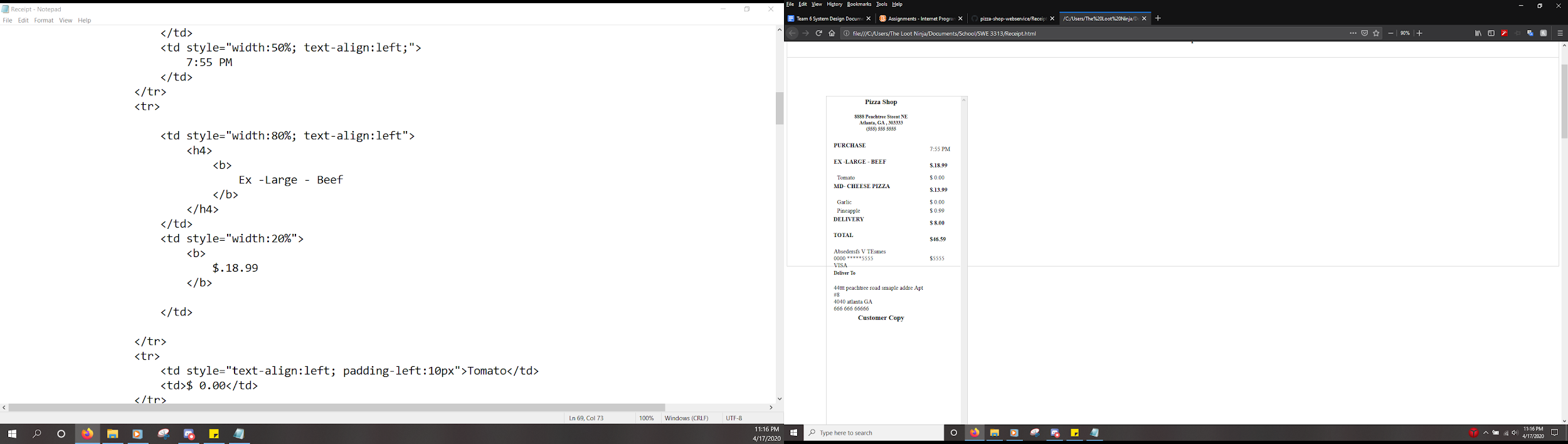
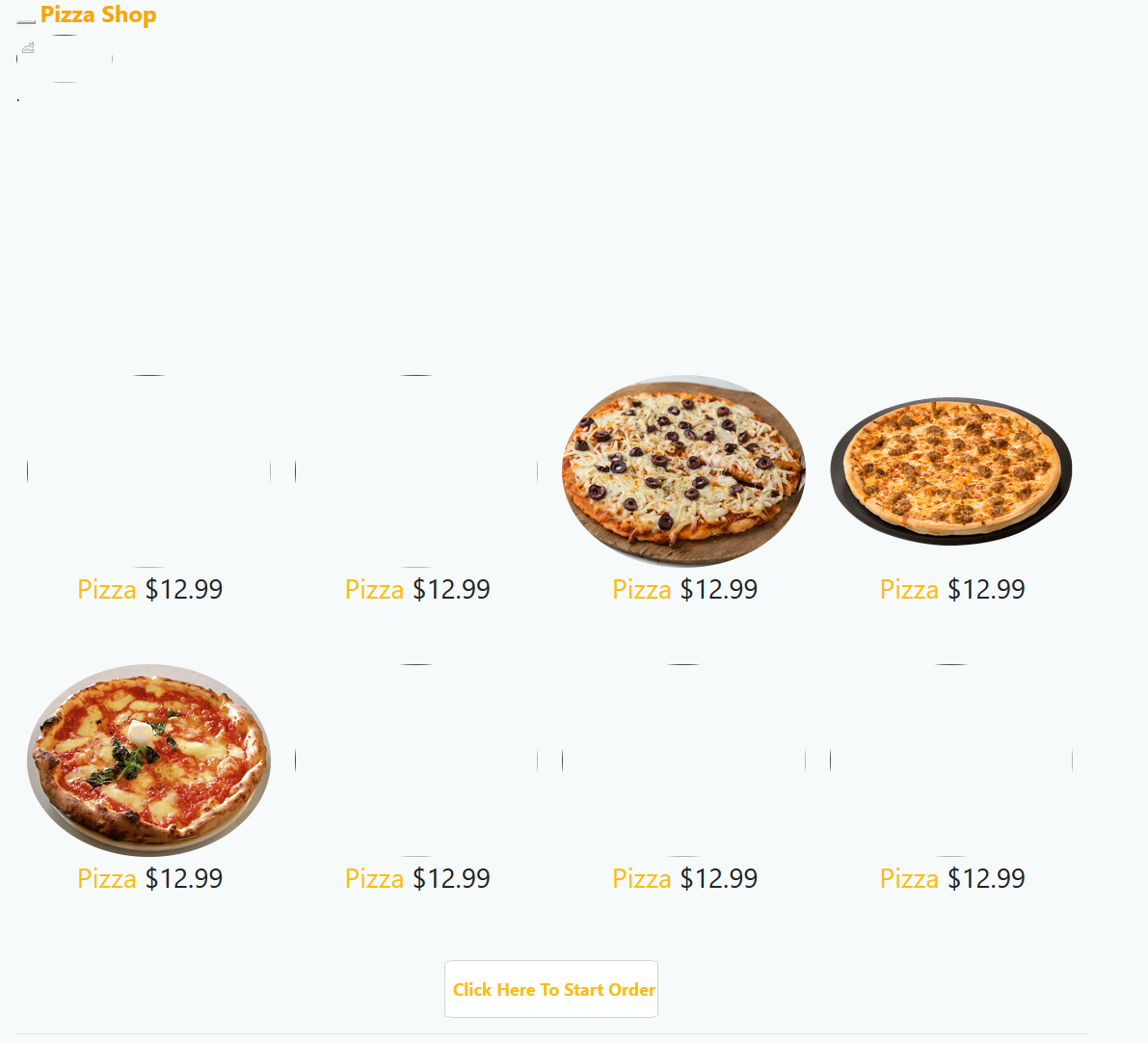
**Team 6 System Design Documents**

