1. **Introduction**

**Potato** is a full-stack **monolithic application** designed to serve both web and mobile users. The backend is developed with **Spring Boot** and provides **RESTful APIs**, a robust and maintainable foundation for client applications. For the frontend, **ReactJS** is used for web interfaces, while **React** **Native** powers the mobile applications, providing an interactive user experience across platforms.

This application shows how to design and implement a scalable monolithic system using well-structured layers, separating concerns between the user interface, business logic, and data access, while enabling multiple client platforms to interact seamlessly via RESTful endpoints.

It is mainly designed by the **Spring** **ecosystem**, leveraging popular libraries that have been widely adopted by the community for years, combined with modern frontend technologies to provide a responsive, maintainable solution for real-world food delivery scenarios.

1. **Business context**

Potato covers the basic business scenarios for Merchant, Menu, Cart, Order, Payment, Rating and Access Control.

**With Merchant**, Customers can explore a list of merchants with options to search by name, filter and sort by type of food, rating, or location. They can also view detailed information of a selected merchant, including its name, introduction, type of food, rating, opening hours, estimated delivery time, and location. Merchant Admin can manage their own merchant details, while Admin has the ability to view all merchants, approve new ones, or deactivate existing ones.

With **Menu**, Customers can browse all the dishes offered by a specific merchant. They can also view detailed information of each available dish such as image, name, description and price. Merchant Admin has full control to managetheirmerchant’s menu.

With **Cart**, Customers can add any dish from themenu or from thedishdetailpage by choosing options provided by the Merchant, if needed. After adding, the dishes will appear in the cart along with their corresponding subtotal cost and a summary panel showing cart total cost. When Customers update the quantity of dishes from the cart, the summary panel and cart are updated accordingly. After all, Customers can proceed to checkout to place an order by clicking Checkout button from the cart view.

**With Order**, Customers can review the order details, including the list of dishes, quantity, price, delivery fee and total amount, before confirming. Customers can also track their order status after placing the order and they are allowed to cancel only when it is still in the “pending confirmation” status. Merchant Admin can confirm orders, update status, and view **order statistics** within a defined period (today, last 7 days, last 30 days), including the number of orders, revenue, and average order value. System Admin can monitor all orders for administration and summary purposes.

