

CPSC 471: Project Final Report

# **My Vietnamese Delight**

Duc Nguyen 30080293

Duong Tran 30113765

Viet Ho 30122283

## + Abstract

Due to the pandemic, not many people are going to the restaurant for dining. Therefore, our project is developing a web-based application to allow customers to look at the menu and pricing from home. Also, we developed a new system for this restaurant to allow customers to create and place orders via online modalities. We used Ruby on Rails and MySQL to deploy the web application. To get the website in the final, we firstly wrote a project proposal to get an overview of how the website will be developed. Then we draw an Entity Relationship Diagram (ERD) and Relational Model based on the project proposal. We found that they are very helpful for us to follow when we developed the website. While we also drew a Hierarchical Input Process Output (HIPO) diagram and Data Flow diagram (DFD) which helped us know what needs to be improved for the website. Thanks to everything, we developed the final website although it is still missing some functions compared to what is in the project proposal. Because of the time shortage and knowledge restrictions, we could not develop a perfect website. In conclusion, we have developed a web-based application where customers can see the menu items, their descriptions, image, and pricing online. Also, customers can add the menu items to the cart and place an order online. With the admin login, the employee can edit the list of employees, the menu items, and the promo items, and can view all orders by customers.

## + Introduction (describe the problem or task your database was designed to address)

Most modern merchant-to-customer interactions are usually handled by the merchant's website, which introduces all the information a customer needs to confidently make an informed choice and just as easily place an order. This information consists of the following specifications: price, nutritional value (if applicable), availability and so forth anything a customer wishes to know immediately.

With this in mind, *My Vietnamese Delight* requires an overhaul of their website to incorporate this information and an implementation of a bespoke ordering platform to easily add and modify items as well as implement a way to manage online orders.

## + Describe briefly the system you have created to address the problem

To achieve a productive and convenient ordering system, the website has to be informative and user-friendly. The website keeps track of customer's order and address information, then sends

it to the admin interface to notify the restaurant of incoming orders. Moreover, the menu and gallery have to be attractive to lure in more customers. Some of the main features displayed on the website include: a default customer view where customers can access the most updated menu and the corresponding images of the dishes. Each item has the specification of price, and its description including ingredients. From there customers can choose their preferred items to add into the order cart where the basket will show the individual prices for each item and the current total price. Once a customer places an order, the data will be transferred to the admin system. Owner or employee of My Vietnamese Delight would have access to the admin login page to update the menu and gallery, and view incoming orders.

## Project Design

For our website, API and database implementation, we focused on creating a website that is solely designed for customers to view menu items, add it to the cart and then checkout leaving their details for pickup or delivery purposes. This implementation also focused on ease of use for a site administrator to quickly use CRUD functionalities on employees alongside being able to update the menu and promotional items page with the exact same CRUD functionality. Thus the below API image provided below shows the internal structure of the site while the ERD further below shows the relationships between each item in the API.

### User Possible Transactions

#### Add Item to Cart

```
def create
  # Find associated product and current cart
  chosen_menu_item = MenuItem.find(params[:menu_item_id])
  current_cart = Cart.last
  # If cart already has this product then find the relevant line_item and iterate
  # quantity otherwise create a new line_item for this product
  if current_cart.line_items.map(&:menu_item).include?(chosen_menu_item)
    # Find the line_item with the chosen_product
    @line_item = current_cart.line_items.find_by(:menu_item_id => chosen_menu_item)
    # Iterate the line_item's quantity by one
    @line_item.quantity += 1
  else
    @line_item = LineItem.new
    @line_item.cart = current_cart
  end
end
```

```

@line_item.menu_item = chosen_menu_item
end

# Save and redirect to cart show path
@line_item.save!
redirect_to cart_path(current_cart)
end

```

## Cart Checkout

```

def create
  @order = Order.new(order_params)
  @current_cart.line_items.each do |item|
    @order.line_items << item
    item.cart_id = nil
  end
  @order.save!
  Cart.destroy(@current_cart.id)
  redirect_to orders_path
end

```

## Create a new Order

```

def create
  @order = Order.new(order_params)
  @current_cart.line_items.each do |item|
    @order.line_items << item
    item.cart_id = nil
  end
  @order.save!
  Cart.destroy(@current_cart.id)
  redirect_to orders_path
end

private
def order_params
  params.require(:order).permit(:name, :email, :address, :pay_method)
end

end

```

## Employee Possible Transactions

Everything included that a customer can do plus

CRUD all Menu Items (Same as the promo items transactions)

```
def create
  @menu_item = MenuItem.new(menu_item_params)

  respond_to do |format|
    if @menu_item.save!
      format.html { redirect_to menu_item_url(@menu_item), notice: "Menu item was successfully created." }
      format.json { render :show, status: :created, location: @menu_item }
    else
      format.html { render :new, status: :unprocessable_entity }
      format.json { render json: @menu_item.errors, status: :unprocessable_entity }
    end
  end
end

# PATCH/PUT /menu_items/1 or /menu_items/1.json
def update
  respond_to do |format|
    if @menu_item.update(menu_item_params)
      format.html { redirect_to menu_item_url(@menu_item), notice: "Menu item was successfully updated." }
      format.json { render :show, status: :ok, location: @menu_item }
    else
      format.html { render :edit, status: :unprocessable_entity }
      format.json { render json: @menu_item.errors, status: :unprocessable_entity }
    end
  end
end

# DELETE /menu_items/1 or /menu_items/1.json
def destroy
  @menu_item.destroy

  respond_to do |format|
    format.html { redirect_to menu_items_url, notice: "Menu item was successfully destroyed." }
    format.json { head :no_content }
  end
end
```

```

    end
end

private
# Use callbacks to share common setup or constraints between actions.
def set_menu_item
  @menu_item = MenuItem.find(params[:id])
end

# Only allow a list of trusted parameters through.
def menu_item_params
  params.require(:menu_item).permit(:name, :description, :image, :price, :ispromo)
end
end

