**Cairo University Faculty of Computers and Artificial Intelligence**

A logo with a yellow and green and blue text

Description automatically generated with medium confidenceA computer and map of the world

Description automatically generated

**Software design specification document**

**2023**

**Project Team**

|  |  |  |
| --- | --- | --- |
| **ID** | **Name** | **Email** |
| 20216133 | Raneem Anwar Mohamed | RaneemAnwar37@outlook.com |
| 20216074 | Amr Emad El deen | moramoda99@gmail.com |
| 20216081 | Karim Yasser Mohamed | karimyasser1510@gmail.com |
| 20216131 | Mohamed Talal Abdelmomen | mohamedtalal881@gmail.com |

**Contents**

[Class diagram design 2](#_Toc3524)

[Class diagram Explanation 3](#_Toc3525)

[requirement exposure webservice……………………………………………………………………………………………………………4,5](#_Toc3525)

postman collection……………………………………………………………………………………………………………………………………6

# 

# Class diagram design :

A screenshot of a computer

Description automatically generated

# Class diagram Explanation:

**Strategy Pattern: (db)**

The Strategy Pattern is applied to handle database operations for entities like Customer, Product, and Order with a specific focus on implementing the in-memory database (Memory db). This approach provides flexibility, maintainability, and separation of concerns in the context of different database implementations.

**Abstract factory Pattern: (Notification)**

When a customer places an order, our application can use the abstract factory to create a notification factory based on the desired notification channel (email or SMS). Once we have the factory, we can use it to create specific notification objects (email or SMS) and send them to the customer.

**Composite Pattern: (Order)**

* Component (Order): Abstract class defining the common abstract for simple and compound orders.
* Leaf (SimpleOrder): Class representing an individual order.
* Composite (CompoundOrder): Class representing a composition of orders.

**Template Pattern: (Order)**

* Abstract Class (Order): Abstract class defining a template method (placeOrder) with a sequence of steps for order processing.
* Concrete Classes (SimpleOrder, CompoundOrder): Classes extending the template class and providing specific implementations for each step.

Requirements Exposure as Web Service API

|  |  |
| --- | --- |
| **Requirement** | **Exposed API** |
| A customer should be able to create an account and put a specific balance using that account. | 1-POST /customer/create  A service to create customer to login and use system. Input: name, email, password, phone number, location, balance. |
| The customer should be able to check if his account is registered. | 2-GET /customer/user/check  A service to check if the user exists. This service returns all user info. Input: email, password |
| This service returns all the registered customers on the system | 3-GET /customer/getAll |
| This service is used to create a product. | 4-POST /product/create  Input: name, vendor, price, category |
| This service returns all the created products. | 5-GET /product/getAll |
| This service returns a product by it’s name. | 6-GET /product/get/{name}  Input: product’s name |
| The customer places a simple order which would include single product or several products. | 7-POST /order/createSimpleOrder  Input: customer's email, product’s names |
| The customer places a compound order which includes various orders for himself or for his friends. | 8-POST /order/createCompoundOrder  Input: customer's emails, product's names |
| This service returns all placed orders by ID. | 9-GET /order/getID |
| This service returns all placed orders. | 10-GET /order/getAll |
| This service sends an email to the customer with his name and the product’s name he bought. | 11-POST /notification/sendEmail  Input: Order ID |
| This service sends an SMS to the customer with his name and the product’s name he bought. | 12-POST /notification/sendSMS  Input: Order ID |
| This service returns all notifications in queue. | 13-GET /notification/get |

**Postman collection link:**

**https://api.postman.com/collections/31547966-ad46be0d-6f43-4f23-9872-90570d3d97ea?access\_key=PMAT-01HK0S5GP6HEV3PEK24H3FM5NQ**

A close up of a thank you sign

Description automatically generated