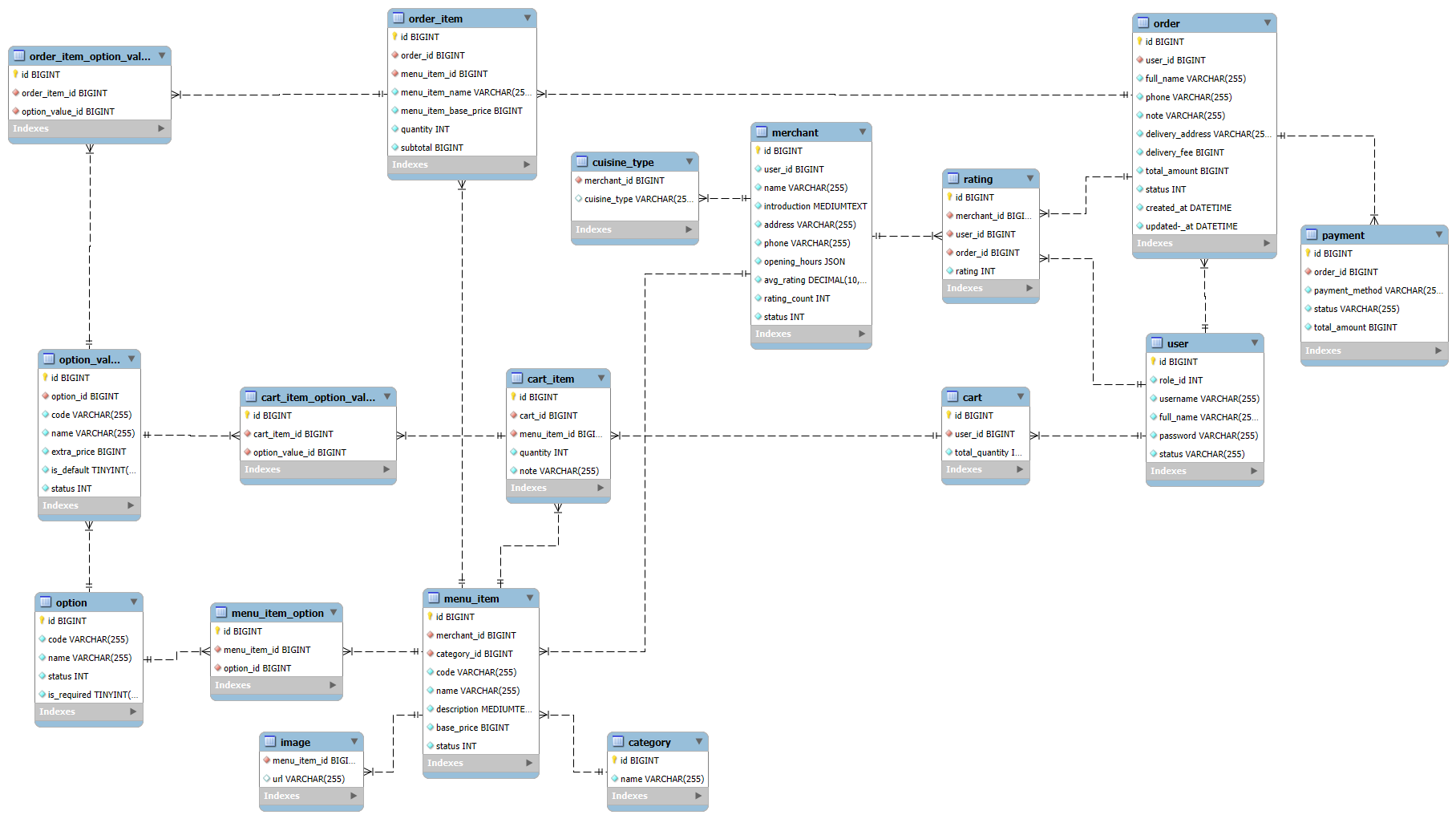
With **Payment**, Customers can pay via supported methods (online payment or cash on delivery) and the order status will be updated accordingly.

With **Rating**, Customers can rate for their orders after receiving them. Their rating will be used for calculating merchant’s average rating.

With **Access Control**, Customers, Merchant Admin, and System Admin can log in and log out the system. Besides, Customers and Merchant Admin can also sign up for a new account. After sign up as a Customer role, system will allow to login or to access protected resource. After sign up as a Merchant Admin role, the System Admin will decide to approve or reject the registration, if it is successful the Merchant Admin will be allowed to login. After login, Customers are directed to the home page, Merchant Admin to the merchant management page, and System Admin to the administration dashboard. When log out, Customers are directed to the home page, Merchant Admin and System Admin are directed to the login page.

Some **one-off setup tasks** when the system starts include creating Admin users, sample Customers, Merchant Admin, Delivery Staff accounts, and populating sample data for merchants, dishes, orders, and deliveries.

1. **Conceptual model**



1. **Business process**



1. **User stories** 
   1. **Merchant**
      1. User story 1.1: Browse merchants with Search, Filters and Sorting

As a Customer

I want to see a list of merchants with search, filtering and sorting

So that I can easily find merchants that match my preferences.

* Whenever searching by name in the search bar, then the list of merchants shall narrow down with appropriate merchants.
* Whenever filtering with any name, type of food, rating, or location of the merchant, then the list of merchants needs to narrow down with appropriate merchants.
* Whenever sorting with descending on rating, or location, then the list of merchants needs to follow with this sorting.
* Whenever both filtering and sorting are in action, then the list of merchants shall be effective by both of descriptions above.
  + 1. User story 1.2: View Merchant Information

As a Customer

I want to view detailed information of a selected merchant

So that I can decide whether to order from it.

* Whenever a Customer clicks on a merchant in the list, then the detail page shall display the name, introduction, cuisine type, rating, opening hours, estimated delivery time, and location of this merchant.
  + 1. User story 1.3: Manage Merchant Information

As a Merchant Admin

I want to update my merchant’s details

so that Customers always see the correct information.

* Whenever updating merchant information such as name, introduction, cuisine type, opening hours and location, then the system shall save the changes and reflect them to the Customer’s view.
  + 1. User story 1.4: View All Merchants

As a System Admin

I want to view all merchants

so that I have an overview of the platform’s available merchants.

