Design Overview for Swin-Bite

Name: Show Wai Yan

Student ID: 105293041

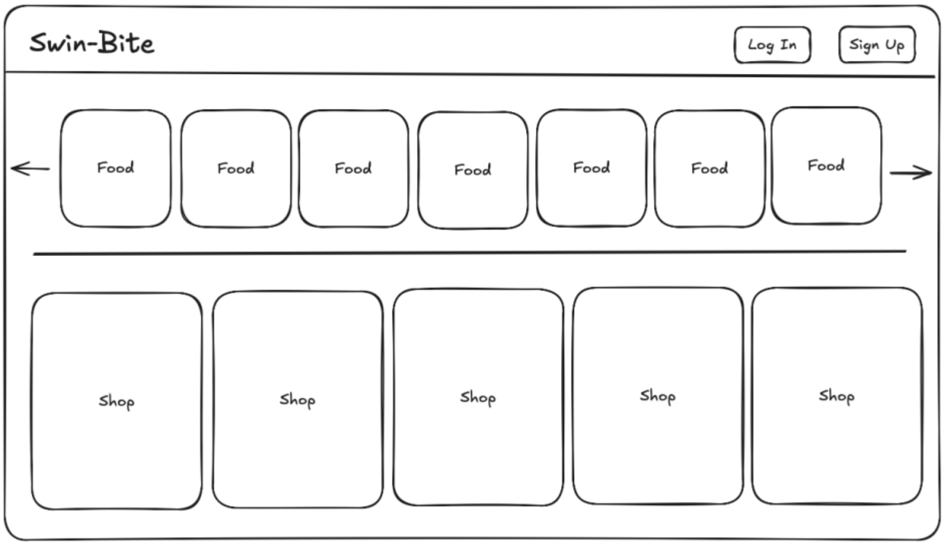
Summary of Program

The Food Delivery Web Application is a comprehensive online platform that connects customers directly with restaurants through a streamlined ordering system. The application allows customers to browse restaurants, view menus with various food categories (dishes, drinks, and snacks), add items to their shopping cart, and place orders with integrated payment processing. Restaurants can register as users on the platform, manage their own establishments, update their menus in real-time, and track incoming orders efficiently.

The system is designed with a clean object-oriented architecture where restaurants operate as direct users of the platform rather than being managed by separate owner entities. This approach eliminates unnecessary complexity while maintaining full functionality. Customers can create accounts, browse multiple restaurants, build custom orders through an intuitive shopping cart system, and complete purchases using secure bank account integration. The platform also includes a delivery service component that handles order fulfilment and provides real-time status updates from order placement to delivery completion.

The application will be implemented using C# and ASP.NET Core Web API for robust backend services, with a separate frontend built using HTML, CSS, and JavaScript to ensure clear separation of concerns for the assessment requirements. The system supports comprehensive user management, menu management, order processing, payment handling, and creating a complete end-to-end food delivery solution.

Sketch of sample output



Required Roles

User Class Details

Responsibility	Type Details	Notes
User authentication	Fields: userId (int), username (string), email (string), password (string)	Base class for all users
Profile management	Methods: Login() -> bool, Logout() -> void, UpdateProfile() -> bool	Common user operations
Access control	Properties: IsAuthenticated (bool), UserType (UserType enum)	Determines user permissions

Customer Class Details

Responsibility	Type Details	Notes
Shopping cart management	Fields: shoppingCart (ShoppingCart), orderHistory (List<Order>)	Inherits from User
Order placement	Methods: AddToCart(Food item) -> void, PlaceOrder() -> bool	Customer-specific functionality
Restaurant browsing	Methods: BrowseRestaurants() -> List<Restaurant>, ViewMenu(Restaurant) -> List<Food>	Food discovery features

