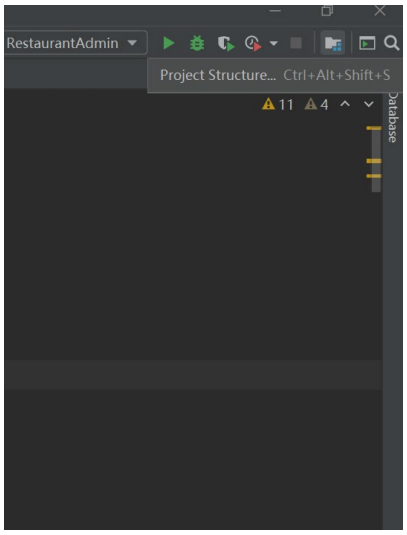
   The new project is based on an idea for my favorite restaurant near by the school—Thaitation. Due to the COVID-19, they once talked about an electronic system will be better for them since they are more like a home restaurant. I get their menu for data, and create customer&cook information from https://www.generatedata.com/ (which only generate Canadian information). I sincerely hope the quarantine be over and they may use my system.

2. **README**

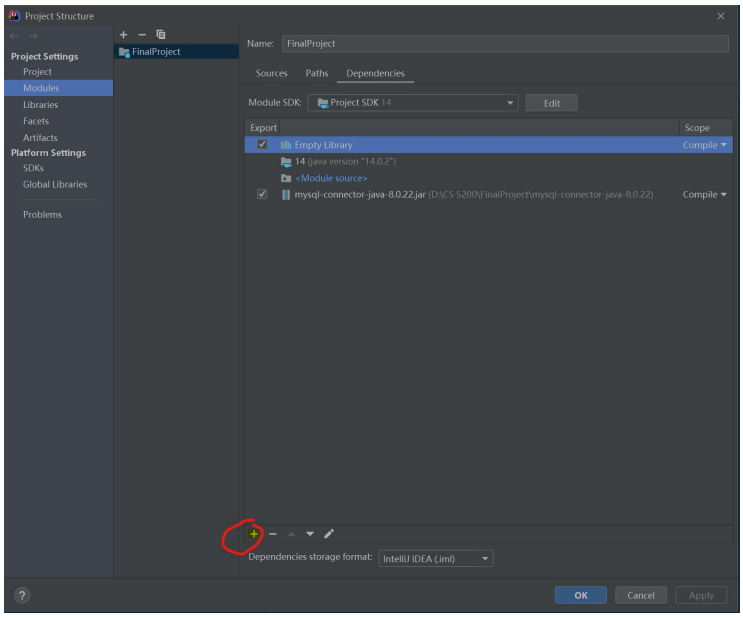
   I used Java 14.0 and MySQL for the system by IntelliJ.

   The essential part will be mysql-connector-java-8.0.22.jar which I have provide.

