

SWE 3313 Mom and Pop's Pizzeria



By: Jason Paek, Zach Stover, Martin Nguyen, Jasmine Brown, Mohammad Umar

Table of Contents:

Project Plan

● Scope:	2
● Schedule:	3
● Gantt Chart:	4
● Team Organization:	6
● Resumes:	7
● Technical Description:	12
● Data management plan:	14
● Test plan:	15

Requirement

● Functional and nonfunctional requirement definitions:	16
● Use Case Flow Documentation:	17
● Class Documentation:	20
● Use Case Diagram:	26
● Class Diagram:	27
● Entity Relation Diagram:	28
● State Transition Diagram:	29

System Design Documents

● Conceptual System Designs:	30
● Technical Design:	50
● Technical Support Specification:	66

Weekly Status Report: 67

Project Log: 78

Scope:

1. Customer Record

- Information Keyed by customer phone number
- Record customer's information
 - 1. Name
 - 2. Address
 - 3. Phone number
 - 4. Type of charge account
- Record pertinent information for locating address
 - Subdivision name, intersections, etc...

2. Payment Process

- Record customer payment information in the form of checks, cash or credit cards.
- Track customer payment type and amount of payment.

3. Customer Database

- User will access the customer database for all consumer transactions
 - Information should come up already for customers showing address and delivery information

4. Order Confirmation / receipt

- Signature option if paid with credit card.
- Receipt
 - if delivery ask for address if pickup give address
- Finalize payment.

5. Restaurant Menu

- The system should contain the whole restaurant menu
 - For order taking purposes
 - Menu must consist of various types of pizzas, various crust options, and beverages
 - User needs to have GUI access to menu items and quickly enter desired orders

Schedule:

Task ID	Work breakdown structure	Planned start	Planned finish	Workload-planned (Hours)	Workload - actual	Progress (%complete)
1_1	Customer Input	10/05	10/10	10	3	100
1_2	Customer information	10/11	10/18	15	4	100
1_3	Locating Information	10/15	10/24	10	3	100
2_1	Payment Process	10/22	10/31	5	3	100
2_2	Payment type/Amount	10/29	11/09	8	4	100
3_1	Customer Transactions	11/05	11/18	7	4	100
4_1	Receipt	10/05	10/18	6	2	100
4_2	Finalize payment	10/20	10/27	6	4	100
5_1	Menu options	10/29	11/10	10	2	100
5_2	User interface	10/29	11/18	15	7	100

Gantt Chart:



GANTT CHART

Pizza Project Plan

PROJECT NAME

Mom & Pop's Pizzeria

CLIENT NAME

05-Oct-20

START DATE

18-Nov-20

END DATE

11/1/2020
11/2/2020
11/3/2020
11/4/2020
11/5/2020
11/6/2020
11/7/2020
11/8/2020
11/9/2020
11/10/2020
11/11/2020
11/12/2020
11/13/2020
11/14/2020
11/15/2020
11/16/2020
11/17/2020
11/18/2020

Team Organization:**Team Roles :**

Jason Paek: Team Leader

Martin Nguyen: Organizer

Jasmine Brown: Project Manager

Mohammad Umar: Lead Document Writer

Zach Stover: Lead Programmer

Resumes:



JASON PAEK



JASONPAEK2000@GMAIL.COM



404-277-5277



<https://www.linkedin.com/in/jasonpaek/>

OBJECTIVE

Determined student at KSU working towards a Bachelor of Science degree in Computer Science and looking for work experience.

SKILLS

Technical Skills:

Limited Knowledge in C#, C++

Proficient in Microsoft Office

Interpersonal Skills:

Experienced in Communication, Teamwork, and Customer Service

EXPERIENCE

WALMART

CUSTOMER SERVICE DESK ASSOCIATE

Nov 2019 – Present

Fast-Paced multitasking experience with:

- Customer Service
- Returns/Claims
- Cashiering/Cash Handling
- Stocking

WENDY'S

RESTAURANT CREW MEMBER

May 2019 – Aug 2019 (4 months)

Fast-Paced multitasking experience with:

- Cashiering/Serving
- Customer Service
- Janitorial Work
- Food Preparation
- Stocking

VOLUNTEER EXPERIENCE OR LEADERSHIP

Jesus Hope Korean Church

Teacher Assistant (May 2016 – Jun 2018)

Provided educational and artistic development in minors

Ivy Springs Manor

Elderly Home Assistant (Jun 2017 – Aug 2017)

Aided Elderly residents with entertainment, food, and other health related activities

EDUCATION

Bachelor of Science Degree/ Computer Science

Kennesaw State University

Extracurricular Involvement in Honors College, Rotaract, Thrive, and Engaged Owl Leaders



Zach Stover:**OBJECTIVE**

I'm a college student interested in an internship over this summer.

EDUCATION

Creekview High School	2015 - 2018
Kennesaw State University	2018- Present

SKILLS

Microsoft Office (Word and PowerPoint)
Efficient in Java
Efficient in C++
HTML/CSS Experience

Awards, Certificates, and Events

A & B Honor Roll	2018
Intramural Chair	2019
Hackathon	2019

Clubs & Organizations

Fraternity (Delta Sigma Phi)	2019 - Graduation
------------------------------	-------------------

Jasmine M Brown

Objective

College student seeking a summer internship in the field of Software Engineering.

EDUCATION

Kennesaw State University, Kennesaw, GA

- Bachelor of Science in Software Engineering
- Minor in Information Technology with a Concentration in Security
- Minor in Spanish
June 2018 - Present
Kennesaw, GA

SKILLS

Coding in Java, C#, SQL, and Python. Web development with Html, JavaScript, and CSS. Usage of the Microsoft office suite.

EXPERIENCE

Kennesaw State SPCEET, Student Assistant

Marietta, GA ---February 3, 2020 - Present

- Assisting with office duties such as filing, printing/scanning, and informational document creation/updates.

Cobb EMC, Intern

Marietta, GA --- January 24, 2018 - April 27, 2018, 13 Weeks

- Research-based internship through the Kennesaw Mountain Magnet Program

Girls Who Code, Student

Atlanta, GA --- June 2016 - July 2016, 7 Weeks

- Coding, worked with computer programs, networked with individuals in businesses and technological companies

AWARDS/HONORS

- W3Schools HTML certification
(2018)
- Member of the National Technical Honor Society
(2016-present)
- Cisco certification
(2015)

VOLUNTEERING

- Ongoing projects with the Kennesaw chapter and Georgia district of Circle K International
(2019-Present)
- Shop With A Mustang - Assisted children from low-income families in holiday shopping
(2015-2018)
- Various projects with local Interact club, Habitat for Humanity Club, and Art Club-Club activities dedicated to helping in the community
(2015-2018)
- Special Needs Dance - Winter dance held at Kennesaw Mountain High School
(2014-2018)

CLUBS

- Circle K International
 - Chapter President
(2020-Present)
 - District Marketing Development and Education Chair
(2019-2020)
 - Chapter Vice President
(2019-2020)
 - Chapter Reservation Delegate
(2019-2020)
- Kennesaw Society of Black Engineers
 - Member
(2020-Present)
- Society of Women in Engineering
 - Member
(2020-Present)
- Interact
 - Member
(2015-2018)
 - Director of Public Relations
(2017-2018)
- Habitat for Humanity
 - Member
(2015-2018)
 - Communications Chair
(2016-2017)
 - Treasurer
(2017-2018)

Martin Nguyen:

Martin Nguyen

Summary

Highly motivated Computer Science student with eagerness to learn and expand upon skill set and knowledge.

123 Lorem Ipsum
Dolor, ST 12345
(123) 456-7890
exampleemail@example.com



EXPERIENCE

IDLogiq, California — Intern Employee

October 2019 - PRESENT

Cryptographic authentication company, worked with a wide variety of assignments from front end UX/UI to business marketing assignments.

Publix, Johns Creek, GA — Deli Clerk

January 2018 - July 2018

Provided customer service in Traditional by slicing meat and cheese, as well as in the Subs section by making subs.

LANGUAGES

English & Vietnamese

SKILLS

- Proficient in JAVA, Python, Javascript, and HTML
- Exceptional Customer Service
- Neat and well-organized
- Hard-working achiever
- Dedicated to provide quality work
- Works well with deadlines
- Efficient and collaborative

EDUCATION

Kennesaw State University, formally Southern Polytechnic State University, Marietta — Working to get Bachelors of Science in Computer Science (Currently: Junior)

January 2020 - May 2022

Georgia Perimeter College, Alpharetta/Dunwoody — Associates in Computer Science

August 2017 - December 2019

Interests/Hobbies

- Drawing
- Gaming
- Sports: volleyball, soccer, tennis, swim, boxing, surf

Mohammad Umar:**OBJECTIVE**

College student interested in summer internship in web design and majoring in computer science.

EDUCATION

Meadowcreek High School	2012 - 2017
Kennesaw State University	2017- Present

SKILLS

Microsoft Office (Word and PowerPoint)
WordPress
HTML
CSS
C#
Teamwork

Awards, Certificates, and Events

A & B Honor Roll	
Web Challenge	2015
Web Challenge	2016
Georgia STEM Day	2015
CIW Site Development Associate	2016

Clubs & Organizations

Future Business Leader of America(FBLA)	2014 - 2016
---	-------------

Technical Description:

We will start by applying the UI onto a website. Within the website, there will be an opening screen consisting of a simple welcoming of the customer. The whole goal of the opening screen is to make it the most appealing to the customer by adding appealing visuals and keeping it simple to use. On the opening page, there will be a menu button that will display all the pizza options the customer can choose from. Customers will be able to choose from many different types of pizza. For example, cheese pizza, pepperoni pizza, veggie pizza, meat pizza with customer choice of sauce and crust. Also on the opening screen, there will be a “login in” button that will direct the customer to the login screen.

Within the login screen, the system will take in the customer’s name, address, phone number, and type of charge account. Importantly, the information will be keyed by the customer phone number and then the system will record pertinent information for locating the address. Once the customer has logged in, the customer will then be directed to the selection screen where the menu will be displayed and the selection will be made. On this page, the layout will be in forms in buttons where the customer will select how many pizzas they want. Within the page, the pizzas will be grouped in cheese pizza, pepperoni pizza, veggie pizza, and meat pizza; the quantity will be marked per pizza and the user can mark what size they want. In addition, the customer will be able to mark what they want for a drink, any additional extra pizza sauce(s), additional meat and cheese, and what kind of crust.

Once the selection is made by the customer, the user now has the decision on what payment they want to do varying from credit card, check, or cash though the “checkout” button. Each payment option will be displayed onto a button waiting for the customer to click on it. Once the payment option is chosen, the user will need to pay the valid amount + taxes to have a

valid payment. Once the payment is made, the system will then track the user payment method and the amount of payment.

Within the payment page, the user will also need to have an option to sign if they have paid with a credit card. Also, there will be a section where the user will input their address in which the pizzas will be shipped if they decide to select delivery; the customer also has an option to pick up in a restaurant. Once the payment is officially made, the customer will get a receipt confirming their order.

Once the order is placed and the payment method is valid, the information will be stored inside a database resulting in the user seeing the address and delivery information. Within the delivery information, there will be information telling the user how long the order will take to prepare and estimated drive time.

To access the website the customer and owner needs access to a phone, laptop, tablet, or desktop with internet connection. The design of the website is going to be simple so it will be easy to use for both customers and employees. All the information will be organized making navigating through the website easy. The website design will have simple colors and fonts making it easier for users to understand all the information.

To program the website we will be using HTML, CSS, C#, Java, and SQL. We will use HTML and CSS to program the front end of the website which will include all the information that the user will see. To program back end of the website we will use C# and Java to create login, payment system, etc. We will use SQL to create a database to store information from the customer. The information will be accessible by the customer and employees.

Data management plan:

The customer would be able to access and update any information pertaining to their account. The manager would have viewing access to that information. The customer is able to access the menu but they are not able change or update any of that data. The manager can change/update the pizza data on the menu. The customer can update their payment (check, cash, credit) at any time. The customer can also update information such as email address, phone number, delivery address. The management can view the customer data but cannot change it. The customer will be able to check order type such as delivery or pickup.

The information will be stored as objects in a database. Customer information will require login information to be correct and Manager information will require separate login information. After a customer login to their account they will be able to access all of their information. Also, the customer will be able to view the menu and will be able to order any pizza they desire. They will be able to make payment and decide whether they want to pick-up the pizza or they want it delivered. If customers choose their pizza to be delivered they will be able to provide the address where they like the pizza to be delivered. After manager login to their account they will be able to access all the information entered by the customer. Information such as their name, email, phone number, order, payment information, delivery address, etc.

Test plan:

First we will test it by running through the applications a couple times to see if there are any inconsistencies. Then we will ask several other people so we can have an outside perspective on our application and get their feedback/input. The criteria for early termination would include bugs within the program and/or missing portions that are caught. If something unexpected does occur our method of terminating will have a specific input to quit the program and terminate it.

For user training we find it quite important because there will be never-ending waves of updates, so the user training programs will be ongoing. User training also allows the design team to identify any problems in the user experience so we can deal with it before it gets finalized and deployed. It is going to be expected to regularly train for at least a few hours a week to stay aware of mandatory training and maintenance alerts. Automation won't be needed because the application itself is going to be fully dependent on human interaction.

On paging issues there can be some problems like page crawling, which is an issue with google not indexing all the pages we would want to due to having a large amount of URLs, and duplicate content, which are several pages that have very similar, or exactly the same content. To help with paging issues we can add page series and a "view all" page that contains clear non-dynamic URLs and no broken links to help with overcoming the page crawling and reducing possibilities of duplicate contents. With data storage we will create a SQL Query that will pretty much store all the imputed information from the customer. Then the customer can access their own information depending on how much access is allowed. For example, the customer's credit card info, name, and address will all be accessible and editable primarily to them. The employers can also access all the information the customers entered because of management access.