* Whenever System Admin views all merchants, then the system shall display the complete list of registered merchants with their current status (approved, pending, deactivated).
  + 1. User story 1.5: Approve New Merchant

As a System Admin

I want to approve new merchants

so that only valid merchants are allowed to operate in the system.

* Whenever System Admin approves a new merchant, then the merchant shall appear as available for Customers.
  + 1. User story 1.6: Deactivate Merchant

As a System Admin

I want to deactivate existing merchants

so that inactive or violating merchants cannot continue serving customers.

* Whenever System Admin deactivates a merchant, then the merchant shall no longer be available for Customers.
  1. **Menu** 
     1. User story 2.1: Browse Merchant Menu

As a Customer

I want to browse all the dishes offered by a specific merchant

so that I can choose which one to order.

* Whenever a Customer opens a merchant’s menu, then the list of dishes shall be displayed in sections base on category.
* Each dish shall show at least its image, name, price and availability.
  + 1. User story 2.2: View Dish Details

As a Customer

I want to view detailed information of a dish  
So that I can decide if I want to order it.

* Whenever a Customer clicks on an available dish in the menu, then the detail page shall display the dish’s image, name, description and price
  + 1. User story 2.3: Manage Merchant Menu

As a Merchant Admin  
I want to manage my merchant’s menu  
So that customers always see the correct and updated food options.

* The system shall allow the Merchant Admin to **manage categories.**
* The system shall allow the Merchant Admin to **manage dishes under a category.**
* The system shall allow the Merchant Admin to **manage option groups** (e.g., Size, Toppings, Drinks)**.**
* The system shall allow the Merchant Admin manage option items under an option group (e.g., Small, Medium, Large; Extra Cheese; Pineapple).
* The system shall allow the Merchant Admin to assign one or many option groups to a dish.
* Whenever creating or updating a category, dish, or option group/item, the system shall save the changes and immediately reflect them in the Customer’s menu view.
* Whenever deleting a category, dish, or option group/item, it shall no longer appear in the Customer’s menu view.
  1. **Cart**
     1. User story 3.1: Add Dishes to Cart

As a Customer  
I want to add a dish into my cart  
So that I can prepare an order.

* Whenever a Customer want to add a dish to the cart, the system will display the details of that dish along with list of options.
* Whenever a Customer adds a dish from the **menu interface**, then the system shall add that dish into the cart with the **default quantity = 1.**
* Whenever a Customer adds a dish from the **dish detail interface**, then the system shall allow the Customer to adjust the desired quantity.
* Whenever a dish is added to the cart (in both cases), the menu interface shall display the quantity correspond to the selected dish.
* Whenever a dish is added to the cart (in both cases), then the **cart view** shall display the corresponding dish(es) with their selected quantities and then the system shall calculate **subtotal** and total.
  + 1. User story 3.2: Update Dish Quantity in Cart

As a Customer  
I want to update the quantity of a dish in the cart

so that I can adjust my order before checkout.

* Whenever a Customer increases or decreases the quantity of a dish from the menu interface or dish detail interface, then the cart shall update the dish quantity accordingly and then the system shall recalculate **subtotal** and total.
  + 1. User story 3.3: Remove a Dish from Cart

As a Customer  
I want to remove a dish in my cart  
So that I can adjust my order before my checkout.

* When the quantity of a dish is 1, and Customer want to decrease it to 0, that dish shall be removed from cart, and then subtotal and total shall be updated accordingly.
  + 1. User story 3.4: View Cart

As a Customer

I want to view my cart

so that I know what I added and the current order value so far.

* Whenever a Customer opens the cart view, then the system shall display the list of selected dishes and their quantities. The summary panel shall show the subtotal cost and the total cost.
  + 1. User story 3.5: Proceed to Checkout

As a Customer

I want to proceed to checkout from the cart

so that I can place an order.

* Whenever a Customer clicks the "Checkout" button from the cart view, then the system shall navigate to the checkout process.
  1. **Order** 
     1. User story 4.1: Review Order

As a Customer

I want to review my order details before confirming

so that I can make sure everything is correct.