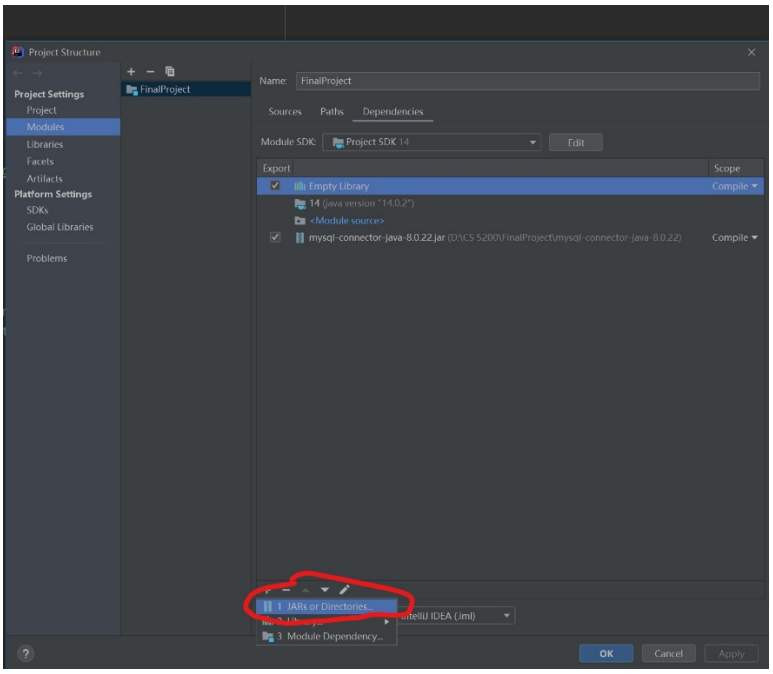
   How to import



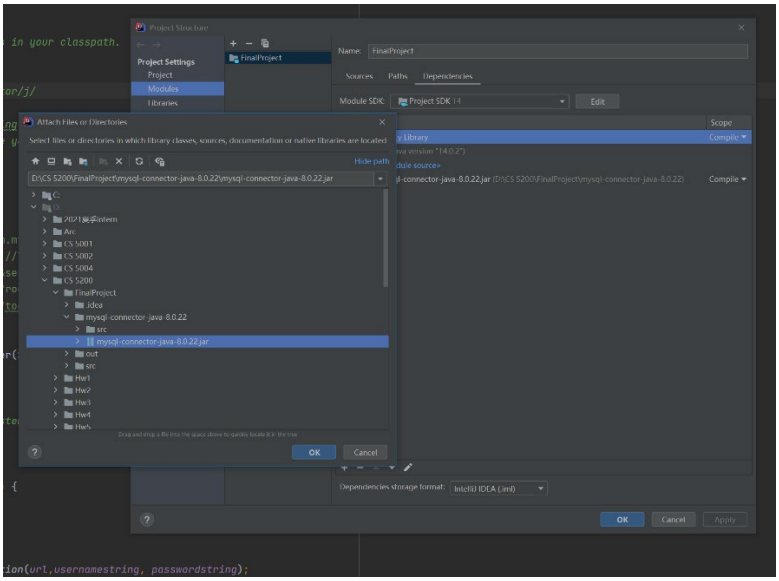
Click the Project Structure.



Click the red circle



Click the red circle again to input the jar file



Other tips:

Please remember to change the username and password if we don't have the same.

