Mama's Pizzeria
System Design Documents
Table of Contents

Register Receipt:

Mama's Pizzeria

(xxx) xxx-xxxx

MM/DD/YYYY   00:00

Order#   0000000000

Customer   Josh Doe

Cashier   John Doe

1	item	$00.00
1	item	$00.00
1	item	$00.00
1	item	$00.00

Subtotal $00.00

Tax (8.00%) $00.00

Total $00.00

Payment   xxxxxxxxxxxx0000

Customer Copy

Order Slip:

Mama's Pizzeria

(xxx) xxx-xxxx

MM/DD/YYYY   00:00

Order#   0000000000

Customer   Josh Doe

Cashier   John Doe

1	item	$00.00
1	item	$00.00
1	item	$00.00
1	item	$00.00

Payment Status   PAID

Order Slip

Daily Business Report/Summary:

Mama's Pizzeria

MM/DD/YYYY	00:00
Summary Date	MM/DD/YYYY
Manager	John Doe

Orders	000

Cash Sales	$0000.00
Check Sales	$0000.00
Card Sales	$0000.00
Total Sales	$0000.00

Payroll	$0000.00
Ingredient Cost	$0000.00
Total Cost	$0000.00

Net Profit	$0000.00

Low Stock Alert	00
Sausage	00
Sausage	00
Sausage	00

Daily Summary
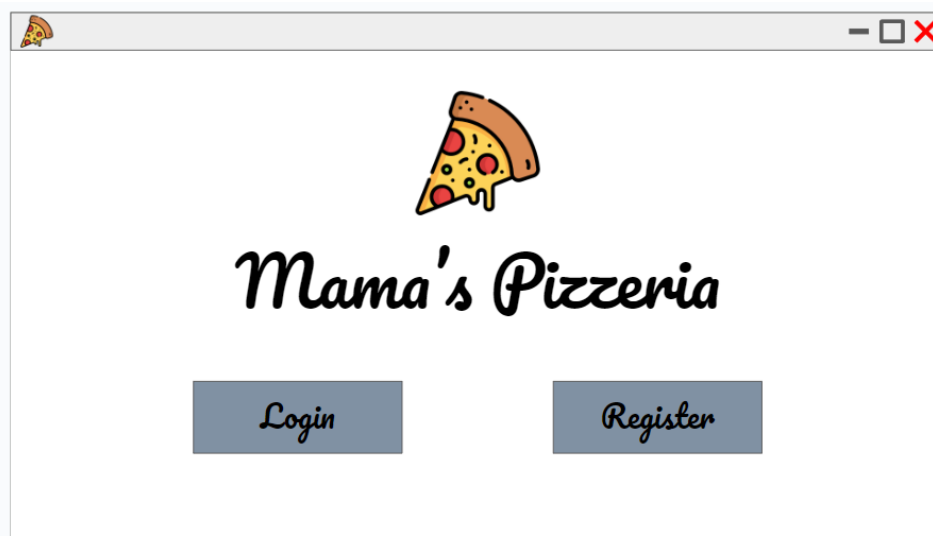
Starting Screen: This allows users to login or register.

Log-in Screen: This allows users to log in to their account.

Register Page

Required (*)

First Name *   Last Name *   Address: (ex 1234 Pizza Street) *

Email: (ex 123@gmail.com) *   Apt: (ex 111)   City: (Ex Marietta)

Phone: (ex 1234567890) *   State: (ex GA)   Zip: (ex 111111) *

Register   Back

Register Page: This allows users to register for an account.

Mama's Pizzeria

Home   Order Online   Personal Information   About Us   Log Out

Pizza Sizes		Toppings		1 Free Topping .50 per topping	Drinks		
10in.	$10.	Pepperoni	Cheese		Small $3	Large $4	2 Liter $5
12in.	$13.	Mushrooms	Sausage		Water		Pepsi
14in.	$15.	Bell Peppers	Jalapeno Peppers		Diet Pepsi		Dr. Pepper
16in.	$17.	Onion	Pineapple		Diet Dr. Pepper		
		Olives					

Stuffed Crust

Thin Crust   Deep Dish

Menu Page: This allows users to view the menu.