```

## CRUD all Employees

```

def create
  @employee = Employee.new(employee_params)

  respond_to do |format|
    if @employee.save!
      format.html { redirect_to employee_url(@employee), notice: "Employee was successfully created." }
      format.json { render :show, status: :created, location: @employee }
    else
      format.html { render :new, status: :unprocessable_entity }
      format.json { render json: @employee.errors, status: :unprocessable_entity }
    end
  end
end

# PATCH/PUT /employees/1 or /employees/1.json
def update
  respond_to do |format|
    if @employee.update(employee_params)
      format.html { redirect_to employee_url(@employee), notice: "Employee was successfully updated." }
      format.json { render :show, status: :ok, location: @employee }
    else

```

```
    format.html { render :edit, status: :unprocessable_entity }
    format.json { render json: @employee.errors, status: :unprocessable_entity }
  end
end

# DELETE /employees/1 or /employees/1.json
def destroy
  @employee.destroy

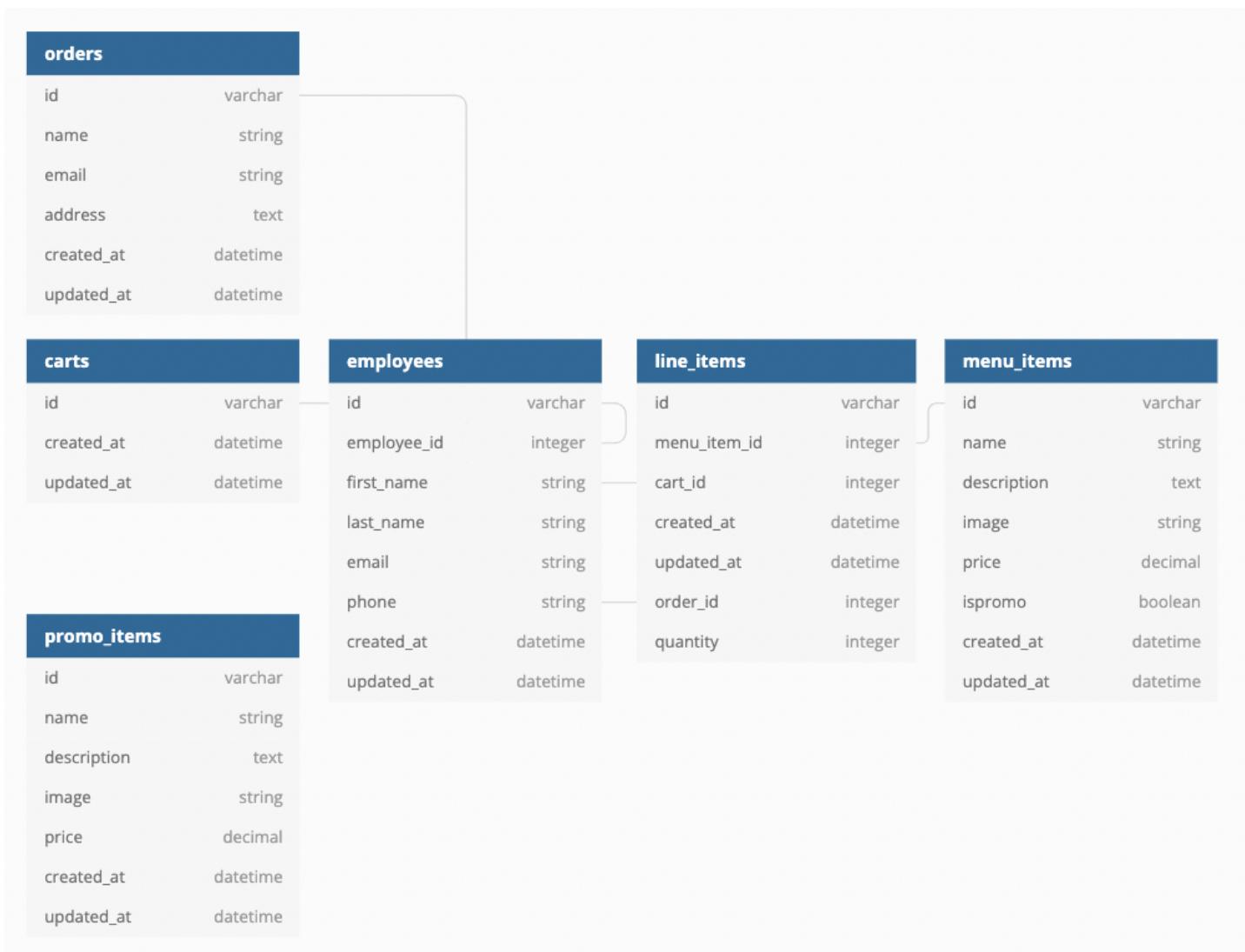
  respond_to do |format|
    format.html { redirect_to employees_url, notice: "Employee was successfully destroyed." }
    format.json { head :no_content }
  end
end

private

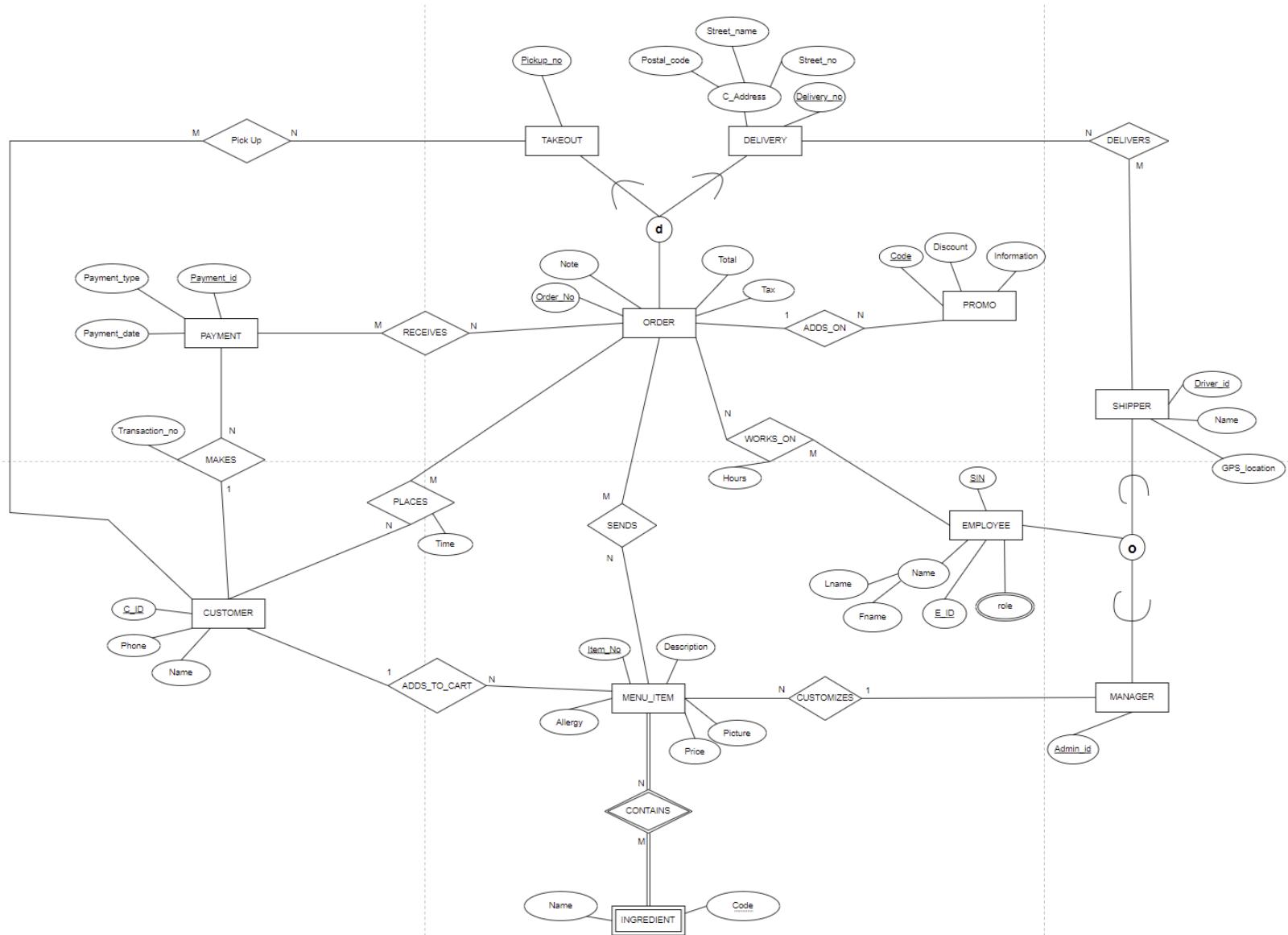
# Use callbacks to share common setup or constraints between actions.
def set_employee
  @employee = Employee.find(params[:id])
end

# Only allow a list of trusted parameters through.
def employee_params
  params.require(:employee).permit(:employee_id, :first_name, :last_name, :email, :phone)
end
end
```

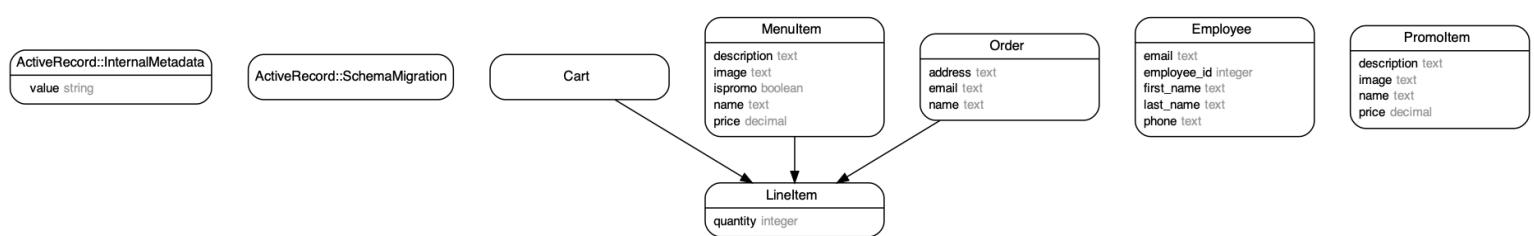
## Functionality offered by the API



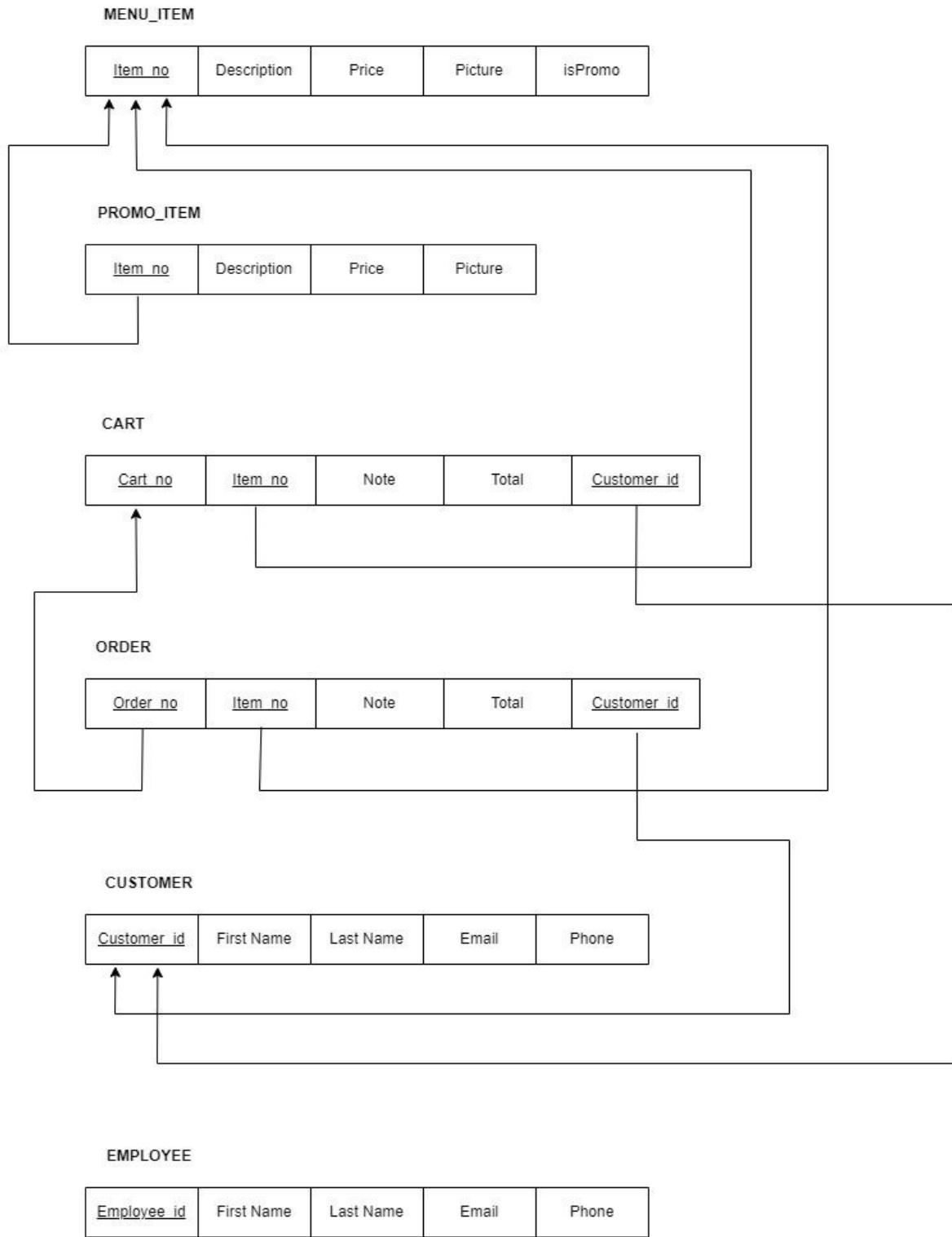
## EER diagram



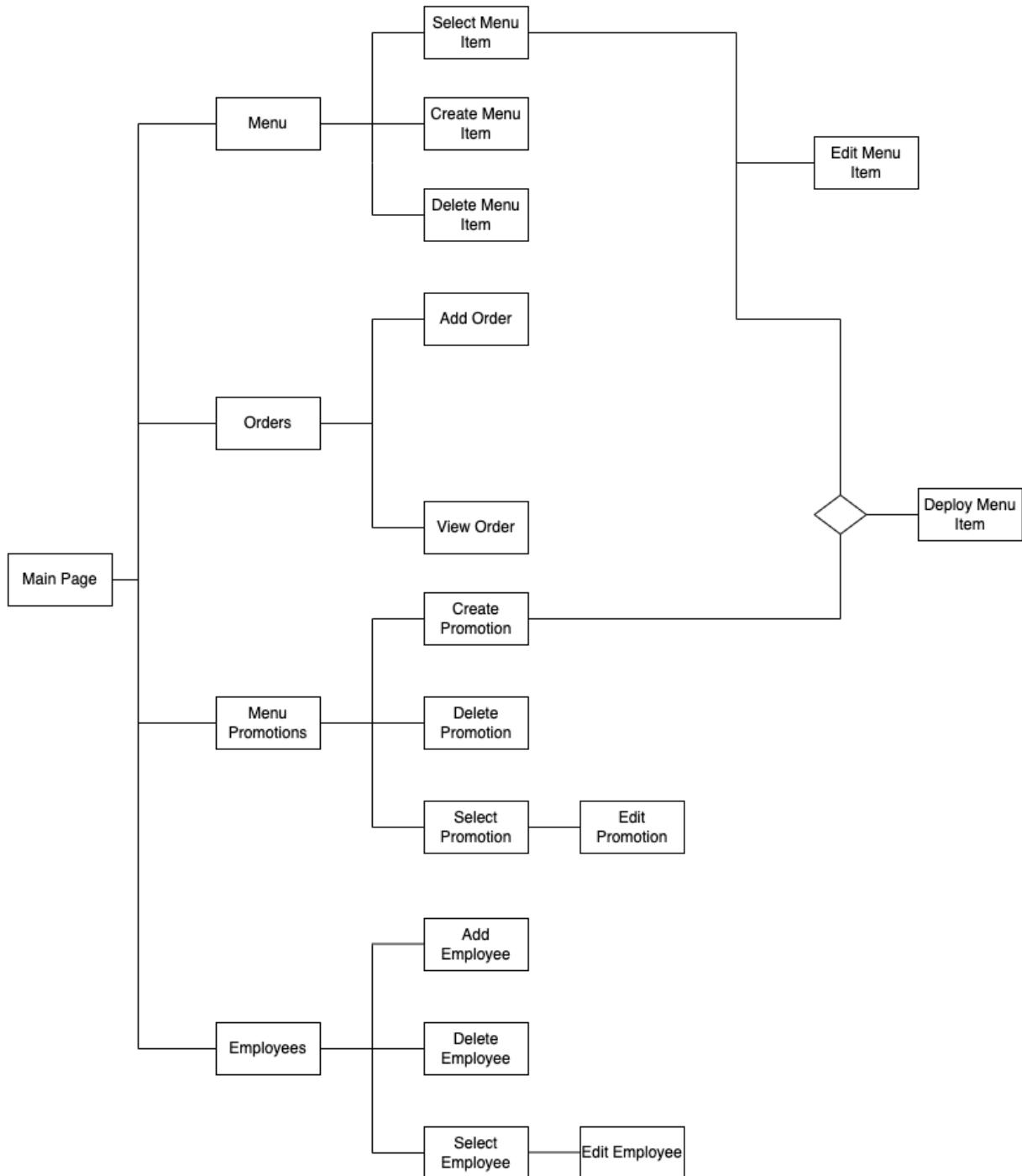
Cpsc471appv2 domain model



## Relational Model



## HIPO



## H IPO Functions

**Function:** Add Menu Item

**Inputs:** @Name, @Description, @Price, @isPromo

**Outputs:** None

**Pseudocode:**

Connect to the database

Query = INSERT INTO Menu\_Item(@Name, @Description, @Price, @isPromo)

**Function:** Edit Menu Item

**Inputs:** @Name, @Description, @Price, @isPromo

**Outputs:** None

**Pseudocode:**

Connect to the database

Query = UPDATE Menu\_Item

SET "attribute"

WHERE (Name = @Name) AND (Description = @Description) AND (Price = @Price) AND  
(isPromo = @isPromo)

**Function:** Select Menu Item

**Inputs:** @Name, @Description, @Price, @isPromo

**Outputs:** MenuItem\_Item

**Pseudocode:**

Connect to the database

Query = SELECT m.Name, m.Description, m.Price, m.isPromo

FROM Menu\_Item as m

WHERE (m.Name = @Name) AND (m.Description = @Description) AND (m.Price = @Price)  
AND (m.isPromo = @isPromo)

**Function:** Delete Menu Item

**Inputs:** @Name, @Description, @Price, @isPromo

**Outputs:** None

**Pseudocode:**

Connect to the database

Query = DELETE m.Name, m.Description, m.Price, m.isPromo

FROM Menu\_Item as m

WHERE (m.Name = @Name) AND (m.Description = @Description) AND (m.Price = @Price)  
AND (m.isPromo = @isPromo)

**Function:** Add Order

**Inputs:** @Order\_No, @Note, @Total, @Tax, @Item\_Name

**Outputs:** None

**Pseudocode:**

**Function:** View Order

**Inputs:** @Order\_No, @Note, @Total, @Tax, @Item\_Name

**Function:** View Order

**Inputs:** @Order\_No, @Note, @Total, @Tax, @Item\_Name

**Outputs:** Order\_No, Note, Total, Tax, Item\_Name

**Pseudocode:**

Connect to the database

Query = SELECT o.Order\_No, o.Note, o.Total, o.Tax, m.Item\_Name

FROM Order as o, Menu\_Item as m

WHERE (o.Order\_No = @Order\_No) AND (o.Note = @Note) AND (o.Total = @Total) AND  
(o.Tax = @Tax) AND (m.Item\_Name= @Item\_Name)

**Function:** Create Promotion

**Inputs:** @Code, @Discount, @Information

**Outputs:** None

**Pseudocode:**

Connect to the database

Query = INSERT INTO Menu\_Promotions(@Code, @Discount, @Information)

**Function:** Deloy Menu Item

**Inputs:** @Name, @Description, @Price, @isPromo, @Promo\_Code

**Outputs:** None

**Pseudocode:**

Connect to the database

Query = UPDATE Menu\_Item

SET "attribute"

WHERE (Name = @Name) AND (Description = @Description) AND (Price = @Price) AND  
(isPromo = @isPromo) AND (Promo\_Code = @Promo\_Code)

**Function:** Delete Promotion

**Inputs:** @Code, @Discount, @Information

**Outputs:** None

**Pseudocode:**

Connect to the database

Query = DELETE p.Code, p.Discount, p.Discount

FROM Menu\_Promotions as p

WHERE (p.Code = @Code) AND (p.Discount = @Discount) AND (p.Information =  
@Information)

**Function:** Edit Promotion

**Inputs:** @Code, @Discount, @Information

**Outputs:** None

**Pseudocode:**

Connect to the database  
Query = UPDATE Menu\_Promotions  
SET “attribute”  
WHERE (Code = @Code) AND (Discount = @Discount) AND (Information = @Information)

**Function:** Select Promotion

**Inputs:** @Code, @Discount, @Information

**Outputs:** Menu\_Promotion

**Pseudocode:**

Connect to the database  
Query = SELECT p.Code, p.Discount, p.Information  
FROM Menu\_Promotions as p  
WHERE (p.Code = @Code) AND (p.Discount = @Discount) AND (p.Information = @Information)

**Function:** Add Employee

**Inputs:** @SIN, @FName, @LName, @E\_ID, @Role

**Outputs:** None

**Pseudocode:**

Connect to the database  
Query = INSERT INTO Employees(@SIN, @FName, @LName, @E\_ID, @Role)

**Function:** Delete Employee

**Inputs:** @SIN, @FName, @LName, @E\_ID, @Role

**Outputs:** None

**Pseudocode:**

Connect to the database  
Query = DELETE e.SIN, e.FName, e.LName, e.E\_ID, e.Role  
FROM Employee as e  
WHERE (e.SIN = @SIN) AND (e.FName = @FName) AND (e.LName = @LName) AND  
(e.E\_ID = @E\_ID) AND (e.Role = @Role)

**Function:** Edit Employee

**Inputs:** @SIN, @FName, @LName, @E\_ID, @Role

**Outputs:** None

**Pseudocode:**

Connect to the database  
Query = UPDATE Employee  
SET “attribute”  
WHERE (SIN = @SIN) AND (FName = @FName) AND (LName = @LName) AND (E\_ID = @E\_ID) AND (Role = @Role)

**Function:** Select Employee

**Inputs:** @SIN, @FName, @LName, @E\_ID, @Role

**Outputs:** Employee

**Pseudocode:**

Connect to the database

```
Query = DELETE e.SIN, e.FName, e.LName, e.E_ID, e.Role  
FROM Employee as e  
WHERE (e.SIN = @SIN) AND (e.FName = @FName) AND (e.LName = @LName) AND  
(e.E_ID = @E_ID) AND (e.Role = @Role)
```

## Changes from Original Proposal

Due to time shortage and knowledge restrictions, we could not implement the Payment, Takeout, Delivery, Manager and Shipper features. We decided to challenge ourselves during this course and semester to continue and build our skill sets by introducing new technologies and languages into our project, but unfortunately our inexperience in this language and creating the entire framework from the ground up required more time than projected.

