

## CIS 611 Final Term Project

Spring 2018

100 Points

### *Task Specifications:*

A local pizza store wants you to develop a Java-based ordering system. The main goal of the system is to reduce the work volume on the cashiers, allows cashiers serve fewer customers in the store while taking phone orders, and gather contact information of the store customers in order to spread out coupons and daily specials.

Customers of the system can place orders through *Kiosks* (Terminal computer machines in store). Customers through the ordering system can browse and order anything on the menu (e.g., pizza slices, 10/14/16 whole pizza, garlic bread, breadsticks, salads, soft drinks, etc). A customer can have multiple items in the same order. A new customer is required to sign up for an account before placing an order, a customer account requires a customer to create user name and password, and other personal information such as full name, mail address, emails, phone number, etc. Returning customers are only required to login in order to place an order.

Once the customer has placed an order, the application must handle the order payments. Payment information is required, it includes payment amount and credit card information. A user is required to enter the credit card information, and this information is validated before the order is completed. For simplicity, if the payment information is valid, then the payment is considered successfully placed and the order is confirmed. Once the order is confirmed, a customer can print a receipt with the order number and customer full name. Customers should be able to view the status (placed, received, in process, completed) for their kiosk orders.

The customers can change their mind (cancel or change the order items) before the order has not been marked as received (or in process) by the Chefs. For example, a customer may cancel the whole order, or delete or add some items in the order. Canceling the order means that the order must be deleted from the system. If the order is deleted, then the associated payment information must be deleted, and if the order is updated, the amount field in the payment information should also be updated accordingly.

For this implementation, the database administrator (DBA) user is responsible to define a menu or modify the menu. As an example, DBA must be able to name the items and their prices. Chefs must be able to browse the orders and mark them as received/in process or completed when the orders are ready. For this project implementation, the delivery part is disregarded, where the delivery drivers take orders at customers' premises where they mark the orders as delivered.

This Java-based ordering system adheres to 3-tier client-server Architecture, where we have multiple clients interacting with a centralized application server connected to a database server. Having this software architecture implemented, the centralized application server handles orders coming from

clients such as Kiosks computers that are in the store, and it communicates with the centralized database to manipulate order data.

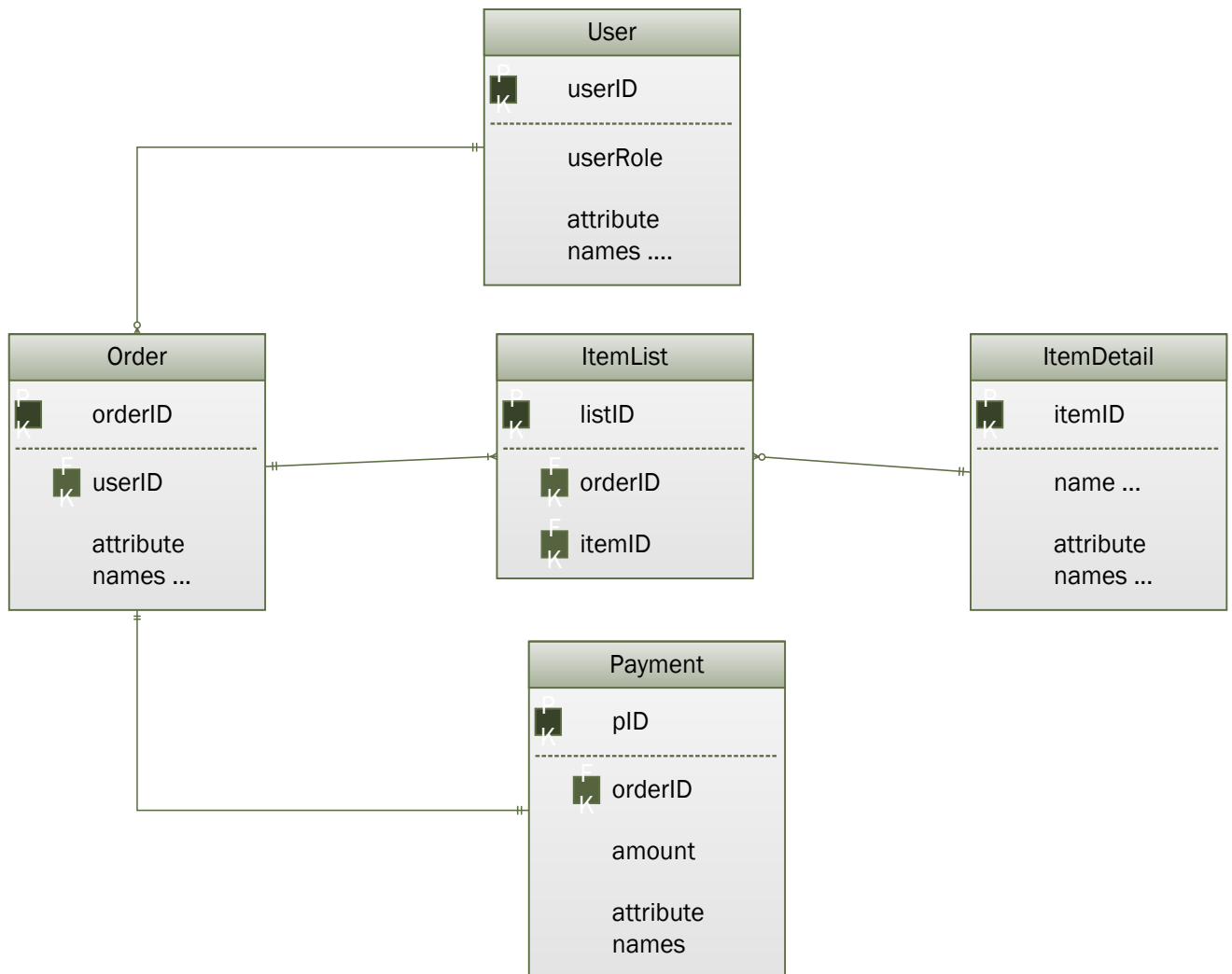
The implementation of this project must have a graphical user interface GUI that uses Java Swing. Basically, the GUI interface is for users to interact with the application. This graphical user interface must principally allow customers to place orders and chefs to take customers' orders. Additionally, a graphical user interface GUI for a system administrator (DBA) should also be provided in order to define, update, or delete store menus. Having that stated, there should be three different GUI frames for the three users' roles, customers, chefs, and system administrator. In the main login GUI, a user should choose the role of customer, chef, or admin in order to login to the system and its related user GUI frame should pop up in order to start interact with the system.