Order Screen: This will allow users to place orders and view orders.

Payment Type Screen: This allows users to select their preferred payment type.

Confirmation Screen: This allows users to view their full order and confirm the order.
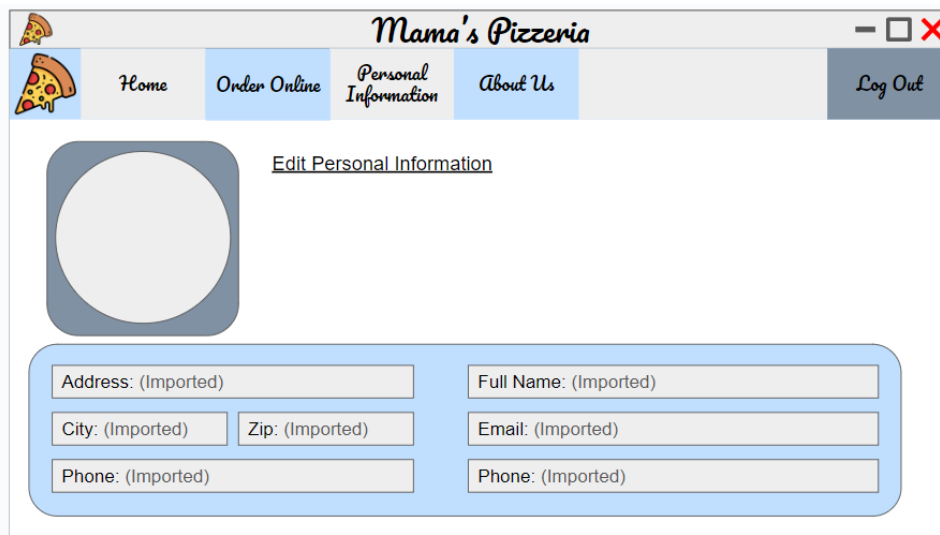

Card Change Screen: This allows users to change their card information.

Thank You Screen: User does nothing here; it is just here as a thank you.

Change Information Screen(Customers): This will allow users to change their personal information.

Change Information(Employees): This will allow managers to change information of employees.

About Us Screen: Users can see store information here.
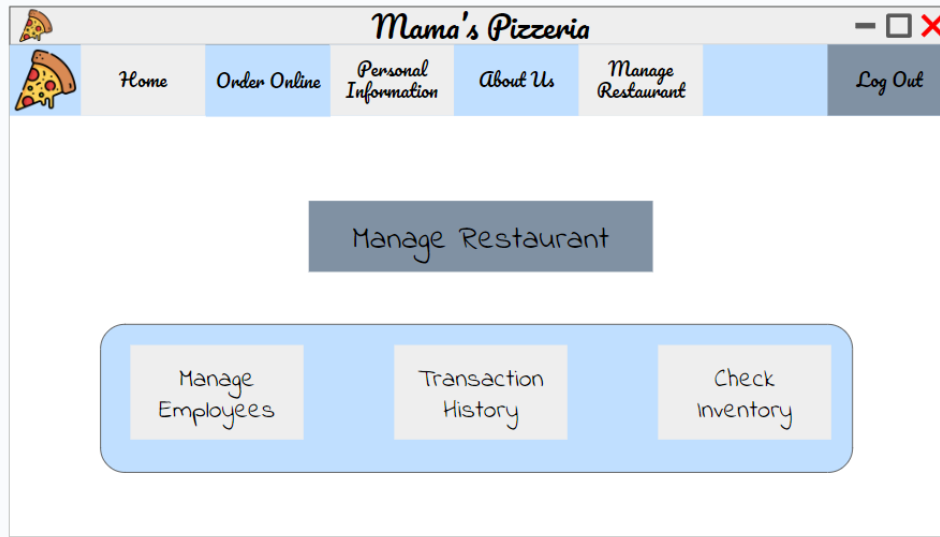
Manager Home Screen: This allows managers to select from different options.