Requirement Definition Documentation

Functional Requirements:

1. The customer will be able to make an account, sign in, and be able to make an order, which will allow the customer to enter and save personal info onto the database. The customer will also be able to order as a guest without an account.
2. The customer on the computer will be able to access and see every item on the menu.
3. The customer will be able to choose food items and the amount of that food item on the menu and be able to create an order by clicking the associated buttons on the menu.
4. The customer will be able to fill in their payment information as well as edit existing information.
5. The customer can access nutritional information and will be able to access further information about the company in the “about us” page.
6. The customer will be able to pick order type on the computer(Delivery/ Pickup).
7. The customer will be able to submit an order, and the order will be placed in a queue which will be organized in priority of time ordered.
8. The Employee will be able to access a customer database that contains the customer's information and the orders placed.
9. The Employee will be able to add, edit, or remove menu items.

Nonfunctional Requirements:

1. The program must be elegant and correlate with the Mom and Pop restaurant: consistent theme, color scheme, formatting, alignment, photos, and etc..
2. Restaurant and Customer data must be secure and only accessible by authorized users.
3. The system should be able to be monitored and maintained by the Mom and Pop management.
4. Management should make sure that every employee at the restaurant should be able to use the program correctly.
5. The system should be simple enough to be used by any level of technical expertise.
6. The UI should be easily navigable and should allow a fluid experience for the user.
7. There should be back ups of all the data in case of failure.

Use Case Flow Documentation

Use Case Flow for ordering, delivery, etc

Precondition

- The website must be fully operational before the use case can start. The payment system has to be able to send and receive information with the required authentication resources.

Main Flow

- The use case starts when the customer enters the link to the Pizza ordering website. A menu of available items are displayed with options for adding each main item to the cart. After the main items are selected, an options menu is displayed for the crust type, toppings, etc. The system then prompts for “Checkout?” If the customer selects yes, Checkout, the user is then prompted again for Registered User information, or New User information. For New Users, a page to enter all delivery and payment information is displayed. For registered users, a Login Page, which will be followed by a display of payment and delivery or pickup information confirmation.
- If delivery, when the driver is out for delivery, a notification will be sent to the user with the phone number indicated in the customer information.
- If pickup, a notification will be sent to the user with the phone number indicated in the customer information containing the address and estimated time to pickup.

Sub-Flows

- Add a User-created item. The system will let the user make their pizza from scratch starting with just a selection of crust type. The user can then choose the toppings to add on the pizza.

Change of User Information

- Upon being directed to confirm User Information, the user has the option to update their Address, Phone Number, etc. After updating their information if they chose to, the user can save the information, and set the updated address as the default for future orders.

Alternative Flows

- E-1: When nothing is added to the cart, the user will be prompted with a text screen with a warning saying that the cart is empty.
- E-2: An invalid Username or password is entered in the create an account page (whether it does not follow the requirements or if the username is already taken). The user will be prompted with the username requirements and a message will be prompted saying that the username is invalid or the password is empty.
- E-3: An invalid Username or password is entered in the sign in page. The user will be prompted with a message stating that the username or password does not match.

- E-4: A required field in customer, payment, or delivery information is not filled out or an option is not selected. An asterisk will appear next to the title field and a small message prompt will appear stating that a required field is not filled out or selected.

Use Case Flow for Edit Menu, Customer database, etc...

Preconditions

- The employee modifying the current menu options must be an administrator or given special privileges to change and update/edit menu items.

Main Flow

- In the administrator mode (or higher), all prices, icons, and names are able to be edited. Selecting specific fields will bring up available attributes for the specific fields.
- The employees can browse and traverse through the customer database to access customer information and order confirmation.

Sub-Flows

- Accessing Customer Database
 - In the drop bar menu of profile, employees and managers will have a unique option of “Customer Database” to browse or navigate through the list of customer information / order confirmation.
- Add a New Item
 - Adding a new item to the menu will prompt the employee with a request for price, name, and icon input to be filled out in order to complete change.
- Delete Item
 - Fast-clicking the edit button on the edit menu interface will allow the user to edit the menu
 - Long-clicking the specific menu item can remove the item or change the price while an empty box will display to take in user-input for additional menu items that are wanting to be added.

Alternative Flows

- E-1: If an invalid price for any menu item is entered in the edit function (whether nothing is entered or the price is lower than the total ingredient cost), a prompt will ask the user to enter a valid price.

- E-2: If an Invalid product name is entered (if it is empty), a prompt will ask the user to enter a valid product name.
- E-3: If an Invalid customer name is entered into the customer database, a prompt will display stating that no customers under that name were found.

Class Documentation

Classes

Home Page

Attributes: home (boolean), about (boolean), contactUs (boolean), nutritionInfo (boolean), logIn (boolean), cart (boolean), and menu (boolean). All of the attributes are of private type.

Methods: selectHome, selectAbout, selectContactUs, selectNutritionInfo, selectLogIn, selectCart, selectMenu. All of the methods are of type public.

About Page

Attributes: aboutUsInfo (string). The attribute is private type.

Methods: No methods for the about page.

Nutritional Information Page

Attributes: home (boolean) and nutritionalInfo (string). All attributes are of private type.

Methods: selectHome of public type.

Contact information Page (w/o Account)

Attributes: email (string), name (string), phoneNumber (string), addressLine (string), additionalAddressInfo (string), city (string), state (string), zipCode (int), submit (bool), back (bool). All attributes are of private type.

Methods: setEmail, setName, setPhoneNumber, setAddressLine, setAdditionalAddressInfo, setCity, setState, setZipCode, selectSubmit, selectBack. All methods are of public type.

Menu Page

Attributes: menuID (int), menuType (string), menuName (string), and back (boolean). All attributes are of private type.

Methods: selectMenuItem and selectBack of type public.

Pizza

Attributes: pizzaID (int), pizzaName (string), pizzaSize (string), pizzaTopping (string), pizzaCrust (string), pizzaPrice (int), addToCart (boolean), quantity (int), and edit (boolean). All attributes are of private type.

Methods: selectPizzaSize, selectPizzaTopping, selectPizzaCrust, selectQuantity, calculatePrice, selectAddToCart, and selectEdit. All methods are of public type.

Sides

Attributes: sidesID (int), sidesName (string), sidesName (string), sidesPrice (int), addToCart (bool), quantity (int), edit (bool). All attributes are of private type.

Methods: selectQuantity, calculatePrice, selectAddToCart, selectEdit. All methods are of public type.

Drinks

Attributes: drinksID (int), drinksName (string), drinksSize (string), drinksPrice (int), addToCart (bool), quantity (int), edit (bool). All attributes are of private type.

Methods: selectQuantity, calculatePrice, selectAddToCart, selectEdit. All methods are of public type.

Dessert

Attributes: dessertID (int), dessertName (string), dessertType (string), dessertPrice (int), addToCart (boolean), quantity (int), and edit (boolean). All attributes are of private type.

Methods: selectQuantity, calculatePrice, selectAddToCart, and selectEdit. All the methods are of public type.

Cart Page

Attributes: itemID (int), itemName (string), itemTypes (string), itemQuantity (int), remove (bool), processOrder (bool), subtotal (double), tax (double), total (double), and back (bool). All attributes are of private type.

Methods: selectRemove, calculateSubTotal, calculateTax, calculateTotal, selectProcessOrder, and selectBack. All the methods are of public type.

Sign In Page

Attributes: username (string), password(string), submit (bool), forgotPassword (string), forgotUsername (string), createAnAccount (string), employeeSignIn (string), back (bool). All attributes are of private type.

Methods: setUsername, setPassword, selectSubmit, selectForgotPassword, selectCreateAnAccount, selectEmployeeSignIn, selectBack. All methods are of public type.

Create An Account Page

Attributes: email (string), password (string), name (string), phoneNumber (string), addressLine1 (string), addressLine2 (string), city (string), state (string), zipCode (int), submit (bool), haveAnAccount (string), and back (bool). All attributes are of private type.

Methods: setEmail, setPassword, setName, setPhoneNumber, setAddressLine1, setAddressLine2, setCity, setState, setZipCode, selectSubmit, selectSignIn, and selectBack. All methods are of public type.

Payment Process Page

Attributes: orderSummary (string), orderItem (string), total (double), tax (double), subTotal: (double) , paymentMethod (string), case (bool), check (bool), creditCard (bool), visa (bool), masterCard (bool), cardNumber (string), expireDate (int), securityCode (int), placeOrder (bool), back (bool). All attributes are of private type.

Methods: selectPaymentMethod, setCardNumber, setExpirationDate, setSecurityCode, selectPlaceOrder, selectBack. All methods are of public type.

Customer Info Page w/o Account

Attributes: email (string), name (string), phoneNumber (string), addressLine (string), additionalAddressInfo (string), city (string), state (string), zipCode (int), submit (bool), haveAnAccount (string), createAnAccount (string), and back (bool). All attributes are of private type.

Methods: setEmail, setName, setPhoneNumber, setAddressLine, setAdditionalAddressInfo, setCity, setState, setZipCode, selectSubmit, selectSignUp, selectlogIn, and selectBack. All methods are of public type.

Order Confirmation/Receipt Page

Attributes: info (string), customerID (int), onlineOrderNumber (int), orderType (bool), delivery (bool), pickup (bool), methodOfPayment (string), estimatedDeliveryTime (int), orderDetail (string), orderItems (string), orderItems (string), subTotal (double), deliveryCost (double), tax (double), total (double), confirm (bool), deliveryAddress (string), back (bool). All attributes are of private type.

Methods: selectOrderType, signature, selectConfirm, selectBack. All methods are of public type.

Customer Database Page

Attributes: customerName (string), addressInfo (string), contact (string), payment (string), other (string), date (int), time (double), paid (double), status (bool), search (string), back (bool). All attributes are of private type.

Methods: classDisplay, setSearch, and selectBack. All methods are of public type.

Employee/ Staff Page

Attributes: home (bool), orderHistory (bool), customerDatabase (bool), editMenu (bool), logout (bool). All attributes are of private type.

Methods: selectHome, selectOrderHistory, selectEditMenu, selectLogout. All methods are of public type.

Relationships

The **Home Page** acts as the “Central Page” in which these pages inheritance relationships with the About Page, Nutritional Page, Contact Information Page, Menu Page, Cart Page, and the Sign in Page.

The **Sign In Page** has an inheritance relationship with the Home Page, Employee/ Staff Page, the Customer Database Page, and Create an Account Page.

The **Employee/ Staff Page** has an inheritance relationship with the Create an Account Page and Cart Page.

The **Customer Database Page** has an inheritance relationship with the Create an Account page and Cart Page.

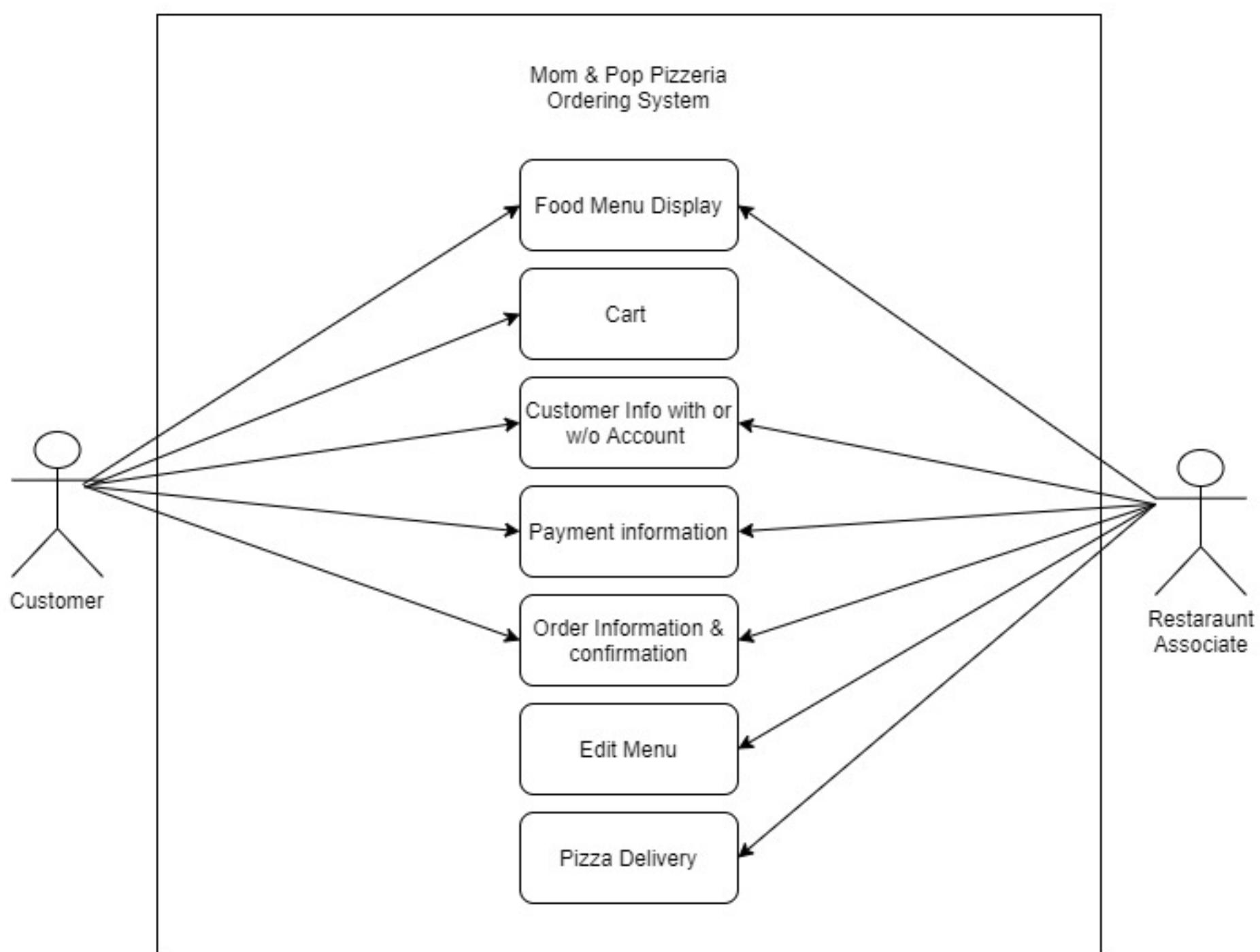
The **Create an Account Page** has an inheritance relationship with the Sign in Page.

