Faculty of Computer and Artificial Intelligence

Cairo University

CS213-OOP

First Semester 2022-2023

**Assignment #2 Report**

**S19/20**

**TA: Aya Ayman**

**Team:**

* **Salah Eddin Mohamed Salah - 20210187**
* **Mohamed Essam - 20210346**
* **Abdelrhman Gamal Ali - 20210223**

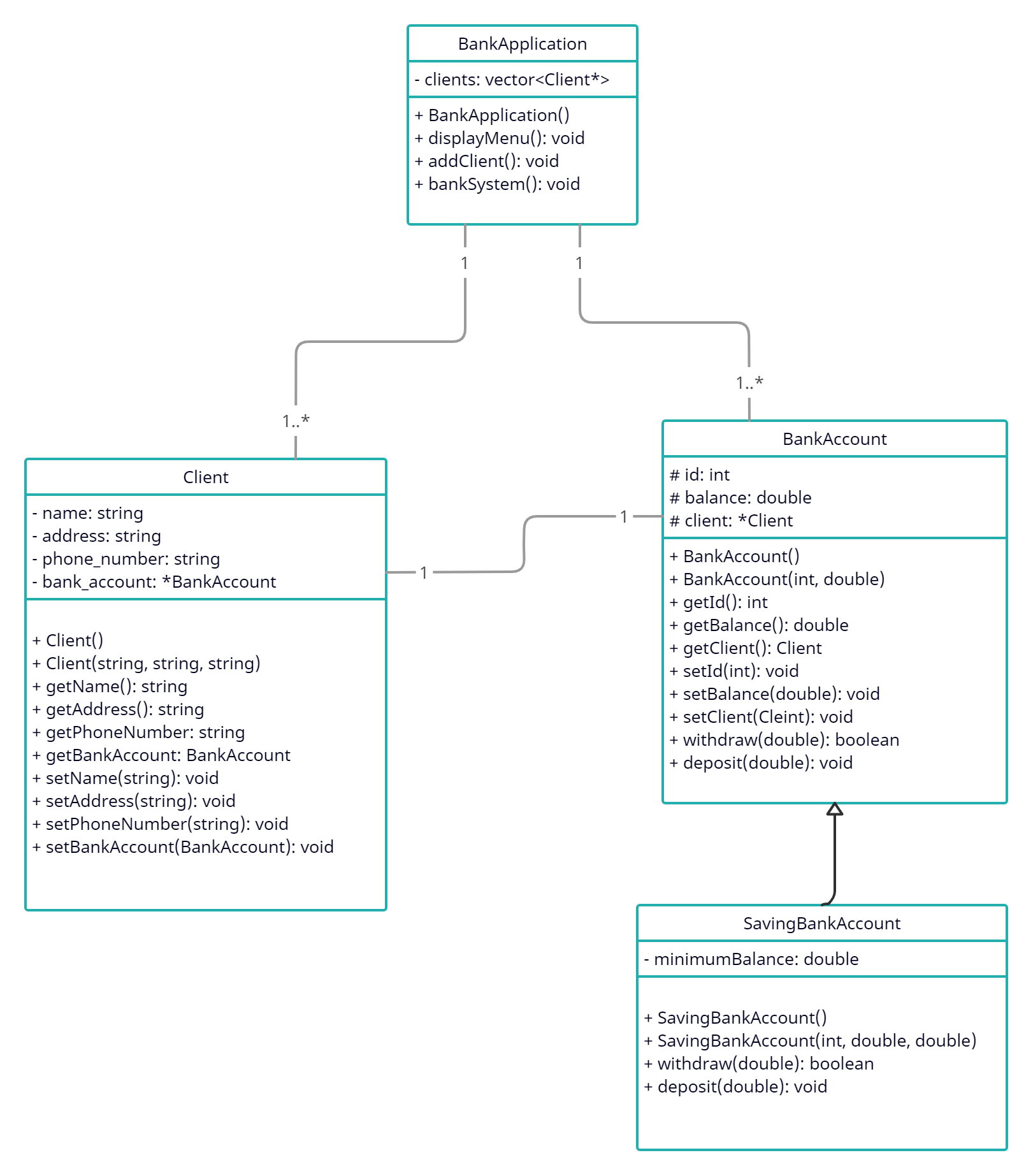
Task 1: Big Real

UML



Task 2: Bank System

UML



Work break-down table

|  |  |
| --- | --- |
| Salah Eddin Mohamed | Bank Account and Saving Bank Account Classes |
| Mohamed Essam | Class UML and Client Class |
| Abdelrahman Gamal | Banking Application Class |

Task 3: Sheet 2

|  |  |
| --- | --- |
| Salah Eddin Mohamed | Problems 0 and 3 |
| Abdelrahman Gamal | Problems 1 and 4 |
| Mohamed Essam | Problems 2 and 5 |

Bonus Task 1: Code analyser

Report from PVS Code Analyser (Bank System):

V818 It is more efficient to use an initialization list 'name(name), address(address), phoneNumber(phoneNumber)' rather than an assignment operator. Client.cpp(7), Client.cpp(8), Client.cpp(9), Client.cpp(10)

V599 The virtual destructor is not present, although the 'BankAccount' class contains virtual functions. Client.cpp(44)

V1010 Unchecked tainted data is used in index: 'id'. BankApplication.cpp(42), BankApplication.cpp(41)

V1010 Unchecked tainted data is used in index: 'id'. BankApplication.cpp(57), BankApplication.cpp(56)

Report from PVS Code Analyser (BigReal):

V832 It's better to use '= default;' syntax instead of empty constructor body. BigDecimalIntClass.h(45)

V813 Decreased performance. The 'input' argument should probably be rendered as a constant reference. BigDecimalIntClass.cpp(20)