Manage Employee Screen: This allows managers to manage employee information.

Transaction History Screen: This will allow managers to view orders to see what is being ordered the most, least, etc.

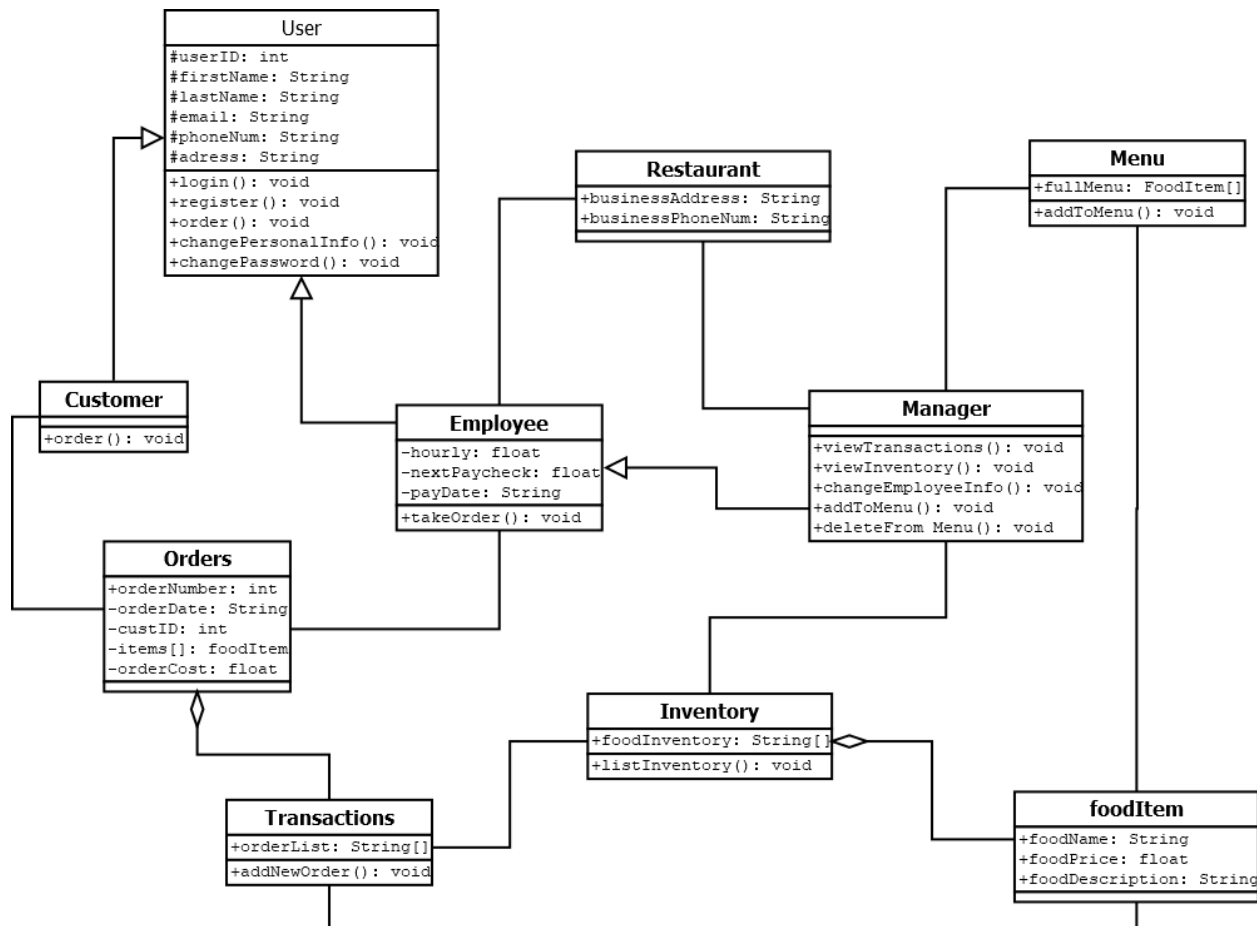Check Inventory Screen: This allows managers to check the inventory of specific ingredients and other products.

