

PIZZA ORDERING SYSTEM

Final Report



Contents

[I. Introduction 2](#_Toc50748767)

[A. Problem Statement 2](#_Toc50748768)

[B. Proposal 2](#_Toc50748769)

[II. System Description 2](#_Toc50748770)

[III. System Requirements Specification 2](#_Toc50748771)

[A. Functional Requirements 2](#_Toc50748772)

[B. Non-functional Requirements 11](#_Toc50748773)

[IV. Use Case Diagram 11](#_Toc50748774)

[V. Class Diagram 12](#_Toc50748775)

[VI. Sequence Diagrams 12](#_Toc50748776)

[VII. State Diagram 21](#_Toc50748777)

[VIII. Activity Diagrams 22](#_Toc50748778)

[IX. Database Design 23](#_Toc50748779)

[A. ER Schema 23](#_Toc50748780)

[B. Table Schema 23](#_Toc50748781)

[X. Conclusion 23](#_Toc50748782)

[XI. Data Dictionary 24](#_Toc50748783)

# Introduction

## Problem Statement

Katie’s Pizza Store came to me with a problem that she needed help setting up a system to deliver a database with a user interface where customers can successfully order pizza online that the pizza store employees can access. Katie’s Pizza Store wants to increase sales by implementing this form of ordering and allow the customers to work more efficiently and quickly with the system. The customers need to be able to order pizza successfully from the system while collecting personal information, payment, and order information. The store employees need to be able to successfully receive the orders among other helpful tasks such as changing and tracing the order status and having access to view a customer’s information for verification.

## Proposal

We propose a user-friendly software system to allow the customers to successfully create and edit a customer account to order pizzas online, trace the status of an order, and view their order history. The software system will also allow the employees of the pizza store to successfully receive new orders, trace the status of an order, and search a customer’s profile in the software system.

# System Description

The system includes two subsystems: an online ordering subsystem and an order management subsystem. The online ordering subsystem allows customers to order pizzas online. The orders shall be kept in a database. The staffs of the store can then handle the orders with the order management subsystems. The status of an order should be kept in the database. The customers should be able to trace their orders online. Other considerations are welcome.

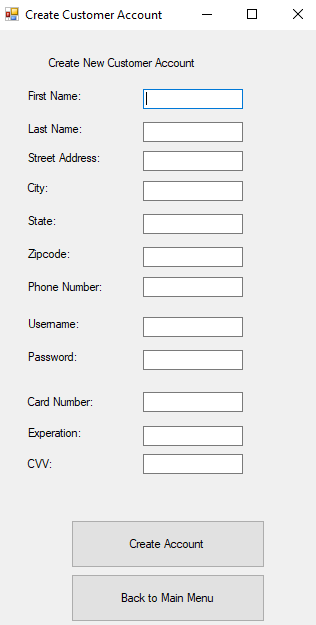
1. Create a customer account (Customer)
2. Change the profile of a customer (Customer)
3. Order pizzas online (including making a payment) (Customer)
4. Trace the status of an order (Customer/Pizza Store)
5. Change the status of an order (including changing the stock of materials and sauces) (Pizza Store)
6. Receive new orders (Pizza Store)
7. Search a customer’s profile (Pizza Store)
8. Allow customer to check their order history (Customer)

# System Requirements Specification

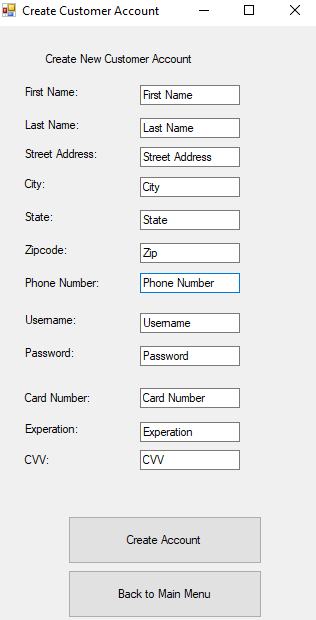
## Functional Requirements

For Customers:

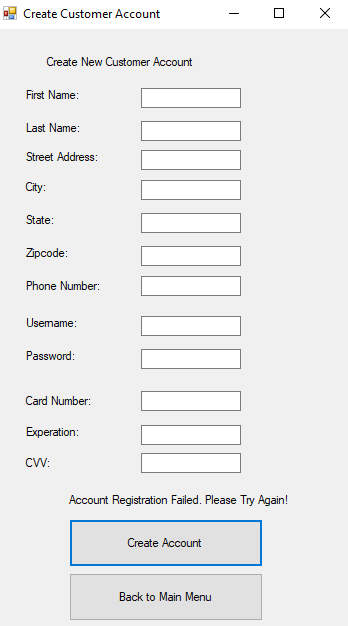
1. The system shall allow a customer to create a customer account with the system.
   1. The system shall allow a customer to enter their personal information into the system to create an account (name, address, phone, username, password, and credit card information).