* Whenever a Customer proceeds to order, then the system shall display the list of dishes, their quantities, unit price, subtotal, delivery fee, and total amount.
* Whenever a Customer confirms the order, then the system shall save the order with status = “pending confirmation”.
  + 1. User story 4.2: Track Order Status

As a Customer

I want to track my order status after placing an order

so that I know the progress of my order.

* Whenever the Merchant Admin updates the order status, then the Customer shall see the updated status in their order tracking view.
  + 1. User story 4.3: Cancel Order

As a Customer

I want to cancel my order

so that I can withdraw before the merchant starts preparing it.

* Whenever a Customer attempts to cancel an order in “pending confirmation” status, then the system shall allow cancellation and update the order status to “cancelled”. The system shall notify to Merchant Admin and remove that order in Merchant Admin’s interface.
* Whenever a Customer attempts to cancel an order in any other status, then the system shall reject the cancellation and display an appropriate message.
  + 1. User story 4.4: Confirm and Update Order Status

As a Merchant Admin

I want to confirm incoming orders and update their statuses

so that I can process customer requests properly.

* Whenever a new order is in “pending confirmation” status, then the Merchant Admin shall be able to confirm it, changing its status to “confirmed”.
* Whenever Merchant Admin processes an order, then they shall be able to update its status (e.g., delivering, completed).
  + 1. User story 4.5: View Order Statistics

As a Merchant Admin

I want to view order statistics within a defined period

so that I can monitor business performance.

* Whenever Merchant Admin selects a time period (today, last 7 days, last 30 days), then the system shall display the total number of orders, total revenue, and average order value for that period.
  + 1. User story 4.6: Monitors All Orders

As a System Admin

I want to monitor all orders in the system

so that I can have an overview for administration and summary purposes.

* Whenever System Admin views the order monitoring page, then the system shall display all orders across all merchants with their details and current statuses.
* Whenever System Admin generates a summary, then the system shall display aggregated data such as total number of orders, total revenue, and distribution by status.
  1. **Payment**
     1. User story 5.1: Pay for Order

As a Customer  
I want to pay for my order via supported methods  
So that I can complete my purchase.

* Whenever a Customer selects online payment, then the system redirects them to the secure payment gateway.
* Whenever an online payment is successful, then the order status shall update to “paid”.
* Whenever an online payment is **unsuccessful**, then the order shall remain in its current status, and the Customer will be notified of the failure.
* Whenever a Customer selects **cash on delivery (COD),** then the order shall be marked as “COD”.
* Whenever a COD order is successfully delivered and confirmed, then the order status will be updated to “completed”.
  1. **Rating**
     1. User story 6.1: Rate Order

As a Customer  
I want to rate my order after receiving it  
So that I can share my experience.

* The system shall allow Customers to rate only when the order status is "delivered".
* The rating scale is 1 to 5 stars.
* The system will use that rating data to calculate merchant’s average rating and rating counts.
  1. **Access Control**
     1. User story 7.1: Log in

As a User (Customer/Merchant Admin /Admin)  
I want to log in  
So that I can access my view.

* Whenever a user with a Customer role does a login, then I will be brought to the home page/merchant page.
* Whenever a user with a Merchant Admin role does a login, then I will be brought to the merchant management page.
* Whenever a user with a System Admin role does a login, then I will be brought to the administration page.
  + 1. User story 7.2: Log out

As a User (Customer/Merchant Admin/System Admin)

I want to log out

so that I can securely end my session.

* Whenever a User clicks “Logout”, then the system shall terminate the current session.
* After logout, Customer will be directed to the home page (public).
* Whenever a Customer tries to access a protected page after logout, then the system shall redirect them back to the login page.
* After logout, Merchant Admin and System Admin will be directed to the login page.
  + 1. User story 7.3: Customer Registration

As a Customer

I want to register for an account

So that I can log in and place orders in the system

* Customer provides valid information (username, unique email/phone, strong password).
* System creates a new Customer account with role = "Customer".
* After successful registration, system allows Customer to login.
  + 1. User story 7.4: Merchant Admin Registration

As a Merchant Admin

I want to register for an account

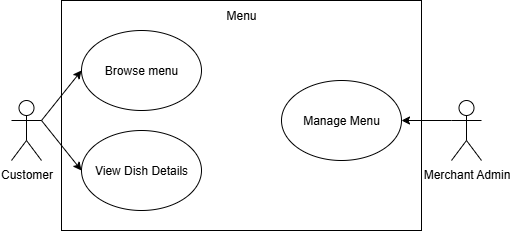
So that I can manage my merchant, menu, and orders on the platform.

