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



**Advanced SE**

**Software design specification document**

**2022**

**Project Team**

|  |  |  |
| --- | --- | --- |
| **ID** | **Name** | **Email** |
| 20201118 | Ali Safwat | as20201118@gmail.com |
| 20200161 | Hussien Mostafa | Zeeka32@gmail.com |
| 20201127 | Amr Riyad | spyvip123@gmail.com |
| 20200800 | Adel Tarek | adeltarek202@gmail.com |

**Contents**

[Class diagram design](#_30j0zll) 2

[**Class diagram Explanation**](#_kywodgtxt1h4)2

[**Sequence diagram design**](#_100v7up9mnmb)4

[Github repository link](#_2et92p0) 11

# Class diagram design

# 

# 

# 

# 

# 

# Class diagram Explanation

* **setters and getters were not included in the class diagram, however, all needed setters and getters were used.**
* **Strategy Pattern: this pattern was used because it lets you define a family of algorithms, put each of them into a separate class, and make their objects interchangeable. The family of algorithms in question are, payment with wallet, payment with credit card, and payment on delivery. Classes that use this pattern are: IPaymentMethod, PayWithWallet, PayWithCreditCards, and CashOnDelivery.**
* **Factory Pattern: we create objects without exposing the creation logic to the client and refer to newly created objects using a common interface, we use this pattern to create new service providers without having the classes using these services be affected by the creation process. Classes that use this pattern are: ServiceFactory, ConcreteMobileFactory, ConcreteDonationFactory, ConcreteInternetFactory, ConcreteLandlineFactory, and these factories create all the service providers.**
* **Decorator Pattern: we used this pattern to apply different discounts to different users and services, classes that use this pattern are: CostManager, CalculatePrice, Discount, Specific, and Overall.**
* **Observer Pattern: we used this pattern to notify users when a refund request was approved or denied, and we also used it to notify the admin when a new refund request was issued. Classes that used this pattern are Admin, User, and IAccount.**