Restaurant Class Details

Responsibility	Type Details	Notes
Restaurant information	Fields: restaurantId (int), name (string), address (string), rating (double)	Inherits from User
Menu management	Fields: menu (List<Food>), category (string), operatingHours (string)	Restaurant operations
Order processing	Methods: ProcessOrder(Order) -> bool, GetMenu() -> List<Food>, AddMenuItem(Food) -> bool	Core restaurant operations
Business management	Methods: ViewOrders() -> List<Order>, UpdateOrderStatus() -> bool, UpdateMenu() -> bool	Restaurant capabilities

Food Class Details

Responsibility	Type Details	Notes
Food item data	Fields: foodId (int), name (string), price (decimal), description (string)	Basic food information
Category management	Fields: category (FoodCategory), restaurant (Restaurant)	Food classification
Availability	Properties: IsAvailable (bool), PrepTime (int)	Order management support

Dish Class Details

Responsibility	Type Details	Notes
Main course items	Fields: servingSize (string), spiceLevel (int)	Inherits from Food
Nutritional info	Fields: calories (int), ingredients (List<string>)	Detailed dish information

Drink Class Details

Responsibility	Type Details	Notes
Beverage items	Fields: volume (string), temperature (Temperature enum)	Inherits from Food
Beverage specifics	Fields: isCarbonated (bool), hasAlcohol (bool)	Drink characteristics

Snack Class Details

Responsibility	Type Details	Notes
Light food items	Fields: packageSize (string), isHealthy (bool)	Inherits from Food
Snack properties	Fields: allergens (List<string>)	Allergy information

Shopping Cart Class Details

Responsibility	Type Details	Notes
Cart management	Fields: items (List<Food>), quantities (Dictionary<Food, int>)	Item storage
Price calculation	Methods: AddItem(Food) -> void, RemoveItem(Food) -> void, CalculateTotal() -> decimal	Cart operations
Order conversion	Methods: Clear() -> void, ConvertToOrder() -> Order	Checkout functionality

Bank Account Class Details

Responsibility	Type Details	Notes
Payment processing	Fields: accountNumber (string), balance (decimal), accountHolder (User)	Financial management
Transaction handling	Methods: ProcessPayment(decimal amount) -> bool, GetBalance() -> decimal	Payment operations
Security	Fields: pin (string), isActive (bool)	Account security

Delivery Service Class Details

Responsibility	Type Details	Notes
Order delivery	Fields: deliveryId (int), estimatedTime (DateTime), status (DeliveryStatus)	Delivery tracking
Delivery management	Methods: UpdateStatus() -> void, CalculateDeliveryTime() -> int	Delivery operations

UserType Enumeration Details

Value	Notes
Customer	Regular customers who place orders
Restaurant	Restaurant users who manage establishments

