CS5200 Final Project Report

1. README Section

To run this project, you should follow the steps below.

1. Extract the project file into your host environment, and you can see the architecture of this project. There are three file boxes inside the zip file.

Application Code:

- model.py – This file encapsulated basic read and write operation for database directly, and you can see that roll back is used when write operation failed.
- controller.py – The main business logic of this project, for each table, there is corresponding logic function, and for some functions, it uses the database procedure.
- view.py – this file is the entry of this project, and it display the system view via command line. Since this project simulates an application, you can see there are some dead loops for this file, so if you want to exit, you can use Ctrl+Z to end this project.
- setting.py – This file contains the config parameter of database connection, before you run this project, make sure you edit this file to connect the database.

SQL:

- dump.sql: This file is exported from my database; you can just load this file to your MySQL workbench to get the same schema. If you want to get the same database step by step, just run the below SQL file.
- create.sql: This file is for creating the database
- trigger_procedure.sql: This file defines the data programming projects in the MySQL database management system.
- insert.sql: This file contains the insert operation for this application.

Resource:

In this file box, you can see the resource of this project, containing conceptual design graph, user flow chart, video, and report.

2. Load the dump database file to your local database. You can import the dump SQL file to you MySQL workbench, or you can open this script and run it to get the same schema. Also, you can get the same schema after running other three sql files step by step.

3. Config the setting file for your database connection, and then run the view.py to start this project. The start of this project will tell you whether the connection is made.

2. Technical Specifications

In this project, we use the following technologies.

Backend (service): Python (need install tabulate module, pymysql module)

Front end: Command line

Database Management System: MySQL

Architecture: MCV design

For the reason to choose this technical specification, the reason shows below:

- This project is aimed to know deeply about database work, and students should write the procedure, trigger, or function in the MySQL rather than use ORM to operate this work.
- Python is easy to do the basic application program since its code is easy to read, so choosing python can focus more on Database operation, for example we can call MySQL procedure easily via python.
- Frond end part, not choosing GUI or website, since for the website, many of framework uses the ORM to hide the basic operation which is violating the initial purpose for this

project. For GUI, PYQT5 is good, but for my project, there are a lot of pages, so if using GUI to develop will cost too much unnecessary time.

- MVC design pattern, this is separate the complex code work into three parts. Model file just define the basic CRUD operations to database, view files define the page to interact with user, and controller file writes the specific logic for selected view and model section.

3. Conceptual design UML

After the feedback of the proposal, the design has been modified for right relationship, and there are many functions added in this project such as edit the personal information and make comment, not just only one placing an order function. The conceptual design of UML shows below.