# 

# 

# 

# 

# 

# 

# 

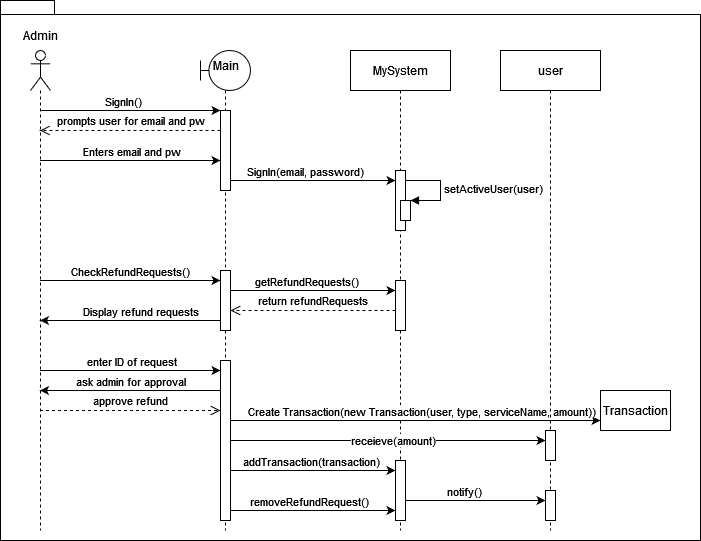
# 

# Sequence diagram design

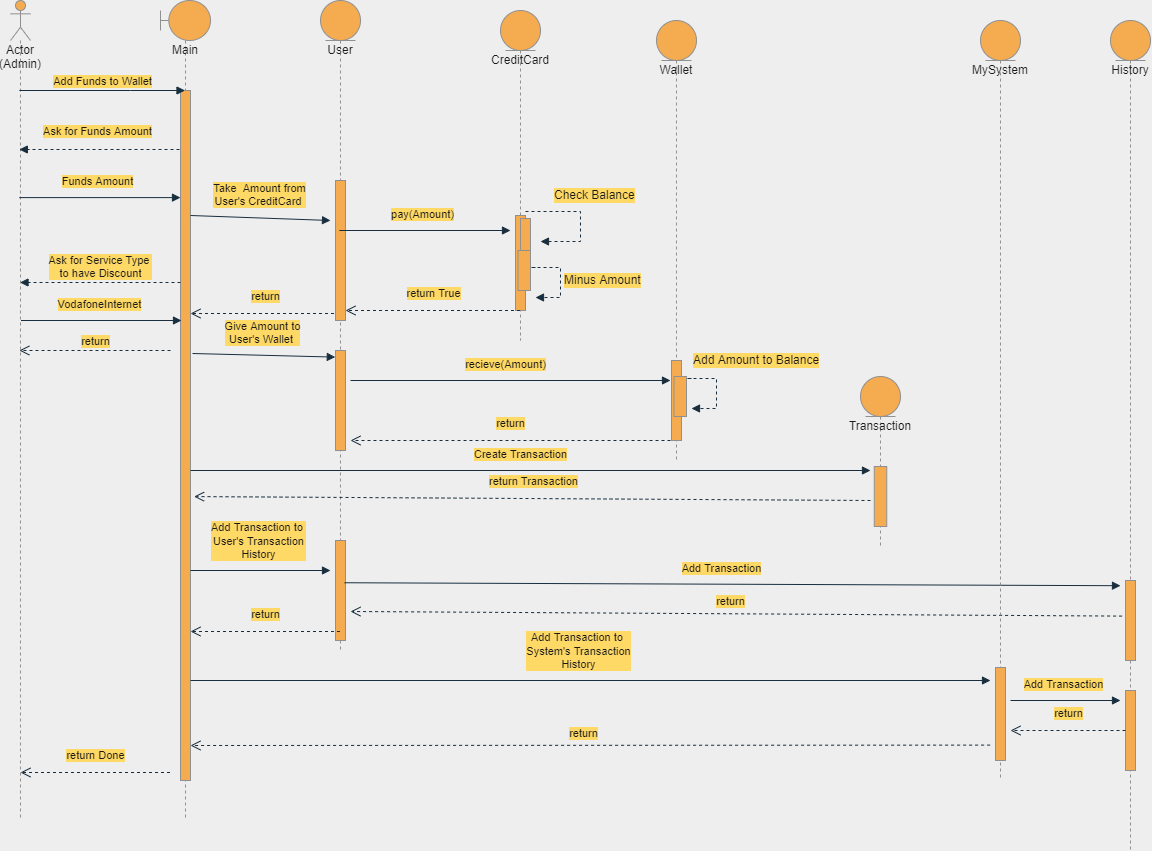
1- user paying for a landline service sequence diagram

