Aplikacja do obsługi giełd kryptowalutowych

Adam Kasperowicz - 279046

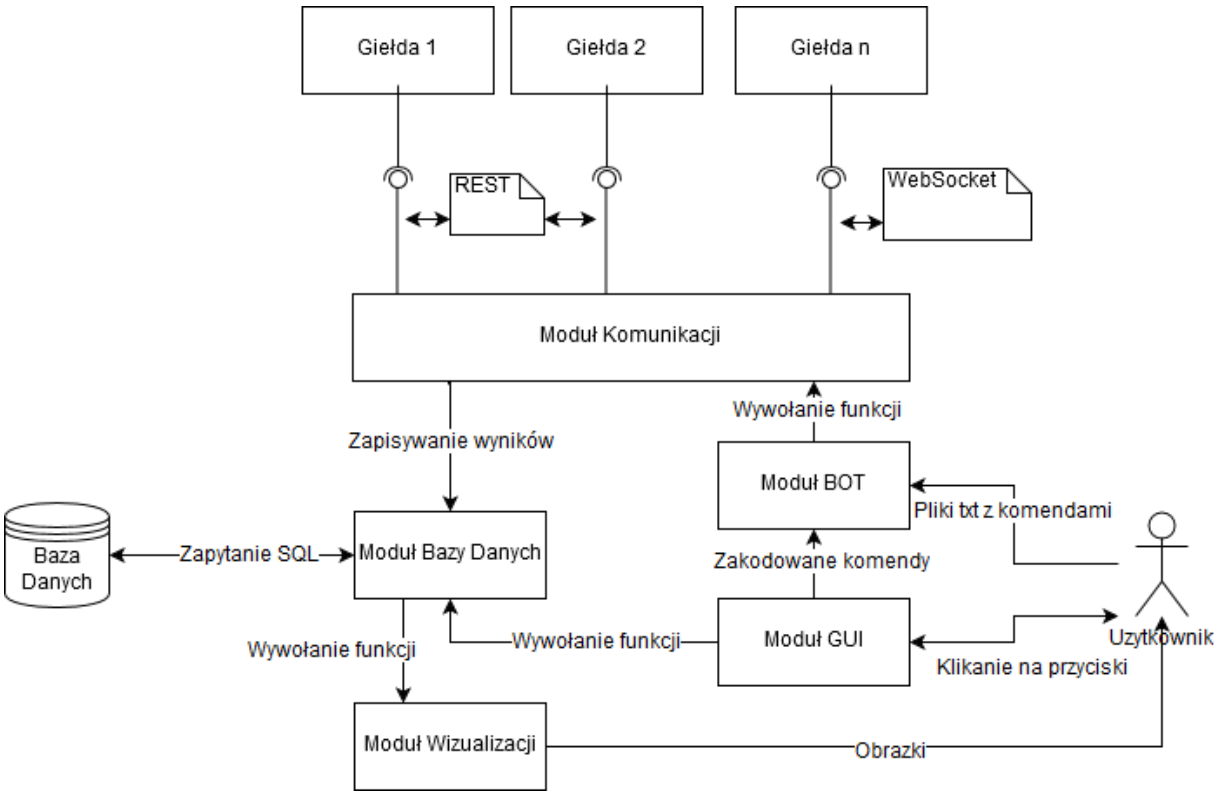
Aplikacja ma za zadanie osiągnięcie następujących celów:

- Uzyskiwanie danych z giełd bez potrzeby wchodzenia na stronę giełdy.
- Wykonywanie transkacji bez potrzeby wchodzenia na giełde.
- Zbieranie danych z giełdy w tle (np. Zapisywanie w bazie danych obecnych wartości danej pary walut co minutę przez nieokreślony czas).
- Jak naprostsze podłączanie nowych giełd do aplikacji.
- Możliwość wyświetlania danych z baz danych w postaci wykresów itp.

Aplikacja powinna posiadać następujące cechy:

- Pełne pokrycie testami (Test Driven Development wykorzystując PyUnit).
- Wykorzystywanie zawsze najbardziej efektywnego protokołu komunikacji z giełdą(jeśli giełda oferuję zarówno REST'a jak i WebSocket'a to korzystamy z drugiej opcji).
- Jak najbardziej przyjazna i jak najprostsza dla użytkownika.