By: Surafel Assefa, Gashaw Amalto, Steven Crowther, and Kane Baldwin

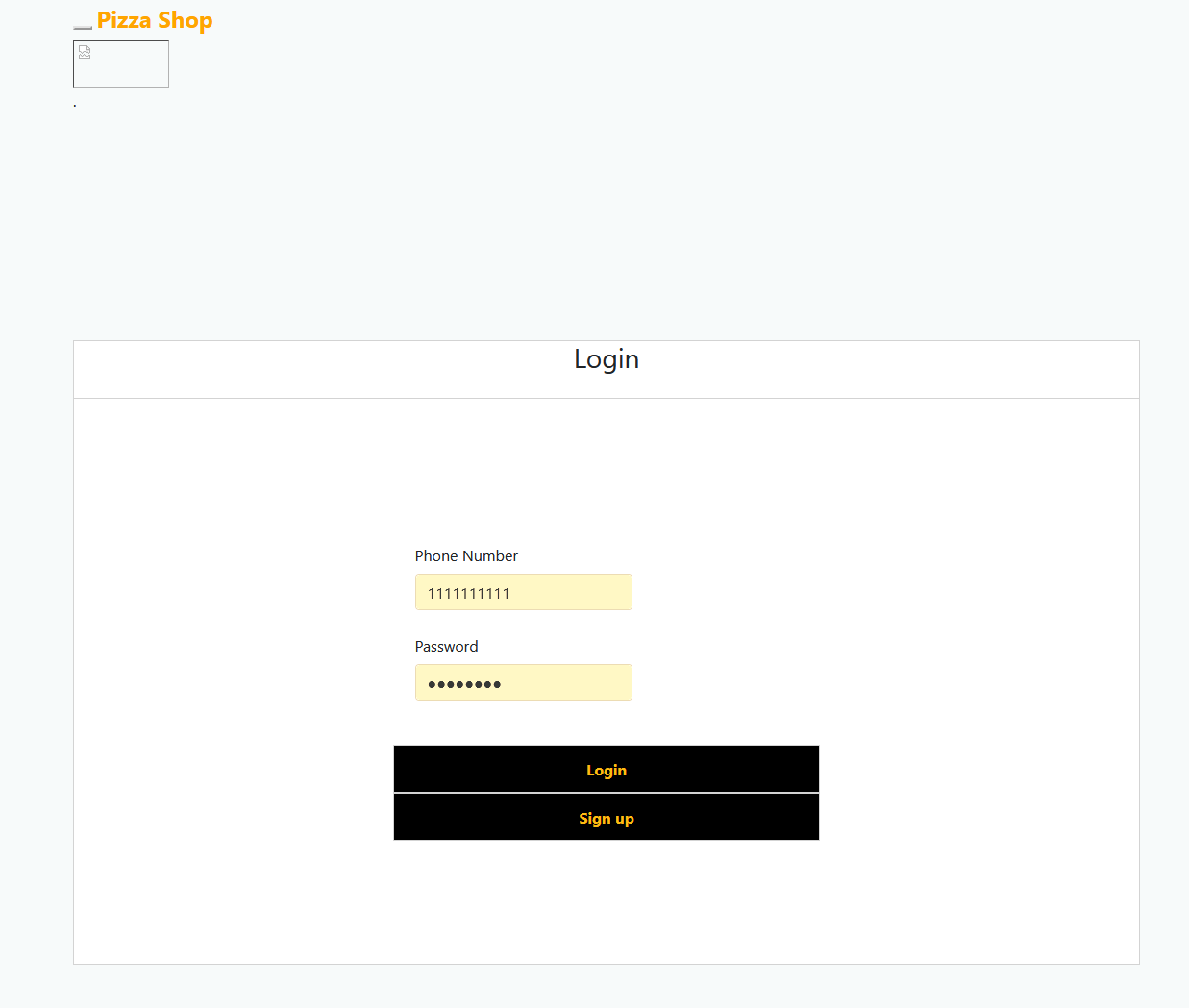
**Receipt Screenshot**



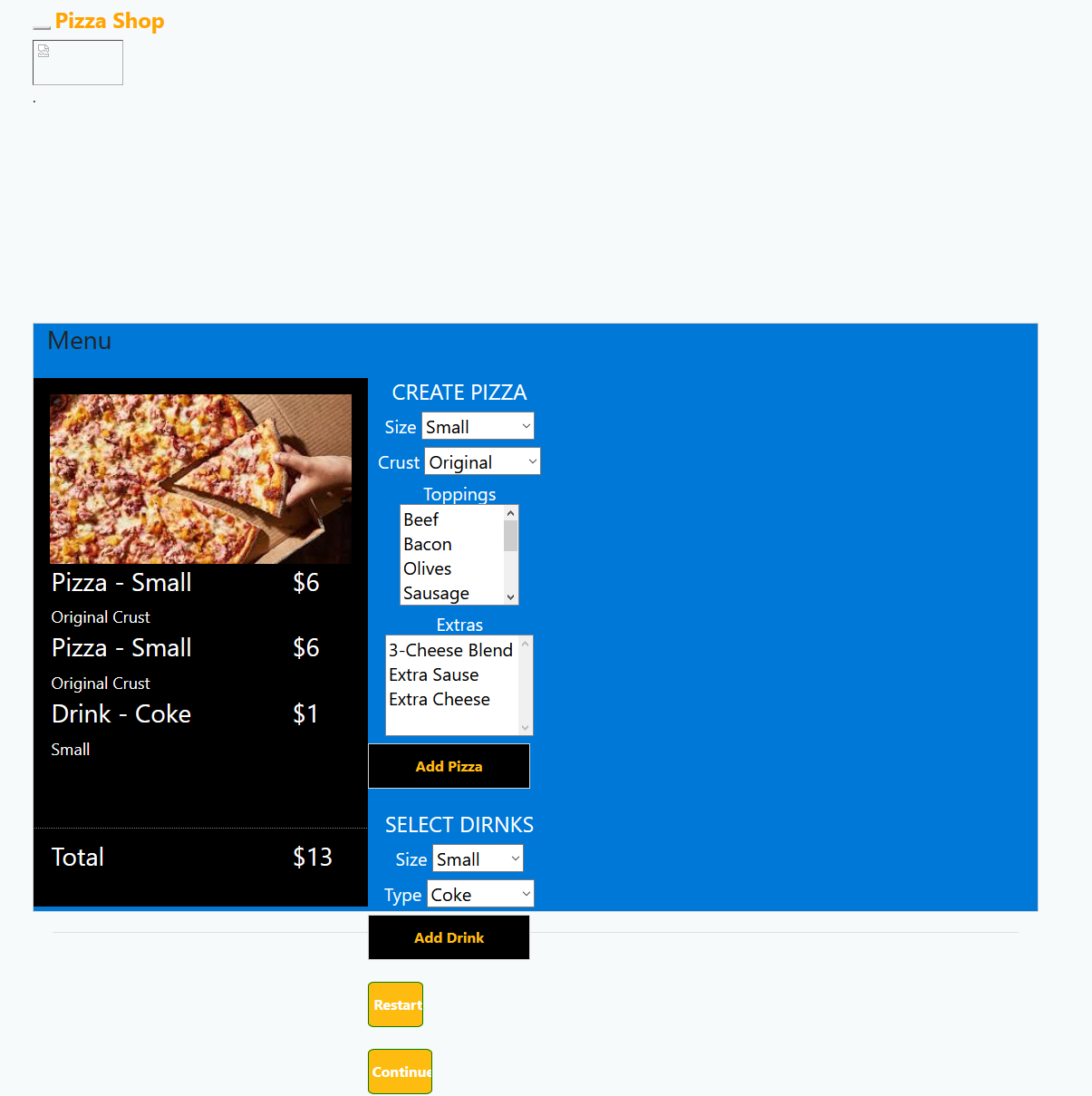
**Conceptual System Design Screenshots**



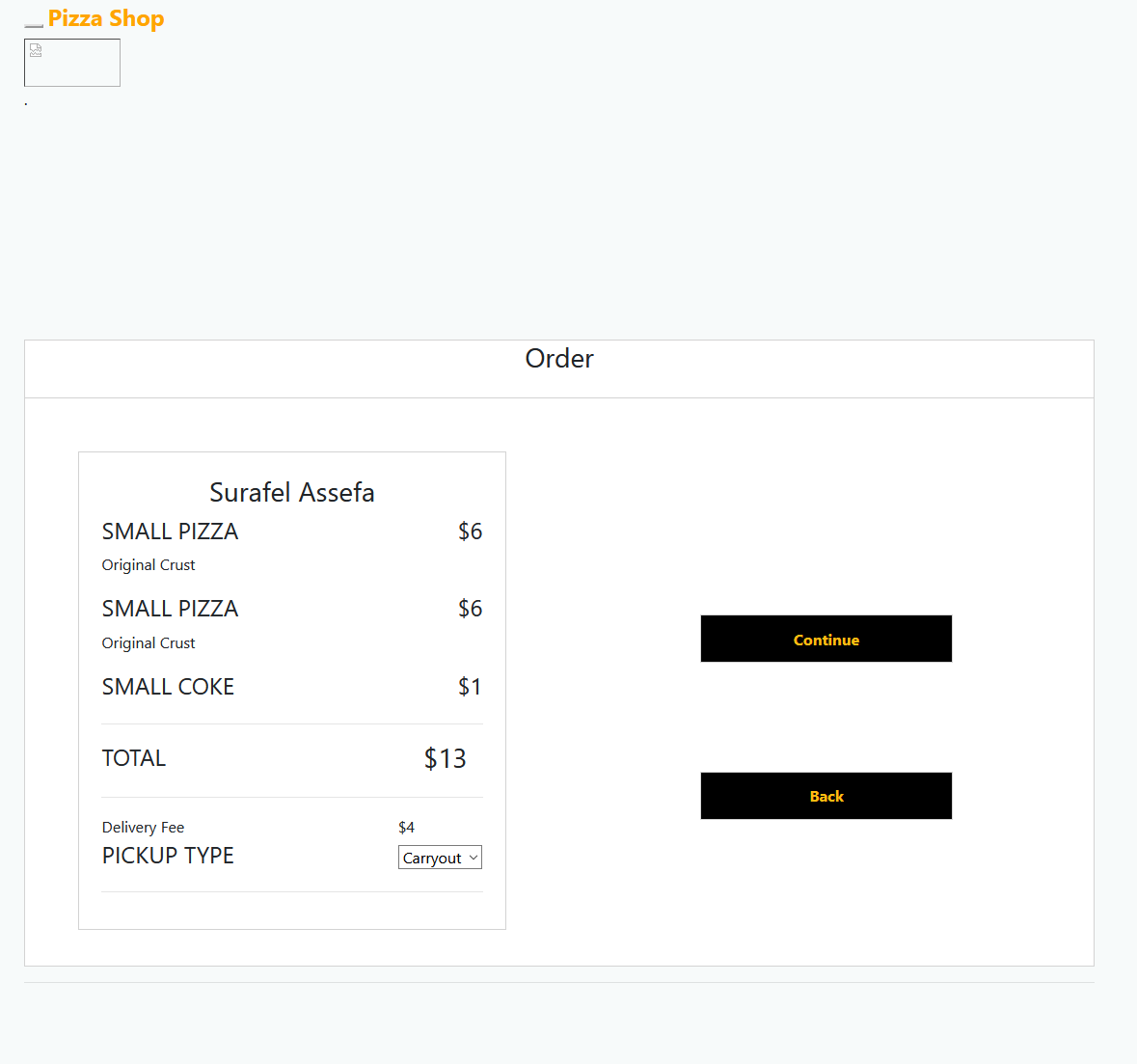
This main page for the Pizza Shop will be used by anybody visiting the website. They may click a pizza listed to start with it as a default order after logging in, or click the button at the bottom labelled “Click Here To Start Order” to start with an empty order.

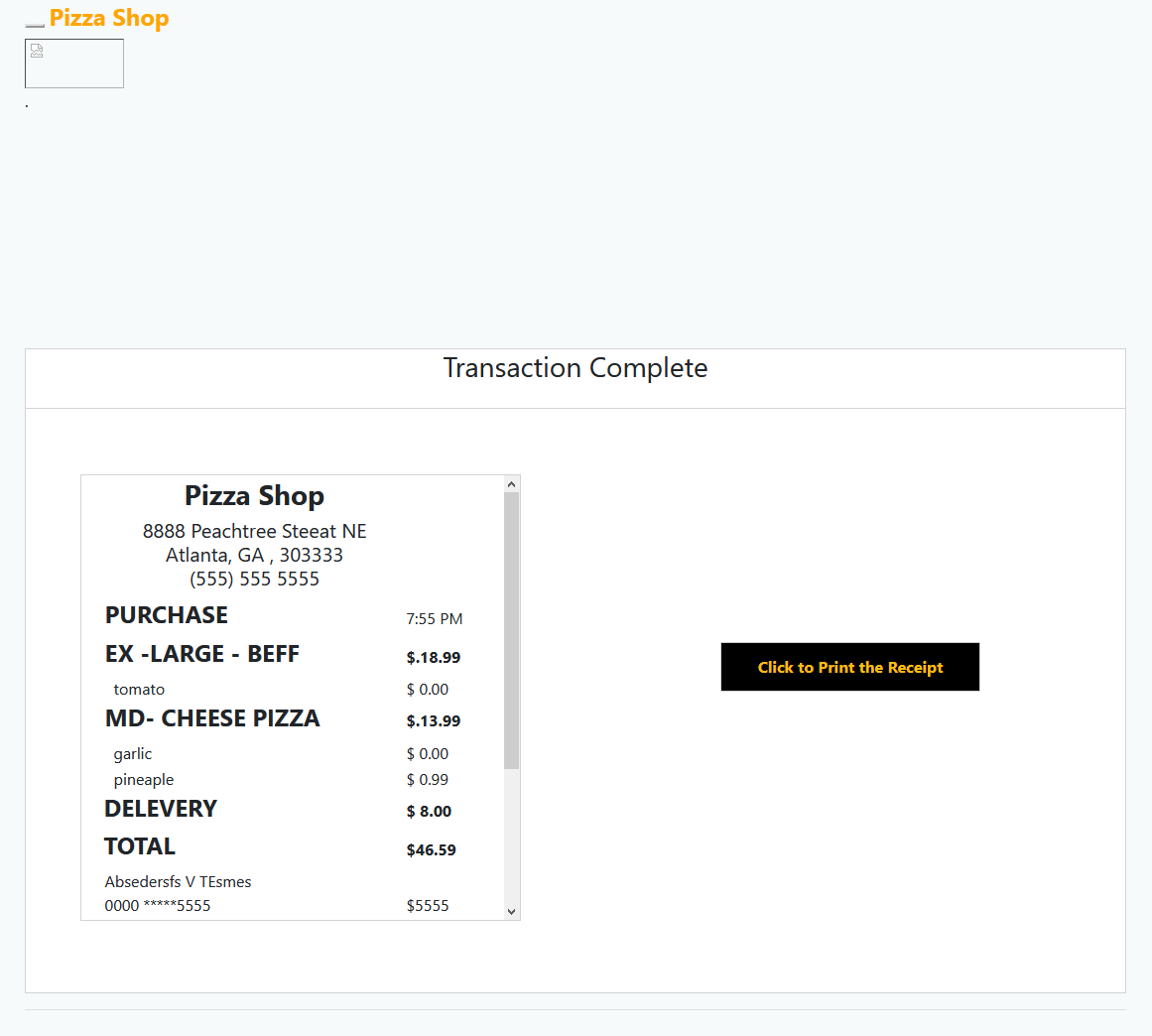


This page is for people to login to the website. For newcomers, they would click the sign up button to create a new account, and previous users would enter their Phone Number and Password. They will then be brought the pizza ordering page.



This is the menu screen. If you try to access this screen without logging in, it will throw you an error. Here you can select the options that you want for your pizza and drink. Merely select from the drop down menus, or control+click on the options menu for the pizza toppings and extras. Once you have a pizza or a drink configured, click the add pizza/add drink button respectively to add the item to your order.

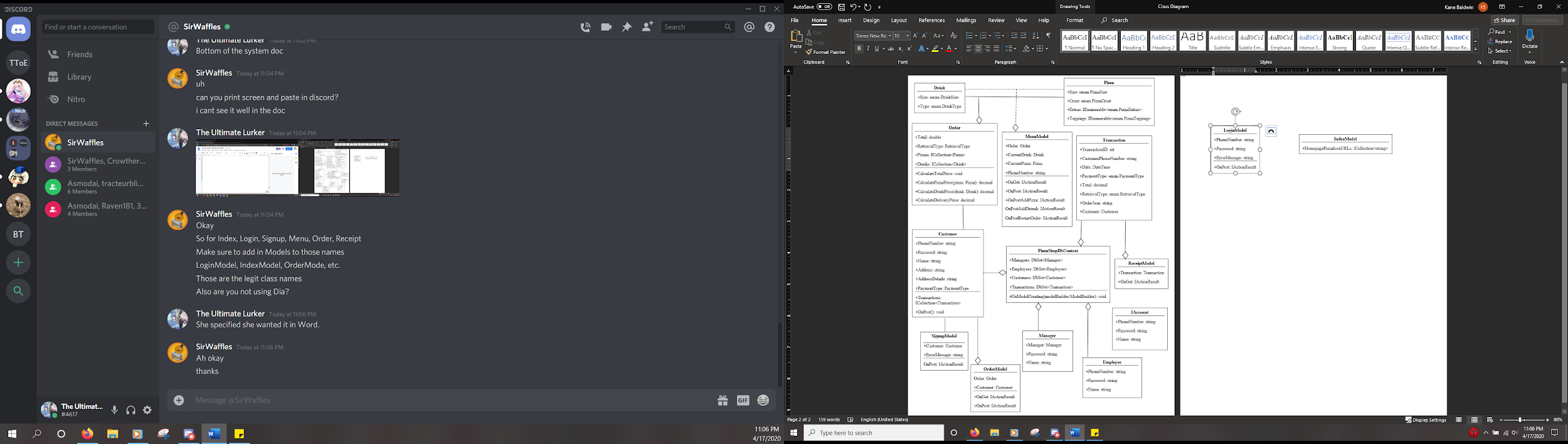
This page shows up after the pizza is input into the ordering page. It lists everything ordered on that page and how much it will all cost. Additionally, it allows the customer to select between carryout and delivery.

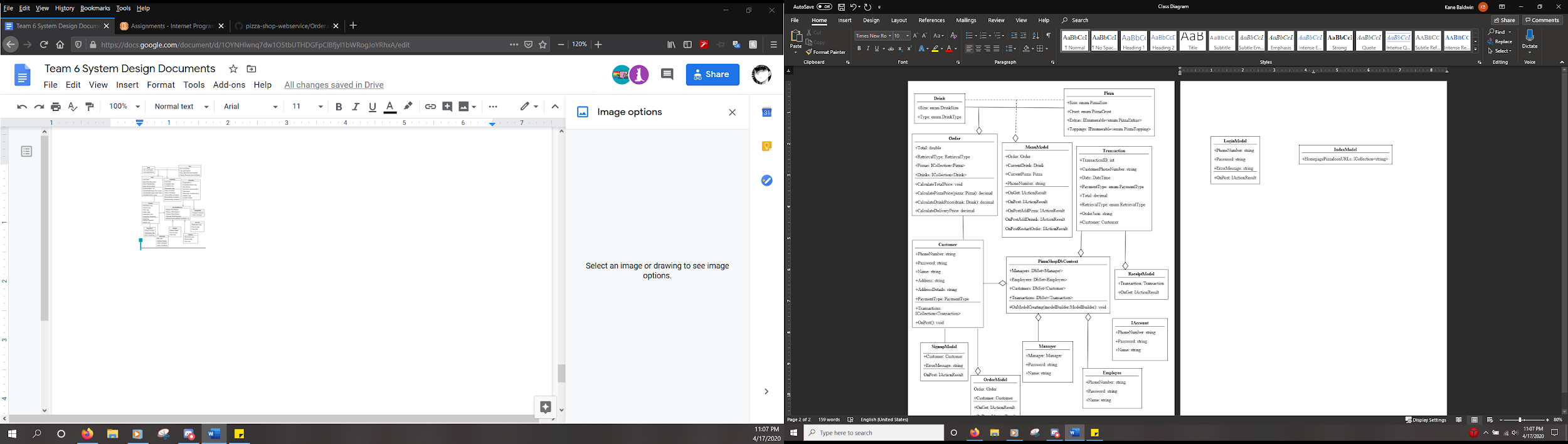
****

The receipt screen shows you a receipt of the order. Presented is a button to print the receipt.

**Technical Design: (High Level)**

* + **IAccount**
    - Interface for required info for accounts.
    - String attributes for phone number, password, and name.
  + **Customer**
    - Represents customer account.
    - Class inherits from IAccount.
    - Class is also used as table and entities.
    - String attributes for phone number, password, name, address, and address details.
    - PaymentType enum attribute for payment type.
    - ICollection<Transaction> attribute. Is used to identify entity relationships in C# EntityFramework, isn't used by developers.
    - One-to-many relationship with Transaction.
    - PhoneNumber is set as the primary key.
  + **Manager**
    - Represent manager account.
    - Class inherits from IAccount.
    - Class is also used as table and entities.
    - String attributes for phone number, password, and name.
    - PhoneNumber is set as the primary key.
  + **Employee**
    - Represents employee account.
    - Class inherits from IAccount.
    - Class is also used as table and entities.
    - String attributes for phone number, password, and name.
    - PhoneNumber is set as the primary key.
  + **Drink**
    - Attributes for DrinkSize and DrinkType enums.
  + **Pizza**
    - Attributes for PizzaSize and PizzaCrust enums.
    - IEnumerable attributes for PizzaExtra and PizzaTopping enums.
  + **Order**
    - Represents customer order.
    - Decimal attribute for Total and DeliveryFee.
    - RetrievalType attribute for RetrievalType enum.
    - ICollection attributes for Drinks and Pizzas.
    - CalculateTotalPrice() calculates the price of all items in this order.
    - CalculatePizzaPrice(Pizza pizza) calculates and returns the price of a single pizza.
    - CalculateDrinkPrice(Drink drink) calculates and returns the price of a single drink
    - CalculateDeliveryPrice() returns the price of delivery.
  + **Transaction**
    - Class is also used as table and entities.
    - Int attribute for TransactionId.
    - DateTime attribute for current date.
    - String attribute for CustomerPhoneNumber and OrderJson.
    - PaymentType attribute for PaymentType enum.
    - Decimal attribute for Total.
    - RetrievalType attribute for RetrievalType enum.
    - Customer attribute for Customer class. Is used to identify entity relationships in C# EntityFramework, isn't used by developers.
    - Many-to-one relationship with Customer.
    - TransactionId is used as the primary key.
    - CustomerPhoneNumber and Customer attributes in conjunction serve to create a foreign key with a Customer entity.
  + **PizzaShopDbContext**
    - Inherits from DbContext
    - DbSet attributes for Managers, Employees, Customers, and Transactions
    - OnModelCreating(ModelBuilder) overrides a protected method in DbContext, is used to register and create tables for Manager, Employee, Customer, and Transaction classes and relations.
  + **IndexModel**
    - ICollection attribute of strings for HomepagePizzaIconURLs. Stores a list of URLs for pizza icons.
    - IndexModel(IPizzaShopConfigDataLoader) constructor uses dependency injection to get an instance of config loader. Sets HomepagePizzaIconURLs.
  + **Customer/LoginModel**
    - IPizzaShopDbHandler attribute.
    - String attributes for PhoneNumber and password.
    - LoginModel(PizzaShopDbHandler) sets the private readonly pizzaShopDbHandler object.
    - IActionResult OnPost() validates the model, and checks to see if the phone number is already in use by another customer account, or a manager or employee account. If so, the Login page is returned. If not, the phone number is saved into a session cookie, and redirects to the menu page.
  + **Customer/SignupModel**
    - IPizzaShopDbHandler attribute.
    - Customer attribute for customer. Attributes of customer are automatically bound to HTML input elements by ASP.NET.
    - String attribute for error message.
    - SignupModel(PizzaShopDbHandler) sets the private readonly pizzaShopDbHandler object.
    - IActionResult OnPost() validates the model, and checks the database to see if the phone number is used in another customer, manager, or employee account. If so, the page is returned and an error message is set warning that the phone number is already in use. If not, the user is redirected to the customer login page.
  + **Manager/LoginModel**
    - IPizzaShopDbHandler attribute.
    - String attributes for PhoneNumber and password.
    - LoginModel(PizzaShopDbHandler) sets the private readonly pizzaShopDbHandler object.
    - IActionResult OnPost() validates the model, and checks to see if the phone number is already in use by another manager account, or a customer or employee account. If so, the Login page is returned. If not, the phone number is saved into a session cookie, and redirects to the manager login page.
  + **Employee/SignupModel**
    - IPizzaShopDbHandler attribute.
    - Employee attribute for employee. Attributes of employee are automatically bound to HTML input elements by ASP.NET.
    - String attribute for error message.
    - SignupModel(PizzaShopDbHandler) sets the private readonly pizzaShopDbHandler object.
    - IActionResult OnPost() validates the model, and checks the database to see if the phone number is used in another customer, manager, or employee account. If so, the page is returned and an error message is set warning that the phone number is already in use. If not, the user is redirected to the customer login page.
  + **Employee/SignupModel**
    - IPizzaShopDbHandler attribute.
    - Manager attribute for manager. Attributes of manager are automatically bound to HTML input elements by ASP.NET.
    - String attribute for error message.
    - SignupModel(PizzaShopDbHandler) sets the private readonly pizzaShopDbHandler object.
    - IActionResult OnPost() validates the model, and checks the database to see if the phone number is used in another customer, manager, or employee account. If so, the page is returned and an error message is set warning that the phone number is already in use.
  + **MenuModel**
    - IPizzaShopDbHandler attribute.
    - Order attribute for order.
    - Drink attribute for CurrentDrink. Attributes of Drink are automatically bound to HTML input elements by ASP.NET.
    - Pizza attribute for CurrentPizza. Attributes of Pizza are automatically bound to HTML input elements by ASP.NET.
    - MenuModel(IPizzaShopDbHandler) sets the IPizzaShopDbHandler attribute. Creates CurrentDrink and CurrentPizza.
    - IActionResult OnGet(). Loads customer phone number from the session cookie and checks to see if there is a customer account. If there is an account, then it loads the page. If not, then it redirects to a blank page with an error message.
    - IActionResult OnPost() loads the order from the session cookie, and changes the RetrievalType, recalculates the total price, and saves the new order in the session cookie.
    - IActionResult OnPostAddPizza() validates the model, loads the order from the session cookie into the Order attribute, adds the current drink to the Drinks collection, then recalculates the total price of the order. The Order attribute is then saved into the session cookie, and the user is redirected back to the menu page.
    - IActionResult OnPostAddPizza() validates the model, loads the order from the session cookie into the Order attribute, adds the current pizza to the Pizzas collection, then recalculates the total price of the order. The Order attribute is then saved into the session cookie, and the user is redirected back to the menu page.
    - IActionResult OnPostRestartOrder() validates the model, removes the order from the session cookie, and the user is redirected back to the menu page.
  + **OrderModel**
    - Order attribute for the order.
    - Customer attribute for the customer.
    - OrderModel(IPizzaShopDbHandler). Creates instance of OrderModel with a new Customer and assigns a pizzaShopDbHandler
    - IActionResult OnGet(). This will set the phone number to the entered phone number. If the person entered no phone number they will be told to enter a phone number. The value of the customer will be set based on their respective phone number. If the customer has no assigned value, the customer will receive a message stating there is no account associated with this phone number.
    - IActionResult OnPost(). This sets the retrieval type of the order. The order’s total price is recalculated and the order is saved in the session cookie. The user is redirected to the payment option page.
  + **PaymentOptionModel**
    - Order attribute for PaymentOptionModel.
    - Customer attribute for PaymentOptionModel.
    - IActionResult OnGet(). This will load the order from the session cookie. It will then get the customer from db using the phone number in the session cookie.
    - IActionResult OnPost() This will redirect the user to the payment confirmation.
  + **PaymentConfirmationModel**
    - String CardNumber attribute for PaymentConfirmationModel.
    - String expiration date attribute for PaymentConfirmationModel.
    - Customer customer attribute for PaymentConfirmationModel.
    - IActionResult OnGet() This will get the customer using the phone number in the session cookie. It will then Autofill the card type and the card holder’s name values from Customer object
    - IActionResult OnPost() This will validate the PaymentConfirmation model. It will then create a transaction object, save it to the database. Lastly it will redirect the customer to the receipt page.