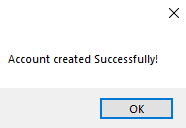
* 1. The customer shall press “Create Account” after entering information.



* 1. The system shall validate the information entered from the customer.
     1. If the username or password is incorrect, the system shall repeat 1.1.

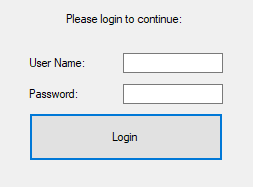


* + 1. If successful, the system shall confirm the information.

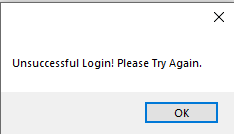


* 1. The system shall present return the customer back to the main menu.

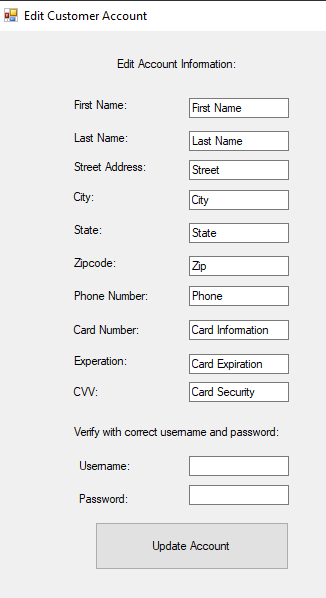
1. The system shall allow a customer to change the information in their customer account with the system.
   1. The system shall allow the customer to login to their account with their username and password and press continue.



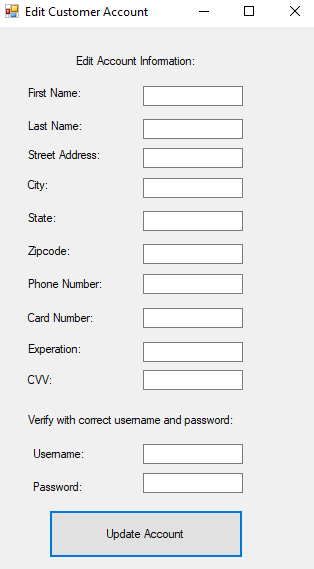
* 1. The system shall validate the information entered from the customer.
     1. If the customer is unsuccessful logging in due to incorrect username or password, the system will tell the customer the credentials are incorrect and repeat step 2.1.



* + 1. If the customer is successful, the system will present the customer with their current account information.



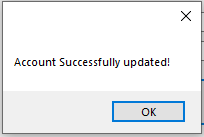
* 1. The system will allow the customer to edit customer account of the above information.



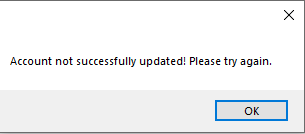
* 1. The customer shall press “Update Account” after providing new Information.



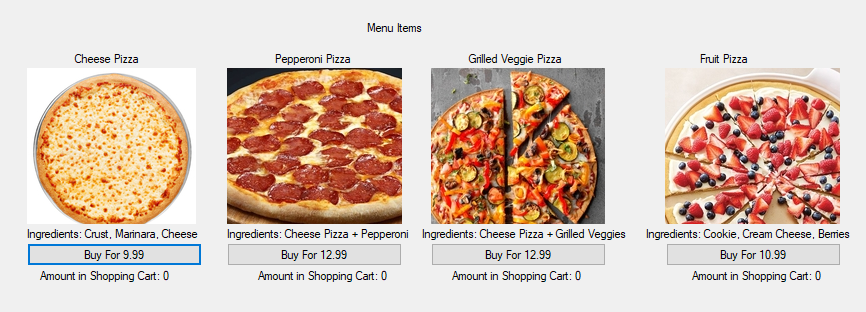
* + 1. If customer enters the password correctly, the system will confirm that the changes were updated and saved.