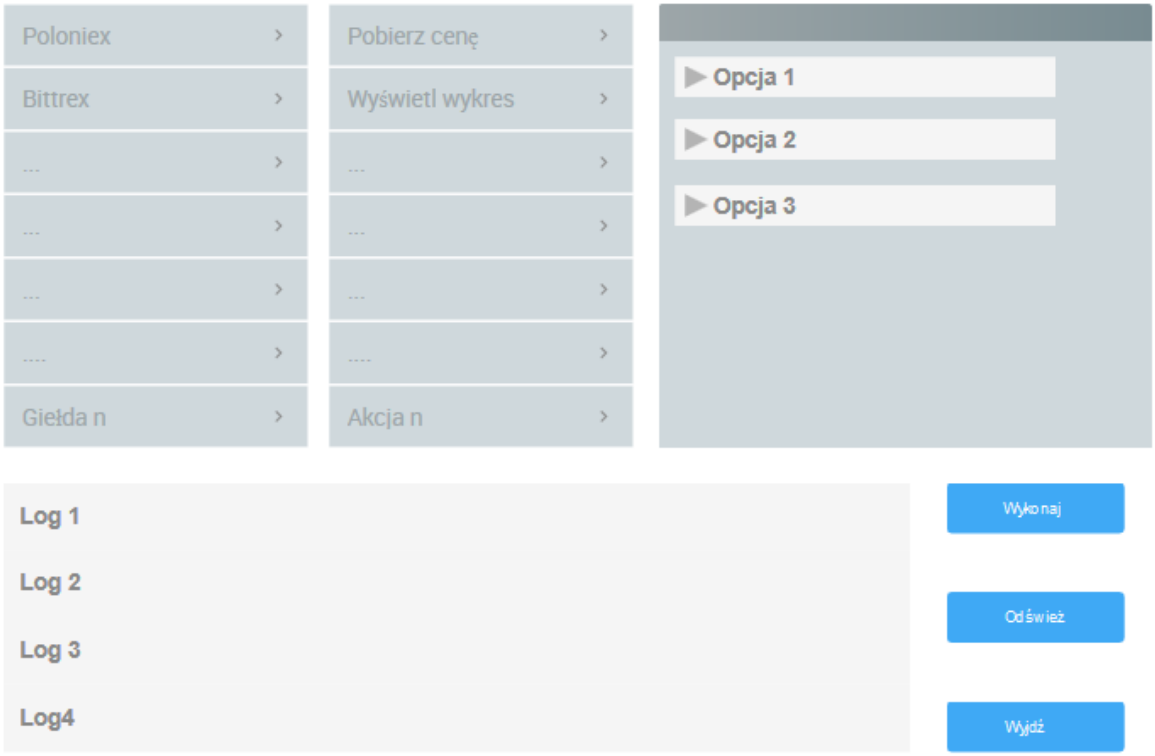
Ogólna struktura aplikacji przedstawi się następująco:



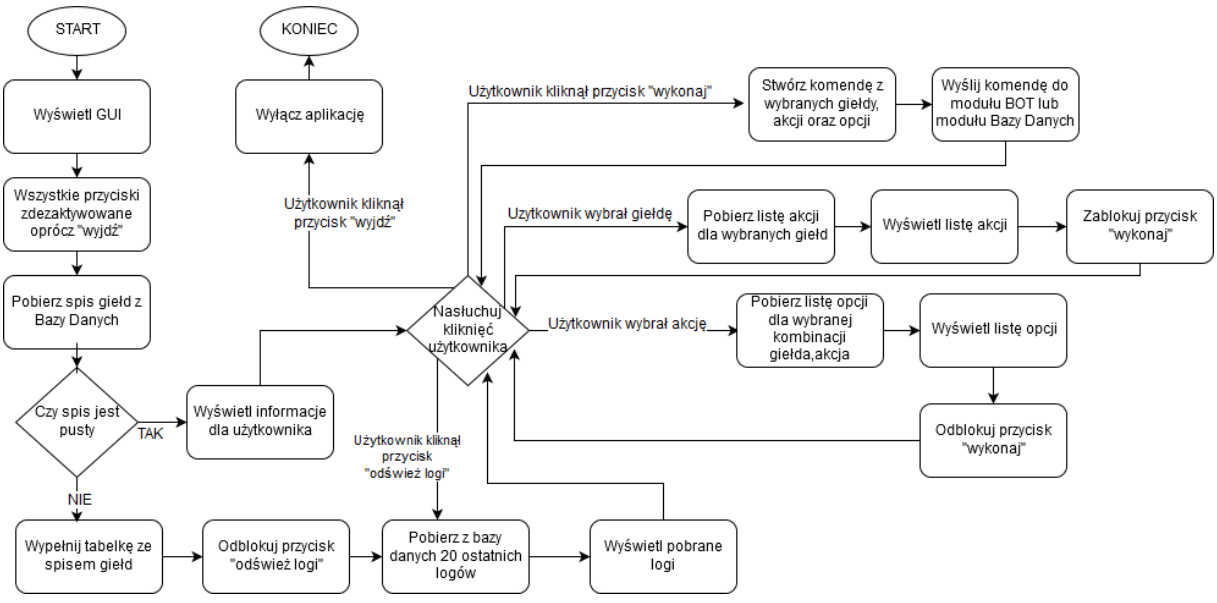
Moduł GUI:

Element pozwala na prostą i intuicyjną obsługę aplikacji. Obsługa polega na wybieraniu jednej lub wielu giełd, następnie wyboru możliwych akcji, podaniem specyfikacji jeśli takie są możliwe/wymagane i wreszcie zatwierdzeniem akcji. Interfejs zostanie zaimplementowany z wykorzystaniem biblioteki PyQt.

Pierwszy prototyp interfejsu graficznego



Algorytm działania modułu GUI



Moduł BOT:

Ten moduł ma za zadanie obsługę wysyłanych do niego komend(zwykłe String'i podążające za ustaloną konwencją). Moduł parsuje następnie otrzymane komendy i przerabia je na wywołania odpowiednich funkcji z modułu komunikacyjnego. Z początku moduł będzie musiał zająć się tylko prostymi formułami typu.

„Poloniex: getTicker" – co zostałoby przetłumaczone na Poloniex.do(„getTicker")

Lecz ostateczna wersja modułu przewiduje funkcjonalności typu:

- Pętle wywołujące komendy co dany interwał czasu
- Warunki pozwalające na programowanie prostych botów
- Wielowątkowość dająca możliwość działania wielu botów jednocześnie

Moduł BOT zostaje wywołany na 2 sposoby:

1. Moduł GUI zleca wykonywanie zadanych mu komend
2. Moduł BOT wczytuję pliki TXT z komendami napisanymi przez użytkownika

Opcja druga istnieje w celu pozwolenia użytkownikowi na wyjście poza ograniczenia narzucane przez Moduł GUI.

Moduł Wizualizacji:

Część programu zajmująca się tworzeniem odpowiednich wykresów dla odpowiednich danych. Na wejściu przyjmuje obiekty DTO(opisane w module Bazy Danych), odczytuję z nich informację o typie przechowywanych danych a następnie tworzy odpowiedni wykresy dla uzytkownika.

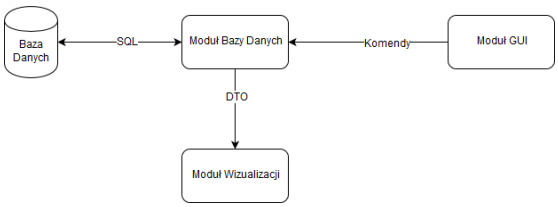
Moduł Bazy Danych:

Następny moduł jest w swojej charakterystyce bardzo podobny do modułu BOT. Moduł Bazy Danych ma za zadanie przetłumaczenie komend wysyłanych przez Moduł GUI oraz Moduł Komunikacji na odpowiednie wywołania funkcji. Tym razem jednak Moduł Bazy Danych sięga do Bazy Danych poprzez zapytania SQL oraz do Modułu Wizualizacji poprzez obiekt DTO. Baza danych wykorzystuje SQLite3.

Obiekt DTO(Data Transfer Object) jest to wzorzec projektowy potrzebny nam do przesyłania dużych ilości skomplikowanych danych między modułami. Służy on nam do zserializowania danych otrzymywanych z giełd. Dzięki temu możemy pobrane informacje łatwo zapisywać w bazie danych a następnie wizualizować.

Moduł Bazy Danych może zostać wykorzystany w 2 sposoby, które zostają przedstawione poniżej:

1. **Moduł GUI chce wyświetlić informacje z Bazy Danych**



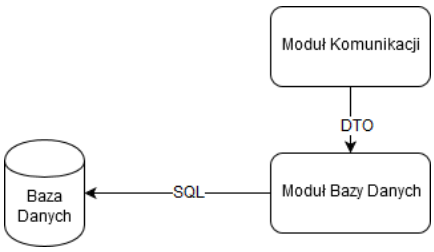
Moduł GUI wysyła komendę „Poloniex showOrderBook".

Moduł Bazy danych zamienia komendę na database.execute(„ zapytanie sql pobierające odpowiednie dane").

Pobrane dane z Bazy Danych tworzą obiekt DTO.

Obiekt ten jest wysyłany do Modułu Wizualizacji.

2. **Moduł Komunikacji chce zapisać informacje do Bazy Danych**



Moduł Komunikacji wysyła obiekt DTO zawierający dane pobrane z giełdy.

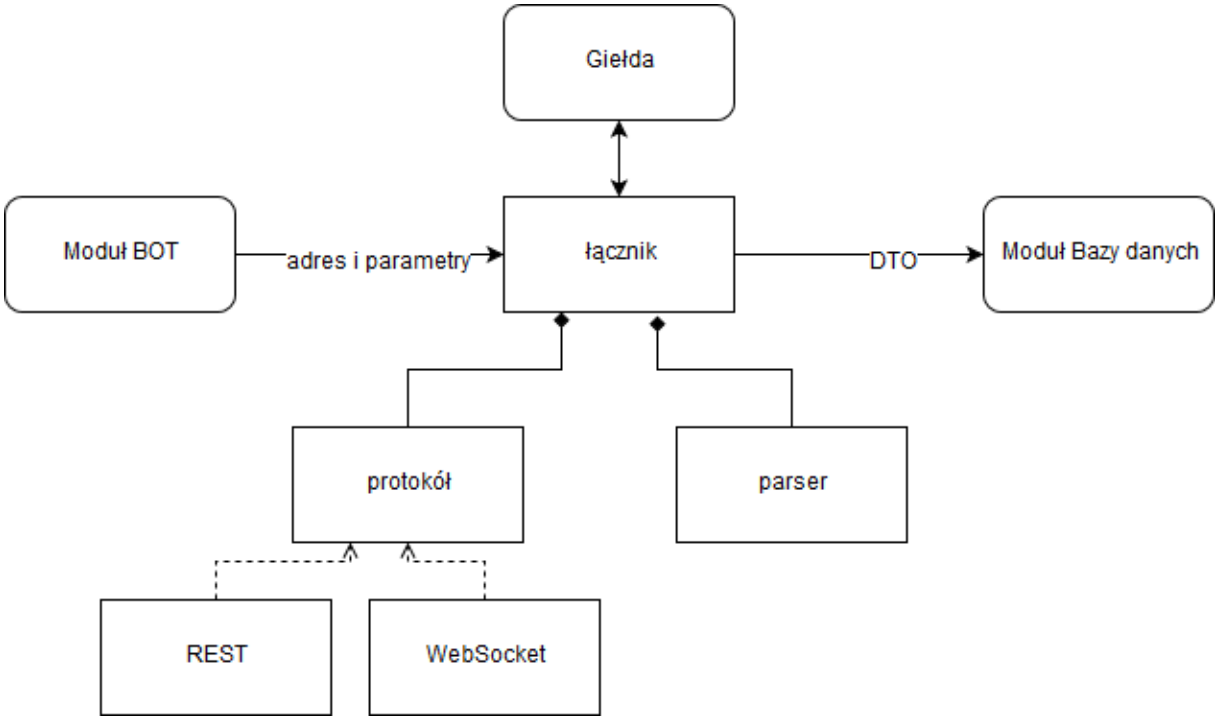
Moduł Bazy Danych dowiaduje się z obiektu DTO jakie rodzaje danych są w nim przechowywane i wykorzystuje tą informację do umieszczenia owych danych w odpowiednich tabelach.

Moduł komunikacji:

Zadaniem tego elementu jest hermetyzacja procesu komunikacji z giełdami. Bardzo ważne jest zaprojektowanie tego modułu w taki sposób by dodawanie nowej giełdy wiązało się z jak najmniejszą ilością kodu. Dlatego wyodrębniamy części zmieniające się w każdej giełdzie i przedstawiamy metodę najlepiej hermetyzujące daną część.

- Różne protokoły – API giełd wykorzystują często inne protokoły niż REST, dlatego dla każdego protokołu tworzymy klasę zajmującą się tylko danym protokołem. Klasa ta powinna przyjmować na wejście tylko adres i parametry zapytań dla danego protokołu. Dzięki temu 2 giełdy wykorzystujące ten sam protokół mogą wykorzystać dokładnie tą samą klasę.
- Różne formaty zwracanych danych – Każda giełda zwraca nam informację w trochę innym formacie. Niestety, tej charakterystyki nie da się zgeneralizować i klasa odpowiadająca za daną giełdę powinna posiadać własny parser, który otrzymane dane zapisuje do wystandaryzowanego DTO.
- Różne adresy i parametry – Istotną cechą różniącą się między giełdami są także adresy zapytań do API. Ten problem rozwiązujemy poprzez ręczne wprowadzanie tych adresów do bazy danych. Następnie podczas używania Modułu GUI adresy te są automatycznie pobierane i przesyłane do Modułu Komunikacji

Podsumowując, w celu skomunikowania się z daną giełdą tworzony jest obiekt, który w diagramie poniżej nazywa się „łącznik”. Dany łącznik tworzony jest przez klasę zarządzającą, która podejmuje pierwszy kontakt z Modułem BOT.

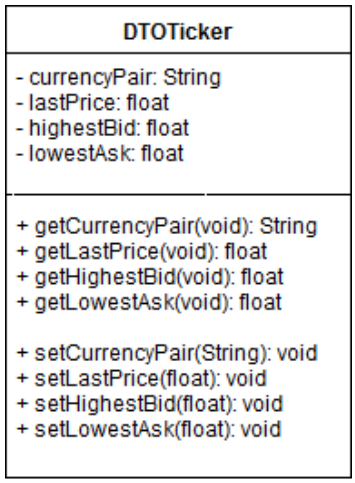


Fazy implementacji

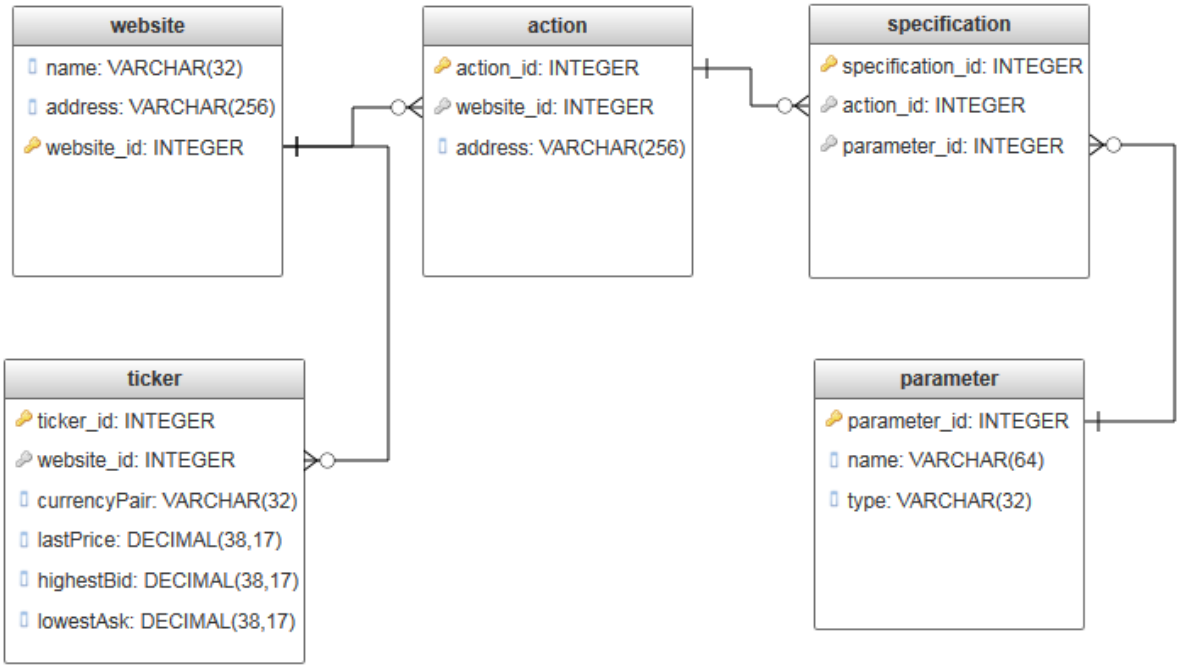
1. Zapis obecnej wartości pary walut w bazie danych dla jednej giełdy.

Pierwsza część wymagać będzie zaimplementowania rdzenia wszystkich modułów, oprócz modułu wizualizacyjnego. Pierwszą giełdą będzie „Poloniex” wykorzystujący protokół WAMP będący podprotokołem WebSocket’a . W celu zdobycia informacji o aktualnej cenie zasubskrybujemy „ticker” i będziemy wyszukiwać pary walut podanych w parametrze „market”.

Dane pobrane zapiszemy do obiektu DTO przedstawionego obok.



Gotowy obiekt DTO prześlemy do Modułu Bazy Danych i tam umieścimy go do samej Bazy Danych, która będzie wyglądać następująco.



2. Możliwość zapisywania obecnej wartości pary walut przez cały dzień i wizualizacja tej wartości w postaci wykresu.
3. Dołączenie dwóch kolejnych giełd.
4. Zaimplementowanie wszystkich ważniejszych danych możliwych do pobrania z giełdy
5. Możliwość zdalnego handlowania na giełdzie.
6. Prosty bot przyjmujący strategię średnich kroczących.