**Class Diagrams**



**Database Table Descriptions**

* + **Customer** table logs individual customers’ information
    - PhoneNumber: string of primary key tracking the customer’s phone
    - Password: encoded string containing the users password
    - Name: string for tracking user’s name
    - PaymentType: enum PaymentType tracking how the customer pays
    - AddressDetails: string containing where the customer lives
  + **Manager** table logs information about Pizza Shop managers
    - PhoneNumber: string primary key to login
    - Password: encoded string to confirm Manager identity upon logging in
    - Name: string to track the identity along with Manager accounts
    - Isolated from other tables
  + **Transaction** table tracks every order made by a Customer
    - TransactionID: integer primary key for each Transaction
    - CustomerPhoneNumber: string that tracks the phone number of the order
    - Date: DateTime value that logs when the order was made
    - PaymentType: PaymentType value for logging payment type
    - Total: double that logs the total cost
    - OrderJson: string in JSON that logs the full order
    - Customer: Foreign key between Transaction and Customer table

**Technical Support Specifications**

* Customers will have a phone number at the bottom of the page that connects them to the restaurant allowing for assistance with ordering.
  + This service will be available as long as the Pizza Shop is open.
* The buttons on the site were designed to be as intuitive as possible, guiding the users in the purpose of each page, and being consistent throughout.
* The pictures are designed to aid the customer in deciding what items to add to their order.
* The text and descriptions are designed to aid the customer in deciding what items to add to their order.