## API Documentation

API Documentation is published at:

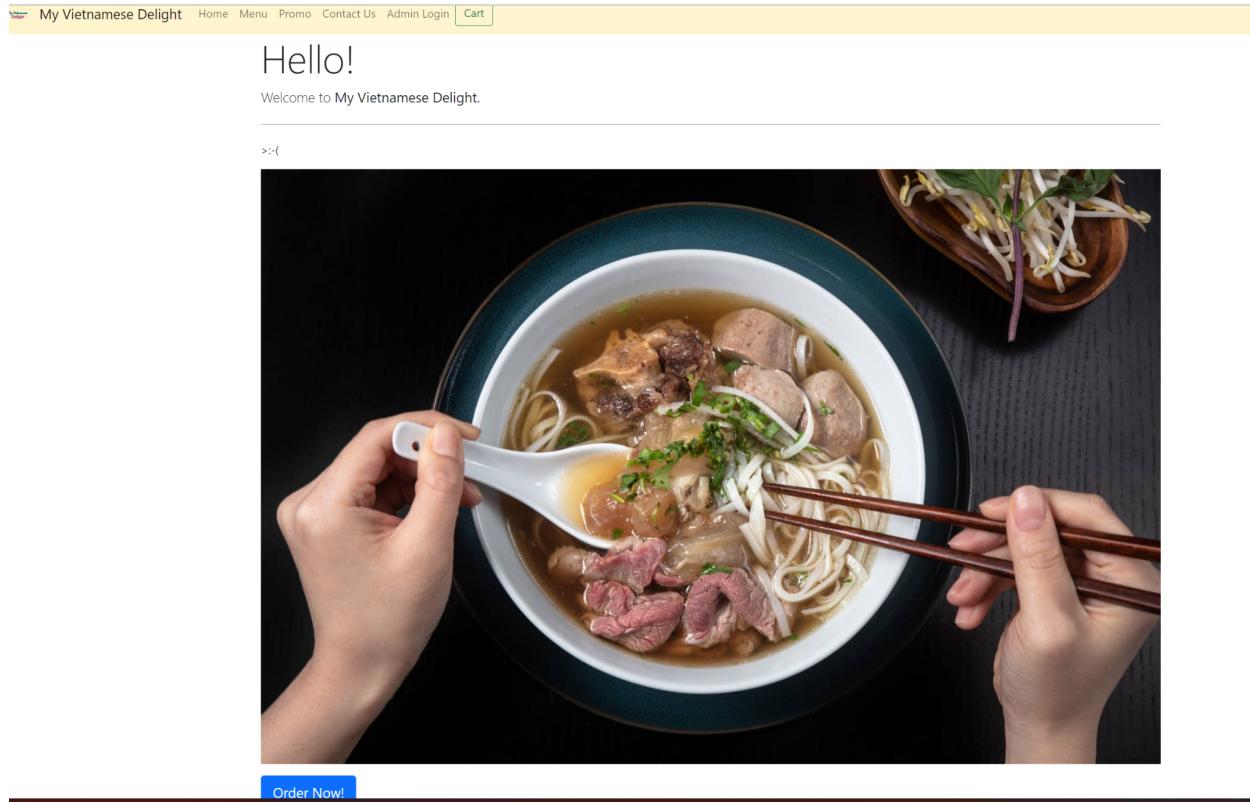
<https://documenter.getpostman.com/view/20444847/UVyysCuD>

# User Guide

## **For Customer:**

To use the system, the customer must go to the site:

<https://stark-dusk-07050.herokuapp.com/home>



This is the welcome page with the restaurant introduction about its culture, cuisine and dedication. There is a picture featuring the most popular dish of the restaurant.

Next, the customer can click on the “Order Now” button or the Menu/Promo tab to select the food items they want to order.

## Menu Page

### pho

wow so yummy

29.99

Add pho to cart



### spring rolls

wow so fresh

12.99

Add spring rolls to cart



In the Menu page, there will be several dishes featured with description, price and photo of the actual dish. The customer can go through the entire menu and when they decide which dish they want, simply click on the “Add to cart” button above the picture of the dish to add it to their cart.

## Promo Items Page

### banh xeo

yellow !

5.99

Add banh xeo to cart



The Promo page is similar with the same functionality of the Menu page, where customers can select the dish if they want to their cart. The main difference between this page and Menu page is that Promo features the limited-time supplied dish or whatever dish that is on discount. Once the customer is finished with their selection, they can click on the “Cart” tab on top to view their current order.

A screenshot of a web page titled "Cart show". At the top, there is a navigation bar with links for Home, Menu, Promo, Contact Us, Admin Login, and a green "Cart" button. The main content area displays a list of items in the cart:

- Item: Pho
  - Quantity: 6
  - Price: \$29.99
- Item: Spring Rolls
  - Quantity: 3
  - Price: \$12.99

## Cart show

- Item: Pho
  - Quantity: 6
  - Price: \$29.99
- Item: Spring Rolls
  - Quantity: 3
  - Price: \$12.99

**Sub total: \$218.91**

[Proceed to checkout](#)

The Cart will show which item they pick and the quantity of each item. It will also calculate the sub-total based on price and quantity of each. Customers can select if they want to delete or add more items to the cart. When the order is ready, customer clicks on “proceed to checkout” blue line to be transfer to the Information page where they fill out their information and decide if they want to take out or delivery

A screenshot of a web page titled "My Vietnamese Delight". At the top, there is a navigation bar with links for Home, Menu, Promo, Contact Us, Admin Login, and a green "Cart" button. Below the navigation bar, there are three input fields labeled "Name", "Email", and "Address", each with a corresponding text input box. At the bottom, there is a "Create Order" button.

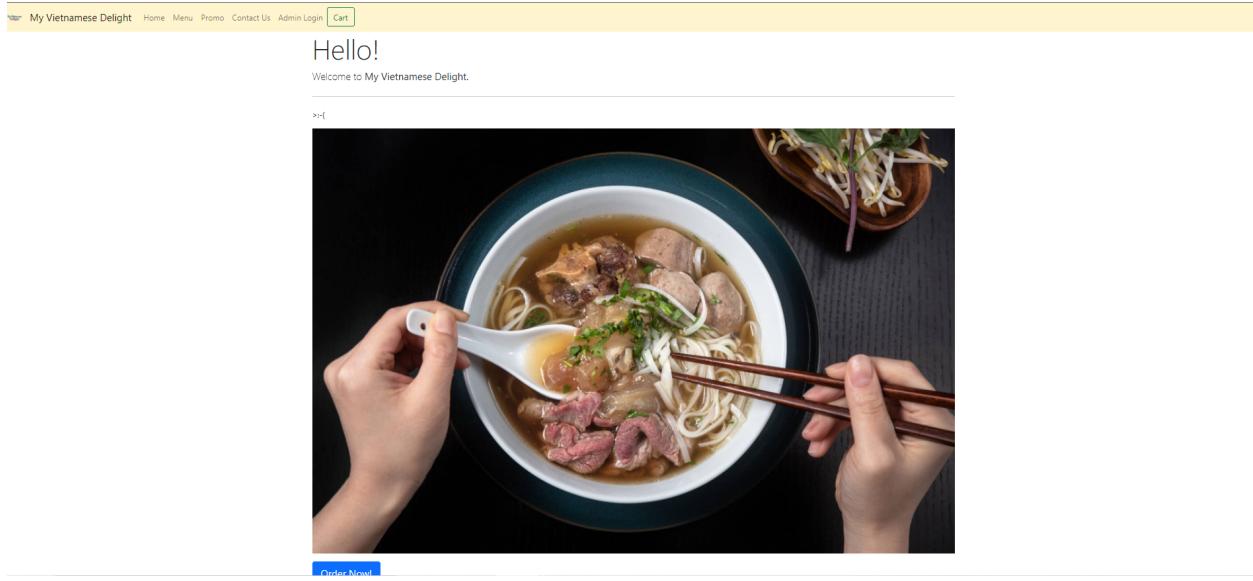
For take-out customer, if they had trouble finding the location of the restaurant, they can click on Contact us page where the restaurant's address and phone number are provided. It also included a GPS map to the restaurant location