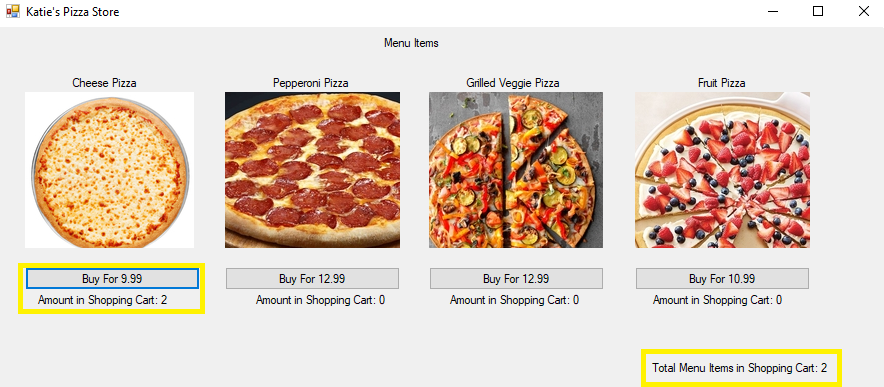
* + 1. If the customer enters the password incorrectly, the system will confirm that the account changes were not updated repeat 2.5.



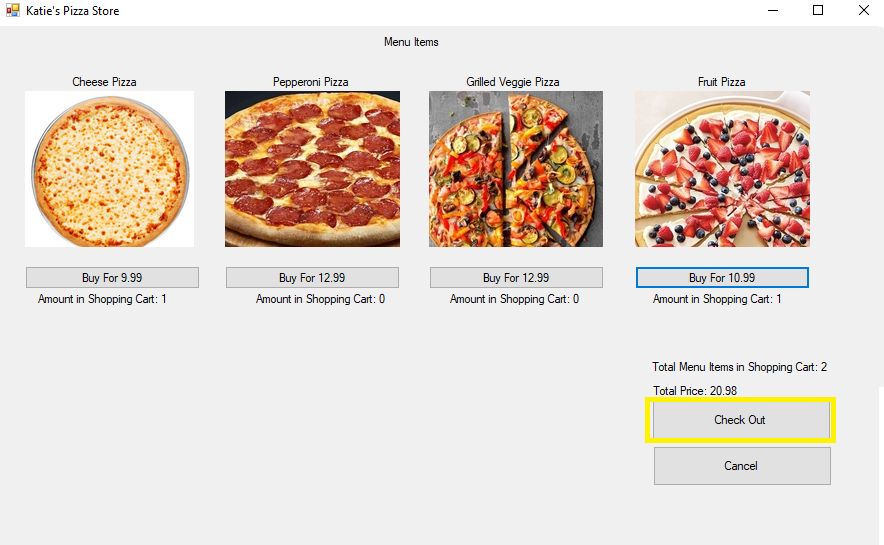
1. The system shall allow a customer to order pizzas with the system.
   1. The system shall display a menu of pizzas to the customer including a picture, name, description, and price.



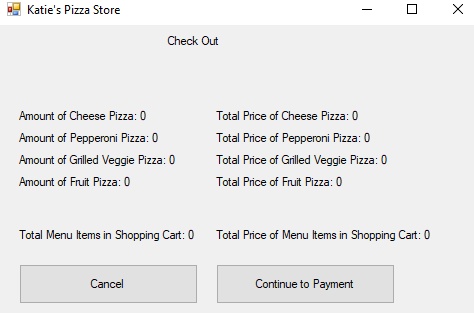
* 1. The customer shall press “Buy For” button besides each menu item to order a specific item from the menu. The item shall be added to a shopping cart.



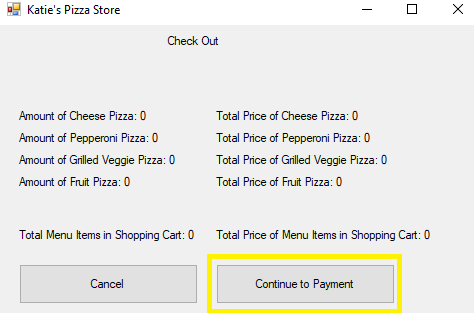
* 1. The customer shall repeat 3.2 to order as many items as he/she wants.
  2. The customer shall press “Check Out” button to confirm the order.



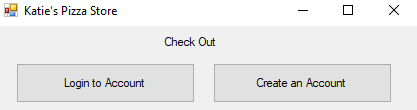
* 1. The system shall display the contents of the shopping cart.



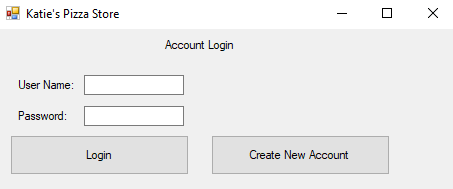
* 1. The customer shall confirm the order.



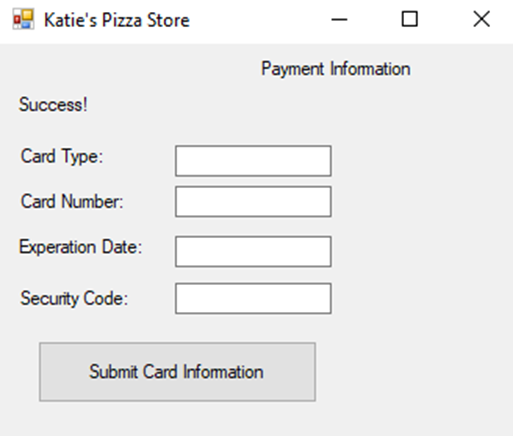
* 1. The system shall request the customer to login to his/her account or create a new one.



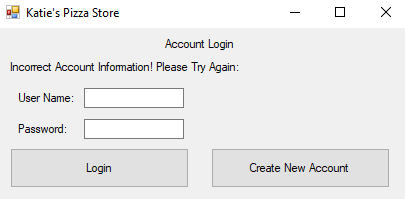
* 1. The customer shall login to the account if they have one.



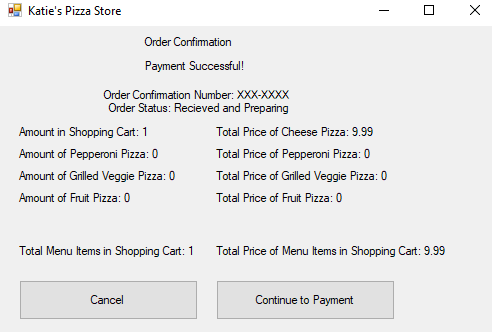
* 1. The system shall validate customer’s account with entered data (username + password).
     1. If the customer login successfully, the system shall request payment type.
        1. If payment selected is credit card the system shall request card information if no saved payment has already been entered (type + cardNumber + expDate + securityCode).



* + 1. If the customer login unsuccessfully, the system shall repeat 3.7.



* 1. The system shall request delivery or pick up option from the customer.
     1. The system shall request delivery address information from the customer if delivery is selected.
  2. The system shall charge the order fee from the card if card is selected.
  3. The system shall generate the order record (pizza info, order number, initialize the status of the order).



* + 1. If the customer selected to pick up the pizza the order record will also include the pick up time.

1. The system shall allow a customer to trace the status of an order with the system.
   1. The system shall allow the customer to enter their order number into the system and press continue.
   2. The system shall validate that the order number exists.
      1. If the order number exists the status will read as “Preparing”, “Out For Delivery”, “Successfully Delivered”, “Unsuccessfully Delivered”, or “Ready For Pickup”.
      2. If the order does not exist, the system will repeat 4.1.
2. The system shall allow a customer to check their order history with the system.
   1. The system shall allow the user to view their order numbers from their entire order history from the system.
   2. The system will display no orders if none have ever existed.
   3. The customer shall select an order number and the system shall allow the customer to view more information for each order number.
   4. The system will present the user with their entire order information for each order ever placed. This includes: Pizza Information, Order Number, Delivery Type, Payment Type.

For Employees:

1. The system shall allow an employee of the pizza store to login to the database with the system.
   1. The system shall allow an employee to login to an account with a username and password provided by the pizza store that every employee uses to access the database.
   2. The system will validate the employee’s credentials.
      1. If the employee enters the credentials incorrectly, the system will tell the employee it’s incorrect and 6.1.
      2. If the employee enters the credentials successfully, the system shall allow the employee to access the database.
   3. The system shall present a menu with actions and allow the employee to choose from the list of actions. This list of actions include: View Orders, Trace the Status of an Order, Change Order Status, Update Menu Item Stock, Search a Customer’s Profile.
      1. If the employee selects view all orders the system shall move to requirement R7 and perform those actions.
      2. If the employee selects trace the status of an order the system shall move to requirement R8 and perform those actions.
      3. If the employee selects change order status the system shall move to requirement R9 and perform those actions.
      4. If the employee selects update menu item stock the system shall move to requirement R9 and perform those actions.
      5. If the employee selects search a customer’s profile the system shall move to requirement R10 and perform those actions.
2. The system shall allow an employee of the pizza store to receive new orders with the system.
   1. The system shall allow an employee to view all current and previous orders from every customer in the system.
   2. The system shall present the newest orders received first and previous orders last.
   3. The system shall present the order number, customer name, status, and delivery type for each order in the list.
3. The system shall allow an employee of the pizza store to trace the status of an order with the system.
   1. The system shall allow the employee to enter an order number into the system.
   2. The system shall validate that the order number exists.
      1. If the order number exists the status will read as “Preparing”, “Out For Delivery”, “Successfully Delivered”, “Unsuccessfully Delivered”, or “Ready For Pickup”.
      2. If the order does not exist, the system will repeat 8.1.
4. The system shall allow an employee of the pizza store to change the status of an order and the quantity of stock of menu items with the system.
   1. The system shall allow an employee to change the status of a current order to “Preparing”.
   2. The system shall allow an employee to change the status of a current order to “Out For Delivery”.
   3. The system shall allow an employee to change the status of a current order to “Successfully Delivered”.
   4. The system shall allow an employee to change the status of a current order to “Unsuccessfully Delivered”.
   5. The system shall allow an employee to change the status of a current order to “Ready For Pickup”.
   6. The system shall allow an employee to update a description information with “Out of Stock” if the quantity of the item is 0.
5. The system shall allow an employee of the pizza store to search for a customer’s profile.
   1. The system shall allow an employee to enter a customer’s name.
   2. The system will validate if the customer exists in the system.
      1. If the customer does not exist, the system will repeat 10.1.
      2. If the customer exists, the system will present the customer’s entire account information to the employee.

## Non-functional Requirements

# Use Case Diagram



Figure . Use Case Diagram

This use case is to describe the services provided to the customers and the employees of the store:

1. Use case Create Customer Account: the system shall allow a customer to create an account for the customer from the system.
2. Use case Change Customer Profile: the system shall allow a customer to edit/change the customer profile from the system.
3. Use case Order Pizza: the system shall allow a customer to order pizzas from the system.
4. Use case Trace Order Status: the system shall allow both a customer and an employee to trace the order status of an order from the customer the system.
5. Use case Change Order Status: the system shall allow an employee to change the order status of the customer’s order from the system.
6. Use case Receive New Orders: the system shall allow an employee to receive new pizza orders from customers from the system.
7. Use case Search Customer Profile: the system shall allow an employee to search a profile from the system.
8. Use case Check Order History: the system shall allow a customer to check the order history of their orders from the system.

# Class Diagram



# Sequence Diagrams

# 

Figure . Sequence Diagram for Create Customer Account Use Case



Figure . Sequence Diagram for Change Customer Account Use Case



Figure . Sequence Diagram for Order Pizza Use Case



Figure . Sequence Diagram for Trace Order Status Use Case



Figure . Sequence Diagram for Change Order Status Use Case



Figure . Sequence Diagram for Receive New Orders Use Case



Figure . Sequence Diagram for Search Customer Profile Use Case



Figure .Sequence Diagram for Check Order History Use Case

# State Diagram



# Activity Diagrams



Figure . Activity Diagram for Use Case Order Pizza



Figure . Activity Diagram for Use Case Receive New Orders



Figure . Activity Diagram for Use Case Create Customer Account

# Database Design

## ER Schema

## Table Schema

# Conclusion

At this point, I have successfully completed both A (Problem Statement) and B (Proposal) of the Introduction. I have also successfully completed System Description, Part A (Functional Requirements) of System Requirements, Use Case Diagram, Domain Diagram, Conclusion and Data Dictionary. I have used the IBM Rational Rhapsody Modeler 7.5 to complete my Use Case and Domain Diagrams. I have also learned to use Visual Studio to complete the GUI Functional Requirements to implement my plan that I created in my proposal. I have also completed the diagrams for my design specification report.

# Data Dictionary

1. Card: contains the customers’ card information for payment.

2. Customer: a class that is an account for keeping customers’ information.

3. Menu Item: items that are displayed in the menu. Another name for menu item is pizza.

4. Order: contains the order information for the specific order placed by the customer.

5. Pizza: a class in the system, which represent special types of pizzas sold at the store.