Requirements Document

Team 6

Surafel Assefa, Gashaw Amalto, Steven Crowther, Kane Baldwin

**Definitions**

Functional Requirements:

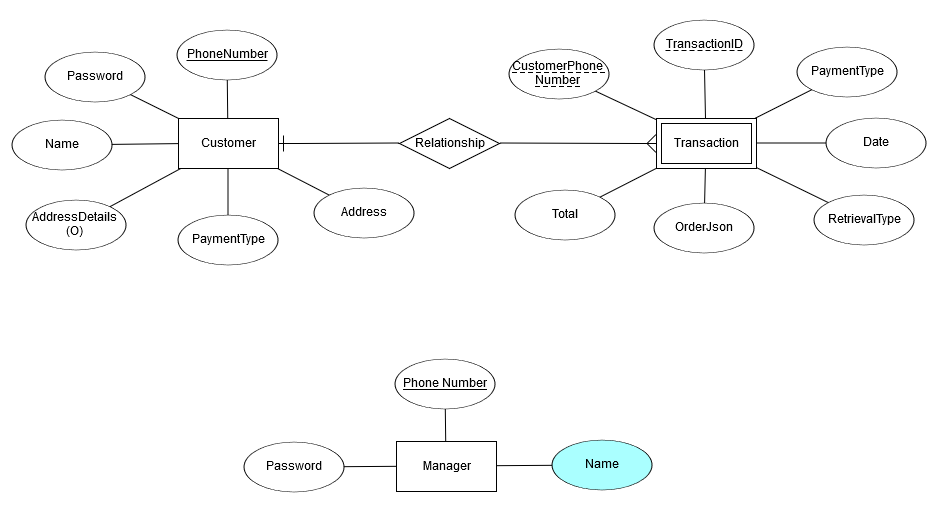
* Should take information given by the user and send it to the pizzeria as an order.
* Should track order, payment, and user between multiple website pages.
* The database should keep each user’s data stored to be accessed by the correct phone number and password.
* Any user or employee should have access to the full menu.
* Only an employee or logged in user should be able to create an order.
* Payment method should be addendable to a user’s profile.
* Online credit card payment should be usable to order a meal.
* Only managers should have access to a restaurant’s full order history.
* Only individual users should have access to their own order history.
* The website should allow the user to access the menu page.
* The website should allow the user to access and create an account.
* The website should allow the user to select payment type from a payment type page.
* The website should allow the user to view the order detail page.
* The website should allow the user to print the receipt page.
* The website should email the user regarding the details of their order.
* The website should only allow the manager to view the transaction page.

Non-Functional Requirements:

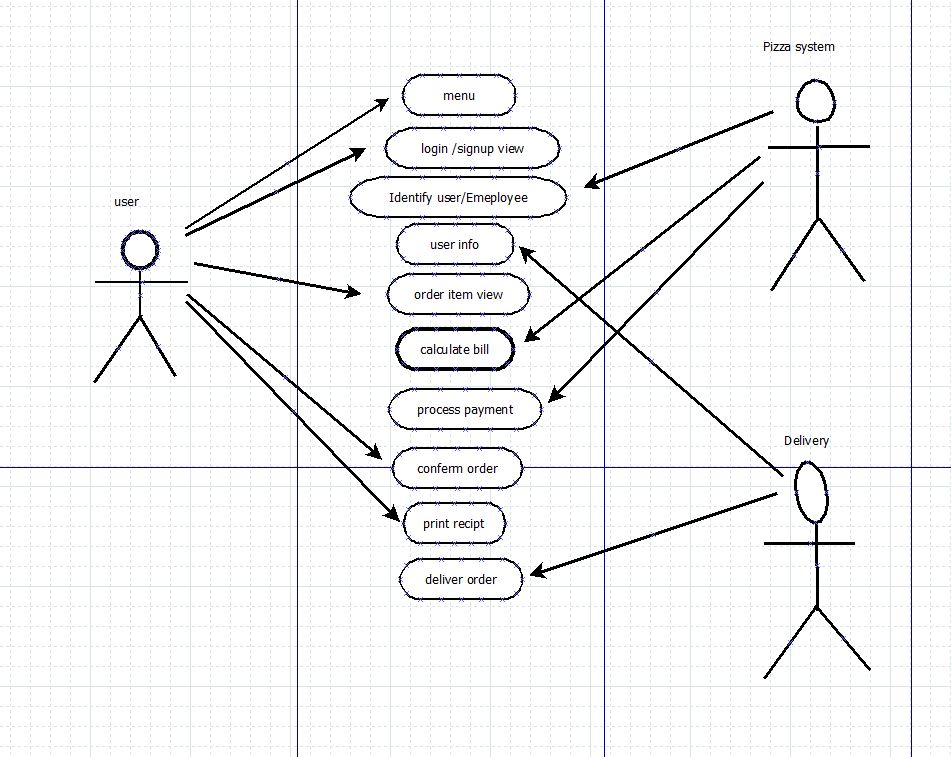
* The website should work on any operating system.
* The website should work on any mobile device.
* The website should be able to process a request at any time of the day.
* The website should send the order to the restaurant as quickly as possible.
* There should be minimal load time between website pages.
* Adding additional toppings, pizzas, or drinks to the website by the business should be simple.
* The website should be easy to navigate.