V813 Decreased performance. The 'num' argument should probably be rendered as a constant reference. BigDecimalIntClass.cpp(26)

V820 The 'num' variable is not used after copying. Copying can be replaced with move/swap for optimization. BigDecimalIntClass.cpp(29)

V801 Decreased performance. It is better to redefine the first function argument as a reference. Consider replacing 'const .. anotherDec' with 'const .. &anotherDec'. BigDecimalIntClass.cpp(100)

V803 Decreased performance. In case 'it1' is iterator it's more effective to use prefix form of increment. Replace iterator++ with ++iterator. BigDecimalIntClass.cpp(131)

V803 Decreased performance. In case 'it2' is iterator it's more effective to use prefix form of increment. Replace iterator++ with ++iterator. BigDecimalIntClass.cpp(132)

V813 Decreased performance. The 'num2' argument should probably be rendered as a constant reference. BigDecimalIntClass.cpp(117)

V813 Decreased performance. The 'num2' argument should probably be rendered as a constant reference. BigDecimalIntClass.cpp(141)

V808 'd' object of 'deque' type was created but was not utilized. BigDecimalIntClass.cpp(193)

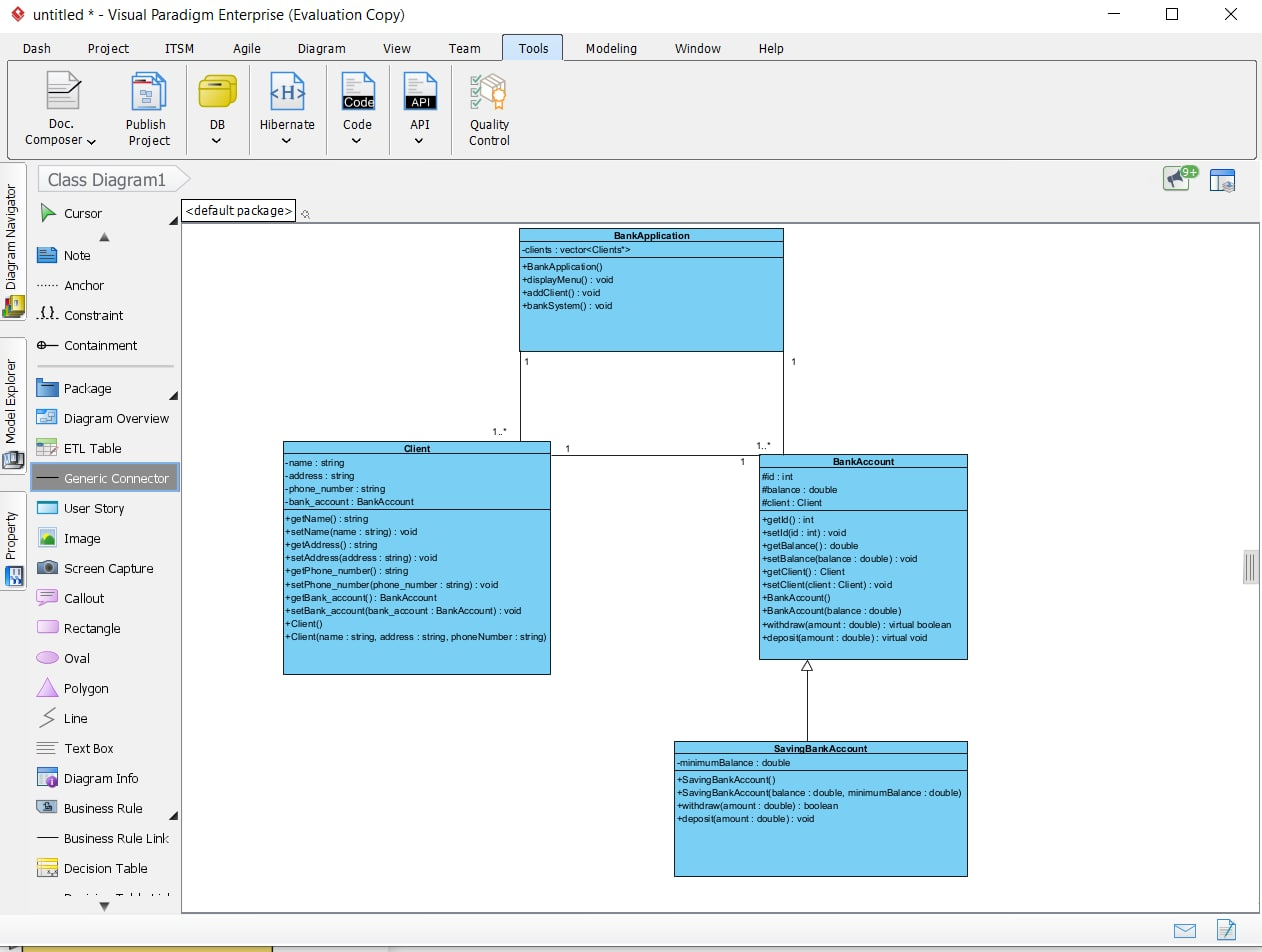
V820 The 'strmin' variable is not used after copying. Copying can be replaced with move/swap for optimization. BigDecimalIntClass.cpp(209)

V820 The 'res' variable is not used after copying. Copying can be replaced with move/swap for optimization. BigDecimalIntClass.cpp(253)

V813 Decreased performance. The 'num' argument should probably be rendered as a constant reference. BigDecimalIntClass.cpp(272)

V730 Not all members of a class are initialized inside the constructor. Consider inspecting: signNumber. BigDecimalIntClass.h(45)

Bonus Task 2: Visual Paradigm



We used it in Task 2 to start our code

It is easy to use but it produces bad unreadable code.