



Software design specification document 2023

Project Team

ID	Name	Email
20200484	Mohamed Yasser Shehta	mohammedyasser0110@gmail.com
20200318	Abdelrhman Mohamed Ahmed	3bdelrhmanelsha3er@gmail.com
20200618	Hala Gamal Meselhy	hg201420142014@gmail.com
20200525	Mariam Alaa Eldeen Ahmed	mariemalaa2002@gmail.com



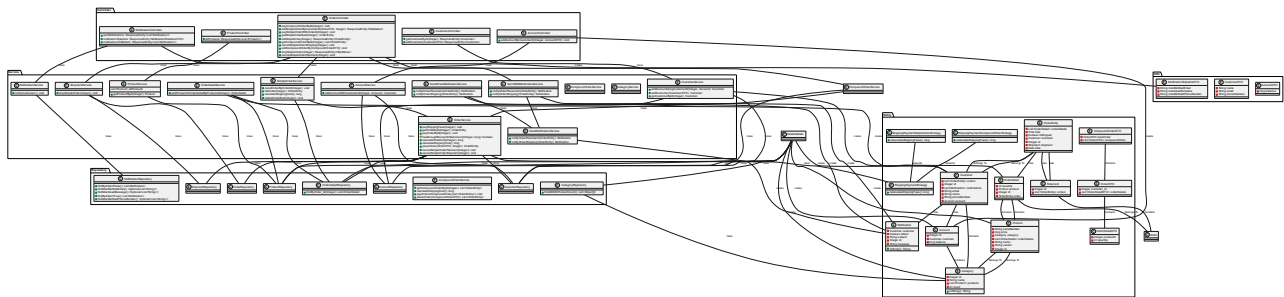
CS352: Sprint SDS – Order and Notifications Sys

SDS document

Contents

Class diagram design	2
Class diagram Explanation	3
Requirements Exposure as Web Service API	4
Requirement	4
Exposed API	4
Project Information	8

Class diagram design





CS352: Sprint SDS – Order and Notifications Sys

SDS document

Class diagram Explanation

- Strategy Pattern:

- **Participating Classes:** ShippingPaymentStrategy, ShippingPaymentSimpleOrderStrategy, ShippingPaymentCompoundOrderStrategy
- **Justification:** The Strategy Pattern is used to define a family of algorithms for calculating shipping fees. ShippingPaymentStrategy is the base interface, and ShippingPaymentSimpleOrderStrategy and ShippingPaymentCompoundOrderStrategy are concrete implementations representing different strategies for calculating shipping fees.

- Observer Pattern:

- **Participating Classes:** Notification, NotificationController, NotificationService, OrderController, OrderService
- **Justification:** The Observer Pattern is implied in the notification mechanism. When an event, such as placing an order or shipping an order, occurs, it triggers notifications. Classes like NotificationController and NotificationService observe these events and handle the notification logic.

- Singleton Pattern:

- **Participating Classes:** DataInitializer, OrderAndNotificationsManagementApplication
- **Justification:** The Singleton Pattern ensures that there is only one instance of the DataInitializer and OrderAndNotificationsManagementApplication classes. This can be useful for scenarios where a single point of initialization or application entry is required.

- Service Layer Pattern:

- **Participating Classes:** AccountService, CategoryService, CustomerService, NotificationService, OrderService, ProductService, ShipmentService, SendEmailNotificationService, SendNotificationService, SendSMSNotificationService, CompoundOrderService, SimpleOrderService
- **Justification:** The Service Layer Pattern is used to encapsulate business logic and orchestrate interactions between different entities. Service classes provide a higher-level interface to the application's functionality, handling complex operations involving multiple entities.

- DTO (Data Transfer Object) Pattern:

- **Participating Classes:** AccountDTO, CustomerDTO, CompoundOrderDTO, OrderDTO, OrderDetailDTO, NotificationStatisticsDTO



CS352: Sprint SDS – Order and Notifications Sys

SDS document

- **Justification:** DTOs are used to transfer data between different layers of the application. For example, OrderDTO represents data that needs to be transferred between the client and the server when placing an order.

Requirements Exposure as Web Service API

Part 1: Exposed Postman Collection

<https://blue-station-897580.postman.co/workspace/My-Workspace~951771ff-1bde-47df-83af-945964a53265/collection/32044519-671ea858-4553-496c-a10c-49ddd747734c?action=share&creator=32044519>

Part 2: Mapping Explanation

Requirement	Exposed API
The System Should Retrieve All Products	<ul style="list-style-type: none">▪ API Operation: GET /product/get-all-products▪ Description: Retrieve a list of all products.
The System Should Retrieve Customer Information by ID.	<ul style="list-style-type: none">▪ API Operation: GET /customer/get-customer-by-id/{id}▪ Description: This API allows you to retrieve detailed information about a customer by specifying their unique ID. <p>Input:</p> <ul style="list-style-type: none">▪ Path parameter: {id} - The unique identifier of the customer.
The System Should Add a New Customer.	<ul style="list-style-type: none">▪ API Operation: POST /customer/add-customer