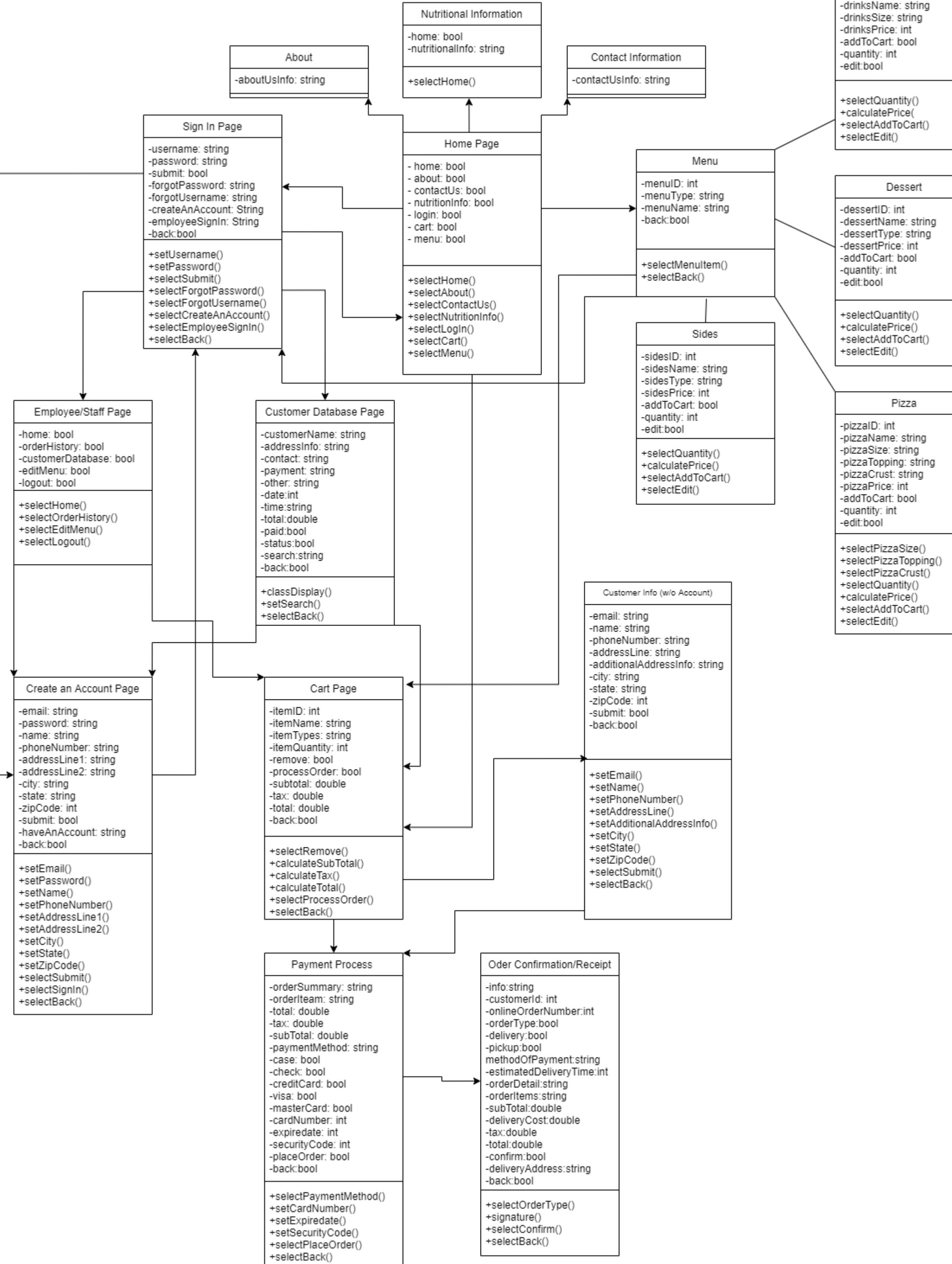
Menu Page has an associative relationship with Pizza, Drinks, Dessert, and Sides. Because those exist within the menu page. Menu page also has an inheritance relationship with cart page.

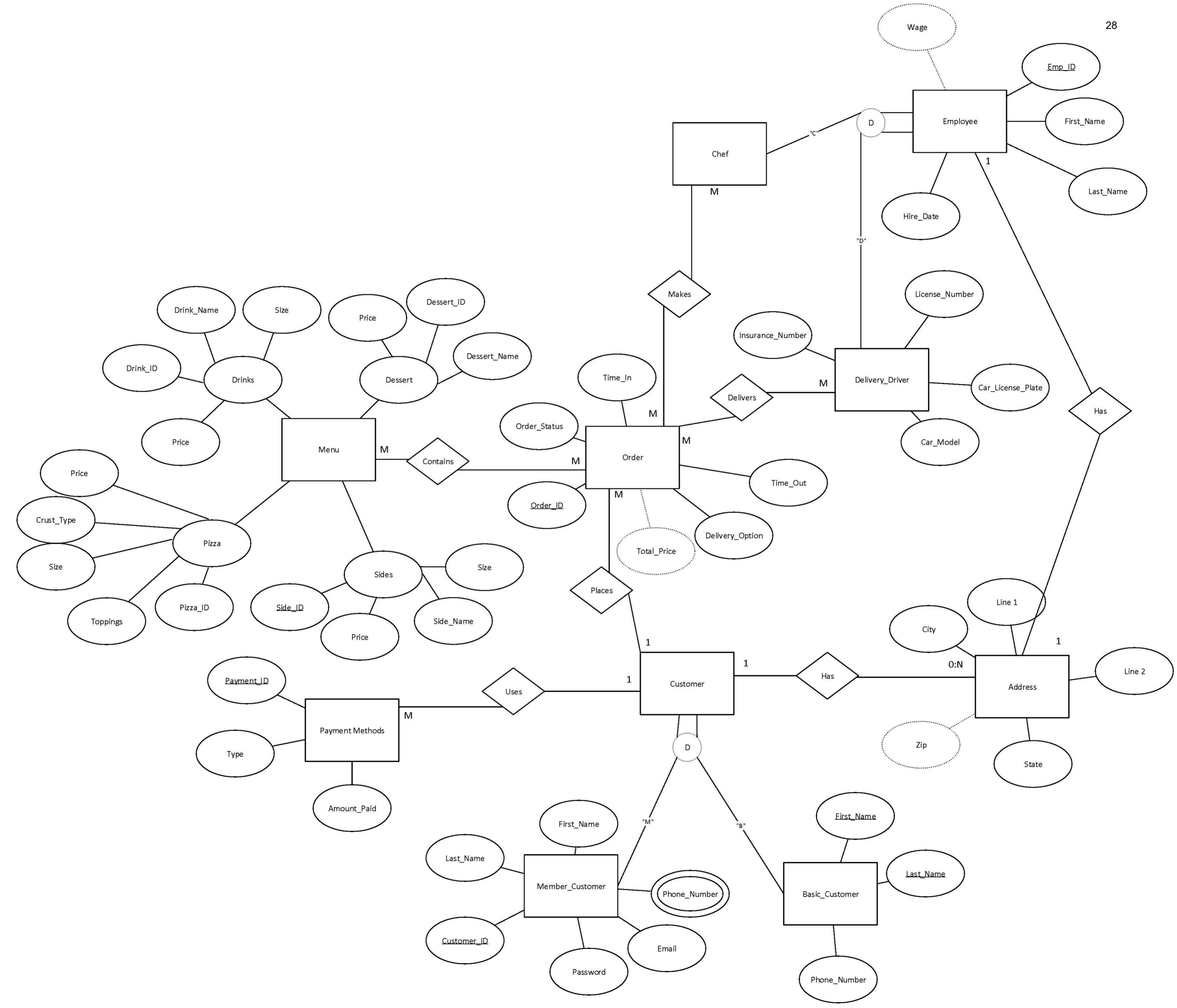
Cart page has an inheritance relationship with Customer Info w/o Account page and Payment Process page.

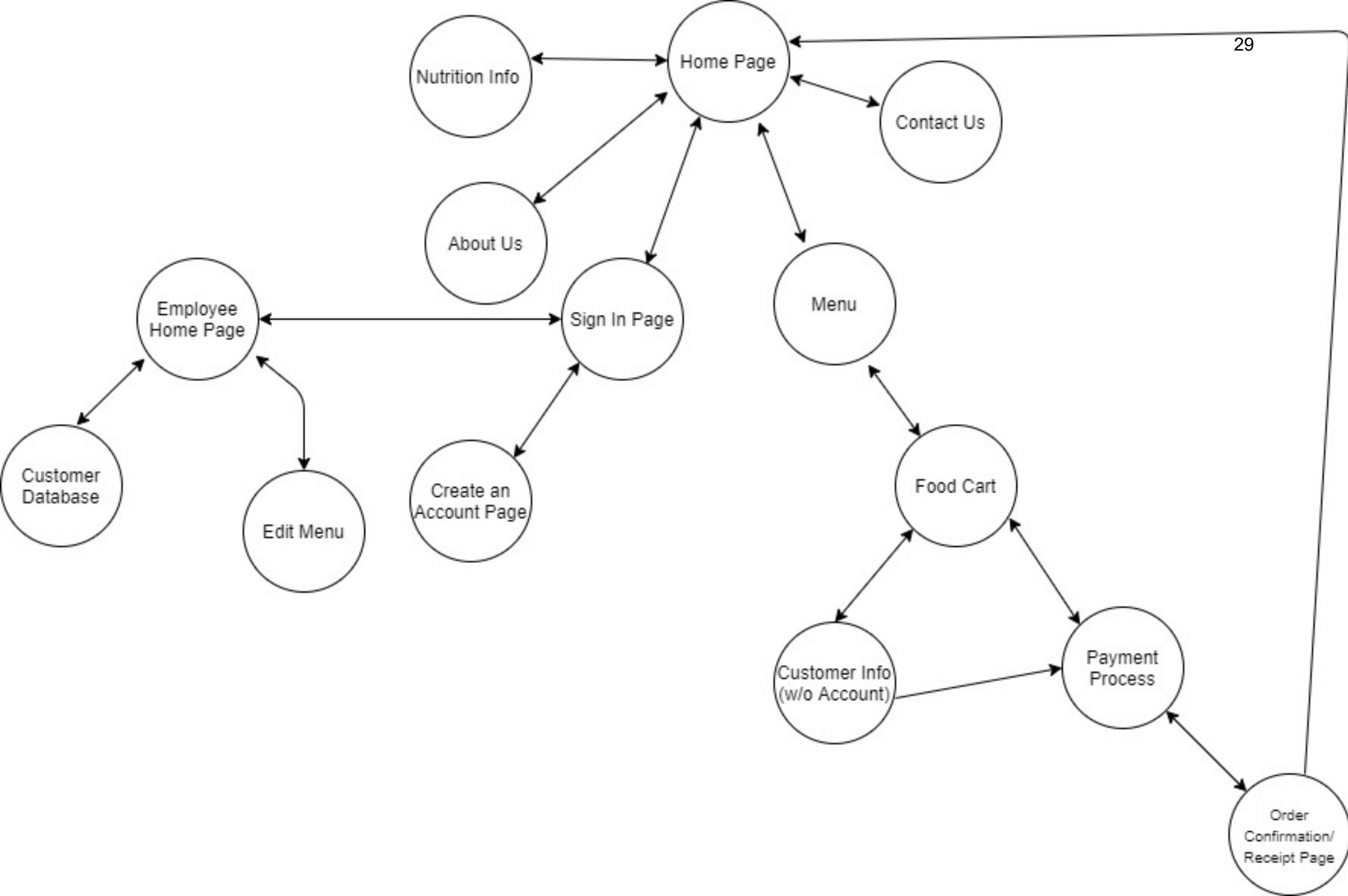
Customer Info w/o Account page has an inheritance relationship with Payment Process.

Payment process page has an inheritance relationship with Order Confirmation/Receipt Page.









