Cairo University Faculty of Computers and Artificial Intelligence



**Software design specification document**

**2022**

**Project Team**

|  |  |  |
| --- | --- | --- |
| **ID** | **Name** | **Email** |
| 20200126 | Ganna Allah Emad AbdElkhalek | gannaemadd@gmail.com |
| 20200605 | Nour Muhammed Anwar | nourmuhammed36@gmail.com |
| 20200446 | Mohamed Samy Atwa | hellosamysamy4@gmail.com |
| 20200231 | Salma Hany Gamal | salma.anany10@gmail.com |

Contents

[Class diagram design 2](#_Toc120811427)

[Class diagram Explanation 3](#_Toc120811428)

[Sequence diagram design 3](#_Toc120811429)

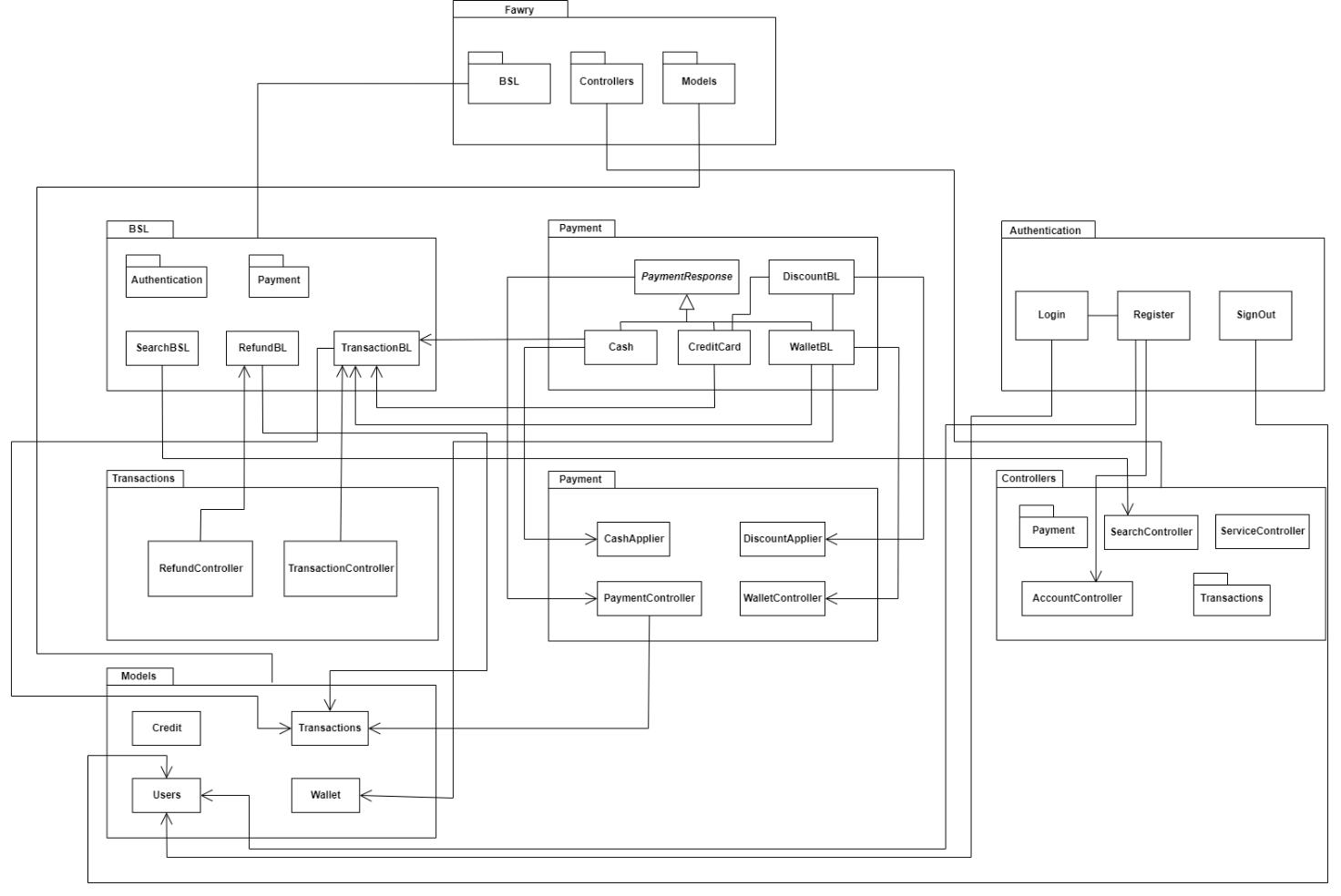
[Github repository link 4](#_Toc120811430)

# Class diagram design [E-PaymentSystem.drawio - diagrams.net](https://app.diagrams.net/#G1c7ub93FhSjpVT1x4KmQxb9gZsHfutHWP)

