Projektowanie oprogramowania – SPRAWOZDANIE

Wydział Inżynierii Metali i Informatyki Przemysłowej, AGH

BANK

Wykonali:

- ***Colleague 1
- ***Colleague 2
- Szymon Rogowski

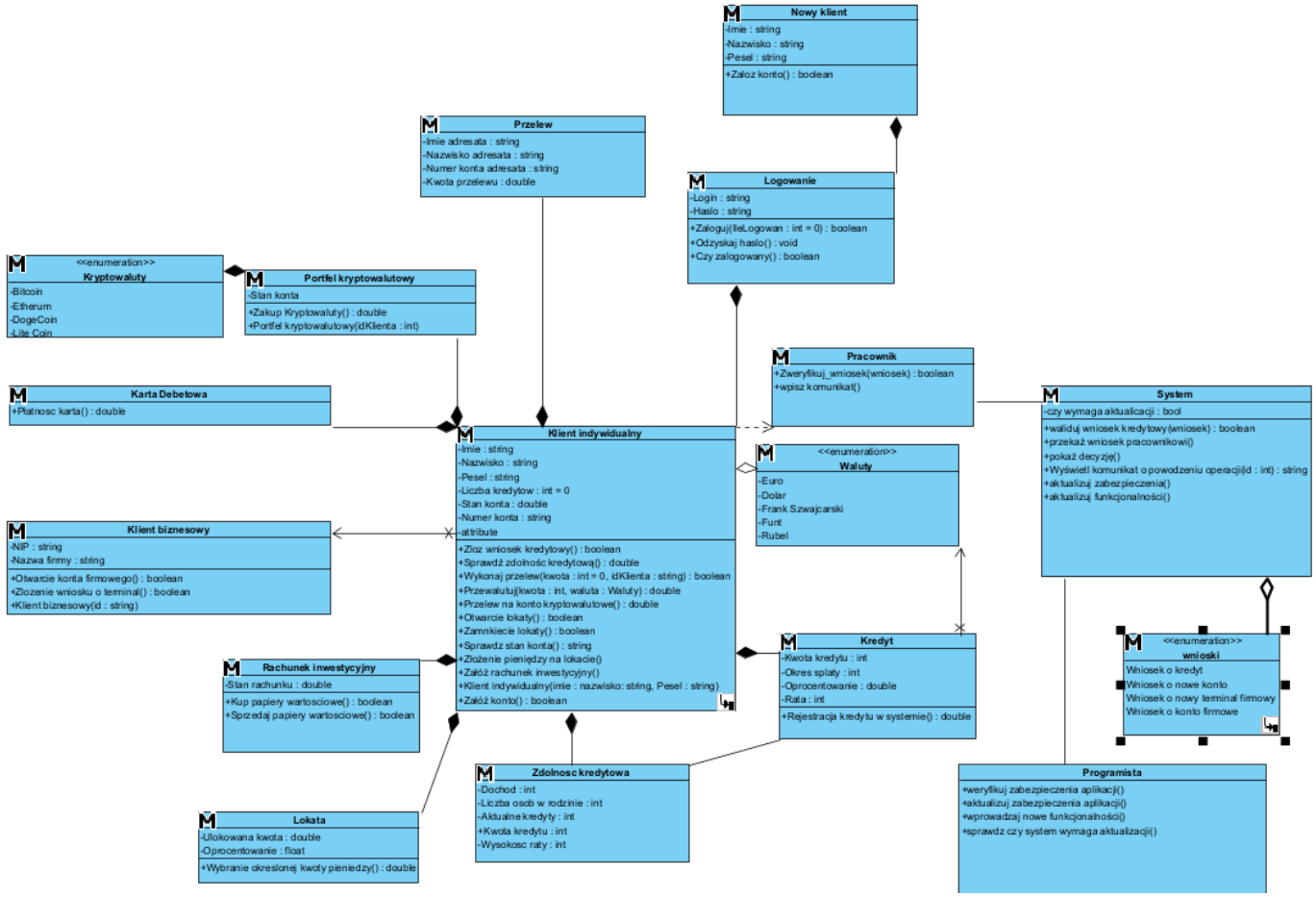
---

1. Opis wykonanego projektu:

Sam projekt jest to pomysł na mobilną aplikacje bankową. W pierwotnym założeniu posiada ona następujące właściwości:

I. Wszelakich podstawowych funkcjonalności takich jak: założenie nowego konta, zalogowanie i wylogowanie z aplikacji, wykonanie przelewu – przygotowanie kwoty do wysyłki, sprawdzenie stanu konta.
II. Bardziej złożonych czynności: otwarcia i zamknięcia lokaty z oprocentowaniem, złożenie pieniędzy na lokacie, złożenie wniosku o kredyt – sprawdzenie i wyliczenie zdolności kredytowej dla danego przychodu.
III. Ale także umożliwia rozszerzenia na klienta biznesowego mogącego: otwarcia konta firmowego czy złożenia wniosku o terminal
IV. Posiada również możliwość przewalutowania pieniędzy pomiędzy walutami: euro, dolar, frank szwajcarski, rubel, funt oraz złotówka.
V. Posiada zdolności otwarcia portfela kryptowalutowego umożliwiającego zakup takich walut jak: Bitcoin, Ethereum, Dodgecoin i Litecoin.
VI. Założenie rachunku inwestycyjnego polegającego na kupnie/sprzedaży papierów wartościowych – możliwego połączenia z giełdą oraz odliczenia transakcji od konta.
VII. Stworzenie enumeracji wiążących pewnych szablonów formularzy np.: wniosek o kredyt, nowe konto, terminal firmowy, konto firmowe.
VIII. Rozszerzenia zdolności aplikacji do obsługi jej przez administratorów w postaci informatyków zdolnych do tworzenia nowych funkcjonalności.
IX. Kont pracowniczych pozwalających np. na weryfikacje wniosków.

Pierwotny schemat klas prezentuje się następująco:



2. Narzędzia projektowe użyte do wykonania projektu:

Projekt został zaplanowany przy pomocy języka **UML** (Unified Modeling Language), używanego do planowania systemów informatycznych. Całość została wykonana przy pomocy środowiska umożliwiającego inżynierie kodu oraz jego generowanie zwanego **Visual Paradigm**.

3. Narzędzia użyte do implementacji:

Cała implementacja projektu opierała się o język programowania **Python**. Interfejs graficzny (GUI) został wykonany przy pomocy biblioteki wbudowanej w Pythonie o nazwie **Tkinter**. Kod został połączony z relacyjną bazą danych **MySQL**. Skorzystaliśmy z programu umożliwiającego graficzną konstrukcje bazy danych – **PHP MyAdmin**.

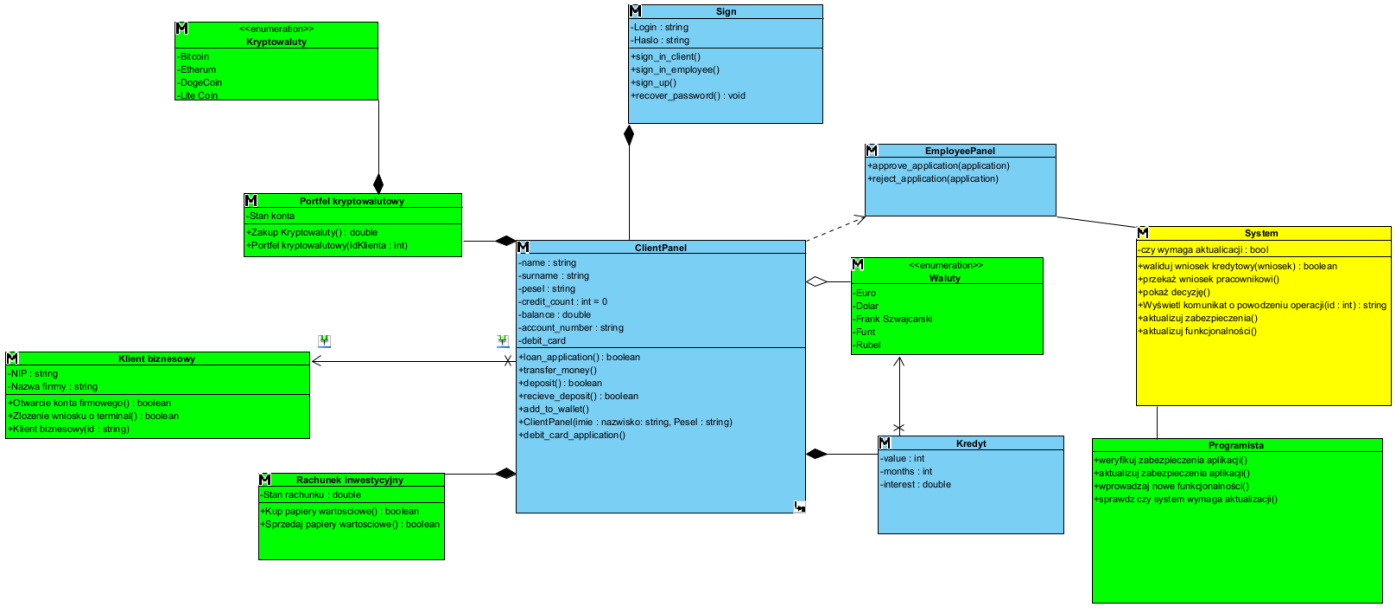
4. Diagram klas po wybraniu fragmentów do implementacji:

Z uwagi na ogrom prac związanych z samą złożonością projektu, wybraliśmy zaledwie część do implementacji wg zaleceń Pana Doktora. Postanowiliśmy w małym stopniu pozmieniać same diagramy oraz zaznaczyć kolorami poszczególne klasy dla przejrzystości diagramu.

Następującymi kolorami zaznaczyliśmy:

- Niebieski – klasy oraz powiązane funkcjonalności które zostały zaimplementowane w kodzie.

- Żółty – klasy oraz powiązane funkcjonalności które zawarte są w postaci przykładowo samej implementacji GUI bądź struktury informatycznej bazy danych o które oparliśmy projekt.
- Zielony – klasy oraz powiązane funkcjonalności które NIE zostały zaimplementowane w kodzie.



5. Test Driven Development:

Testowanie aplikacji wymagało by wstawienia wszystkich przykładowych wartości ręcznie, co wiąże się z bardzo dużym nakładem pracy z uwagi na budowę biblioteki Tkinter.

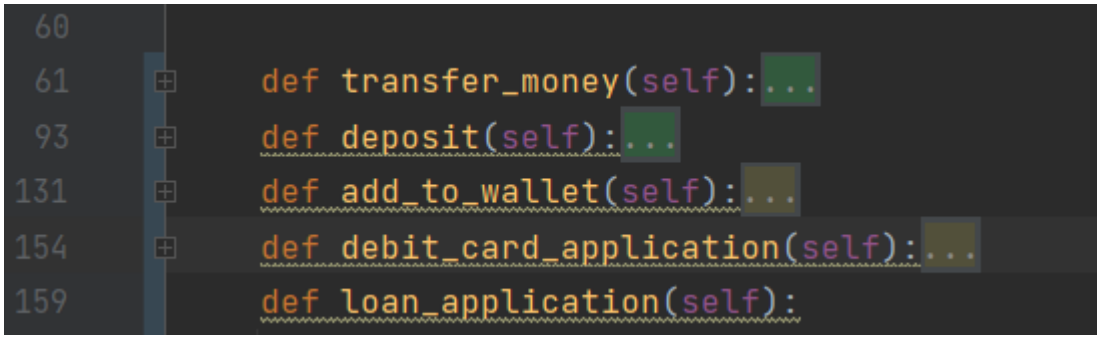
6. GRASP/**SO**LID:

Należało wybrać 2 dowolne założenia i je opisać. W naszym wypadku w wybraliśmy: S i O.

- S (single responsibility principle) –> zasada jednej odpowiedzialności:

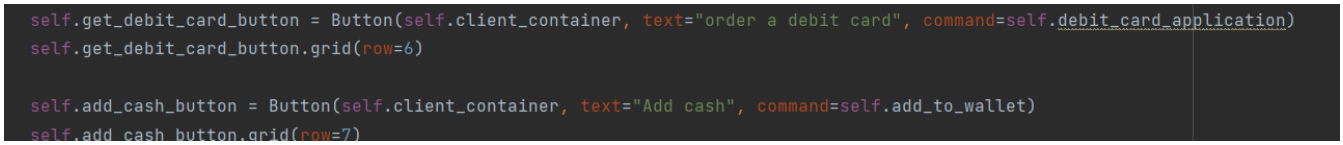
  Nigdy nie powinno być więcej niż jednego powodu istnienia klasy, bądź metody. Każda metoda w zaimplementowanych przez nas klasach np.: w klasie ClientPanel posiada dokładnie jedno zastosowanie spełniając jedną funkcje aplikacji bankowej. Metody tejże klasy prezentują się następująco:

```
+loan_application() : boolean
+calculate_
creditworthiness() : double
+transfer_money()
+deposit() : boolean
+recieve_deposit() : boolean
+check_balance() : string
+add_to_wallet()
```



- O (Open/Closed principle) –> zasada otwarte-zamknięte:

Aplikacja po ukończeniu i pełnym przetestowaniu jest otwarta na rozszerzenia a jej obecne funkcjonalności zamknięte na modyfikacje. Za przykład również posłuży klasa ClientPanel do której bez żadnych przeszkód można dodawać coraz to kolejne metody wraz z realizującym je interfejsem, które będą dodatkowe opcje klientom naszej aplikacji bankowej.



Wystarczą 2 linijki dla dodania nowego „guzika” do realizacji metody. Zarazem ukończone już metody spełniają swoją funkcjonalność dlatego dopóki nie zajdzie potrzeba radykalnego przebudowania systemy, można je uznać za zamknięte na modyfikacje.