III. System Design Documents:

The system design documents will consist of the following:

Conceptual system design:

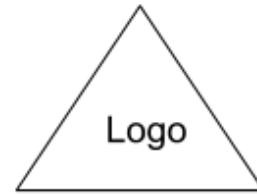
Report Format

Register Receipt

<p style="text-align: center;">Mom And Pop's Pizzeria Receipt</p> <p>Our Location: 680 Arnston Rd, Suite 156 Our Number: 770-555-1212</p> <p>Date: XX/XX/XXXX Order Type: (Delivery/Pick Up) Payment Method: (Credit/Cash/Check) Estimated Ready Time: XX:XX(AM/PM) xxxxxxxxxxxxxxxxxx</p> <p>Order Detail:</p> <table><tr><td>Item 1</td><td>\$Price</td></tr><tr><td>Item 2</td><td>\$Price</td></tr><tr><td>Item 3</td><td>\$Price</td></tr></table> <hr/> <table><tr><td>Subtotal</td><td>\$X.XX</td></tr><tr><td>Tax</td><td>\$X.XX</td></tr><tr><td>(If) Delivery Fee</td><td>\$X.XX</td></tr></table> <hr/> <table><tr><td>Total</td><td>\$X.XX</td></tr></table>		Item 1	\$Price	Item 2	\$Price	Item 3	\$Price	Subtotal	\$X.XX	Tax	\$X.XX	(If) Delivery Fee	\$X.XX	Total	\$X.XX
Item 1	\$Price														
Item 2	\$Price														
Item 3	\$Price														
Subtotal	\$X.XX														
Tax	\$X.XX														
(If) Delivery Fee	\$X.XX														
Total	\$X.XX														

Online Order Receipt

Online Receipt



Our Location: 680 Arnston Rd, Suite 156

Our Number: 770-555-1212

Receipt #: XXXXXXXX

Receipt Date: XX/XX/XXXX

Order Type: (Delivery/Pick Up)

Payment Method: (Credit/Cash/Check)

Estimate Time: XX:XX (AM/PM)

QTY	DESCRIPTION	PRICE
X	ITEM 1	\$X.XX
X	ITEM 2	\$X.XX
X	ITEM 3	\$X.XX
X	ITEM 4	\$X.XX
X	ITEM 5	\$X.XX

	Subtotal	\$X.XX
	Tax	\$X.XX
	(If)Delivery Fee	\$X.XX

	Total	\$X.XX

Thank you for your order, please keep this for your record.

Pizza Order Slip

Mom And Pop's Pizzeria

Order Date: _____

Est Time: _____

Order #:_____

Sizes

- Small
- Medium
- Large
- Extra Large

Crust

- Thin
- Regular
- Pan

Toppings

- Cheese Pineapple Tomato
- Mushroom Sausage
- Onion Ham
- Pepperoni Green Pepper

Daily Business Report/Summary

Mom and Pops Pizzeria
 680 Arnston Rd, suite 156
 Phone 770-555-1212

Daily Report	Date: XX/XX/XXXX
--------------	------------------

Total Qty Sales	Avg. Time on Order	Total Labor Hours	NET profit
###	XXX	###	XXX

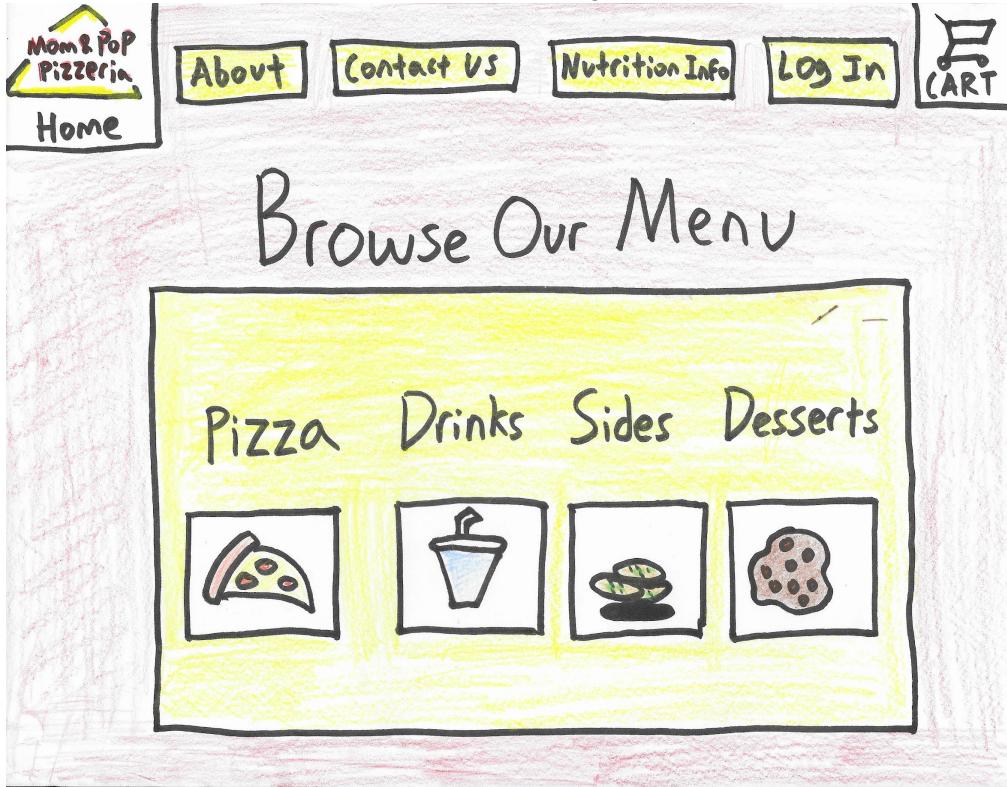
Name of Product	Qty Sold of Item	Total profit Margin
Pizza XXXX	###	\$XX.XX
Drink XXXX	###	\$XX.XX
Dessert XXXX	###	\$XX.XX

Important notices, changes, or occurrences....

<p>E.G. Employee XXXX is fired for not working Grill XXXX is broken Water is not working</p>

Screen layouts/shots

Home Page:



Requirements: The main home page should visually show the user how to navigate the home page to access the about page, the contact page, nutritional info page, log in page, cart page, or the menu page.

Main Flow:

- 1.0: The user can click any of the icons to navigate to its corresponding page/location.

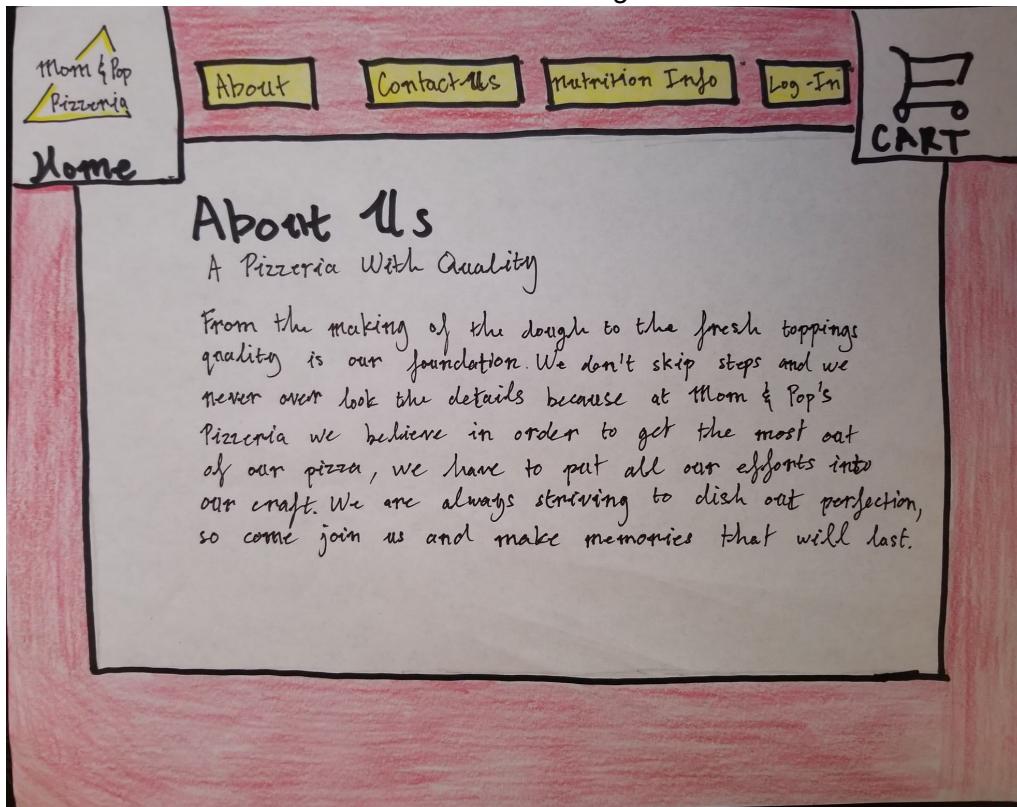
Sub-Flow:

- S-1: If the user selects the Pizza, Drinks, Sides, or Desserts icon, it will direct them to the menu page.
- S-2: If the user clicks the about icon, it will direct them to the about page.
- S-3: If the user clicks the contact us icon, it will direct them to the contact us page.
- S-4: If the user clicks the nutrition info icon, it will direct them to the nutritional information page.
- S-5: If the user clicks the cart icon, it will direct them straight to the cart even if there is nothing in the cart.
- S-6: If the user clicks the Login icon, it will direct them to the Login page.
- S-7: If the user clicks the Home Page icon, it will refresh the page to update the contents.

Alternative Flow:

- N/A

About Us Page:



Requirements: The About Us page should have a simple textbox with a message from the Mom and Pops Store talking about their mission statement. The upper portion of the page should look like the home page with its navigational icons.

Main Flow:

- 1.0: The user can click any of the icons to navigate to its corresponding page/location.

Sub-Flow:

- S-1: If the user clicks the about icon, it will refresh the page.
- S-2: If the user clicks the contact us icon, it will direct them to the contact us page.
- S-3: If the user clicks the nutrition info icon, it will direct them to the nutritional information page.
- S-4: If the user clicks the cart icon, it will direct them straight to the cart even if there is nothing in the cart.
- S-5: If the user clicks the Login icon, it will direct them to the Login page.
- S-6: If the user clicks the Home Page icon, it will direct them to the Home Page.

Alternative Flow:

- N/A

Contact Us Page:



Requirements: The Contact Us page should have a simple textbox that provides the user with the address and the phone number of the store. The upper portion of the page should look like the home page with its navigational icons.

Main Flow:

- 1.0: The user can click any of the icons to navigate to its corresponding page/location.

Sub-Flow:

- S-1: If the user clicks the contact us icon, it will refresh the page.
- S-2: If the user clicks the about us page, it will direct them to the about us page.
- S-3: If the user clicks the nutrition info icon, it will direct them to the nutritional information page.
- S-4: If the user clicks the cart icon, it will direct them straight to the cart even if there is nothing in the cart.
- S-5: If the user clicks the Login icon, it will direct them to the Login page.
- S-6: If the user clicks the Home Page icon, it will direct them to the Home Page.

Alternative Flow:

- N/A

Nutritional Info Page:

Mom & Pop Pizzeria
Home

Nutritional Info

Pizza Size	Small	Medium	Large	Extra Large	Thin Crust
Serving Size	6 slices	8 slices	10 slices	12 slices	N/A
Total Calories	180	190	260	280	220
Total Fat	6g	7g	9g	10g	8g
Total Carbs	26g	27g	39g	41g	21g
Total Protein	7g	7g	10g	11g	8g

Toppings	Calories	Total Fat	Cholesterol	Sodium	Protein
Pepperoni	110	9g	30 mg	440 mg	7g
Sausage	210	16g	60 mg	680 mg	16g
Ham	150	9g	40 mg	800 mg	14g

Requirements: The nutritional info page should have a textbox with two different tables each providing info about the pizza and topping nutritional information. The upper portion of the page should look like the home page with its navigational icons (we will fix that in the final prototype).

Main Flow:

- 1.0: The user can click any of the icons to navigate to its corresponding page/location.

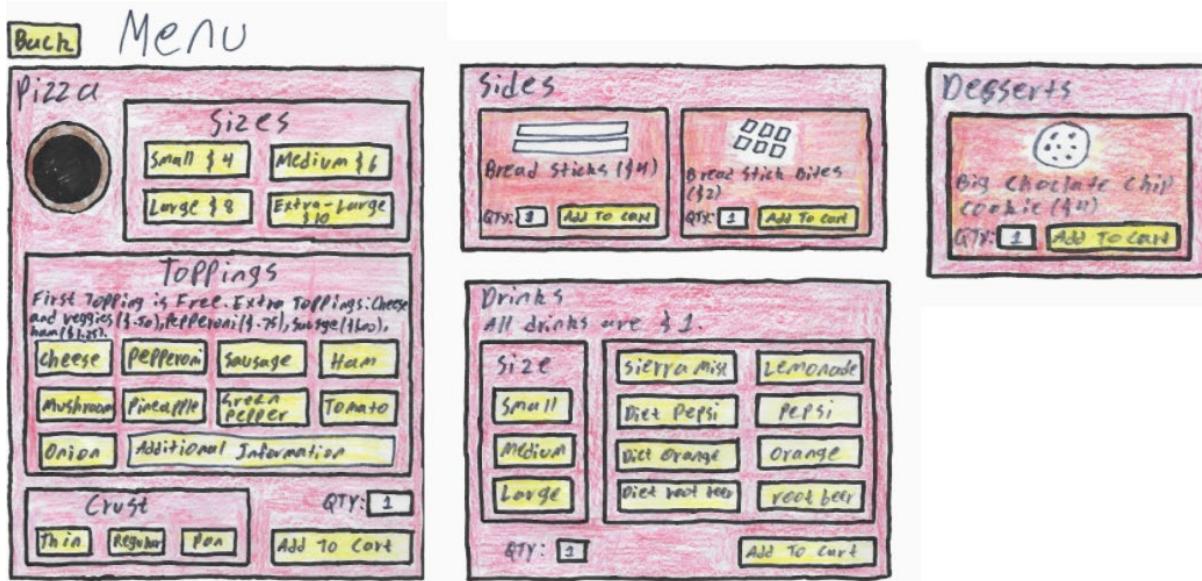
Sub-Flow:

- S-1: If the user clicks the contact us icon, it will direct them to the contact us page.
- S-2: If the user clicks the about us page, it will direct them to the about us page.
- S-3: If the user clicks the nutrition info icon, it will refresh the page
- S-4: If the user clicks the cart icon, it will direct them straight to the cart even if there is nothing in the cart.
- S-5: If the user clicks the Login icon, it will direct them to the Login page.
- S-6: If the user clicks the Home Page icon, it will direct them to the Home Page.