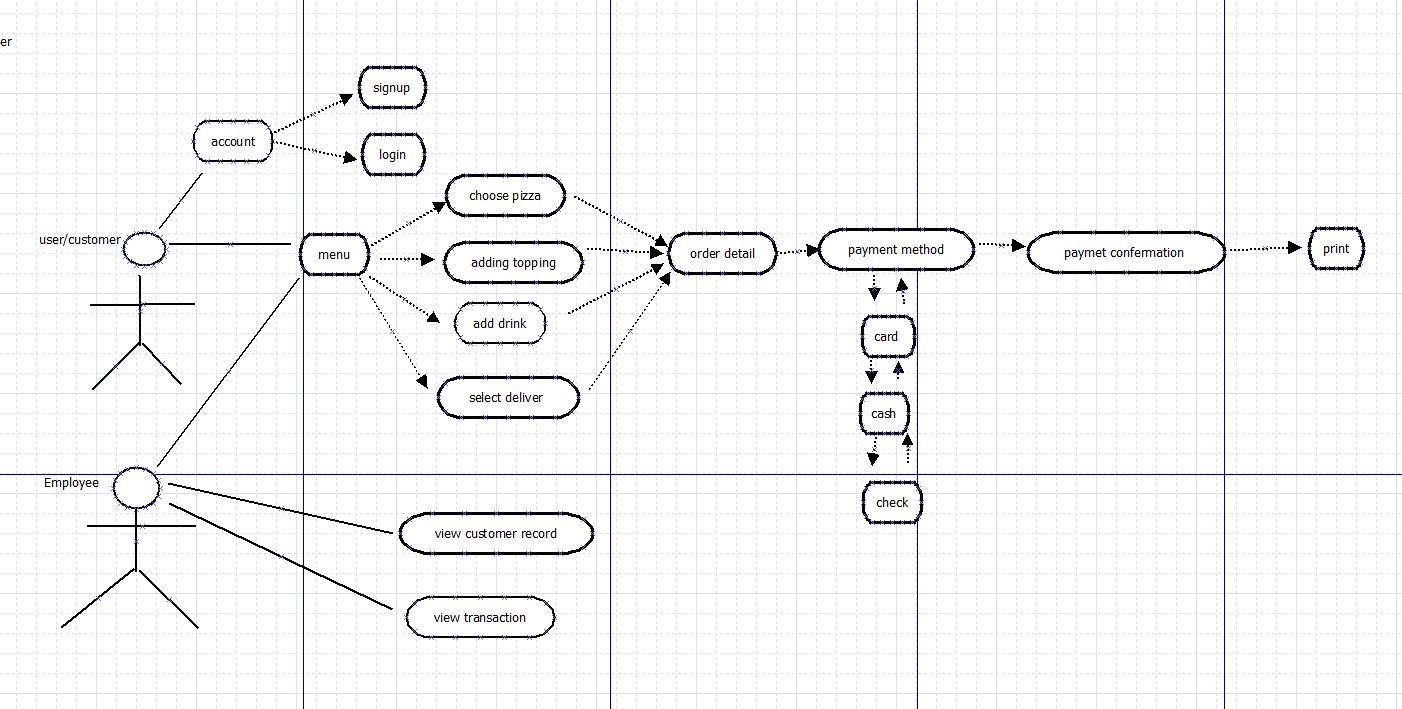
**Specifications**

E-R Diagram:

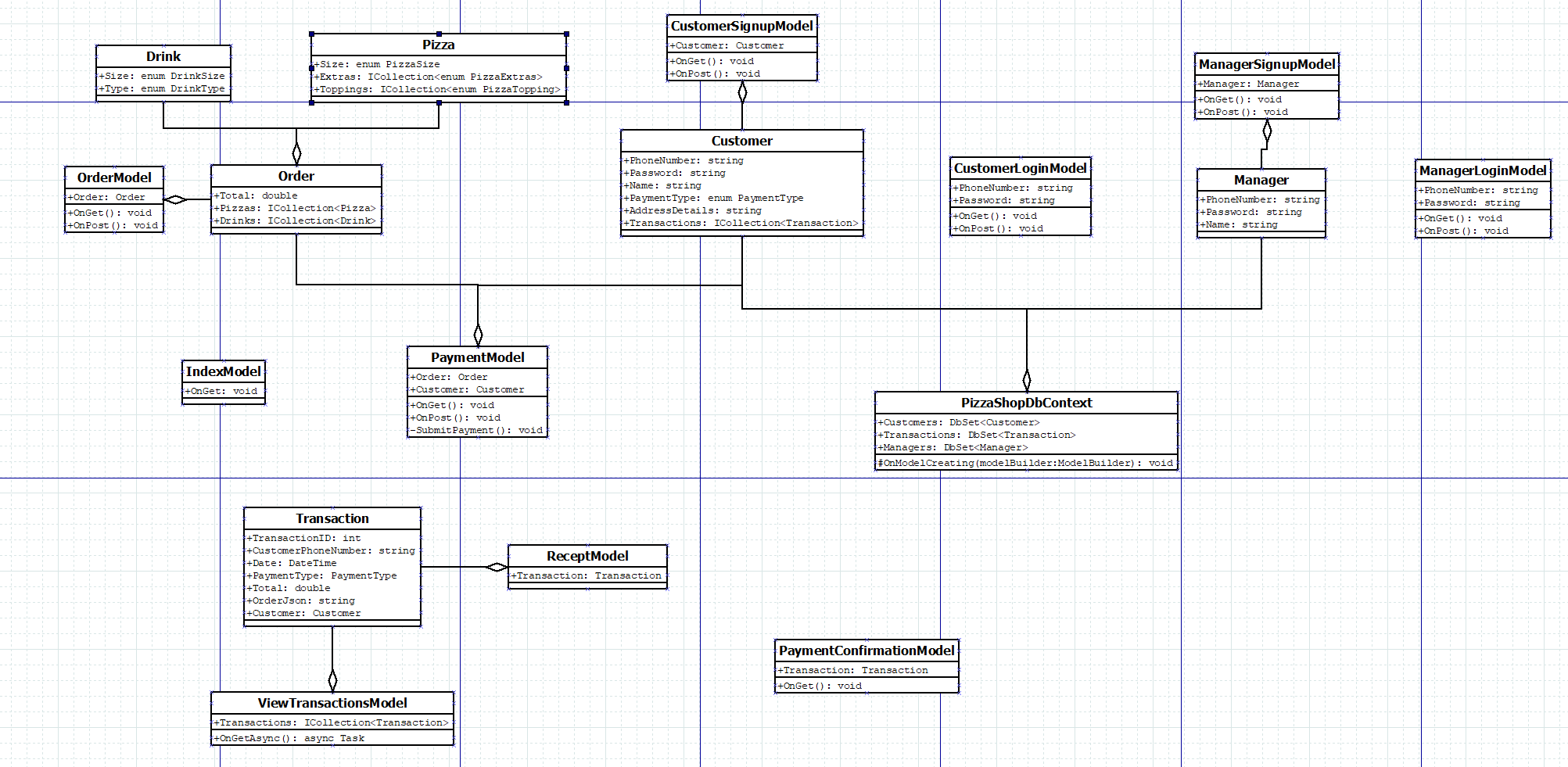


Use Case Diagram:

Use Case Flow Diagram:



Class Diagram:



Class Documentation:

The models in this class diagram are backends for different pages. Therefore, there aren’t many members and methods for these model classes since ASP.NET handles most operations for us.

Pizza - A class representing a pizza and it’s attributes.

Drink - A class representing a drink and it’s attributes.

Order - A class to store a collection of pizzas and drinks, as well as the total for the order.

Customer - A class representing a customer and it’s attributes.

IndexModel - Model for the Index page.

ManagerSignupModel - Model for registering a manager page. Has a Manager class member to fill out information for the manager, which will be added to the database when OnPost is called.

ManagerLoginModel - Model for the manager login page. Has the phone number and password string members for querying the manager in the database.

CustomerSignupModel - Model for the customer signup page. Has a Customerclass member to fill out information for the customer, which will be added to the database when OnPost is called.

CustomerLoginModel - Model for the customer login page. Has the phone number and password string members for querying the customer in the database

PaymentModel - Model for the payment page.

PizzaShopDbContext - A class that maintains the database context for the application

Transaction - Tracks number of and type of transactions enacted by customers

ReceiptModel - Uses Transaction to create a receipt for printing

ViewTransactionsModel - Allows viewing of Transactions

PaymentConfirmationModel - Confirms the user can pay with their given system

State Transition Diagram:

