Restaurant Dine-In and Reservation Management System

CS-5200 Final Project Report

Info:

Group Name: KoppoluSSenthilkumarK
Members of Group: Sameer Prasad Koppolu, Kailash Senthilkumar

README:

For MySQL:

- Use MySQL Workbench to import the Database Schema and the Database Dump. Run the SQL .sql file titled 'KoppoluSSenthilkumarK_dump'. This dump includes the creation of sample data - including a customer account and some placed orders.

For Application:

- Install the following Python Libraries: pymysql, pandas, datetime
  Run the command "pip install pymysql" in the terminal to install pymysql
  Run the command "pip install pandas" in the terminal to install pandas
  Run the command "pip install datetime" in the terminal to install datetime
- Preferably use VS Code as the IDE
- Run the application 'user_application.py' and you will first be prompted to connect to the Database using the appropriate login credentials to access the localhost server in MySQL in your system. Thereafter, you will again be prompted for a username - this is the username for a customer account. Since a first-time user of the application will not have a customer account, you will be prompted to create a customer account with a unique username.
- The application 'user_application.py' can be run within the IDE or via the terminal using the command 'python user_application.py'. In case you are running the application via the terminal, make sure you are in the same directory as that of the 'user_application.py' file.
- Documentation for pymysql: https://pymysql.readthedocs.io/en/latest/user/index.html
- Documentation for pandas: https://pandas.pydata.org/docs/
- Documentation for datetime: https://docs.python.org/3/library/datetime.html

Technical Specifications:

For persistence and storage, the project makes use of a relational database (SQL) on MySQL Workbench. The user application uses a Python programming language script. The application can be run and interacted with using a Command Line Interface.

Conceptual Design:

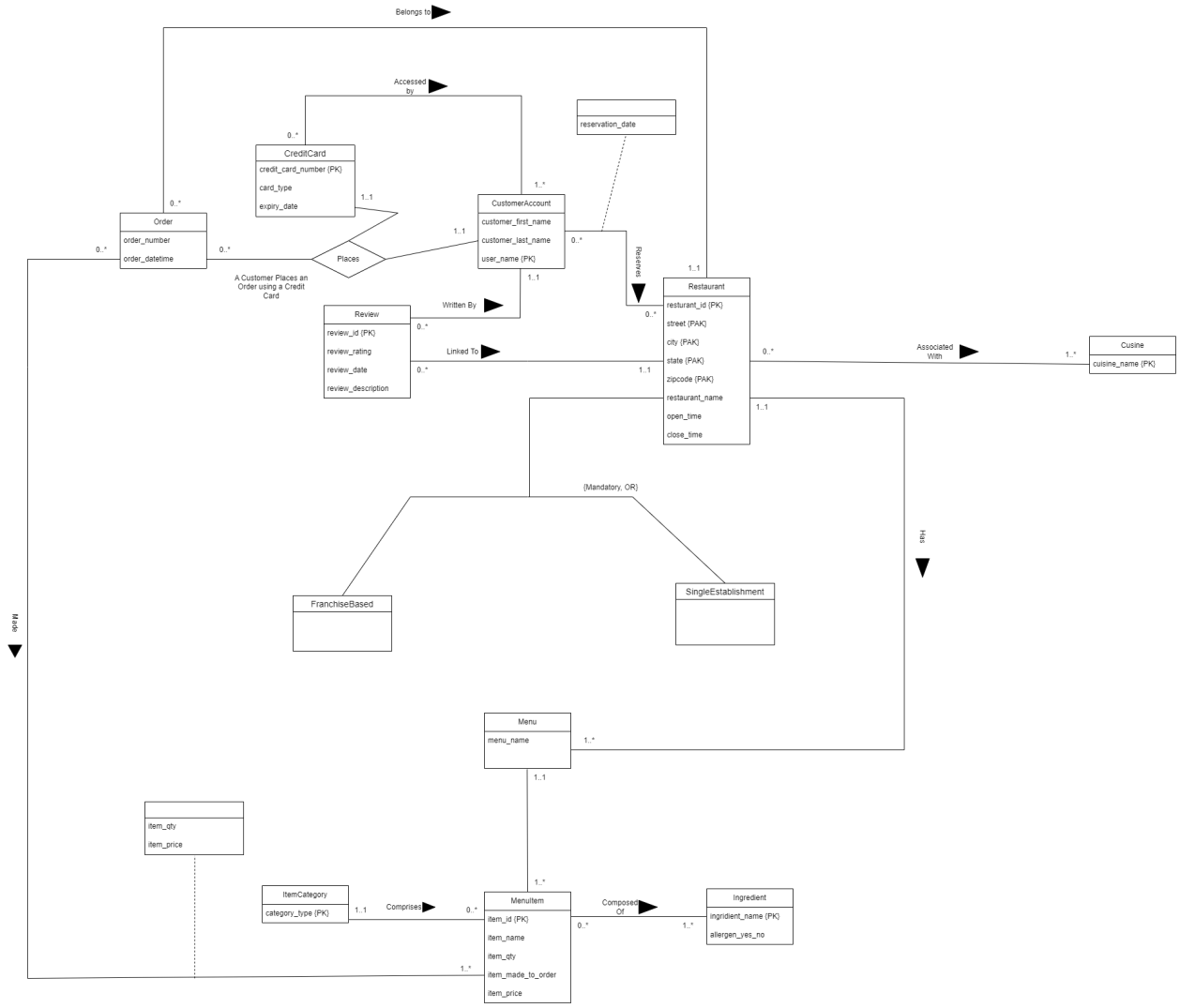

Figure 1: Conceptual Design Using UML Notation

Logical Design:

Attached as PDF with Submission. File is titled ‘logical_design.pdf’

User Flow:

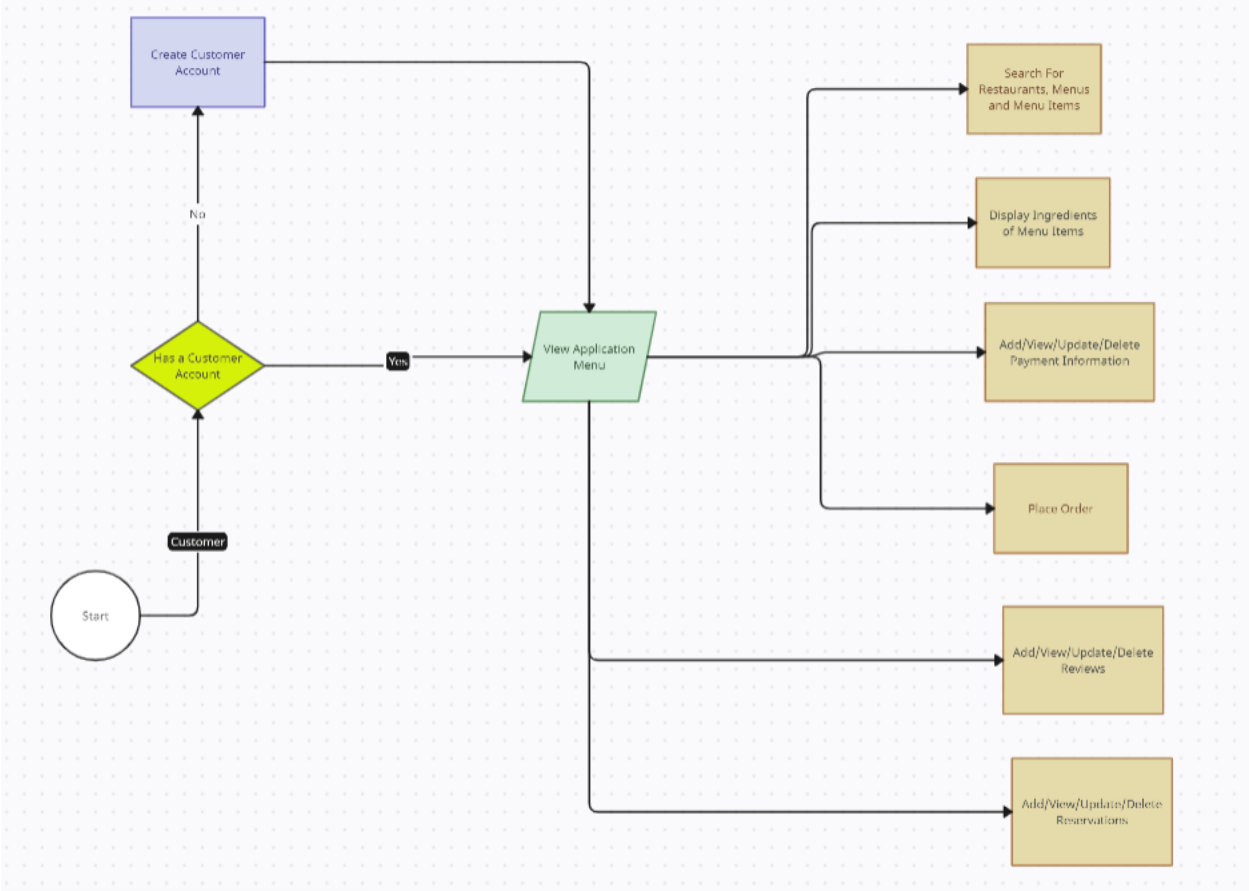

Figure 2: User Flow in the Application

- When the application is started and once the connection to the DB is achieved, the user will be prompted for a username. At this point, if a first-time user is interacting with the application, no username will be found and the user will have to create a customer account. If the user is not a first-time user, then he/she will be shown directly to the application menu.
- The Application Menu consists of 26 options that a user can pick. These encapsulate the CRUD operations that a user can perform. After every option is chosen and executed, the user will be prompted again to make a choice or exit the application
- A user can search for restaurants, view the menus of each restaurant, the respective cuisines of each restaurant, the menu items in each menu of each restaurant, and the ingredients in each menu item.

- The User can add payment information, update payment information, and delete payment information. The payment information includes the credit card details of the user.
- The User can also place orders for menu items from specific restaurants using their payment information
- The User can also create, view, update, and delete reviews for a restaurant. A user can only create a review for a restaurant if he/she has placed at least 1 order at that restaurant.
- The user can also create reservations, update and delete any upcoming reservations, and also view upcoming and past reservations.
- For each of the 26 options, the user will have to type in the information that they are being prompted for.
- A user also has the option to delete their customer account, and if they decide to do so, their account will be deleted and they will be exited from the application. The next time they open the application, they will be prompted for a username, and since they have no valid username, they will be asked to create one. Thereafter, the application menu with the 26 options will be displayed.

Future Scope:

Our project's future development includes the introduction of a loyalty program where customers accumulate redeemable points. Additionally, we will implement promotional codes and special offers, as is customary. We will establish a dedicated customer care team specifically tasked with addressing feedback and reviews provided by our patrons. Moreover, we plan to integrate delivery partner profiles, complete with a distinct review system that allows customers to rate their service. Based on these ratings, delivery partners may receive bonuses, augmenting their income. We also aim to design an interactive and user-friendly front-end interface that enhances customer interaction. We will be planning to build a front-end application for admins to manage the database system. However, the procedures that are required are already written; while building applications, they can be called as required. To manage the substantial volume of data generated, we intend to utilize the Hadoop architecture and facilitate real-time data streaming through Spark, supported by Kafka's messaging services.

Lessons Learned:

- **Technical expertise gained:**

Completing this project has provided us with extensive experience in writing SQL queries, as well as developing stored procedures and functions. To propel this project forward, we implemented Python code for the front-end applications, allowing us to interact seamlessly with the created database. Additionally, we have gained valuable insights into creating a database application tailored to specific requirements.

- **Insights, time management insights, data domain insights:**

The primary reason we chose this project stems from our shared experiences as international students keen to travel and explore new cuisines at top restaurants. We aimed to create a system that would serve as a one-stop application, providing comprehensive information about various restaurants and their offerings. This platform is designed to assist users like us in making informed decisions about dining venues, enabling them to easily make reservations and place orders.

- **Realized or contemplated alternative design / approaches to the project:**

Initially, we planned to incorporate functionality that would enable staff members to view and perform CRUD (Create, Read, Update, Delete) operations on the menu and restaurant tables. However, we decided that perfecting the customer side of the application was paramount before expanding to include more functionalities for the admin side. However, the procedures that are required for the admin are already defined; while building applications, they can be called as required.