Alternative Flow:

- N/A

Menu Page:



Requirements: The menu page should display four separated boxes each associated with the pizzas, sides, drinks, and desserts. This page should allow the user to choose food items, to choose the size/toppings/types of how they want the food items, and to choose the quantity. It should also have a way to navigate out of the page by either going back to the main menu or by going to the cart.

Main Flow:

- 1.0: All options on the menu should be clickable, changeable in quantity with the QTY. box, and should be addable to the cart with a button.
- 1.1: There should be a cart button so the user can navigate into the cart. (we will have it in our final prototype.)

Sub-Flow:

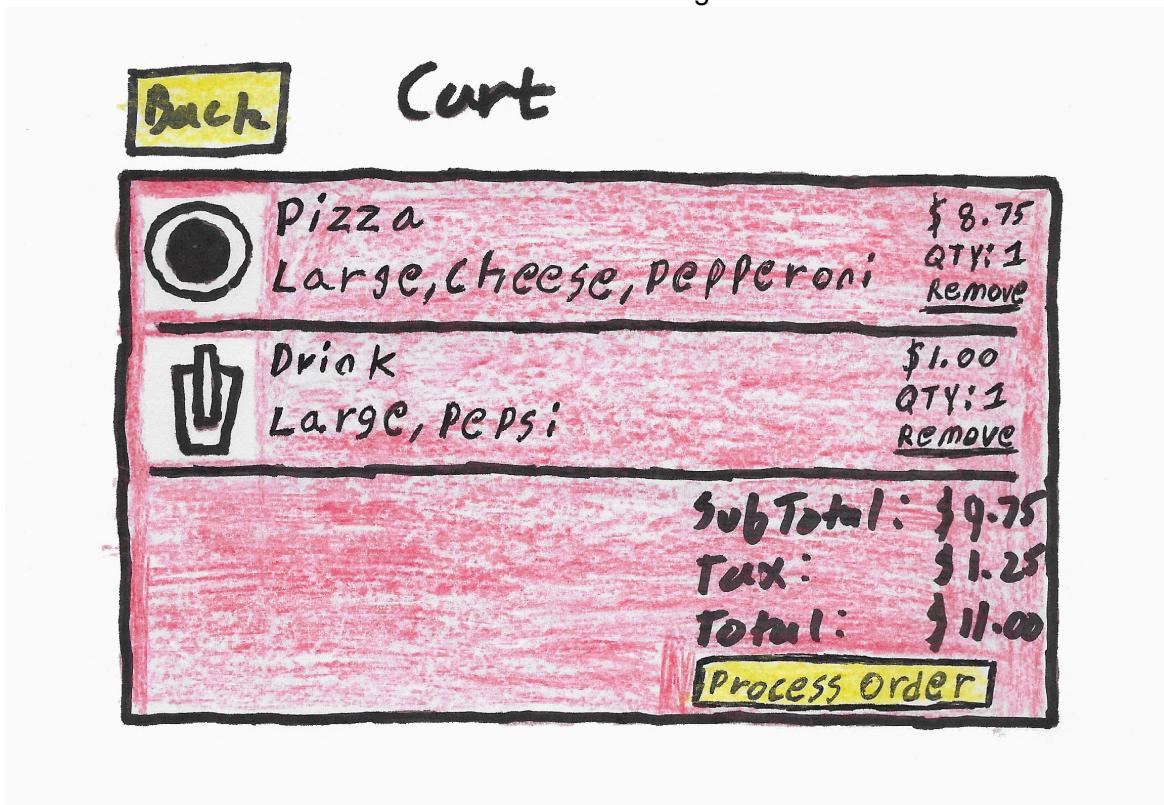
- S-1: The pizza option box should show sizes, toppings, and crust types. Each option in this box should have a box corresponding to the different options on the pizza and a button that adds the pizza (and the amount with the quantity box) into the cart.
- S-2: The sides box should show the different sides, have a quantity box where the user can input the amount they want, and should have a button that adds it to the cart.
- S-3: The drinks box should show and allow the user to click on the different drinks available, have a clickable box corresponding to the different drink sizes, have a quantity box that allows the user to type in the amount of that drink they want, and will have an add to cart button that goes to the cart page.
- S-4: The desserts box should show the different desserts, have a quantity box where the user can input the amount they want, and should have a button that adds it to the cart.
- S-5: There is a back button so the user can navigate out of the menu page and back to the home page.

Alternative Flow:

- E-1: If the user clicks the add to cart button without clicking on any option, an error display will tell the user that an item has not been chosen properly yet.

- E-2: If the user inputs any number less than 0 or any symbol including letters into the quantity box, an error display will tell the user that the quantity is invalid.

Cart Page:



Requirements: The cart page should display the food items chosen on the menu, the Variation/Size/Quantity of the item, the price of the item, and the total of the items added up together. There should be a way to navigate out of the cart page by either going to the menu again or by going to the payment process page.

Main Flow:

- 1.0: The user can view the items in the cart, change the quantity of the items, and can remove an item that they choose that they do not want.
- 2.0: The user can navigate out of the page by clicking the process order button to go to the payment process page or by clicking the back button to go back to the menu.

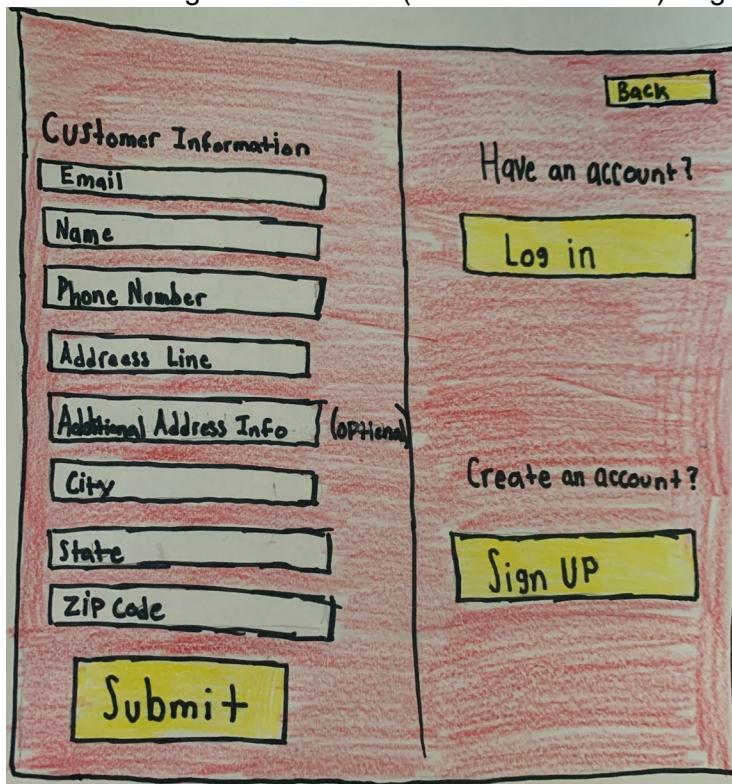
Sub-Flow:

- S-1: The user can change the quantity of the chosen item that they want by clicking the QTY box next to the item and typing their updated amount.
- S-3: The user can remove an item that they do not want by clicking the remove underlined text.

Alternative Flow:

- E-1: If the user types in an updated quantity that is less than zero or is a symbol or letter, an error will display telling the user that an invalid quantity has been selected.
- E-2: If nothing is in the cart, the user cannot process the order and will be notified with an error display saying that nothing is in the cart after clicking on the process order button.

Recording Customer Info (without an account) Page:



Requirements: The recording customer info page (w/o account) page should appear only if the user has not signed into their account in the main menu already. It can record and document the user's contact information or can direct them to the login/sign up pages. Its purpose mainly serves to record customer information if the user does not want to make an account. The user should also be able to navigate back to the cart page if they decide to change their mind.

Main Flow:

- 1.0: The user could choose to log into an account if they already have one, sign up for an account if they want to, or can input all their information to the left if they do not want to make an account.

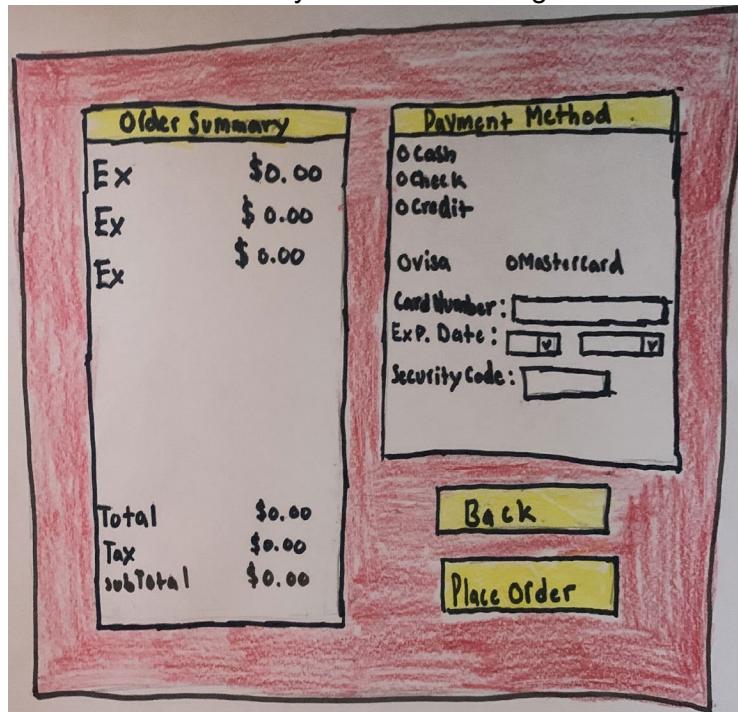
Sub-Flow:

- S-1: The customer will fill out all the fields listed to provide sufficient customer information, once all the fields are filled, they will hit submit and their information will be recorded in the customer database.

Alternative Flow:

- E-1: If the customer inputting information leaves a unique field blank, the customer will be prompted with a "Required field" message that must be filled in before they can submit.
- E-2: If the customer inputs a phone number with improper length, then the customer will be prompted with a "Invalid number" message.
- E-3: If the customer inputs a Zip Code with improper length, then the customer will be prompted with an "Invalid Zip Code" message.

Payment Process Page:



Requirements: The page should resemble a pre-confirmation page by showing the payment inputs on one side with the items and prices on one side. There should also be a button for the customer to travel back to the previous page, and a button to finalize the payment.

Main Flow:

- 1.0: The user inputs their card number, expiration date, and security code in the "Payment Method" box to place the order. Once the customer reviews the Order Summary, they can finalize the payment by clicking on the "Place Order" button.

Sub-Flow:

- S-1: If the user wants to change an item, they can click on the "Back" button to travel back to the previous page to change the needed items.

Alternative Flow:

- E-1: If there is no information in the Card Number slot, Expiration Date slot, and Security Code slot, then the user will be prompted with a red message under each slot's "*Required field"
- E-2: If the customer does not select the payment type (Cash, Credit, or Check), the customer will be prompted with a "Please select one" message before proceeding.
- E-3: If the customer does not select a Credit card type, then the customer will be prompted with a "Please select one" message.

Order Confirmation Page:

Mom & Pop Pizzeria

Order Confirmation

Go Back

Hello (name),
Thank you for placing your order with Mom & Pop.
The following are your order details:

Customer ID: xxxxxx	Online Order Number: xxxxxx
Order Type: <input checked="" type="radio"/> Delivery • <input type="radio"/> Pick Up <input type="checkbox"/> Delivery Address	Method of Payment: <input type="radio"/> Edit <input checked="" type="checkbox"/> E-Signature for Credit
Estimated Ready Time:	
Order Detail:	
1 x Large Pizza (Cheese, Pepperoni)	\$ 8.75
1x Large Drink (Pepsi)	\$ 1.00
Subtotal	\$ 9.75
Delivery	\$ 0.00
Tax	\$ 0.39
Total	\$ 10.14

Confirm

Our location: 120 Main St, Suite 156 | 770-555-1212 | MomandPopPizzeria.com

Requirements: This page displays all the order detail, the customer information, variations of the order information, and estimated time.

Main Flow:

- 1.0: The customer can review their order type to make sure they selected the proper method in which they will receive their order. The customer is also prompted to give a signature should they choose to pay with Credit by filling out the box under "Method of Payment." Once everything is reviewed and filled out(if needed), then the customer can hit confirm to finalize everything.

