Cairo University Faculty of Computers and Artificial Intelligence



**Software design specification document**

**2023**

**Project Team**

|  |  |  |
| --- | --- | --- |
| **ID** | **Name** | **Email** |
| 20210430 | NourEldin Ahmed Hussien | nour.sehs.3@gmail.com |
| 20210430 | Mohanad Hisham Al-Tahawy |  |
| 20210049 | Ahmed Yehia Abdel-Moaty |  |
| 20210301 | Karim Mohammed Abd-Elmoain |  |

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)

# Class diagram design

**Several blue folders with text

Description automatically generated**

# Class diagram Explanation

* **Class Manager is responsible for taking input from the user, showing menus, and connecting packages.**
* **Database interface has the methods required when using any type of database.**
* **InMemoryDataBase class is an implementation for the Database interface.**
* **Account abstract class puts the structure required for any accounts that are present or may be added to the system.**
* **Every account must have ServiceProvder because in this system there are two types of service providers wallet and bank service providers, and for every account it has different options depending on its service provider and every service provider type requires different way of connecting to its API, so strategy pattern is used here that way for every service provider type present or may be added a different implementation for transfer, inquire and verify will be created depending on the behavior required by the service provider.**
* **To add the feature of letting the user pay a bill there must be a billing provider, the bill provider is represented in the BillProvider class which has Account in its attributes because the billing provider must have an account to transfer the payment to, it also has an address which is the URL to its API to tell its servers that some bill was paid successfully. The BillProvider class is an abstract class because there are different types of billing providers present and may be added to the system and each one requires different data to distinguish between bills and has a different way of connecting to its API.**
* **Bill class represents bills that must have a BillProvider for which the bill should be paid, an amount that is decided by the BillProvider, account which wants to pay the bill, and deductionData which is the data required from the BillProvider to distinguish between bills.**

# 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.**



# GitHub repository link

* <https://github.com/TBK-2002/Clone_Instapay>