**A Nail-Salon Web App Point-of-Sales System**

Service Layers

Thi Da Thuy Pham

**Overview:**

My project using Express and Node.js involves setting up a server to handle the requests and manage data. Express.js, a framework for Node.js, simplifies the process of building web applications and APIs. With my project,  a common structure involves three primary layers:

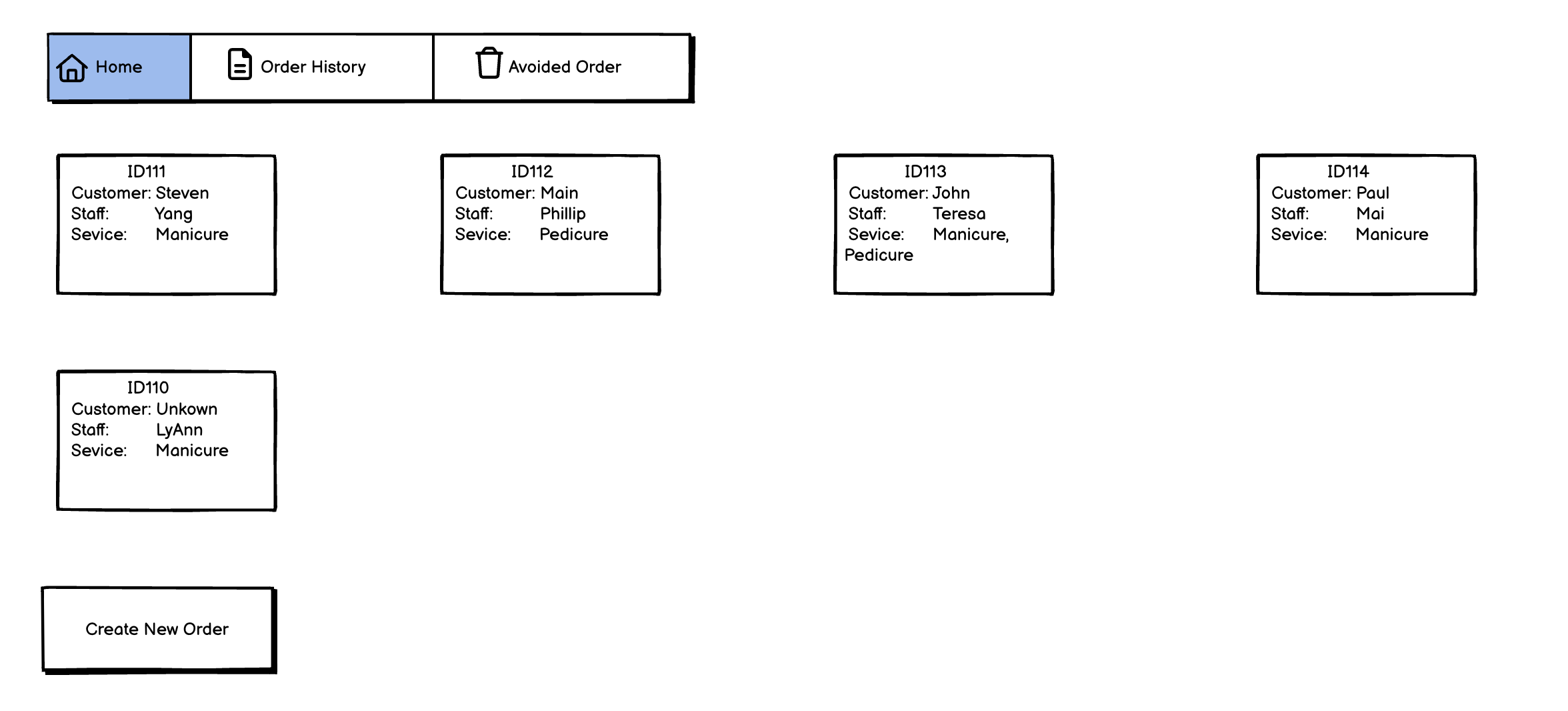
* API layer: This layer handles HTTP requests and responses. It receives requests, validates inputs, and calls the service layer to process the request.
* Service layer: This layer contains the core business logic of the application, it receives requests from the API layer.
* Data access layer: This layer is responsible for interacting with the database or any other data storage mechanism. It provides an abstraction over the underlying data storage, allowing the service layer to access data without needing to know the specifics of the database.

1. **Home page**:

* Method: GET
* URL : /api/v1/orders?status=active
* Purpose: When a cashier open web-app , this endpoint will called all active orders to Home page
* Example:

A screen shot of a computer program

AI-generated content may be incorrect.



SQL Server

GET:/api/v1/orders?status=active

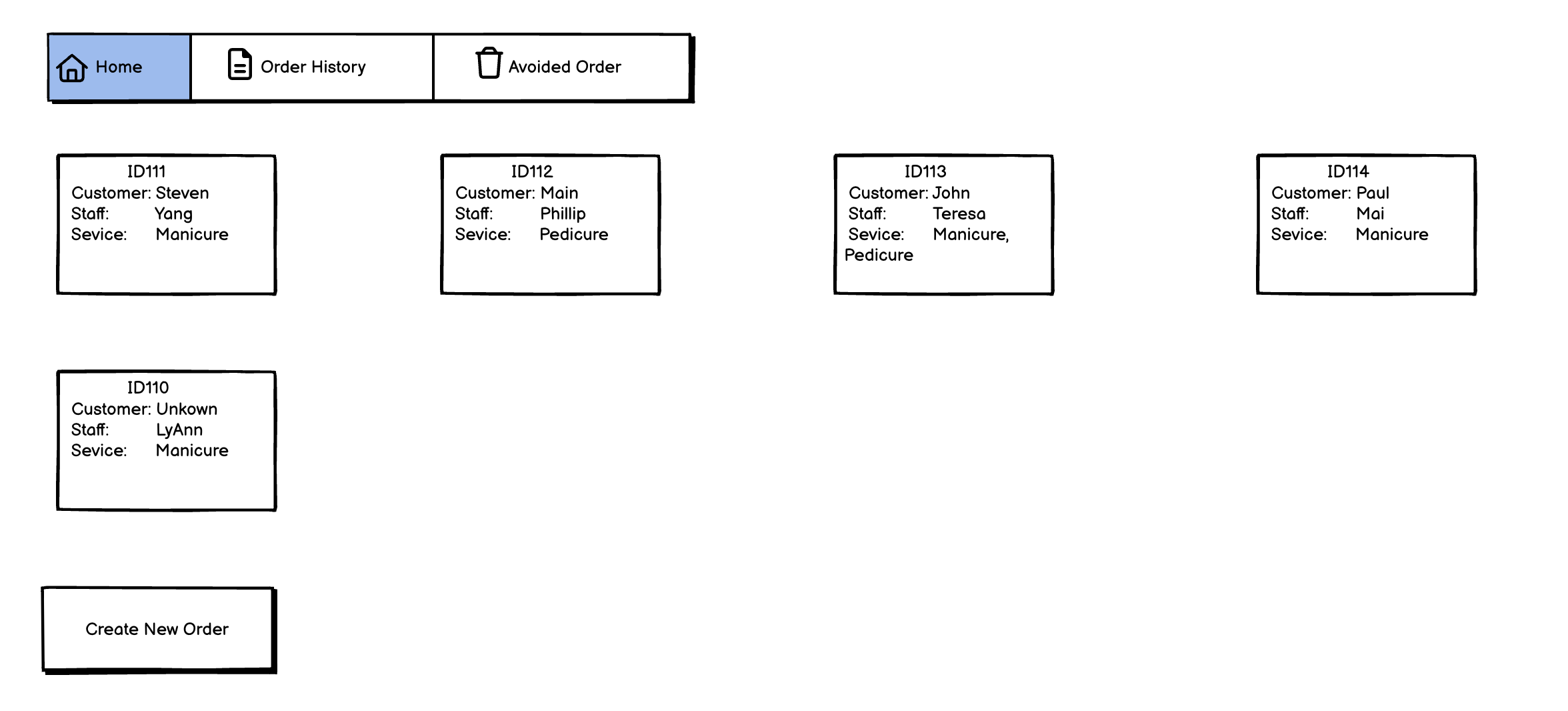
API

1. **Create New Order**:

* Method: POST
* URL : /api/v1/order
* Purpose: When a cashier click “Create New Order” button, move to “enter customer’s phone number” page.
* Example:

A screen shot of a computer

AI-generated content may be incorrect.



SQL Server

POST:/api/v1/order

API

1. **Click “Skip” button**:

This is an Unknown customer order => Move to Menu page

* Method: PUT
* URL : /api/v1/order/{order\_id}
* Purpose: When a cashier click “Skip” button, move to “Menu” page.
* Example:

A screenshot of a computer program

AI-generated content may be incorrect.

A phone number and a box

AI-generated content may be incorrect.

SQL Server

PUT: api/v1/order/{order\_id}

API

1. **Click “OK” button at “Enter customer’s phone number” page**:

* Method: GET
* URL : / api/v1/customers?phone={customer\_phone}
* Purpose: When cashier create new order with customer information. This endpoint will called get all information of order, and it’ll check if :
* There’s no customer with that phone number:
  + - * POST /api/v1/customer
      * To create a new customer
      * Move to “enter customer’s name” page
* There’s a customer with that phone number:
  + - * PUT /api/v1/order/{order\_id}
      * To update that customer to the order
      * Move to Menu page
* Example:

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A phone number and a box

AI-generated content may be incorrect.