### ***Task Instructions:***

- You may use the client/server database project code example as a good resource.
- It is required for this project to create GUIs, use your own creativity to create the Client/Server GUIs, use some code from the examples and projects.
- For each group one group database will be created for you in COB's eLab. Each group must use the group database for this assignment, which will be created for you in the COB's eLab. Student must review the MYSQL DATABASE CONNECTIVITY documents posted on Canvas under the Resources page.
- Recommended to use elab for this assignment for both Java coding and MySQL. Instead of writing Java code in your personal machine, please write the Java code in Eclipse found in eLab for this assignment. If you do so, you will have both the Java code and the MySQL database inside eLab, that may reduce any connectivity issue that you might have faced, if you wrote the Java code in your personal machine and attempted to connect to MySQL database running inside elab.
- Use the following tutorial link for creating a new MySQL Connection in Workbench : <https://dev.mysql.com/doc/workbench/en/wb-getting-started-tutorial-create-connection.html> (Link to an external website.)
- Use the following tutorial link to create a new scheme model (your project database and tables) in Workbench: <https://dev.mysql.com/doc/workbench/en/wb-getting-started-tutorial-creating-a-model.html> (Link to an external website.)
- Spend some time to review the project submission and code review document in Canvas; the final project is a single attempt

## ***Entity-Relation Diagram (ERD)***

An intuitive attribute names are left to student perceptions.



## ***Evaluation Criterion:***

1. All tasks must be completed to receive complete credit for this project
2. The application should perform all the requirements correctly, including read user inputs from GUI, store data in a database, updating the user GUI information

3. The application should notify the user for improper inputs or empty text fields for validation
4. Users of the system should interact with a friendly graphical user interface, GUIs should be informative, simple, and easy to use
5. The program should view a correct data to the users
6. Users should be capable to create new accounts and perform log in operations
7. The application should not crash from improper input, accessing database to retrieve, delete or update data, or any other user's interactions
8. The application database should be created correctly
9. The client should be implemented to receive information from users and request services from the server side and populate the responses to the client in a user friendly format
10. The client side should only perform the interface operations including the data validation before passing the data to the server side of the application
11. The client side of the system should not have any business logic code such as a code to execute queries on the database server, all the business logic code should be implemented in the server side

## *Submissions & Review:*

Please review the class syllabus in Canvas for due dates and review policies

### **Java Final Project– Grading Rubric**

**Student Names:** \_\_\_\_\_

<b>Evaluation Criteria</b>	<b>Comments</b>	<b>Max Points Allowed</b>	<b>Points Earned</b>
Customers can browse the store menu		<b>5</b>	
A new customer can sign up for an account; a customer is required to create a user name and password		<b>5</b>	
Authentication, a customer must login before placing an order		<b>5</b>	
A customer can place an order, and an order can have multiple items		<b>5</b>	

Use of appropriate validation checks (e.g., required fields, correct data types, range checks) to ensure an error-free user experience		<b>5</b>	
Credit Card Payment validation		<b>5</b>	
A customer can print a receipt with the order number and customer full name. Displays the order details in a text area		<b>5</b>	
Customers should be able to view the status (placed, received, in process, completed) for their kiosk orders		<b>5</b>	
A customer can cancel or update (remove/add) the order items under the business rule that an order is not in process/completed. The update functionality must change the total order amount/price		<b>10</b>	
A DBA user must have access to add/update menu item information		<b>10</b>	
A chef user must be able to browse the orders update the status of the orders		<b>10</b>	
The application program must implement a 3-tier client-server Architecture, it implements request-response client server applications, with a server program interacting with a database server in order to provide services		<b>10</b>	
A client side a graphical user interface GUI for users to interact with the application. No business rules are implemented in the client programs. A client only requests services from a server program		<b>10</b>	

A database is created in a database server to store the application data, and data must be manipulated as a response to the SQL queries sent from the Java server program		<b>10</b>	
<b>Total</b>		<b>100</b>	

Total = \_\_\_\_\_/100