# Class diagram Explanation

* Factory design pattern in the services part as we will create many types of services, but they belong to one single family so the factory is well used here as it is a creational design pattern that provides an interface for creating objects in a superclass, but allows subclasses to change the type of objects that will be created.
* Observer pattern is used in notifying user when he requests to refund a transaction and the transaction request accepted or rejected
* Strategy design pattern is behavioral design pattern that is used in determining the payment method by user at runtime for different methods strategy (cash if acceptable - credit card -wallet) so we will be able to define a family of payment algorithms, put each of them into a separate class, and make their objects interchangeable.
* Template design pattern is also used in authentication and many methods in different classes

# Decomposition diagram Design



# Sequence diagram Design

1. Search sequence diagram:

Diagram

Description automatically generated

2) Pay With Credit Card sequence diagram:

**Diagram

Description automatically generated**

3) Sign In Sequence Diagram:

Diagram, table

Description automatically generated

4) Sign Up Sequence Diagram:

Diagram

Description automatically generated

5) Refund Transaction Sequence Diagram:

Diagram

Description automatically generated

6) Accept Refund Sequence Diagram:

Diagram

Description automatically generated

7) Add Money To Wallet Sequence Diagram:

Diagram

Description automatically generated

8) Discount Sequence diagram:

Diagram

Description automatically generated with low confidence

# Requirements Exposure as Web Service API

**https://elements.getpostman.com/redirect?entityId=24890605-14933fcb-5cb8-4bc6-a9a0-791c83b318e2&entityType=collection   
  
Part 2:**

|  |  |  |
| --- | --- | --- |
| Requirement | Input Example | Exposed API |
| The system registers the user and check if the username or the email is registered before. | **username: n**  **Password: n**  **email: nour@gmail.com**  **Wallet: 0**  **Credit :1000** | 1- POST auth/user/register  A service to check if the user exists or not if not register him. This service returns if the registration completed or not  Input: username, password, email, wallet, credit. |
| The system let the user login to use the services if the user has account. | username: n Password: n | 2- POST auth/user/login  A service to check if the user authenticated or not. This service returns the authorized menu. **Input: username, password.** |
| The system let the user search for services. | Reqservice: land | 3- GET /user/search/ A service for search for services provided. This service returns the service name. **Input**: **required service name**. |
| The system let the user pay for services. | serviceName: mobile serviceMoney: 100 method: credit | 4- POST /user/search/ A service allow user to pay service. This service return Transaction Done. Input: Service name, Payment Method, Service Money |
| The System notify the user for the accepted/rejected Refunds he has requested | No input required | 5-GET user/refunds/notifications A Service update the user Refund request notifications. This service return Refunds Transaction and its state. Input: No Input Required |
| The system allow user to add money from the credit to his wallet | walletValue: 100 | 6-PUT user/wallet/add A service add money to the user wallet. This service returns if the transaction done or not  Input: Required amount to add |
| The system allow user to request a refund for a transaction | Id: 2630  id may differ according to transaction made | 7-PUT user/refunds/add A service changes a transaction state to be refunded, returns refund state Input: Transaction ID |
| The system allows user/admin to sign out from the system | No input required | 8-PUT user/signoutPUT admin/signout A service changes user state from logged in to signed out  returns account state Input: no input |
| The system allow admin to login to the system | username: admin1Password: admin1 you can use admin1:3 | 9-POST auth/admin/login A service to check if the user authenticated or not. This service returns the authorized menu. **Input: username, password.** |
| The system allow admin to Accept a refund request for user and return money to his credit | Id: 2630  id may differ according to transaction made | 10-PUT admin/refunds/acceptA service to accept a refund request and change its state. This service returns the state of the operations. Input: Transaction ID |
| The system allow admin to reject a refund request for user. | Id: 2630  id may differ according to transaction made | 11-PUT admin/refunds/rejectA service to reject a refund request and change its state. This service returns the state of the operations. Input: Transaction ID |
| The system allow admin to apply discount on specific service | ServiceName : mobile | 12-PUT admin/apply/discount A service to apply the specific discount on a service. This service returns the state of the operations. **Input: Service name** |
| The system allow admin to list all Transaction done | No input required | 13-GET admin/wallet/alltrans/ A service list all the transactions on wallet. This service returns list of transactions containing username and amount added to wallet. **Input: no input** |
| The system allow admin to list all refunds requests done | No input required | 14-GET admin/transactions/ **A service list all the transactions. This service returns list of transactions for services**. **Input: no input** |
| The system allow admin to list all wallet transaction done | No input required | 15-GET admin/refunds/ **A service list all the refunds requests. This service returns list of refunds**. **Input: no input** |
| The system allow admin to add property for service provider for applying the cash | CashService: we | 1**6**-Post admin/services/cash/apply  **A service** add property for service provider for applying the cash. **Input: service provider name** |

# GitHub repository link

* [nourmuhammed20/Fawary\_E-Payment (github.com)](https://github.com/nourmuhammed20/Fawary_E-Payment)