SQL Server

GET:/api/v1/customers?phone={customer\_phone}

API

1. **Click “OK” button at “Enter customer’s name” page**:

2 APIs has been called:

1. To add name for a new customer
   * Method: PUT
   * URL : /api/v1/customer/{customer\_id}
   * Purpose: The endpoint has called after cashier add name and click OK.
   * Example:

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

SQL Server

PUT:api/v1/customer/{customer\_id}

API

1. To update a new customer to the order
   * Method: PUT
   * URL : /api/v1/order/{order\_id}
   * Purpose: To add name of customer into order.
   * Example:

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

SQL Server

PUT:api/v1/order/{order\_id}

API

1. **Menu page**:
2. Get all staffs :
   * Method: GET
   * Url: /api/v1/staffs
   * Purpose: To get all staff to Menu page
   * Example:

A screen shot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer screen

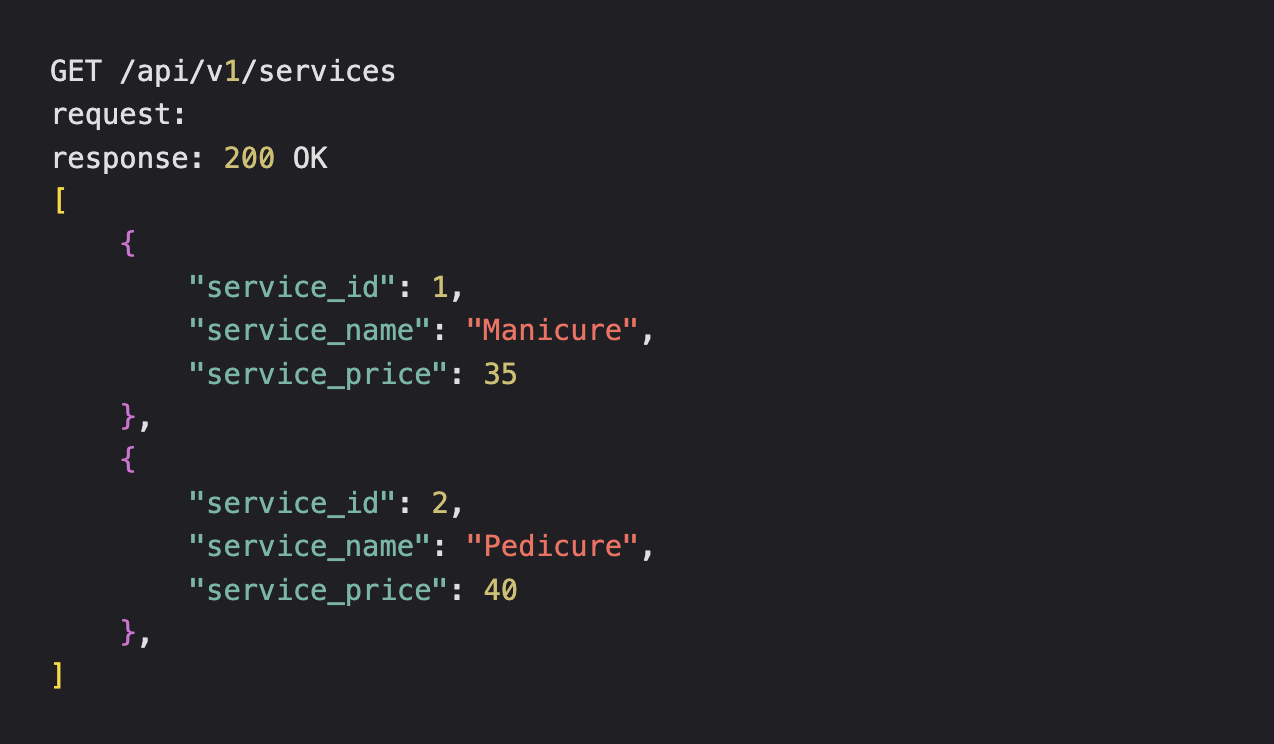
AI-generated content may be incorrect.

SQL Server

Get: /api/v1/staffs

API

1. Get all service:
   * Method: GET
   * Url: /api/v1/services
   * Purpose: To get all services to Menu page
   * Example:



A screenshot of a computer screen

AI-generated content may be incorrect.

SQL Server

Get: /api/v1/services

API

1. **Select “staff”:**

* Method: PUT
* Url: /api/v1/order/{order\_id}
* Purpose: To update staff to order
* Example:

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

SQL Server

PUT: api/v1/order/{order\_id}

API

1. **Select service**:

* Method: PUT
* Url: /api/v1/order/{order\_id}
* Purpose: To add service into order
* Example:

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

SQL Server

PUT: api/v1/order/{order\_id}

API

1. **Click “Check-out” button**:

* Method: PUT
* Url: /api/v1/order/{order\_id}/status
* Purpose: To update status of order from “active” to “done”
* Example:

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

SQL Server

PUT: api/v1/order/{order\_id}/status

API

1. **Open Order History page:**

* Method: GET
* Url: /api/v1/orders?status=done
* Purpose: Get all order with status equal done.
* Example:



A screenshot of a computer

AI-generated content may be incorrect.

SQL Server

GET: api/v1/orders?status=done

API

1. **Open Voided order page**:

* Method: GET
* Url: /api/v1/orders?status=voided
* Purpose: Get all order with status equal voided.
* Example:

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

SQL Server

GET: api/v1/orders?status=voided

API

1. Login page:

* Method: POST
* Url: /api/v1/login
* Purpose: Login to app.
* Example:

A screen shot of a computer code

AI-generated content may be incorrect.

A login screen with black text and white text

AI-generated content may be incorrect.

SQL Server

GET: api/v1/orders?status=voided

API