# 

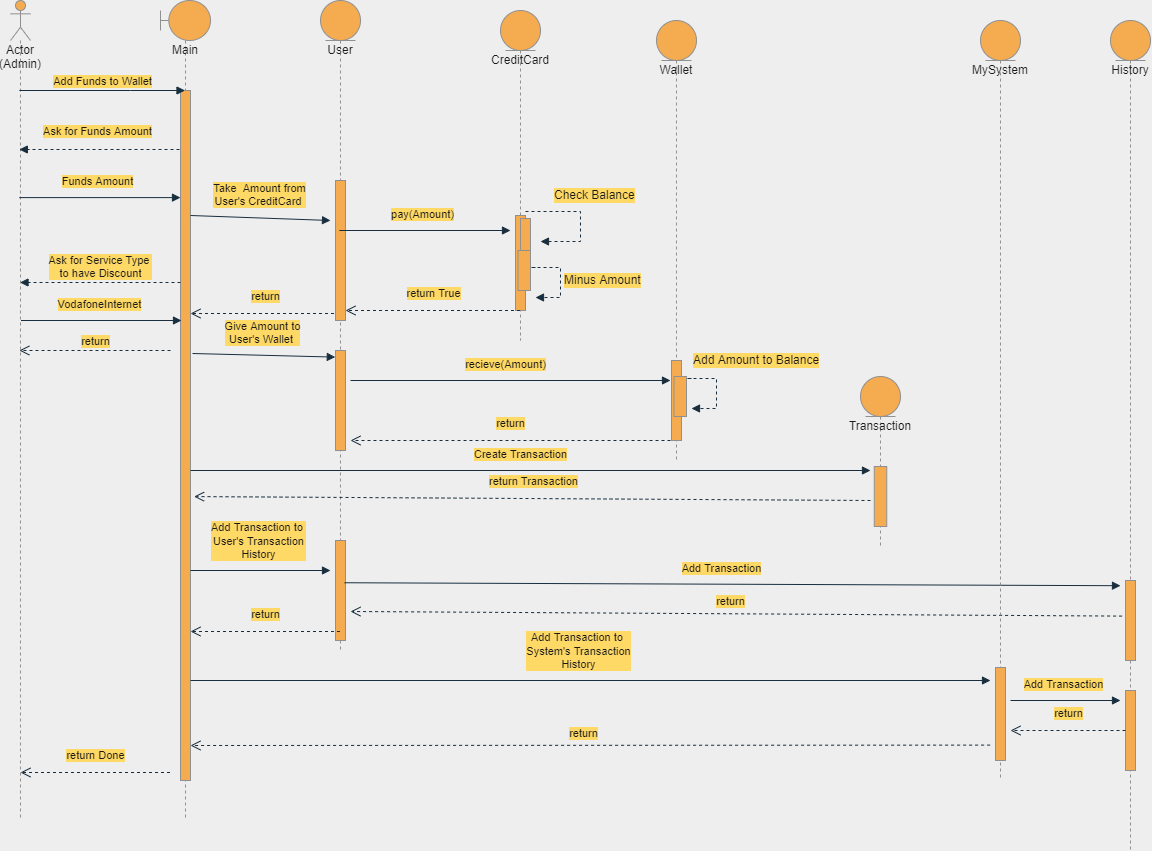
2- Admin Approving Refund Request



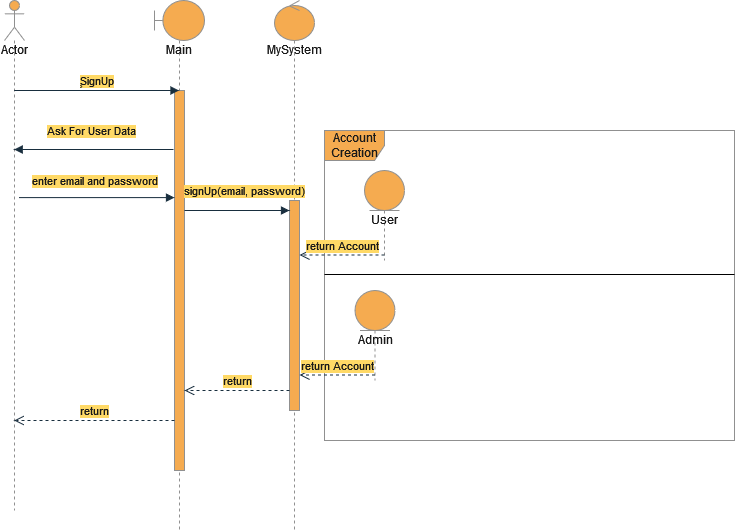
3- user adding funds to wallet



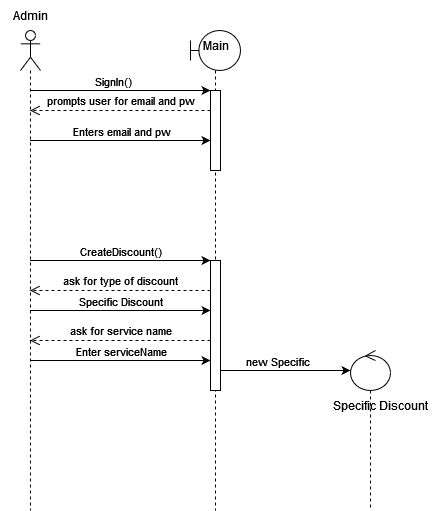
4- ask for refund

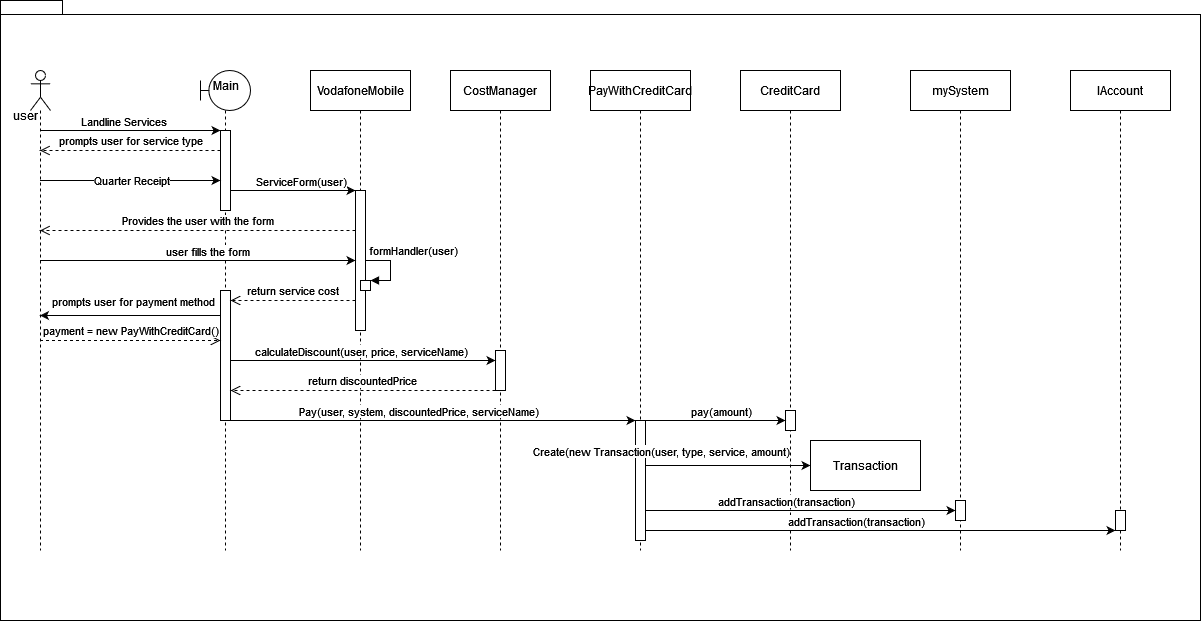


