

**Drozdowski Hubert, 238328**  
**Mróz Jan, 242440**  
**Grupa Wtorek 13:15**

## **Inżynieria Systemów Baz Danych**

# **CryptoFever**

*Prowadzący:*  
*mgr Hanna Mazur*

<b>Użytkownik</b>	<b>Login</b>	<b>Hasło</b>
Gość	-	-
Inwestor	inwestor	^inwestor2020\$
Inwestor	richman4	testpassw
Administrator	admin	^/admin/secret!

**Wrocław, 14.01.2020 r.**

*Wszystkie tabele oraz rysunki w tym dokumencie są autorskie.*

# Spis treści

<b>1.</b>	<b>TEMAT, CEL, ZAKRES I UŻYTKOWNICY .....</b>	<b>3</b>
1.1	TEMAT .....	3
1.2	CEL PROJEKTU .....	3
1.3	ZAKRES PRZEDSIĘWZIĘCIA .....	3
1.4	UŻYTKOWNICY SYSTEMU .....	3
<b>2.</b>	<b>OPIS WYMAGAŃ DLA PROJEKTOWANEGO SYSTEMU .....</b>	<b>4</b>
2.1	WYMAGANIA FUNKCYJNALNE .....	4
2.2	WYMAGANIA NIEFUNKCYJNALNE .....	4
<b>3.</b>	<b>LISTA TRANSAKCYJ DLA POSZCZEGÓLNYCH GRUP UŻYTKOWNIKÓW .....</b>	<b>5</b>
3.1	GOŚĆ .....	5
3.2	INWESTOR .....	5
3.3	ADMINISTRATOR .....	6
<b>4.</b>	<b>DOKUMENTACJA ZAIMPLEMENTOWANEJ BAZY DANYCH .....</b>	<b>8</b>
4.1	DEFINICJE PREDYKATOWE ENCJI .....	8
4.2	DEFINICJE PREDYKATOWE ZWIĄZKÓW .....	8
4.3	DIAGRAM ZWIĄZKÓW ENCJI .....	10
4.4	DEFINICJE SCHEMATÓW RELACJI I PRZYKŁADOWE DANE .....	11
<b>5.</b>	<b>NARZĘDZIA IMPLEMENTACYJNE .....</b>	<b>15</b>
<b>6.</b>	<b>OPIS APLIKACJI BAZODANOWEJ .....</b>	<b>16</b>
6.1	TYTUŁ .....	16
6.2	PRZEZNACZENIE .....	16
6.3	UŻYTKOWNICY .....	16
6.4	ZAKRES .....	16
6.5	SPECYFIKACJA WYMAGAŃ FUNKCYJNALNYCH .....	16
6.6	SPECYFIKACJA WYMAGAŃ NIEFUNKCYJNALNYCH .....	18
6.7	MAPA WYWOