Cunagin, Trevor

CSC 340  09/19/2020

Pizza ordering system

Contents

[I. Introduction 2](#_Toc51323627)

[A. Problem Statement 2](#_Toc51323628)

[B. Proposal 2](#_Toc51323629)

[II. System Description 2](#_Toc51323630)

[III. System Requirements Specification 2](#_Toc51323631)

[A. Functional Requirements 2](#_Toc51323632)

[B. Non-functional Requirements 3](#_Toc51323633)

[IV. Use Case Diagram 3](#_Toc51323634)

[V. Class Diagram 4](#_Toc51323635)

[VI. Sequence Diagrams 4](#_Toc51323636)

[VII. State Diagram 5](#_Toc51323637)

[VIII. Activity Diagrams 6](#_Toc51323638)

[IX. Database Design 7](#_Toc51323639)

[A. ER Schema 7](#_Toc51323640)

[B. Table Schema 7](#_Toc51323641)

[X. Conclusion 8](#_Toc51323642)

[XI. Data Dictionary 8](#_Toc51323643)

# Introduction

## Problem Statement

1. Customers should be able to easily order pizzas through an online ordering system. Ordering through a phone call can cause miscommunications in the order, can take a long time, and is more difficult for both the customer and employee. An online ordering system allows a customer to easily pick what they want and pay through the app.

## Proposal

1. We propose a software system to take orders from the customer and insert the order into the management database. It will also store the order status, customer accounts, and order history.

# 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 staff 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
2. Change the profile of a customer
3. Order pizzas online (including making a payment)
4. Trace the status of an order
5. Change the status of an order (including changing the stock of materials and sauces)
6. Assign a delivery staff to deliver an order
7. Search a customer’s profile
8. List the order history of a customer
9. List the business volume of a particular day or month
10. Edit a delivery staff’s profile

# System Requirements Specification

## Functional Requirements

## Functional Requirements

R1. The system shall allow a customer to order pizzas with the system.

* 1. The system shall display a menu to the customer.
  2. The customer shall press “Order Now” button beside each menu item to order a particular item from the menu. The item shall be added to a shopping cart.

A screenshot of a cell phone

Description automatically generated

Figure 1: Menu

* 1. The customer shall repeat 1.2 to order as many items as they want.
  2. The customer shall press “Check Out” button to continue the order.
  3. The system shall display the contents of the shopping cart.
  4. The customer shall confirm the order.
  5. The system shall request the customer to login to their account.
  6. The customer shall login to the account.
  7. The system shall validate customer’s account with entered data.
     1. If the customer logins successfully, the system shall request card information.
     2. If the customer logins unsuccessfully, the system shall repeat 1.7.
  8. The system shall charge the order fee from the card.
  9. The system shall generate the order record (pizza info, order number, initialize the status of the order).
  10. The system shall send the order number to the customer.

R2.

2.1 The system shall request card information from the customer.

2.2 The customer shall enter his/her card information.

2.3 The system shall display a menu to the customer.

2.4 The customer shall select what pizzas to order, including quantity.

## Non-functional Requirements

# Use Case Diagram

This diagram shows how the users of the ordering system will interact with it. It shows the actions that both customers and employees can perform.



# Class Diagram

This diagram illustrates the various classes involved in the ordering system as well as their relationships.



# Sequence Diagrams

This diagram illustrates the various functions involved in the ordering system and how they work in real time.



# State Diagram

This diagram showcases the various states the ordering system can undertake and their interactions.



# Activity Diagrams

This diagram shows the relationships between various user actions.



# Database Design

These are an illustration of the design of the database that works under the pizza ordering system.

## ER Schema



## Table Schema

Customer(ID, Address, Name, Password, Phone)

Pizza(Name, Price)

Order(Amount, Quantity, Status, Date)

Employee(Username, Password)

# Conclusion

This report proposes a simple approach for customers to be able to order pizzas through an online

ordering system. It uses a flexible framework that can be utilized by numerous businesses, both those

that sell pizzas and other items. It is designed to be easy for the user to navigate, offering many

convenient solutions for the user to seamlessly order what they want. It will provide a more elegant

solution for restaurant ordering compared to prior methods.

# Data Dictionary

1. Menu item: items that are displayed in the menu.

2. Pizza: a class in the system, which represents types of pizzas sold in the store.

3. Cart: a class in the system that represents the pizza orders the customer has entered

4. Order: a class in the system that corresponds to the pizzas the customer wants, sent to the cart

5. Card: a class that represents the payment info of the customer

6. Customer: a class that contains the information for the person ordering pizza

7. Menu: a class that contains all the pizzas sold in the store