CS352: Sprint SDS – Order and Notifications Sys

SDS document

		<ul style="list-style-type: none"> ▪ Description: This API operation is used to add a new customer to the system with the provided customer details. <p>Input:</p> <ul style="list-style-type: none"> ▪ Request Body: CustomerDTO : JSON payload with customer details (name, email, address).
The System Should Create a New Simple Order for A Specific Customer.		<ul style="list-style-type: none"> ▪ API Operation: POST /order/add-simple-order/{customer_id} ▪ Description: This API operation is used to place a new simple order for a specific customer, and it notifies the customer via SMS and email about the order placement. <p>Input:</p> <ul style="list-style-type: none"> ▪ Path parameter: {customer_id} - The unique identifier of the customer. ▪ Request Body: OrderDTO containing order details.
The System Should Create a New Compound Order.		<ul style="list-style-type: none"> ▪ API Operation: POST /order/add-compound-order ▪ Description: This API allows you to place multiple orders as a compound order. It pays for the orders within the compound order and notifies customers about the order placement. <p>Input:</p> <ul style="list-style-type: none"> ▪ Request Body: CompoundOrderDTO containing details of the compound order.



CS352: Sprint SDS – Order and Notifications Sys

SDS document

<p>The System Should Mark a Simple Order as Paid.</p>	<ul style="list-style-type: none"> ▪ API Operation: POST /order/add-compound-order ▪ Description: This API allows you to place multiple orders as a compound order. It pays for the orders within the compound order and notifies customers about the order placement. <p>Input:</p> <ul style="list-style-type: none"> ▪ Path parameter: {order_id} - The unique identifier of the simple order.
<p>The System Should Pay All Orders Within a Compound Order.</p>	<ul style="list-style-type: none"> ▪ API Operation: PUT /order/pay-compound-order/{order_id} ▪ Description: This API allows you to pay all orders within a compound order. <p>Input:</p> <ul style="list-style-type: none"> ▪ Path parameter: {order_id} - The unique identifier of the compound order.
<p>The System Should Retrieve Information About a Simple Order by ID.</p>	<ul style="list-style-type: none"> ▪ API Operation: GET /order/list-simple-order/{order_id} ▪ Description: This API allows you to retrieve details of a simple order by specifying its unique ID. <p>Input:</p> <ul style="list-style-type: none"> ▪ Path parameter: {order_id} - The unique identifier of the simple order.
<p>The System Should Retrieve Information About a Compound Order by ID.</p>	<ul style="list-style-type: none"> ▪ API Operation: GET /order/get-compound-order/{order_id}



CS352: Sprint SDS – Order and Notifications Sys

SDS document

		<ul style="list-style-type: none"> ▪ Description: This API allows you to retrieve information about a compound order by specifying its unique ID. <p>Input:</p> <ul style="list-style-type: none"> ▪ Path parameter: {order_id} - The unique identifier of the compound order.
The System Should Cancel a Simple Order.		<ul style="list-style-type: none"> ▪ API Operation: DELETE /order/cancel-order-placement/{order_id} ▪ Description: This API operation is used to cancel the placement of a simple order. <p>Input:</p> <ul style="list-style-type: none"> ▪ Path parameter: {order_id} - The unique identifier of the simple order.
Add Account to Customer		<ul style="list-style-type: none"> ▪ API Operation: POST /account/add-account/{customer_id} ▪ Description: This API allows you to add a new account to a specific customer. <p>Input:</p> <ul style="list-style-type: none"> ▪ Path parameter: {customer_id} - The unique identifier of the customer. ▪ Request Body: AccountDTO containing account details.
Get Notifications to Be Sent		<ul style="list-style-type: none"> ▪ API Operation: GET /notification/to-be-sent



CS352: Sprint SDS – Order and Notifications Sys

SDS document

		<ul style="list-style-type: none"> ▪ Description: This API allows you to retrieve a list of notifications that are scheduled to be sent.
Get Sent Notifications		<ul style="list-style-type: none"> ▪ API Operation: GET /notification/to-be-sent ▪ Description: This API allows you to retrieve a list of notifications that are scheduled to be sent.
Get Notification Statistics		<ul style="list-style-type: none"> ▪ API Operation: GET /notification/statistics ▪ Description: This API allows you to retrieve statistics related to notifications, including the most notified email, phone number, and the most used message template.

Project Information

Database:

This project uses the H2 database for storage. H2 is a lightweight, in-memory database that is convenient for development and testing purposes. It's important to note the following details about the database:

- **Database Type:** H2 Database Engine
- **Configuration:** In the current setup, the database is configured to run in-memory, meaning that any data stored will be lost when the application is stopped.
- **Persistence:** For a production environment, it is recommended to configure a more robust and persistent database.

Project Port:

The application is configured to run on a specific port. Please make sure to set the appropriate port when making requests. By default, the port is set to 8081, but it can be configured.

Notes:



CS352: Sprint SDS – Order and Notifications Sys

SDS document

Base URL: Ensure that you set the appropriate base URL before making requests. The default base URL is <http://localhost:8081/>.