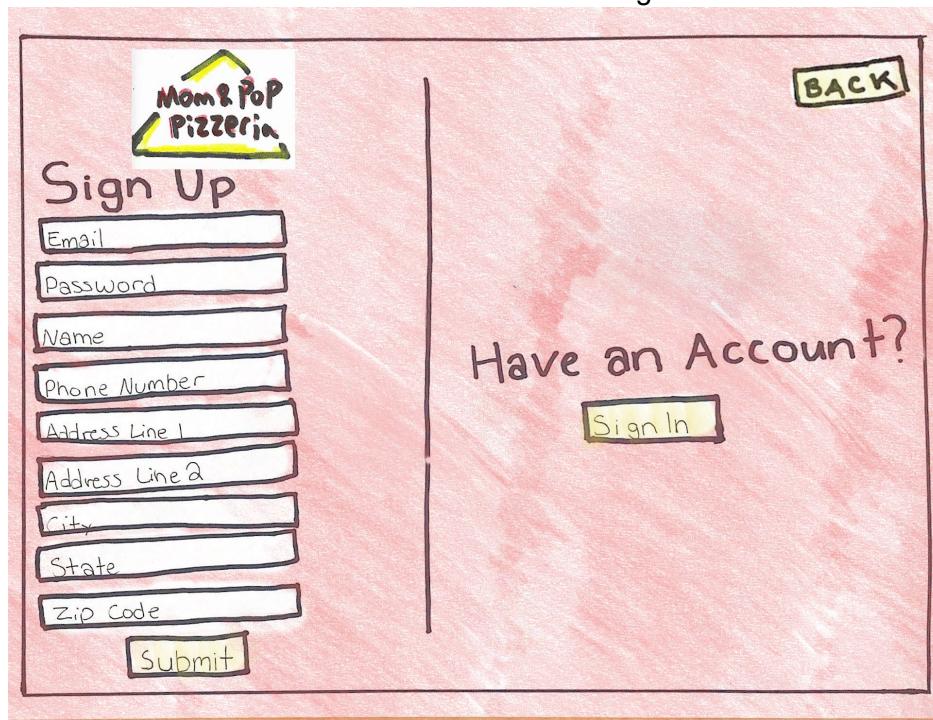
Sub-Flow:

- S-1: If the customer needs to change their Order Type, they can click on the blue "Edit" button and select either Delivery or Pick Up. If the selection was Delivery, the customer can keep or change the Delivery Address.
- S-2: If the customer wants to change the Method of Payment, they can click on the blue "Edit" button and select a different form of payment. Depending on which method of payment they choose, they will be prompted to give a signature.
- S-3: If the customer wants to go back to the previous page, they can select the yellow "Go Back" at the very top center of the Order Confirmation page.

Alternative Flow:

- E-1: If the customer selects Credit as the form of payment, and does not give a signature, then the customer will be prompted with a message “Required field.” before they are allowed to Confirm the order.

Create an Account Page:



Requirements: Prompts the customer for required information to create an account.

Main Flow:

- 1.0: The customer will fill out all the fields listed to create an account, once all the fields are filled, they will hit submit and their account will be created and information will be recorded in the customer database.

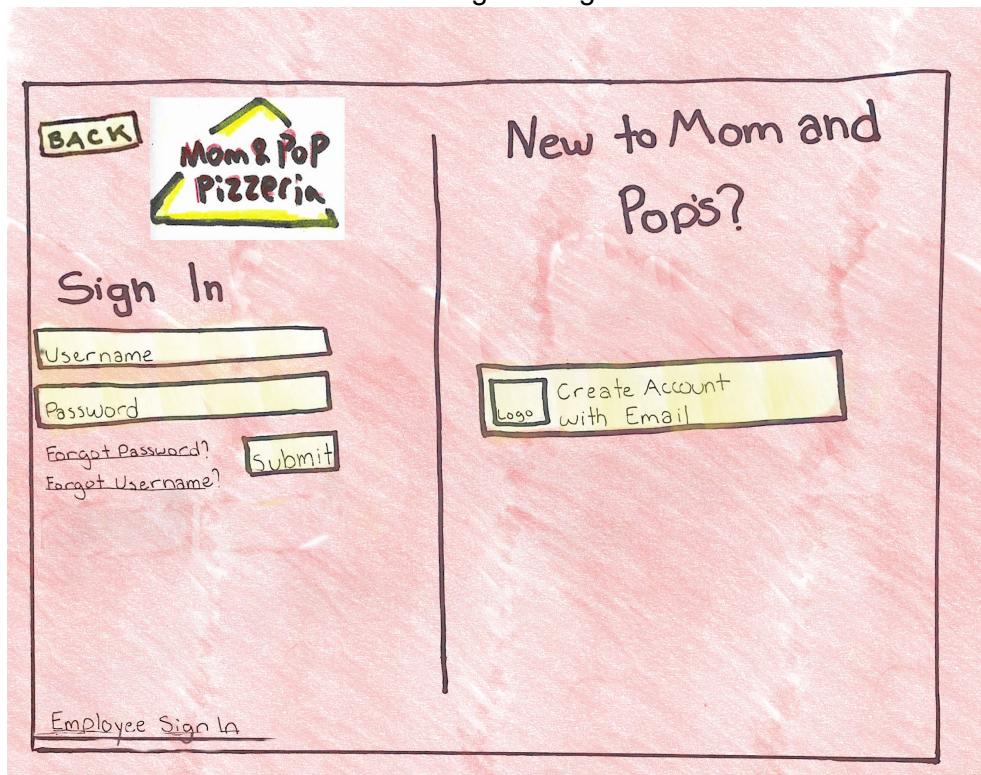
Sub-Flow:

- S-1: If the customer already has an account, they can select the "Sign In" option.

Alternative Flow:

- E-1: If the customer signing up leaves a unique field blank, the customer will be prompted with a "Required field" message that must be filled in before they can submit.
- E-2: If the customer inputs a phone number with improper length, then the customer will be prompted with a "Invalid number" message.
- E-3: If the customer inputs a Zip Code with improper length, then the customer will be prompted with an "Invalid Zip Code" message.

Sign in Page:



Requirements: Customers should be able to log into an existing account or retrieve forgotten information. The employee will also be directed to a unique sign in page.

Main Flow:

- 1.0: The customer inputs their username and password to log into their own personal account. The employees will click "Employee Sign In" at the bottom left corner of the page to sign into their employee account.

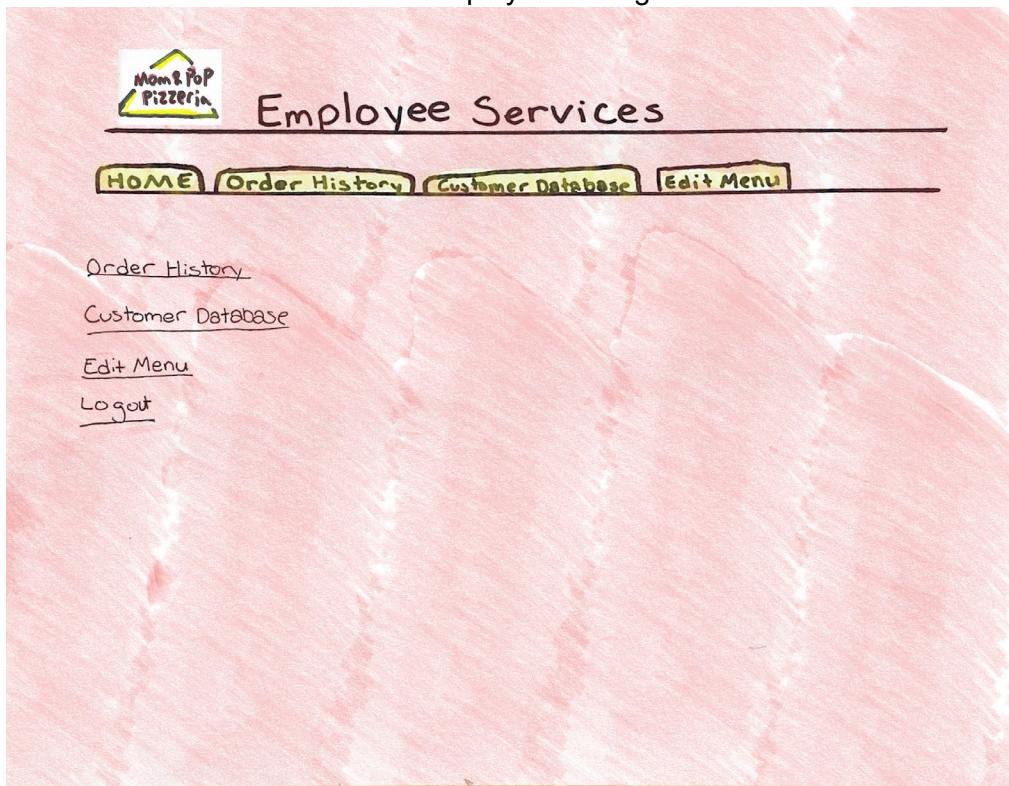
Sub-Flow:

- S-1: If the customer does not have an account, they can create one by clicking "Create Account with Email" button.

Alternative Flow:

- E-1: If the username and password are not correct or do not match, then the customer will be prompted with an "Invalid username/password" message.

Employee UI Page:



Requirements: This page should only be accessible by employees with the options to access Order History, Customer Database, and Edit Menu.

Main Flow:

- 1.0: When signed in through the unique Employee sign in page, the employee is able to access administrative features such as viewing the Order History, Customer Database, and editing the menu through the “Edit Menu” button.

Sub-Flow:

- S-1: If the employee selects the “Home” button they will be redirected to the Employee Services page.
- S-2: If the employee selects the “Order History” button they will be redirected to the Order History page where they can look at all current and past orders.
- S-3: If the employee selects the “Customer Database” button they will be able to access and view the list of customer information that can not be edited.
- S-4: If the employee selects the “Edit Menu” button they will be redirected to the Menu with additional features to change the necessary items.

Alternative Flow:

- N/A

Customer Database Page:

Customer Database

Name	Address Info	Contact	Payment	Other
A. Jones	135 Over Dr.	404 111 3124	Paypal	...
B. William	1751 headhill.	582 131 1242	Credit BoA	...
B. Phil	3BZ Amber	131 121 1211	Debit BoA	...
D. Brown	13B Hampton	123 145 1311	Credit Chase	...

NAME	Date	Time	Total	Paid?	Status
B. William	10/8/20	2:00 PM	21.15	No	unfulfilled
A. Jones	10/8/20	12:15 PM	3.12	Yes	Fulfilled

Requirements: Must be an employee to access.

Main Flow:

- 1.0: The employees can browse and traverse through the customer database to access customer information and order confirmation.

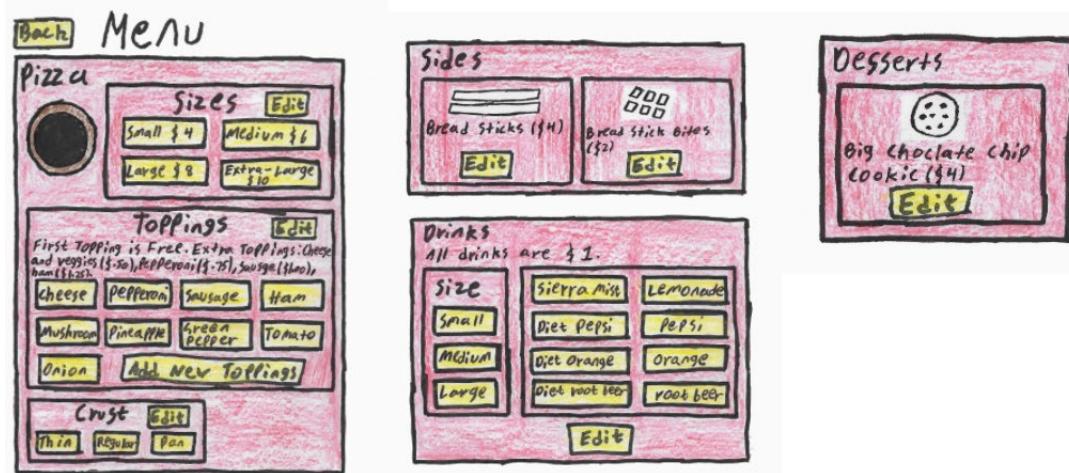
Sub-Flow:

- S-1: The employee or personnel with administrative authorization can use the arrows to reorganize in alphabetical or numerical order (the payment, other, and contact, address, and total should not have arrows, will be fixed)
- S-2: Alternative way to access the Customer Database is in the drop bar menu of profile. Employees and managers will have a unique option of "Customer Database" to browse or navigate through the list of customer information / order confirmation.

Alternative Flow:

- E-1: If an Invalid customer name is entered into the customer database, a prompt will display stating that no customers under that name were found.

Edit Menu Page:



Requirements: The employee modifying the current menu options must be an administrator or given special privileges to change and update/edit menu items.