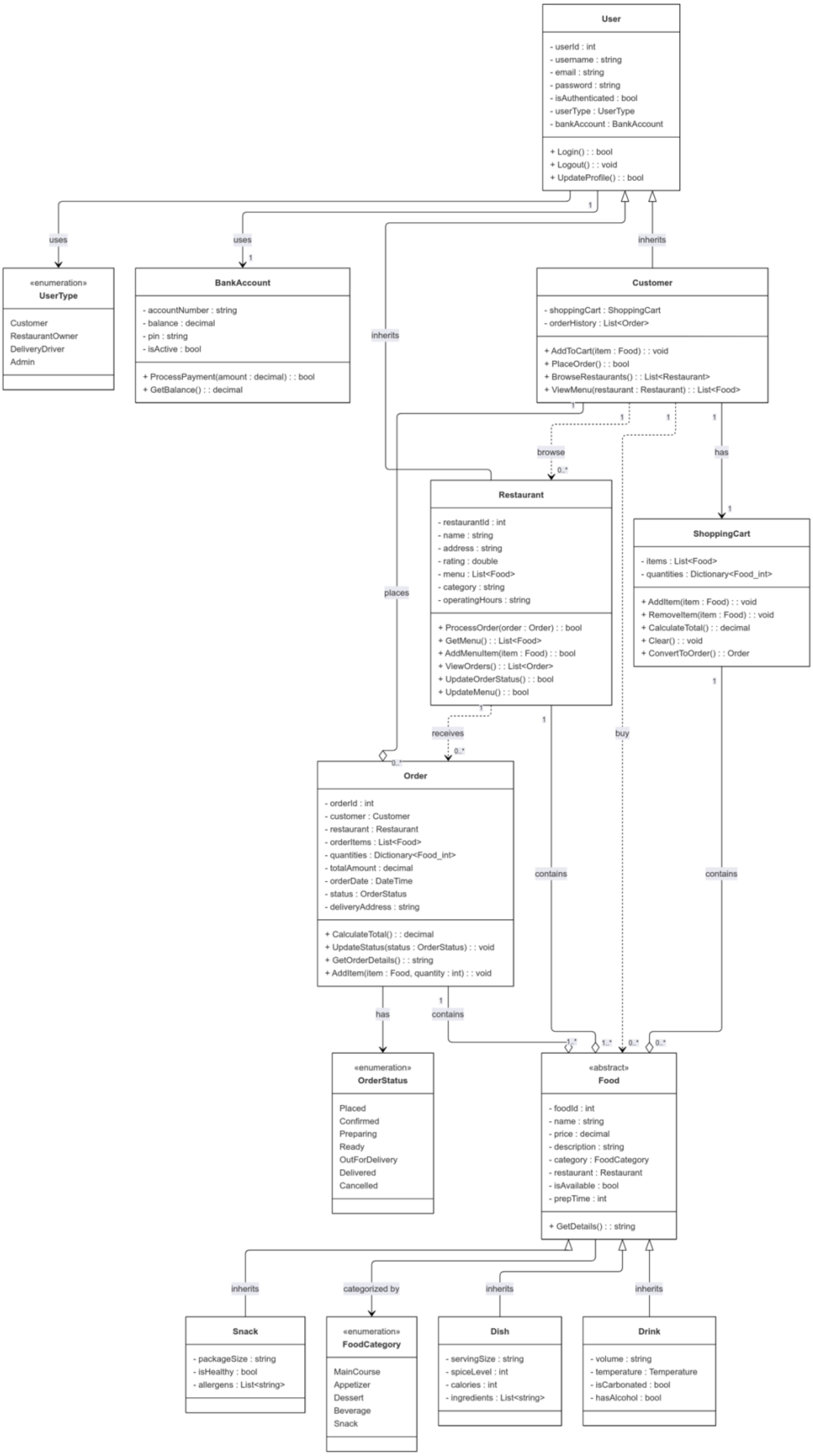
FoodCategory Enumeration Details

Value	Notes
Dish	Primary dishes and meals
Beverage	All types of drinks
Snack	Light food items