The screenshot shows a website header with "My Vietnamese Delight" logo, "Home", "Menu", "Promo", "Contact Us", "Admin Login", and a "Cart" button. Below the header, the page title is "Contact Page". Under the title, there is a section titled "Information" containing the phone number "+15873318073" and the address "328 Centre St S #146". To the right of this information is a Google Map of a downtown area. The map highlights "My Vietnamese Delight" at its location (328 Centre St S #146, Calgary, AB T2G 4X6). It also shows nearby landmarks like "The Bow", "Delta Hotels by-Marriott Calgary Downtown", and "CPA Convention Centre Parkade". Other restaurants visible on the map include "Silver Dragon Restaurant (Calgary)", "Gong Cha Chinatown - Bubble Tea", "OEB Breakfast Co", "Vo Kitchen & Bar", "SAIGON STAR Ultimate Subs &...", "Thi Thi Vietnamese Submarine", "1 Pot", and "ii - Bullet Bar". The map includes street names like 2 St SW, 4 Ave SW, 5 Ave SW, 6 Ave SW, 4 Ave SE, 5 Ave SE, 6 Ave SE, and 316 Macleod T SE - Lot #185. A "Bow River" is also shown flowing through the area.

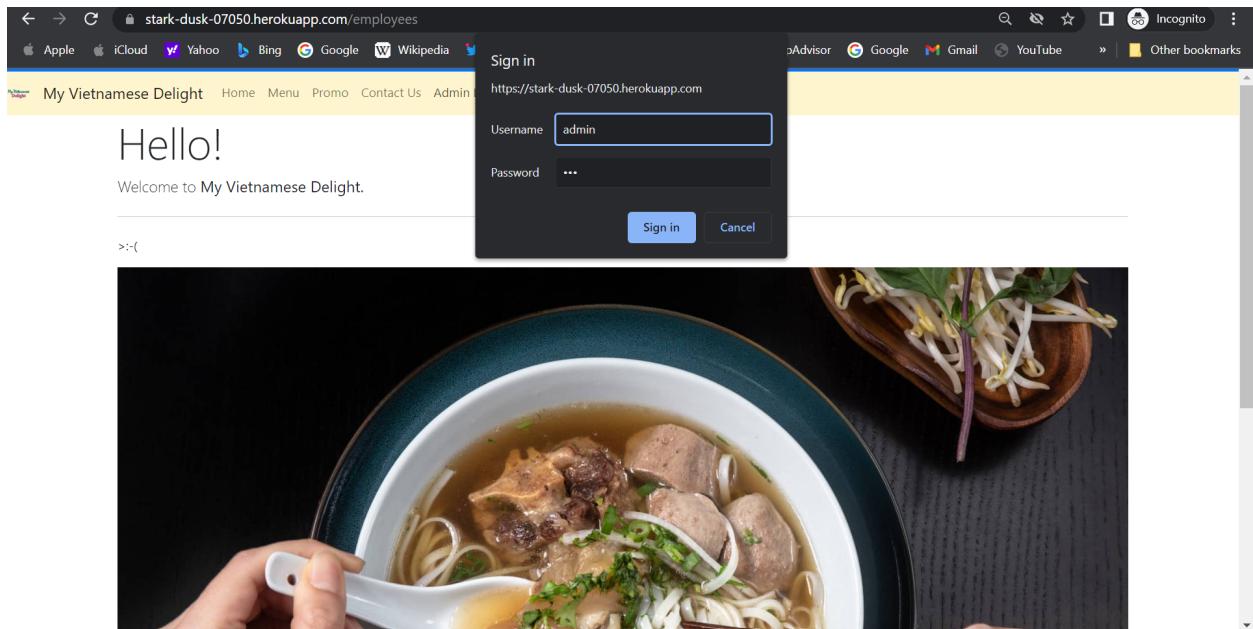
### For Employee:

To use and control the system, the employee need to go the site:

<https://stark-dusk-07050.herokuapp.com/home>



Next, the employee needs to press the admin login button at the top of the page and enter admin for the username and 123 for the password to sign in.



After press sign in, it will direct to the employees page.

My Vietnamese Delight Home Menu Promo Contact Us Admin Login Cart

## Employees

Employee: 1

First name: colty

Last name: nguyen

Email: test@test.com

Phone: 12345678910

[Show this employee](#)

[New Employee](#)

[Modify Menu](#)

[Modify Promos](#)

[View Orders](#)

In the employees page, there is information of all employees (which are employee number, first name, last name, email and phone number). There are also 5 buttons that the employee can press, which are “Show this employee”, “New Employee”, “Modify Menu”, “Modify Promos” and “View Orders”. If the employee presses on “Show this employee” blue line, it will direct to a page that we can edit or destroy this employee which we pressed on.