Main Flow:

- 1.0: In the administrator mode (or higher), all prices, icons, and names are able to be edited. Selecting specific fields will bring up available attributes for the specific fields.
-

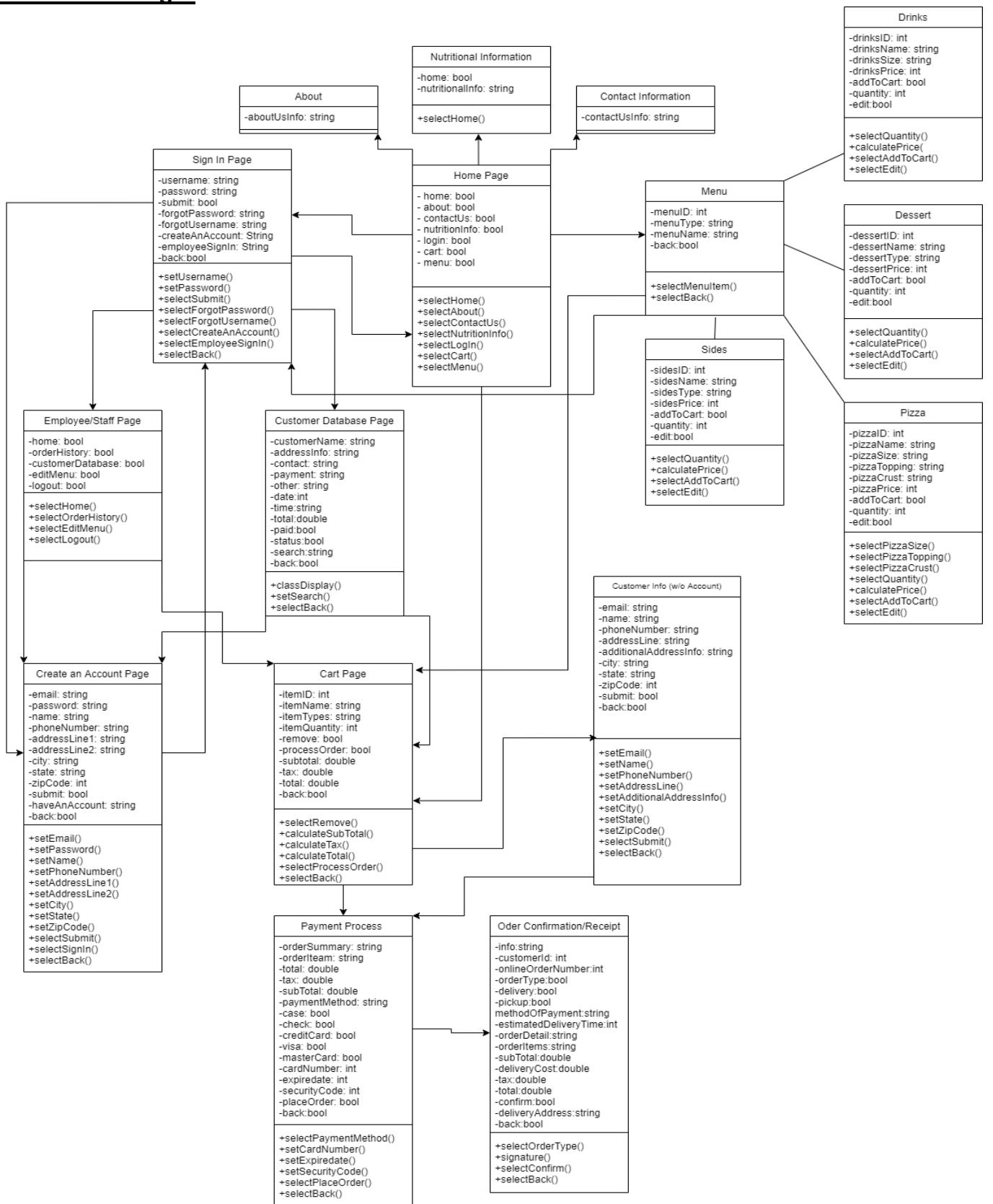
Sub-Flow:

- S-1: Adding a new item to the menu will prompt the employee with a request for price, name, and icon input to be filled out to complete change.
- S-2: Fast-clicking the edit button on the edit menu interface will allow the user to edit the menu
 - 2.1: Long-clicking the specific menu item can remove the item or change the price while an empty box will display to take in user-input for additional menu items that are wanting to be added.

Alternative Flow:

- E-1: If an invalid price for any menu item is entered in the edit function (whether nothing is entered or the price is lower than the total ingredient cost), a prompt will ask the user to enter a valid price.
- E-2: If an Invalid product name is entered (if it is empty), a prompt will ask the user to enter a valid product name.

Technical design:



Class Name: Home page

Category: Data

Hierarchy:

 Superclass (s): Data

Operation Name (s):

 selectHome()

 selectAbout()

 selectContactUs()

 selectNutritionInfo()

 selectLogIn()

 selectCart()

 selectMenu()

Public Member of: Home Page

Documentation:

 /* Stores the Menu and links to the sign in page, about page, nutritional page, contact info page, menu page, and the cart page */

Preconditions: None

Private Attributes: None

Public Attributes:

 home

 about

 contactUs

 nutritionInfo

 logIn

 cart

 and menu

Class Name: About page

Category: Data

Hierarchy:

 Superclass (s): Data

Operation Name (s):

 selectHome()

 selectAbout()

 selectContactUs()

 selectNutritionInfo()

 selectLogIn()

 selectCart()

 selectMenu()

Public Member of: About

Documentation:

 /* Stores the “About Information” about the restaurant and links to the sign in page, about page, nutritional page, contact info page, menu page, and the cart page. */

Preconditions: None

Private Attributes: None

Public Attributes:

aboutUsInfo

Class Name: Nutrition page

Category: Data

Hierarchy:

Superclass (s): Data

Operation Name (s):

selectHome()

Public Member of: Nutrition

Documentation:

/* Stores the “Nutrition Information” about the restaurant and links to the Home Page. */

Preconditions: None

Private Attributes: None

Public Attributes:

home

nutritionallInfo

Class Name: Contact Information page

Category: Data

Hierarchy:

Superclass (s): Data

Operation Name (s):

selectHome()

selectAbout()

selectContactUs()

selectNutritionInfo()

selectLogIn()

selectCart()

selectMenu()

Public Member of: Contact Information

Documentation:

/* Stores the “About Information” about the restaurant and links to the sign in page, about page, nutritional page, contact info page, menu page, and the cart page. */

Preconditions: None

Private Attributes: None

Public Attributes:

contactUsInfo

Class Name: Menu page

Category: Data

Hierarchy:

Superclass (s): Data, Home Page

Operation Name (s):

selectMenuItem()

selectBack()

Public Member of: Menu

Documentation:

/* Stores all the information about the menu (Pizza, Sides, Dessert, and Drinks) */

Preconditions: None

Private Attributes: None

Public Attributes:

menuID

menuType

menuName

back

Class Name: Pizza

Category: Data

Hierarchy:

Superclass (s): Data, Menu Page

Operation Name (s):

selectPizzaSize()

selectPizzaTopping()

selectPizzaCrust()

selectQuantity()

calculatePrice()

selectAddToCart()

selectEdit()

Public Member of: Pizza

Documentation:

/* Displays the information about the pizza (Size, Toppings, Crust, Quantity, and Price) */

Preconditions: None

Private Attributes: None

Public Attributes:

pizzaID

pizzaName

pizzaSize

pizzaTopping

pizzaCrust

pizzaPrice

addToCart

quantity
edit

Class Name: Sides

Category: Data

Hierarchy:

Superclass (s): Data, Menu Page

Operation Name (s):

selectQuantity()
calculatePrice()
selectAddToCart()
selectEdit()

Public Member of: Sides

Documentation:

/* Displays the information about the Sides (Quantity) */

Preconditions: None

Private Attributes: None

Public Attributes:

sidesID
sidesName
sidesPrice
addToCart
quantity
edit

Class Name: Drinks

Category: Data

Hierarchy:

Superclass (s): Data, Menu Page

Operation Name (s):

selectQuantity()
calculatePrice()
selectAddToCart()
selectEdit()

Public Member of: Drinks

Documentation:

/* Displays the information about the Drinks () */

Preconditions: None

Private Attributes: None

Public Attributes:

drinksID
drinksName
drinksSize
drinksPrice

addToCart
quantity
edit

Class Name: Dessert

Category: Data

Hierarchy:

Superclass (s): Data, Menu Page

Operation Name (s):

selectQuantity()
calculatePrice()
selectAddToCart()
selectEdit()

Public Member of: Dessert

Documentation:

/* Displays the information about the dessert (Quantity, and Price) */

Preconditions: None

Private Attributes: None

Public Attributes:

dessertID
dessertName
dessertType
dessertPrice
addToCart
quantity
edit

Class Name: Cart Page

Category: Data

Hierarchy:

Superclass (s): Data, Menu Page

Operation Name (s):

selectRemove()
calculateSubTotal()
calculateTax()
calculateTotal()
selectProcessOrder()
selectBack()

Public Member of: Cart

Documentation:

/* Displays the order information (Price, Removal, Tax, Subtotal, and Total,)*/

Preconditions: None

Private Attributes: None

Public Attributes:

itemID

itemName
 itemTypes
 itemQuantity
 remove
 processOrder
 subtotal
 tax
 total
 Back

Class Name: Sign In Page

Category: Data

Hierarchy:

Superclass (s): Data, Home Page

Operation Name (s):

setUsername()
 setPassword()
 selectSubmit()
 selectForgotPassword()
 selectCreateAnAccount()
 selectEmployeeSignIn()
 selectBack()

Public Member of: Sign In

Documentation:

/* Displays the information related to Signing In (UserName, Password, Submit, Forgot Password option, Create An Account, Employee Sign in) */

Preconditions:

//Sign in information
 Username is not NULL
 Password is not NULL

Private Attributes: None

Public Attributes:

username
 password
 submit
 forgotPassword
 forgotUsername
 createAnAccount
 employeeSignIn
 back

Class Name: Create An Account Page

Category: Data

Hierarchy:

Superclass (s): Data, Sign In Page

Operation Name (s):

- setEmail()
- setPassword()
- setName()
- setPhoneNumber()
- setAddressLine1()
- setAddressLine2()
- setCity()
- setState()
- setZipCode()
- selectSubmit()
- selectSignIn()
- selectBack()

Public Member of: Create An Account

Documentation:

/* Displays the information to Create An Account (Email, Password, Name, Address Line 1 &2 (optional), City, State, Zip Code) */

Preconditions:

- // Create An Account
- Email is not NULL
- Password is not NULL
- Name is not NULL
- Phone Number NULL
- Address Line 1 is not NULL
- City is not NULL
- State is not NULL
- Zip Code is not NULL

Private Attributes: None

Public Attributes:

- email
- password
- name
- phoneNumber
- addressLine1
- addressLine2
- city
- state
- zipCode
- submit
- haveAnAccount

back

Class Name: Payment Page

Category: Data

Hierarchy:

Superclass (s): Data, Cart Page, and Customer Info Page

Operation Name (s):

- selectPaymentMethod()
- setCardNumber()
- setExpirationDate()
- setSecurityCode()
- selectPlaceOrder()
- selectBack()

Public Member of: Payment

Documentation:

/* Displays the information about the Payment Process (Payment Method, Card Number, Expiration Date, and Security Code) */

Preconditions:

- //Payment Info
- Payment method must be selected
- //If Card selected
- Card Number is not NULL
- Expiration Date is not NULL
- Security Code is not NULL

Private Attributes: None

Public Attributes:

- orderSummary
- orderItem
- total
- tax
- subTotal
- paymentMethod
- cash
- check
- creditCard
- visa
- masterCard
- cardNumber
- expireDate
- securityCode
- placeOrder
- back

Class Name: Customer Info Page (w/o Account)

Category: Data

Hierarchy:

Superclass (s): Data, Cart Page

Operation Name (s):

- setEmail()
- setName()
- setPhoneNumber()
- setAddressLine()
- setAdditionalAddressInfo()
- setCity()
- setState()
- setZipCode()
- selectSubmit()
- selectSignUp()
- selectLogIn()
- selectBack()

Public Member of: Customer Info

Documentation:

/* Displays the information if the user has decided not to create a account (Email, Name Phone Number, Address Line 1 & 2 (optional), City, State, Zip Code, Submit) */

Preconditions:

- //Customer Info
- Email is not NULL
- Name is not NULL
- Phone Number NULL
- Address Line is not NULL
- City is not NULL
- State is not NULL
- Zip Code is not NULL

Private Attributes: None

Public Attributes:

- email
- name
- phoneNumber
- addressLine
- additionalAddressInfo
- city
- state
- zipCode
- submit
- haveAnAccount
- createAnAccount
- Back

Class Name: Order Confirmation/ Receipt Page

Category: Data

Hierarchy:

Superclass (s): Data, Payment Page

Operation Name (s):

- selectOrderType()
- signature()
- selectConfirm()
- selectBack()

Public Member of: Order Confirmation/Receipt

Documentation:

/* Displays the information after the order has been placed (Order Type, and Signature)*/

Preconditions:

- //OrderConfirmation
- Needs to order something
- Signature is required when card is used
- Must select Order Type

Private Attributes: None

Public Attributes:

- info
- customerID
- onlineOrderNumber
- orderType
- delivery
- pickup
- methodOfPayment
- estimatedDeliveryTime
- orderDetail
- orderItems
- orderItems
- subTotal
- deliveryCost
- tax
- total
- confirm
- deliveryAddress
- back

Class Name: Customer Database Page

Category: Data

Hierarchy:

Superclass (s): Data, Sign In Page

Operation Name (s):

- classDisplay()
- setSearch()
- selectBack

Public Member of: Customer Database**Documentation:**

/* Displays the information about the Customer Database */

Preconditions: None**Private Attributes:** None**Public Attributes:**

- customerName
- addressInfo
- contact
- payment
- other
- date
- time
- paid
- status
- search
- back

Class Name: Employee/Staff Page**Category:** Data**Hierarchy:**

Superclass (s): Data, Sign In Page

Operation Name (s):

- selectHome()
- selectOrderHistory()
- selectEditMenu()
- selectLogout()

Public Member of: Employee/Staff**Documentation:**

/* Displays the information about the Employee/Staff */

Preconditions: None**Private Attributes:** None**Public Attributes:**

- home
- orderHistory
- customerDatabase
- editMenu

Database Table

Menu	
Label	Type
Pizza(Size)	string
Pizza(Price)	double
Pizza(Pizza_Id)	int
Pizza(Topping)	string
Pizza(Crust_Type)	string
Drinks(Size)	string
Drinks(Price)	double
Drinks(Drink_ID)	int
Drink(Drink_Name)	string
Dessert(Price)	double
Dessert(Dessert_ID)	int
Dessert(Dessert_Name)	string
Sides(Price)	double
Sides(Side_ID)	int
Sides(Side_Name)	string
Sides(Size)	string

Order	
Label	Type
Order ID	int
Order Status	string

Time In	date
Time Out	date
Delivery Option	string
Total Price	double

Employee	
Label	Type
Employee ID (Emp_ID)	int
First Name	string
Last Name	string
Hire Date	date
Wage	double

Delivery Driver	
Label	Type
License Number	string
Car License Plate	string
Car Model	string
Insurance Number	string

Address	
Label	Type
City	string
Address Line 1	string
Address Line 2	string
State	string
Zip	string

Basic Customer	
Label	Type
First Name	string
Last Name	string
Phone Number	string

Member Customer	
Label	Type
First Name	string
Last Name	string
Phone Number	string
Email	string
Password	string
Customer ID	string

Payment Methods	
Label	Type
Payment ID	int
Type	string
Amount Paid	double

Technical Support Specifications:

Technical Support Specifications

If you experience any issues with this application, please contact the technical support team at:

Mom and Pop's Pizzeria

Operating hours:

Sunday - Thursday 11am-9pm

Friday and Saturday 11am – Midnight

MomAndPopPizzeria.com

680 Arntson Rd, suite 156

770-555-1212

System must run:

Windows 7+

Critical - Arises when the product when the product is failing to perform to the specifications and is severely impacting the user's operations and/or security	2 hours
Urgent - Arises when the product is failing to perform to the specifications. Neither operations nor security is at risk, but productivity is negatively impacted.	4 hours
Standard - Arises when the product is failing to perform to specifications. The user is only moderately impacted if at all.	12 hours
Low - Reserved for general inquiries and enhancement requests.	24 hours

Weekly Status Report

Week 1:

1.

TO: Professor. Gesick

FROM: Jason Paek, Martin Nguyen, Jasmine Brown, Mohammad Umar, Zach Stover

Subj: Project Status

Date: 10/6/2020

Time: 7:15PM - 9:00PM

Activities Completed this week:

- > Discuss and understand the requirements and what needs to be done on the prototype.
- > Assign different pages to different people

Activities Needed to be completed next week:

- > Continue working on the prototype and at least have the overall design, font, and color decided and made.

Issues that require attention:

- > We need to pay attention to the programming logic and the flow of the UI so we don't mess up the GUI later on in the project.

Other Comments or notes:

- > We need to make sure that everyone understands the GUI flow and everything makes sense.

Week 2:

2.

TO: Professor. Gesick

FROM: Jason Paek, Martin Nguyen, Jasmine Brown, Mohammad Umar, Zach Stover

Subj: Project Status

Date: 10/13/2020

Time: 5:00PM - 6:00 PM

Activities Completed this week:

> Went over flow logic, decided on overall design, and started finalizing the project.

Activities Needed to be completed next week:

> We need to begin the Requirement documentation and assign the roles.

Issues that require attention:

> Fix major issues with the design (colors, lines, and fonts were inconsistent)

Other Comments or notes:

> Zach and Jason had concerns with the logical flow of the Gui concerning how the flow would work for a user with and a user without an account.

Week 3:

3.

TO: Professor. Gesick

FROM: Jason Paek, Martin Nguyen, Jasmine Brown, Mohammad Umar, Zach Stover

Subj: Project Status

Date: 10/20/2020

Time: 5:00PM - 5:20PM

Activities Completed this week:

- > Assigned Requirement documentation roles
- > Discussed and Understood the requirement documentation.

Activities Needed to be completed next week:

- > Share progress of Requirement documentations to see progress and ask questions if needed.
- > Start working on the Requirement documentation.

Issues that require attention:

- > Jasmine had issues with the ER diagram so Mohammad helped.

Other Comments or notes:

- > Martin and Jason work on Use case Flow documentation and Use case Diagram. Also work on the state transition diagram.
- > Zach and Mohammed work on Class documentation and class diagram
- > Jasmine work on ER diagram
- > Everyone work on Requirement documentation

Week 4:

4.

TO: Professor. Gesick

FROM: Jason Paek, Martin Nguyen, Jasmine Brown, Mohammad Umar, Zach Stover

Subj: Project Status

Date: 10/27/2020

Time: 5:00PM - 6:00PM

Activities Completed this week:

- > Looked over Prototype one together and turned it in.
- > Discussed on how to approach the Requirements documentation.
- > Assigned parts of the documentation to groups.
- > Began working on the Requirements documentation.

Activities Needed to be completed next week:

- > finish the Requirement documentation

Issues that require attention:

- > Nothing to note.

Other Comments or notes:

- > Jasmine is waiting on feedback from the teacher to get technical support information.
- > Everyone has completed their work so we have completed the documentation.

Week 5:

5.

TO: Professor. Gesick

FROM: Jason Paek, Martin Nguyen, Jasmine Brown, Mohammad Umar, Zach Stover

Subj: Project Status

Date: 11/03/2020

Time: 5:00PM - 5:30PM

Activities Completed this week:

- > Shared their work progress on the Requirements documentation.
- > Worked together on the last section of Requirements documentation as a team.

Activities Needed to be completed next week:

- > Finish and review the Requirements documentation for turn in

Issues that require attention:

- > There were no issues so far.

Other Comments or notes:

- > Everyone began and worked without any problems.

6.

TO: Professor. Gesick

FROM: Jason Paek, Martin Nguyen, Jasmine Brown, Mohammad Umar, Zach Stover

Subj: Project Status

Date: 11/05/2020

Time: 5:00PM - 6:00PM

Activities Completed this week:

- > Shared and reviewed the work for Requirements documents.
- > Formatted and compiled the Requirement Documents and turned the documents in.

Activities Needed to be completed next week:

- > Start on the System Design Documents

Issues that require attention:

- > Some minor formatting issues.

Other Comments or notes:

- >Everyone finished their work on time for the Requirements documents.

Week 6:

7.

TO: Professor. Gesick

FROM: Jason Paek, Martin Nguyen, Jasmine Brown, Mohammad Umar, Zach Stover

Subj: Project Status

Date: 11/12/2020

Time: 5:00PM - 5:20PM

Activities Completed this week:

- > Discussed the System Design Documents
- > Divided the work between team members(2 for conceptual, 2 for technical design, and 1 for technical support.

Activities Needed to be completed next week:

- > Show progress of what we have so far and see if anyone one needs help with something from their part of the work.

Issues that require attention:

- > No issues or problems.

Other Comments or notes:

- > Nothing to note.

Week 7:

8.

TO: Professor. Gesick

FROM: Jason Paek, Martin Nguyen, Jasmine Brown, Mohammad Umar, Zach Stover

Subj: Project Status

Date: 11/17/2020

Time: 5:00PM - 5:20PM

Activities Completed this week:

- > Showed progress so far and how much of the System Design Documents were completed to keep track of time and time management.

Activities Needed to be completed next week:

- > Start on the software implementation.
- > Submit the System Design Documentation prior to the start of next week.

Issues that require attention:

- > No issues thus far!

Other Comments or notes:

- > So far everyone is on track with work and schedule.

9.

TO: Professor. Gesick

FROM: Jason Paek, Martin Nguyen, Jasmine Brown, Mohammad Umar, Zach Stover

Subj: Project Status

Date: 11/19/2020

Time: 5:00PM - 5:45PM

Activities Completed this week:

- > System Design Documents were completed and compiled to turn in.
- > Made sure all group members understood how to use Live share in Visual studios.

Activities Needed to be completed next week:

- > The coding aspect of the GUI and making sure that the UI is proper.
- > Format and compile the Final notebook.

Issues that require attention:

- > Live share does not allow every member to see the GUI application page so Mohammed (the project owner) is the only one able to see the project when it is running. We have to do everything with Mohammed.

Other Comments or notes:

- > Everyone needs to meet with Mohammed to work on the GUI since liveshare requires the host. If anyone misses the meeting, please refer to Mohammed to work on the QUI.

Week 8:

10.

TO: Professor. Gesick

FROM: Jason Paek, Martin Nguyen, Jasmine Brown, Mohammad Umar, Zach Stover

Subj: Project Status

Date: 11/24/2020

Time: 5:00PM - 6:15PM

Activities Completed this week:

- > Completed the coding aspect of the GUI design to make sure the buttons were functional.
- > Formatted and compiled everything required for the Final notebook submission.

Activities Needed to be completed next week:

- >Create a video presentation with all the group members.
- >Review what has been done for the final notebook.

Issues that require attention:

- > Mohammad needs to review content with Zach for missing the meeting.

Other Comments or notes:

- >Everyone has done all of their work!

11.

TO: Professor. Gesick

FROM: Jason Paek, Martin Nguyen, Jasmine Brown, Mohammad Umar, Zach Stover

Subj: Project Status

Date: 12/01/2020

Time: 4:50PM-8:00PM

Activities Completed this week:

- > final notebook completed
- > Presentation
- > Peer evaluation
- >Prototype 2

Activities Needed to be completed next week:

- > Nothing

Issues that require attention:

- > Nothing really

Other Comments or notes:

- > Nothing.

Project Log																				
Zach Stover	DATE	TIME	DESCRIPTION	Mohammad Umar	DATE	TIME	DESCRIPTION	Jason Peek	DATE	TIME	DESCRIPTION	Jasmine Brown	DATE	TIME	DESCRIPTION	Martin Nguyen	DATE	TIME	DESCRIPTION	
			Assigned Prototype Section - To be completed 10/13				Assigned Prototype Section - To be completed 10/13				Assigned Prototype Section - To be completed 10/13				Assigned Prototype Section - To be completed 10/13				Assigned Prototype Section - To be completed 10/13	
	10/6	5:00 - 7:30				10/6 7:15-8:55					7:15PM - 9:00PM				10/6 8:00-8:50				10/6 7:30pm-9:00pm	
																			Hand Draft of First sprint creating for contact us, nutritional info, order confirmation, and about us.	
	10/10	4:00 - 6:50	Hand Draft of Prototype Created			10/13 5:00-6:00	Prototype Finalization			10/10 PM	5:00PM - 7:00	Digital Draft of prototype created and worked on. Stayed later with Martin to fix minor issues and certain pages		10/13 3:00-4:00	Hand Draft of Prototype Created				10/10 6:00PM-7:45PM	Digital Draft of prototype created and worked on. Stayed later with Jason to fix certain pages
			Digital Draft of Prototype Created and worked on			10/20 5:00-5:20	Assigned Documentation Section			10/13 PM	5:00PM - 8:00			10/13 11:30-1:00	Digital Draft of Prototype Created				10/13 5:00-8:00	
																			Finalized the design of prototype and stayed late with Jason to look over slides, fix minor issues, and submit the prototype.	
	10/15	5:00-6:00	Prototype Finalization			10/27 5:00-6:04	Groundwork and Groups assigned for documentation			10/15 30PM	5:00PM - 6:00	Going over and assigning roles for the documentation		10/13 5:00-6:00	Prototype Finalization				10/15 30PM	Finalized the design of prototype and stayed late with Jason to look over slides, fix minor issues, and submit the prototype.
	10/20	5:00-5:20	Assigned Documentation Section			11/3 5:00-5:30	Documentation Check			10/20 5:00 - 5:20 PM				10/20 5:00-5:20	Assigned Documentation Section				10/20 5:00PM-5:20PM	Assigned Documentation Section
	10/27	5:00-6:04	Groundwork and Groups assigned for documentation			11/5 5:00-6:00	Final Requirements Checked and Submitted			10/27 5:00-6:04		Groundwork and Groups assigned for documentation		10/27 5:00-5:30	Groundwork and Groups assigned for documentation				10/27 5:00PM-6:04PM	Groundwork and Groups assigned for documentation
	11/3	5:00-5:30	Documentation Check			11/12 5:00-5:20	System Requirements Assigned			11/3 5:00-5:30		Documentation Check		11/3 12:00-2:00	ER Diagram Completed				11/3 5:00PM-5:30PM	Documentation Check
	11/5	5:00-6:00	Final Requirements Checked and Submitted			11/17 5:00-5:20	System Requirements Check in			11/5 5:00-6:00		Final Requirements Checked and Submitted		11/3 5:00-5:30	Documentation Check				11/5 5:00PM-6:00PM	Final Requirements Checked and Submitted
	11/12	5:00-5:20	System Requirements Assigned			11/19 5:00-5:45	Live Visual Studio Setup			11/12 5:00-5:20		System Requirements Assigned		11/4 1:15-1:30	ER Diagram Fix				11/12 5:00PM-5:20PM	System Requirements Assigned
	11/17	5:00-5:20	System Requirements Check in			11/24 5:00-6:15	GUI Creation and Finalization			11/17 5:00-5:20		System Requirements Check in		11/5 5:00-6:00	Final Requirements Checked and Submitted				11/17 5:00PM-5:20PM	System Requirements Check in
	11/19	5:00-5:45	Live Visual Studio Setup			11/19 5:00-5:45	Live Visual Studio Setup			11/19 5:00-5:45		Live Visual Studio Setup		11/12 5:00-5:20	System Requirements Assigned				11/19 5:00PM-5:45PM	Live Visual Studio Setup
																			GUI Creation and Finalization. Stayed later with Martin to work on compiling the final notebook	
	11/24	5:00-6:15	GUI Creation and Finalization			11/24 5:00-8:00	GUI Creation, comments for code, Finalization Prototype 2			11/24 5:00-8:00PM				11/19 5:00-5:45	Live Visual Studio Setup				11/24 5:00PM-8:00PM	GUI Creation and Finalization. Stayed later with Martin to work on compiling the final notebook
																				Final Presentation and submitting prototype 2
	12/1	5:00-8:00	Finishing Presentation			12/1 5:00-8:00 PM	Finished presentation, prototype 2			12/1/2020 5:00-8:00 PM				11/24 5:00-6:15	GUI Creation and Finalization				12/1 5:00-8:00pm	
																				Finished presentation and submission