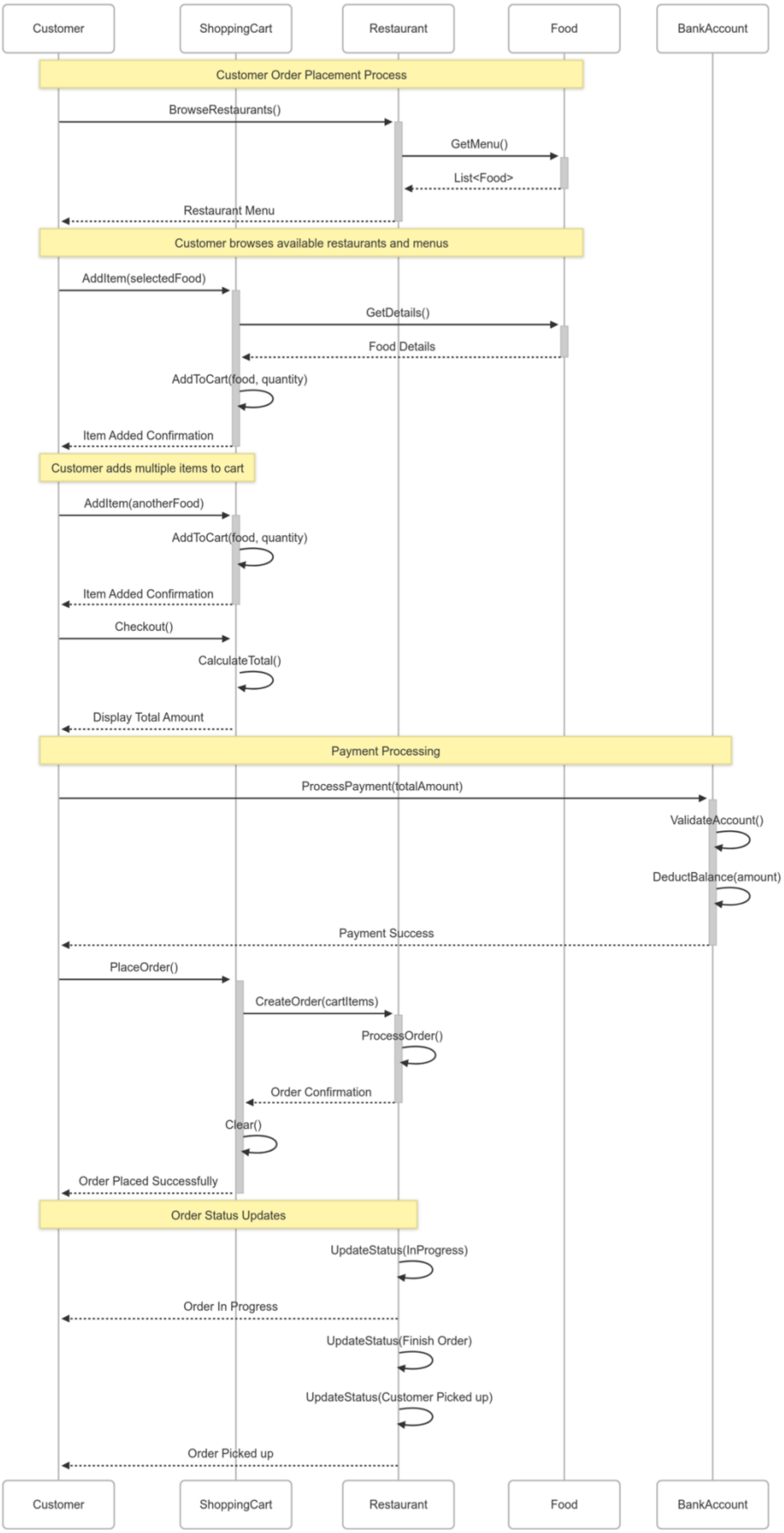
DeliveryStatus Enumeration Details

Value	Notes
Pending	Order received, not yet assigned
InProgress	Order being prepared
OutForDelivery	Order out for delivery
Delivered	Order successfully delivered
Cancelled	Order cancelled

Class Diagram



Sequence Diagram


Customer
ShoppingCart
Restaurant
Food
BankAccount
Customer Order Placement Process
BrowseRestaurants()
GetMenu()
List<Food>
Restaurant Menu
Customer browses available restaurants and menus
AddItem(selectedFood)
GetDetails()
Food Details
AddToCart(food, quantity)
Item Added Confirmation
Customer adds multiple items to cart
AddItem(anotherFood)
AddToCart(food, quantity)
Item Added Confirmation
Checkout()
CalculateTotal()
Display Total Amount
Payment Processing
ProcessPayment(totalAmount)
ValidateAccount()
DeductBalance(amount)
Payment Success
PlaceOrder()
CreateOrder(cartItems)
ProcessOrder()
Order Confirmation
Clear()
Order Placed Successfully
Order Status Updates
UpdateStatus(InProgress)
Order In Progress
UpdateStatus(Finish Order)
UpdateStatus(Customer Picked up)
Order Picked up
Customer
ShoppingCart
Restaurant
Food
BankAccount