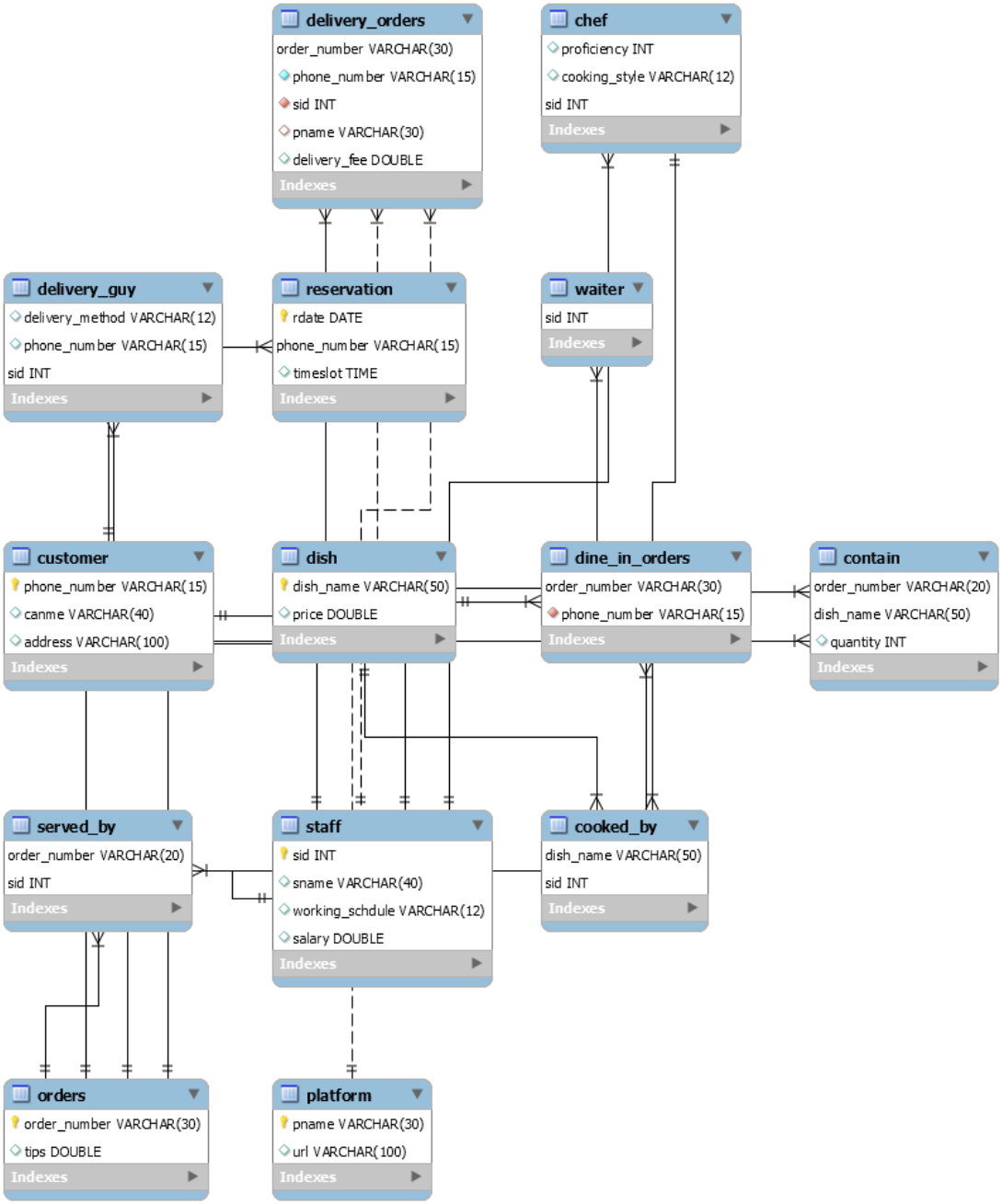
😊

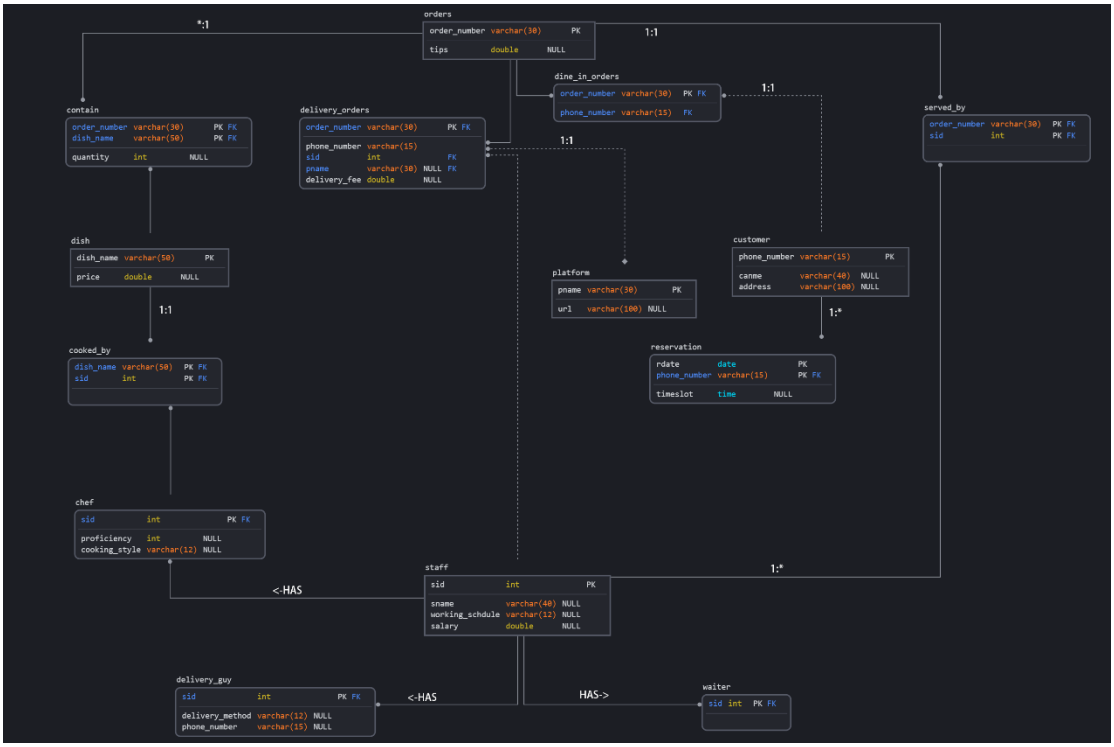
```
public class RestaurantAdmin {

    static final String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
    static final String url = "jdbc:mysql://localhost:3306/restaurant?useSSL=false&" +
            "allowPublicKeyRetrieval=true&serverTimezone=EST";
    static final String usernamestring = "root";
    static final String passwordstring = "toor";
    private static boolean notDone = true;

    private static Scanner sc = new Scanner(System.in);
```

