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



**Software design specification document**

**2022**

**Project Team**

|  |  |  |
| --- | --- | --- |
| **ID** | **Name** | **Email** |
| 20200828 | Ibrahim Ismail Ibrahim | ibrahimisma3il@icloud.com |
| 20200323 | Abdallah Emad Abdelhamid | abdallaballo20@gmail.com |
| 20200591 | Nada Mohamed Aref |  |
| 20200301 | Abdallah Ahmed Shouker |  |

# Class diagram design

***Diagram, engineering drawing

Description automatically generated-Updating*** ***Class*** ***Diagram***

# 

# *Decomposition* *Class* *Diagram*

# 

# Class diagram Explanation

*  **strategy design pattern,We found that we need to create something specific in a lot of different ways and extract all of these algorithms into separate classes.**

**Participating:1. Payment<interface> implements WalletPayment, Ondelivery, CreaditCard.**

**2. IRecipe <interface> implements Monthly, Quarter.**

* **Factory method**

**We found that we need to create objects in a superclass, but allows subclasses to alter the type of objects that will be created.**

**Participating:**

**1. ServiceFactory, Service, MobileRechargeSerice, InternetPaymentService, LandLineService, DonationService.**

* **Abstract Factory method**

**We found that we need to produce families of related objects without specifying their concrete classes.**

**Participating: FactoryOf ServiceProviderFactory, LandlineProivderFactory, MobileProviderFactory, InternetProviderFactory, , ServiceProviderFactory, DonationProviderFactory ,ServiceProvider, We, Vodafone, Etisalat, Orange, MothlyRicpitLandLine, QuarterRicpitLandLine,Schools, CancerHospital, MGOs.**

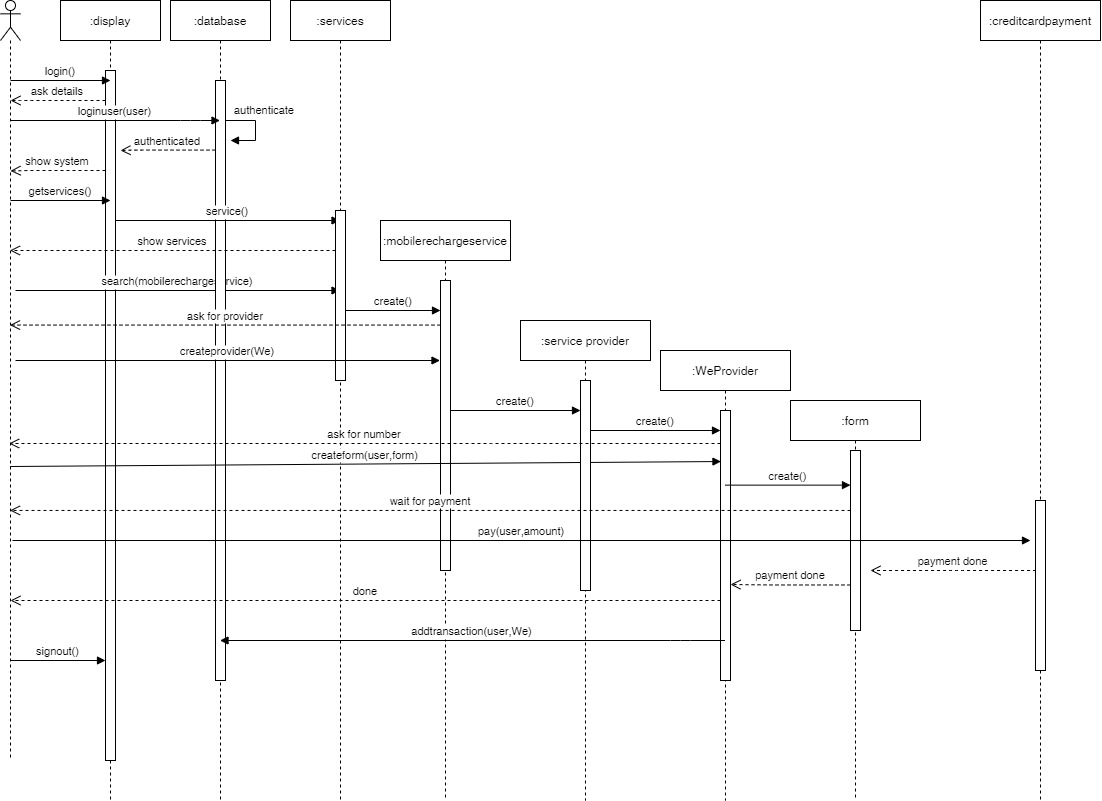
* **Decorator design pattern**

**We found that we need to attach new behaviors to objects by placing these objects inside special wrapper objects that contain the behaviors.**

