Cairo University Faculty of Computers and Artificial Intelligence



**Software design specification document**

**2022**

**Project Team**

|  |  |  |
| --- | --- | --- |
| **ID** | **Name** | **Email** |
| 20200742 | Zeyad Mohamed Ali | zeyadmohamedali@gmail.com |
|  | Rana Hossam |  |
|  | Esraa Khalifa |  |

Contents

[Instructions[To be removed] 2](#_Toc120811426)

[Class diagram design 2](#_Toc120811427)

[Class diagram Explanation 3](#_Toc120811428)

[Sequence diagram design 3](#_Toc120811429)

[Github repository link 4](#_Toc120811430)

# Instructions[To be removed]

* **IMPORTANT. Rename this document to** Phase2\_LabGroupNumber\_Phase1\_StudentID1\_StudentID2\_StudentID3\_StudentID4**\_SDS Document.docx**
* **Remove the following notes and any red notes**

# Class diagram design

* **You should provide clean version for your class diagram design.**
* **Class diagram is a static diagram and should not represent any dynamic flow of events.**
* **Put Relationships between classes and the types of the relationships.**
* **Put multiplicity.**
* **Put relationship name (e.g. faculty "offer" course).**
* **Put attributes in the classes.**
* **Put functions &Put parameters.**
* **Put data types of each attributes and the parameters.**
* **Highly perfered: Each class has a corresponding interface**
  + **Let all objects parameters and returns be of interface type.**
* **See Shopping Cart Case Study**

**Diagram

Description automatically generated**

# Class diagram Explanation

* **Explain here the design pattern(s) that you used and your justification for using them, and the participating classes for each pattern.**

# Sequence diagram design

* **List Sequence diagrams for the most important user story (according to your opinion).**
* **Make sure that each object in the sequence diagram has a corresponding class in the class description table above. If not, it will be REJECTED.**
* **Put actual function calls with proper parameters and return types corresponding to class diagrams.**
* **Following are couple of examples for small / meduim examples. We expect such diagrams, however there is a missing thing in them. Most of calls don’t have parameters. Please always specify the parameters in the call, matching the class diagram.**



# Requirements Exposure as Web Service API

**Part 1:** [**https://www.postman.com/zozmo/workspace/fawry-requests/collection/25105422-d1a6373e-5f06-4ab3-afe6-2822b9fd5706?action=share&creator=25105422**](https://www.postman.com/zozmo/workspace/fawry-requests/collection/25105422-d1a6373e-5f06-4ab3-afe6-2822b9fd5706?action=share&creator=25105422)

**Part 2:**

**Explain here the exact mapping between every single requirement and its corresponding web service API operation. A sample example is provided to better explain the concept.**

|  |  |  |
| --- | --- | --- |
| Requirement |  | Exposed API |
| The system check if the customer signed up before if he did NOT\_ACCEPTABLE statue will be returned if he did not signed up before OK statues will be returned |  | 1- /Customer/SignUp  This service takes customer in the body of the request and return http status (Ok or NOT\_ACCEPTABLE) |
| The system search for the customer how signed in with his email and password |  | 2-/Customer/Signin This service takes customer in the body of the request and returns Customer object of the Customer who signed in |
| The system search for service with same name which customer entered. |  | 3-/Customer/SearchService This service takes customer in the body of the request and take string of the service name in the http url and returns http status OK the services is exist or NOT\_FOUND if the service dose not exist |
| The system search for the transaction with the same id that the use entered and when it find it he make refund request process |  | 4-/Customer/RefundRequest This service takes customer in the body of the request and take the id of the transaction which the customer wants to refund in the http url and return http status OK if the transactions exist or NO\_FOUND if it doesn’t exist |
| The system take customer data and the amount of money he wants to add and check if his credit card has enough money to be added to his wallet if his credit card has enough money the process will be completed if not the process will be canceled |  | 5-/Customer/AddtoWallet This service takes customer in the body of the request and takes the amount of money the user want to add in the http url and returns http status (Ok or NOT\_ACCEPTABLE) |
| The system search for the service according to the customer choice and check its discount |  | 6-/Customer/checkServiceDiscount This service takes customer in the body of the request and takes the  Number of the service he wants to check in the http url the customer has to put a number between 1 and 3  And the service returns the discount amount  1-MobileRecharge  2-InternetPayment  3-LandLine |
| The system takes customer data and his choice and make a payment transaction to him according to his choices and check if the customer credit card or wallet has enough money to this transaction if not the system will cancel the transaction and return null |  | 7-/Customer/makeService This service takes customer in the body of the request and takes the  ServiceChoice and receipChoice and serviceProviderChoice and paymentChoice and the amount of money the customer wants to pay  And the service returns a payment transaction that the user had made  The customer has to chose a number between 1 and 4 to select his servers as  1-MobileRecharge (if he chose this service the service provider choice has to be between 1 and 4) as  1-we  2-orange  3-etisalat  4-vodafone    2-InternetPayment(if he chose this service the service provider choice has to be between 1 and 4) as  1-we  2-orange  3-etisalat  4-vodafone  3-LandLine(if he chose this service the receipt choice has to be between 1 and 2) as  1-Monthly receipt  2-Quarter receipt  4-Donation(if he chose this service the receipt choice has to be between 1 and 3) as  1-Schools  2-NGO  3-Cancer Hospital  The Payment choice has to be a number between 1 and 3 as  1-Credit card  2-Wallet  3-Cash |
| The system get all payment transactions made by customers and return it as array list |  | 8-/Admin/ListPaymentTransactions This request returns an array list of all payment transaction made by customers |
| The system get all refund request transactions made by customers and return it as array list |  | 9-/Admin/ListRefundTransactions This request returns an array list of all refund requests transactions made by customers |
| The system get all add to wallet transactions made by customers and return it as array list |  | 10-/Admin/ListAddtoWalletTransactions This request returns an array list of all add to wallet transactions made by customers |
| The system takes all admin choices and make a discount. |  | 11-/Admin/MakeDiscount This request take discount choice and service choice and discount amount in the http url this request returns http status OK  The admin has to enter the number of discount choice 1 or 2 as  1-Over all discount  2-Sepcific discount (if he choice this discount he has to choice service number between 1 and 3 as)  1-MobileRecharge  2-InternetPayment  3-LandLine |
| The system take the Admin choices and search for the transaction with the same id if the transaction exist the process will be completed according to Admin choice if not it will return null |  | 12-/Admin/HandleRefund This request take transaction id and the state of the transaction that the Admin wants to give to it in the http url and returns http status OK if the process completer or NOT\_FUND if the transaction not found |

# Github repository link

https://github.com/ZeyadMoAli/Software\_Project