My Vietnamese Delight Home Menu Promo Contact Us Admin Login Cart

Employee: 1

First name: colty

Last name: nguyen

Email: test@test.com

Phone: 12345678910

[Edit this employee](#) | [Back to employees](#)

[Destroy this employee](#)

In the above page, if we press “Destroy this employee”, then it will destroy the employee and go back to the employees page. Or if we press “Back to employees”, it will go back to the employees page without destroying this employee. If we press “Edit this employee”, then it will direct to the Editing employee page.

## Editing employee

Employee

First name

Last name

Email

Phone

[Show this employee](#) | [Back to employees](#)

In the Editing employee page, we can change all the information of the employee (employee number, first name, etc). Then when we press “Update Employee”, all the new information will be saved and it will go back to the last page. If we press “Show this employee”, it will go back to the last page. And if we press “Back to employees”, it will go back to the employees page. In the employees page, if we press “New Employee”, it will direct to the New employee page.

## New employee

Employee

First name

Last name

Email

Phone

[Back to employees](#)

In the New employee page, we can add a new employee to the system which includes all the information of the employee which are employee number, first name, last name, email and phone number. After entering all the information, we can press “Create Employee” to create and save all the information we entered and go back to the employees page. If we press “Back to employees”, it will go back to the employees page without changing anything. In the employees page, if we press “Modify Menu”, it will direct to the Menu items page.

The screenshot shows a website header with the title "My Vietnamese Delight" and navigation links for Home, Menu, Promo, Contact Us, Admin Login, and Cart. The "Cart" link is highlighted with a green border.

## Menu items

Name: pho

Description: wow so yummy

Image: <https://www.inspiredtaste.net/wp-content/uploads/2016/06/Vietnamese-Pho-Soup-Recipe-1.jpg>

Price: 29.99

IsPromo: false

[Show this menu item](#)

Name: spring rolls

Description: wow so fresh

Image: <https://healthynibblesandbits.com/wp-content/uploads/2020/02/Vietnamese-Spring-Rolls-Recipe-Card.jpg>

Price: 12.99

IsPromo: false

[Show this menu item](#)

[New menu item](#)

In the Menu items page, we can press “Show this menu item” to make any change to the item we pressed on or we can press “New menu item” to add a new menu item. If we press on “Show this menu item”, it will go to a page where we can see all information about this item and all buttons that we press to make change for this item.

The screenshot shows the details of the "pho" menu item. It includes the name, description, image URL, and price. At the bottom, there are two buttons: "Edit this menu item" and "Destroy this menu item".

Name: pho

Description: wow so yummy

Image: <https://www.inspiredtaste.net/wp-content/uploads/2016/06/Vietnamese-Pho-Soup-Recipe-1.jpg>

Price: 29.99

IsPromo: false

[Edit this menu item](#) | [Back to menu items](#)

[Destroy this menu item](#)

In the above page, we can see all information about pho since we pressed on “Show this menu item” of pho in the Menu items page. If we press “Destroy this menu item”, it will destroy this menu item and go back to the Menu items page. Or if we press “Back to menu items”, it will go back to the Menu items page without changing anything. If we press “Edit the menu item”, it will direct to a page where we can edit the item.

My Vietnamese Delight Home Menu Promo Contact Us Admin Login Cart

## Editing menu item

Name  
pho

Description  
wow so yummy

Image  
https://www.inspiredtaste.

Price  
29.99

Ispromo

[Show this menu item](#) | [Back to menu items](#)

In the Editing menu item page, we can edit all the information about the item we pressed on. After entering all the information we want to update, we can press “Update Menu Item” to update and save all the information we entered, and it will go back to the last page. If we press “Show this menu item”, it will go back to the last page. If we press “Back to menu item”, it will go back to the Menu items page. In the Menu items page, if we press New menu item, it will go to a page where we can add a new item.

My Vietnamese Delight Home Menu Promo Contact Us Admin Login Cart

## New menu item

Name

Description

Image

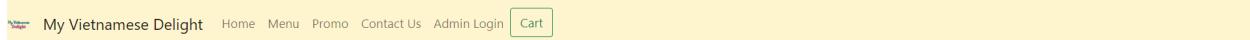
Price

Ispromo

[Back to menu items](#)

In the New menu item, we can enter all the information of an item which are name, description, image, price and ispromo. After entering everything, we can press “Create Menu item” to create and save all the information of the new item to the menu, and it will go back to the Menu items page. If we want to go back to the Menu items page, we can press “Back to menu items”. In the

employees page, we can press “Modify Promos”, it will direct to the Promo items page where we can see all the promo items.



The Promo items page is similar with the same functionality of the Menu items page, where we can see all the information of all promo items, update them, destroy them by selecting “Show this promo item” and add a new promo item by selecting “New promo item”



When we press “Show this promo item”, it will direct to the above page where we can edit the promo item by selecting “Edit this promo item”. If we want to destroy the promo item, we can press “Destroy this promo item”. And if we just want to go back to the Promo items page, we just need to press “Back to promo items”.

## Editing promo item

Name

Description

Image

Price

[Show this promo item](#) | [Back to promo items](#)

When we press “Edit this promo item”, it will direct to the Editing promo item page where we can update any information we want for the item. After entering all the information we want to update for the promo item, we can press “Update Promo item” to update and save information, and go back to the last page. We can press “Show this promo item” to go back to the last page. And we can press “Back to promo items” to go back to the Promo items page.

## New promo item

Name

Description

Image

Price

[Back to promo items](#)

In the Promo items page, if we press “New promo item” to add a new promo item, it will direct to the above page where we can enter all the information for the new item, and press “Create Promo item” to create and save all the information we enter, and go back to the Promo items page. If we want to go back to the Promo items page, we just simply press “Back to promo items”. In the employees page, we can press “View Orders”, it will direct to the All Orders page

where we can see all the orders and their information (Order ID, Name and email of customers, and the items and the quantity of them were ordered).

The screenshot shows a web application titled "My Vietnamese Delight". The top navigation bar includes links for Home, Menu, Promo, Contact Us, Admin Login, and Cart. The "Cart" link is highlighted with a green border.

## All Orders

Order ID: **2**  
Name: dan  
Email: dp@gmail.com

Line Items:

- 6 x pho
- 3 x spring rolls

---

Order ID: **1**  
Name: Duc Nguyen  
Email: test@test.com

Line Items:

- 1 x pho