**Participation: Service , DiscountDecorator,**

# Sequence diagram design

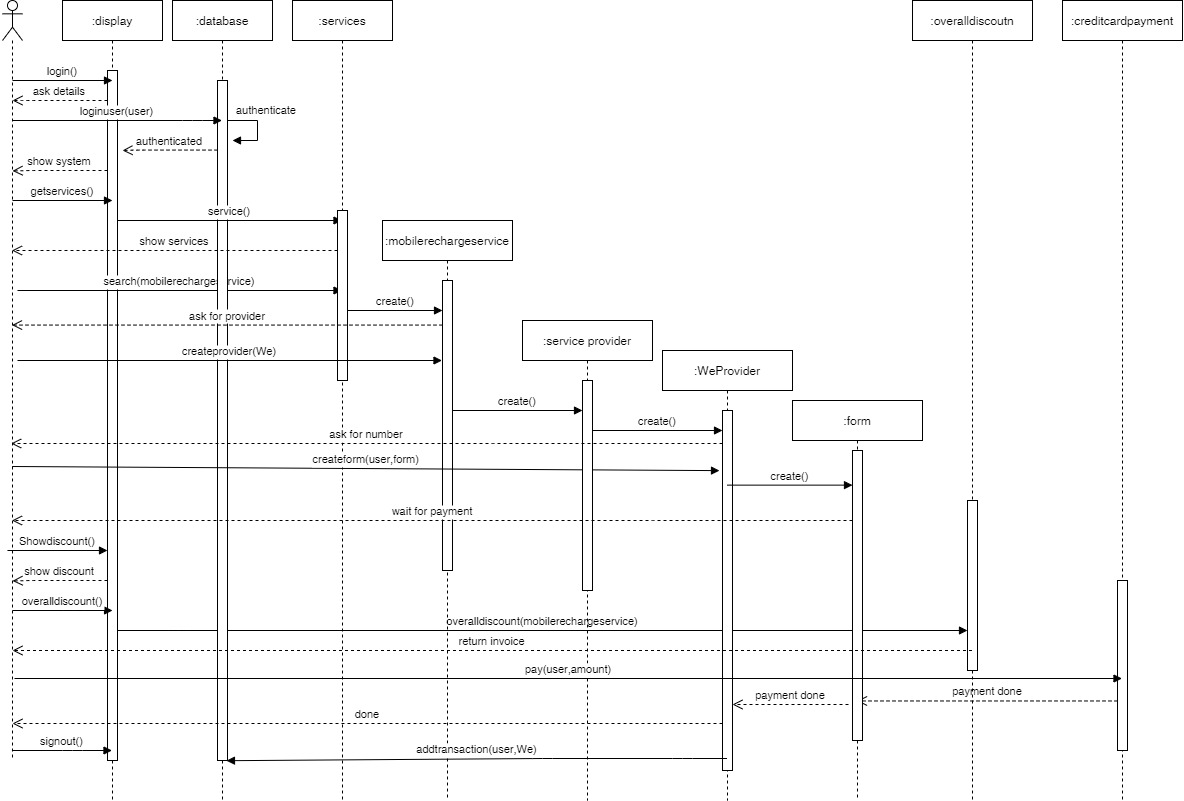
**(1)**



**(2)**

Diagram

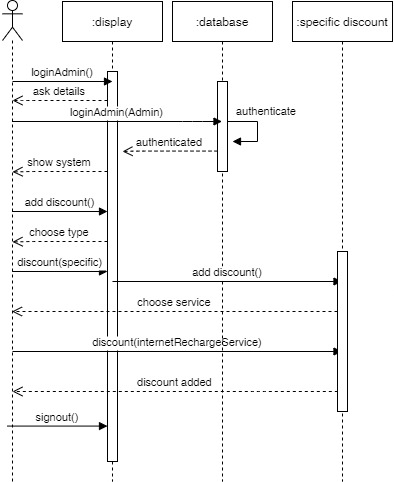
Description automatically generated

**(3)**

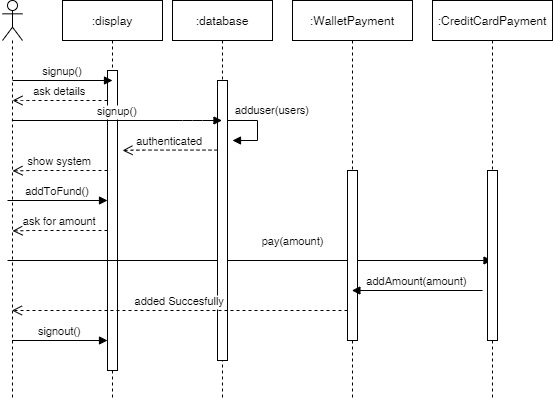
Diagram, schematic

Description automatically generated**(4)**

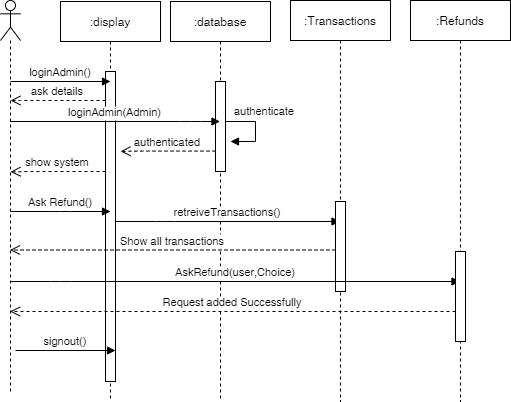
**(5)**



## (6)

**(7)**

**(8)**



## Requirements Exposure as Web Service API

**Part 1: Exposed Postman Collection**



**Part 2:**

|  |  |  |
| --- | --- | --- |
| **Requirement** |  | **Exposed API** |
| The user should be able to sign-in to the system. Given the user’s email and a password, the user can login to the system and use any of the system functionalities. |  | POST/loginUser |
| The user should be able to sign up to the system. The user should provide his username, email and password. The system should check if the username or the email is registered before, if they are not registered before then the signup process should complete successfully, if not, the system will show an error to the user |  | POST/signUp |
| The user or admin can signout |  | POST/signout |
| Admin can log in |  | POST/loginAdmin |
| The admin should be able to add specific discount to the system. |  | PUT/specificDiscount?servicename=internet |
| The admin should be able to add over All Discount to the system. |  | PUT/overAllDiscount |
| The admin should be able to list all refund requests. |  | GET/refunds |
| The admin should be able to accept or reject any refund request and if any refund request got accepted a refund transaction should be processed. |  | GET/refund?id=11&stauts=Accept |
| The admin should be able to list all user Transactions |  | GET/ShowAllUsersTransactions |
| The admin should be able to list all user Refunds |  | GET/ShowAllUsersRefunds |
| The user should be able to add any funds to the wallet. |  | PUT/AddToFund |
| The user can pay for any service in the system. The system should prompt the user to the payment form when the user asks to pay for any service. The default way is to pay via credit card. The system should allow the user to consume from the wallet (check Req. 6) for this payment. If the service that should to receive the payment accepts cache on delivery, then this option should be visible too. |  | GET/services/internet/we |
| The user can show all his refund and check their status (accepted or rejected) |  | GET/ShowmyRefunds |
| The user can show all his transactions |  | GET/transactions |
| The user should be able to search for any service in the system. The user can type the service name and the system will return all services that match the user query. |  | GET/services/internet |
| The user should be able to check any discount for any service in the system |  | GET/discounts |
| The user can ask for a refund for any complete transaction to any given service. Which Id=number of transaction |  | POST/askRefunds?id=1 |

# Github repository link

<https://github.com/Abdallah-Helal/paymentDifferentServices>