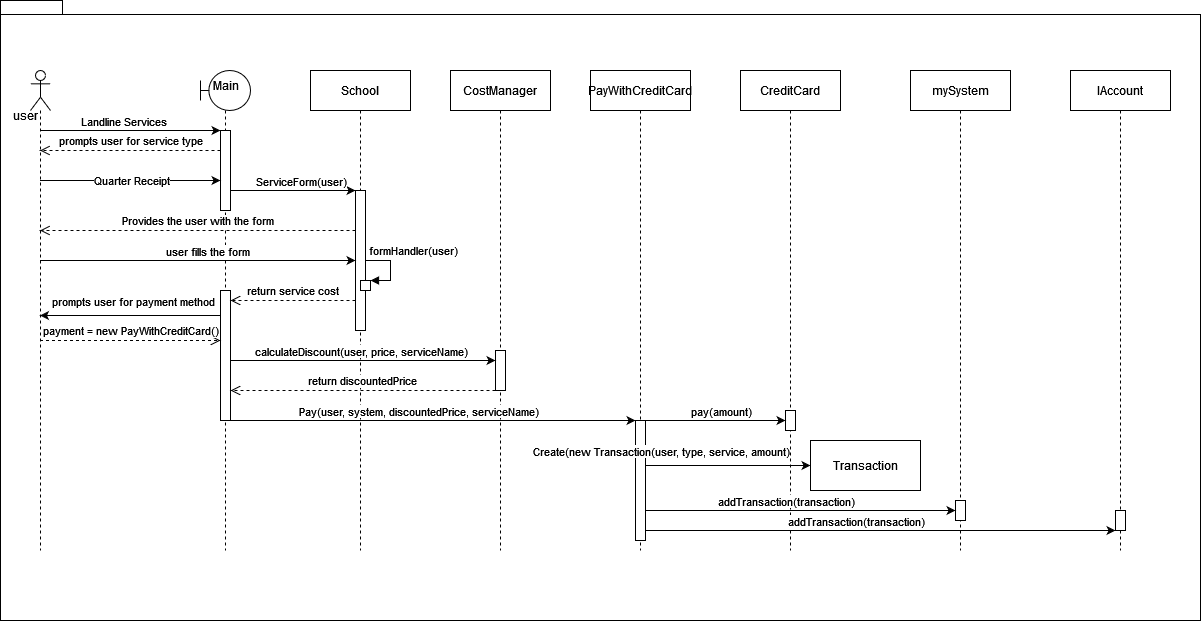
5- Account Creation Process



6- add specific discount



7- Do service with vodafone mobile with credit card

8- Do school donation 

# Github repository link

* **https://github.com/Zeeka32/Software-Engineering-2-Project**