3. Technical Specifications

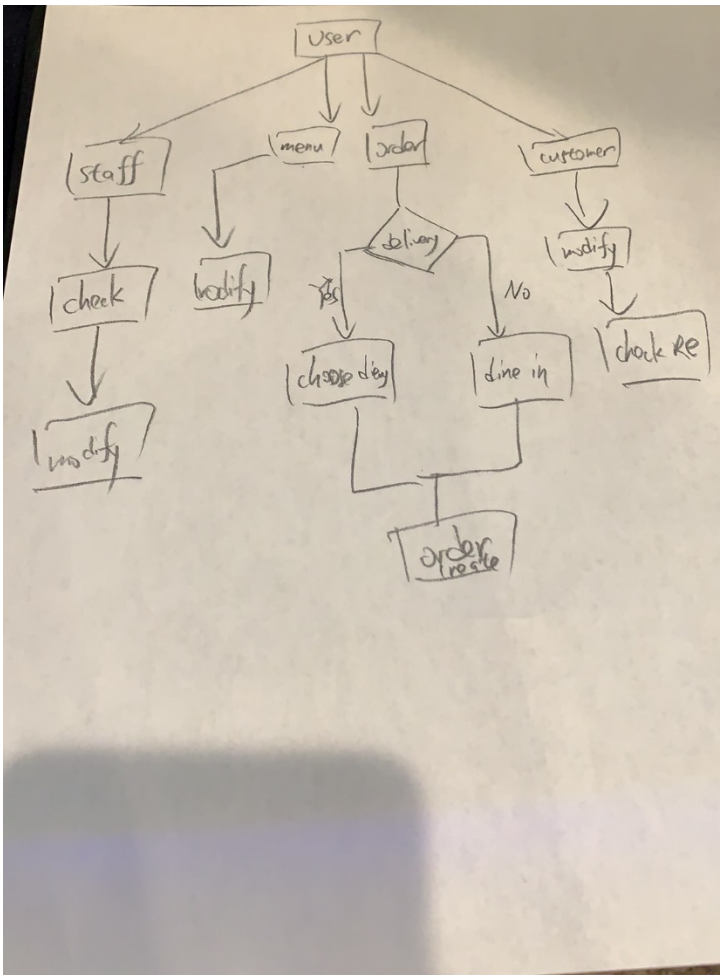
User can view all the staff member, modify any of them and their salary, schedule and proficiency of the cook.
The customer and reservation can also be modified.
The dish and its price can be modified.
The order part can be chosen by dine-in or take out, also can be add tip.

4. UML and EER




orders
order_number varchar(30) PK
tips double NULL
*:1
1:1
dine_in_orders
order_number varchar(30) PK FK
phone_number varchar(15) FK
1:1
contain
order_number varchar(30) PK FK
dish_name varchar(50) PK FK
quantity int NULL
delivery_orders
order_number varchar(30) PK FK
phone_number varchar(15)
sid int FK
pname varchar(30) NULL FK
delivery_fee double NULL
1:1
served_by
order_number varchar(30) PK FK
sid int PK FK
dish
dish_name varchar(50) PK
price double NULL
1:1
platform
pname varchar(30) PK
url varchar(100) NULL
customer
phone_number varchar(15) PK
canme varchar(40) NULL
address varchar(100) NULL
1:*
cooked_by
dish_name varchar(50) PK FK
sid int PK FK
reservation
rdate date PK
phone_number varchar(15) PK FK
timeslot time NULL
chef
sid int PK FK
proficiency int NULL
cooking_style varchar(12) NULL
<-HAS
staff
sid int PK
sname varchar(40) NULL
working_schdule varchar(12) NULL
salary double NULL
1:*
delivery_guy
sid int PK FK
delivery_method varchar(12) NULL
phone_number varchar(15) NULL
<-HAS
HAS->
waiter
sid int PK FK

5. User Flow


User
staff
menu
order
customer
check
modify
deliver
modify
Yes
No
check re
order

6. Lessons Learned

This is the first time I dive into the database design part, in previous working experience I only just user the built CRUD to access or modify the database. I leant the SQL language the connection of MySQL and Java&Python, the UMLmodel helps the design application a lot. Also the CAP theorem and timestamp algorithm help me understand more about the data structure and the how it react within the system.

What's more in data domain, I realized In a normalized data model, the reference domain is typically specified in a reference table.

I should create more connection between tables, this project after I finished I found out that I there were lots of function could be add-on.

There is the part of the code doesn't work:

I tried to use python to visualize the dish.

```
import csv
price_range = {"cheap": 0, "medium": 0, "expensive": 0}
with open('dish.csv', newline='') as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
        price = float(row['price'])
        if price < 20:
            price_range["cheap"] += 1
        elif price < 40:
            price_range["medium"] += 1
        else:
            price_range["expensive"] += 1

import numpy as np
import matplotlib.pyplot as plt

ranges = price_range.keys()
y_pos = np.arange(len(ranges))
number = price_range.values()

plt.bar(y_pos, number, align='center', alpha=0.5)
plt.xticks(y_pos, ranges)
plt.ylabel('Number of dishes of each range in the menu')
plt.xlabel('Price range')
plt.title('Number of dishes in different price ranges')

for i, v in enumerate(number):
    plt.text(i - 0.05, v + 0.1, str(v))

plt.show()
```

7. **Future work**

   As I mentioned in the intro I will try to polish it and give it to my favorite restaurant a try.

   There will be the cook assign part and order tracking function to be built in future.