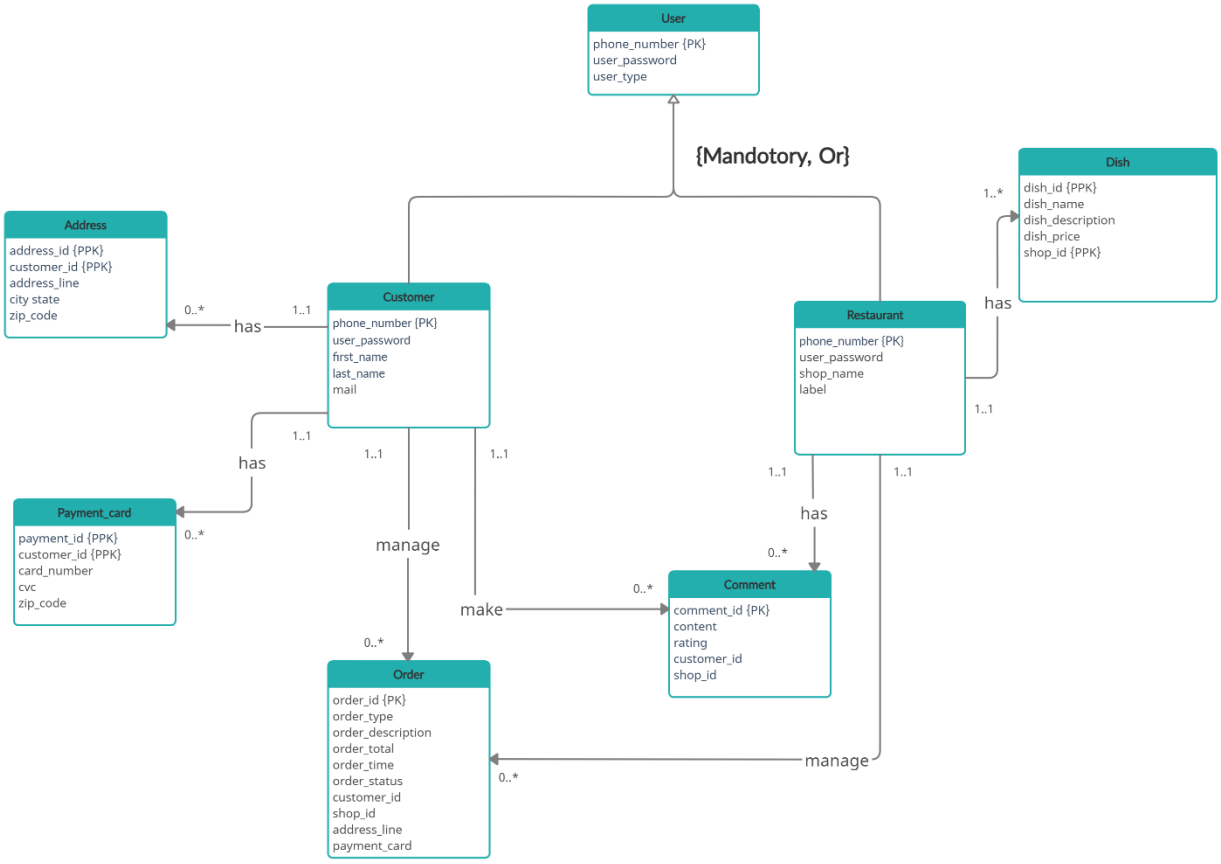

Figure 1: Conceptual design UML

4. Logical design

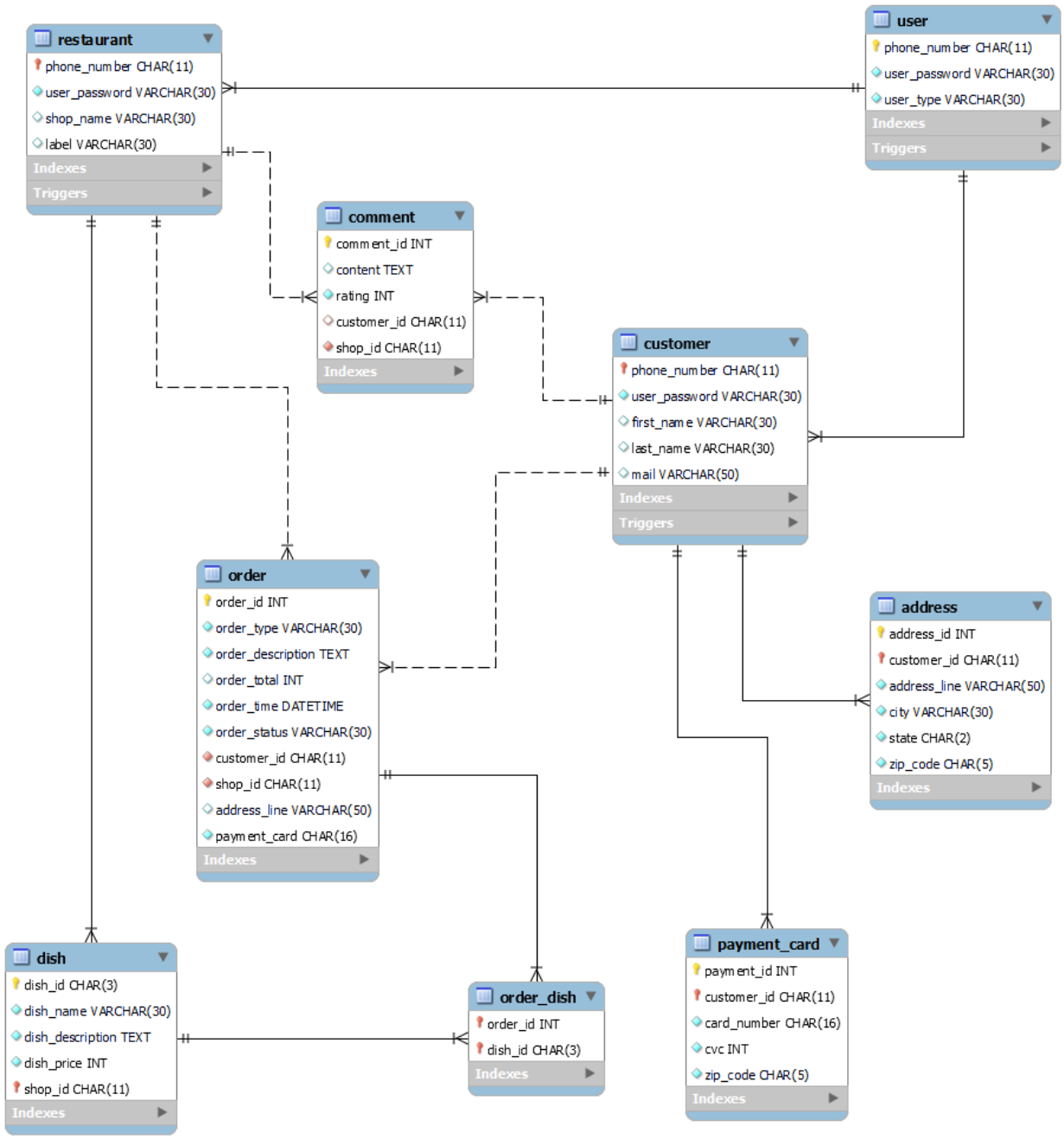

Figure 2: Logic design by Reverse engineer in MySQL workbench

5. User flow of the system

Here is the final user flow of this system, as you can see, there are customer view and restaurant view. This project is designed by different pages, each page has several options, choosing different options will go to the different page or flow.