Class Documentation:

User Class contains all attributes and methods that any user of the system will require.

- **Attributes**
  - **userID** contains a unique ID for every registered user
  - **firstName** and **lastName** contains each user's first and last name
  - **email**, **phoneNum**, and **address** contain Strings holding their respective data
  - **username** and **password** contain specific Strings for each user (encrypted)
- **Methods**
  - void **login()** takes in user's imputed username and password and compares it with their saved username and password, and lets them into the system if their info matches exactly
  - void **register()** takes in firstName, lastName, email, phoneNum, address, username, and password, and creates a new user
  - void **changePersonalInfo()** takes in firstName, lastName, email, phoneNum, and address, and replaces each existing variable value
  - void **changePassword()** takes in a username and a new password, and replaces the existing password with the new one.

Employee Class inherits from User and contains new attributes and a new method specific to employees.

- **Attributes**
  - **Hourly** contains the hourly salary of the employee if applicable(otherwise is zero)
  - **nextPaycheck** contains the sum of money being paid out directly in the employee's next paycheck if applicable (otherwise is zero)
  - **payDate** contains a String listing the date of the employee's next paycheck
- **Methods**
  - void **takeOrder()** takes in an order date, customer ID, a cost, and a list of items and creates a new Transaction object

Manager Class inherits from Employee and contains more methods specific to managers.

- **Methods**
  - void **viewTransactions()** allows the manager to view all past Orders and their details
  - void **viewInventory()** prints a list of each item in stock and it's quantity
  - void **changeEmployeeInfo()** takes in a user ID, first name, last name, email, phone number, and address , and replaces each existing variable value for the user whose ID was used
  - void **addToMenu()** takes in a food name, price, and description, and creates a new FoodItem object
  - void **deleteFromMenu()** takes in a food name and removes the corresponding FoodItem from the menu

Customer Class inherits from User and contains an order method for online ordering.

- **Methods**
  - void **order()** takes in an order date, customer ID, a cost, and a list of items and creates a new Order object

Orders Class contains all the details of each order.

- **Attributes**
  - **orderNumber** is a unique ID for each order
  - **orderDate** is a String that lists the date that an order was made
  - **custID** saves the CustomerID that placed the oder
  - **items** contains a list of all the FoodItems that were ordered
  - **orderCost** contains the total price paid for the order

Transactions Class keeps track of all past transactions.

- **Attributes**
  - **orderList** keeps a list of orders
- **Methods**
  - void **addNewOrder()** is called by order() and takeOrder() and creates a new Order object and save it to orderList

Menu Class keeps track of the full menu.

- **Attributes**
  - **fullMenu** contains a list of FoodItems that represents the full menu
- **Methods**
  - void **addMenuItem()** takes in a FoodItem and adds it to fullMenu

FoodItem Class contains the details of each individual menu item.

- **Attributes**
  - **foodName** contains a String for the name of the food item
  - **foodPrice** contains a float for the price of the food item
  - **foodDescription** contains a String listing the item's details

Inventory Class keeps track of the stock of ingredients in the restaurant's storage.

- **Attributes**
  - **foodInventory** is a list of numbers For each ingredient necessary in the inventory
- **Methods**
  - void **listInventory()** lists each ingredient and its Quantity

Restaurant Class keeps important information about the restaurant.

- **Attributes**
  - **businessAddress** contains a String for the restaurant's address
  - **businessPhoneNum** is a String containing the restaurant phone number

Technical Support Specification:

New Users can expect 24/7 support from our help desk, and FAQ. Our response time varies but strive to deliver results within an hour at the minimum. Some services provided can include troubleshooting new user creations, logins, and helping a user remember credentials of any kind. In addition, editing account details, and other credentials are necessary.

Existing Users can expect the same support as new users but will not have as many push notifications on how to setup their accounts, rather than a critical support for remembering said credentials. Should we update our system, all users including employees are REQUIRED to update their systems. For existing users, the time frame for support is 1 hour minimum, along with 24/7 support.