* Merchant Admin provides valid business information.
* System creates a new Merchant Admin account with role = " Merchant Admin ".
* Merchant Admin account requires approval from System Admin before activation.
* After successful registration (or approval), system allows Merchant Admin to login.

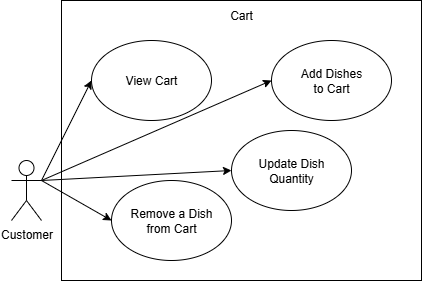
1. **Use cases**
   1. **Merchant**



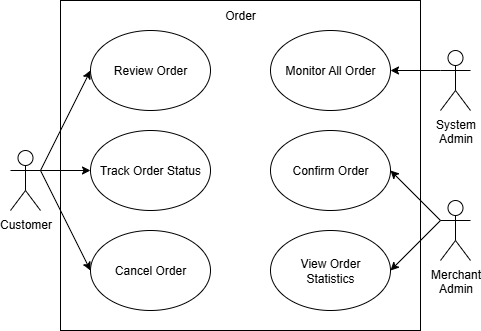
* 1. **Menu**



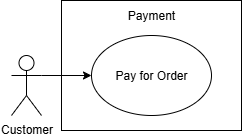
* 1. **Cart**



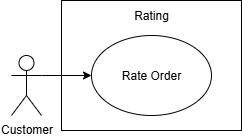
* 1. **Order**



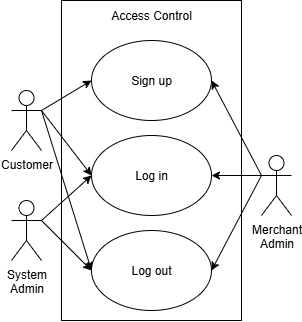
* 1. **Payment**



* 1. **Rating**



* 1. **Access Control**



1. **Rest APIs**
   1. **Merchant**

* Customer see a list of merchants with search, filtering and sorting 🡪 get merchants by criteria (GET)
* Customer view detailed information of a selected merchant 🡪 get details of a specific merchant (GET)
* Merchant Admin update merchant’s details 🡪 update (PUT)
* System Admin view all merchants 🡪 get all merchants (GET)
* System Admin approve new merchants 🡪 create merchant (POST)
* System Admin deactivate existing merchants 🡪 update merchant’s status (PUT)
  1. **Menu**
* Customer browse all the dishes offered by a specific merchant 🡪 get all dishes (GET)
* Customer view detailed information of a dish 🡪 get details of a specific dish (GET)
* Merchant manage merchant’s menu -> create (POST), retrieve (GET), update (PUT), delete (DELETE)
  1. **Cart**
* Customer add a dish to cart 🡪 create a cart with selected dish (POST)
* Customer update quantity of a dish 🡪 update (PUT)
* Customer remove a dish 🡪 delete (DELETE)
* Customer view cart 🡪 get cart (GET)
* Customer proceed to checkout 🡪 create an pre-order (POST)
  1. **Order**
* Customer track order status 🡪 get details of an order (GET)
* Customer cancel order 🡪 update status of an order (PUT)
* Merchant Admin confirming incoming orders and update their statuses 🡪 update status of an order (PUT)
* Merchant Admin view order statistics within a defined period 🡪 get order statistics (GET)
* System Admin monitor all orders 🡪 get all orders (GET)
  1. **Payment**
  2. **Rating**
* Customer rate order 🡪 create rating for an order (POST)
  1. **Access Control**
* User (Customer/Merchant Admin/System Admin) log in (POST)
* User (Customer/Merchant Admin/System Admin) log out (POST)
* User (Customer/Merchant Admin) register (POST)

1. **User Interfaces**
   1. **Home page**

Theese are some user interfaces with role Customer, meeting several requirements: searching, cart, filtering with cuisine type, list of active merchants, log in/register