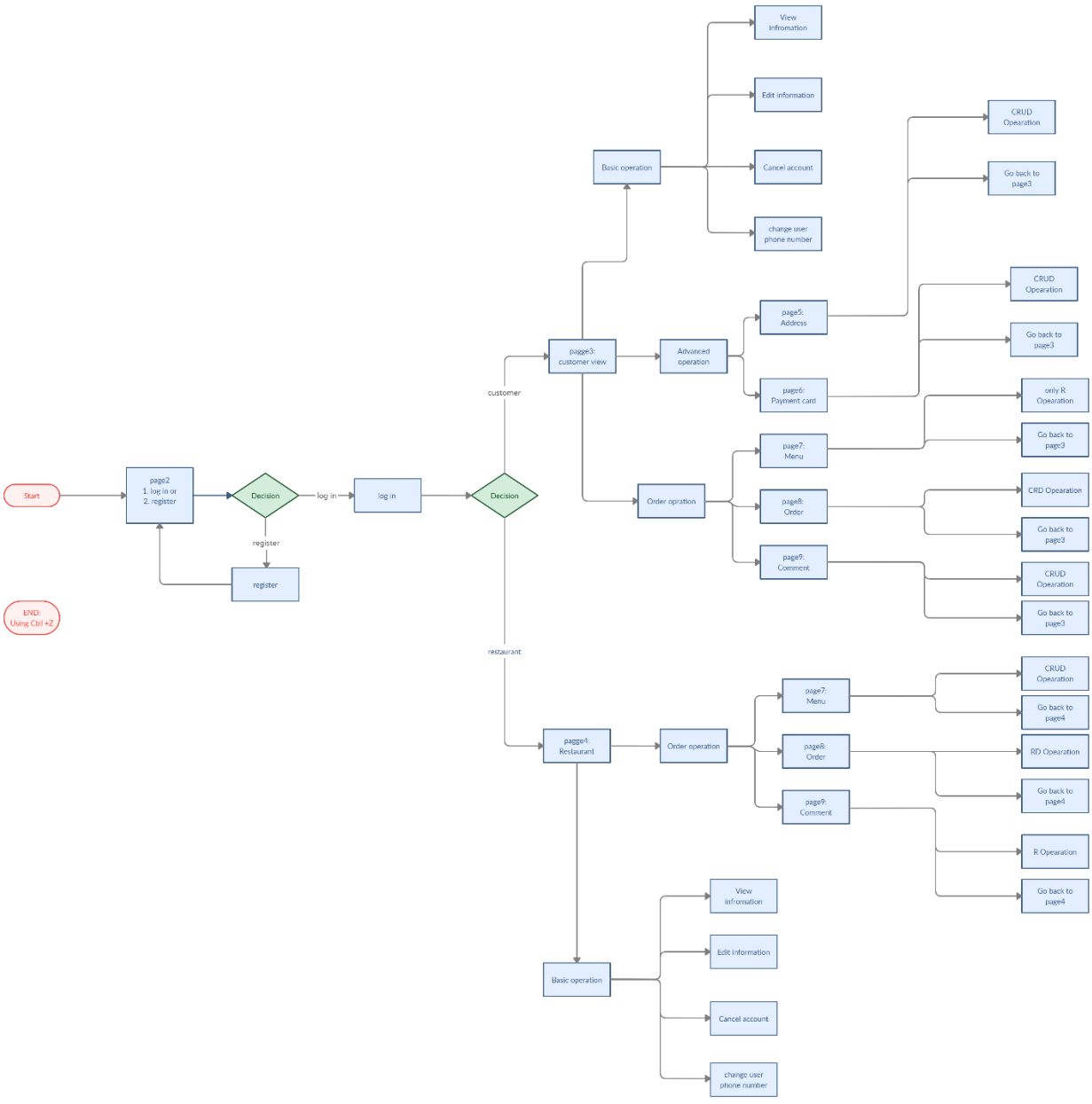

Figure 3: Final user flow chart

6. Lessons Learned

In this project, I have gained the Python programming skills to database, and MVC design pattern for the application design. The time exceed my expectation. Even if there are just CRUD operations and the database size is not large. Since for each table, we should handle the legal input and combine the crud operation to specific logic. Now I have recognized why there are framework for programmer, since it can reduce the time, also bring more safe and efficient work.

7. Future work

This database can be easily used for the canteen food order system. Also, since the customer information is very complete, it can be also used for similar type application, just modify some part is OK. However, there is some potential work, this application did not offer the delivery part even if I design the order type in the corresponding filed. In future, this application can be redone by framework and upload on the website.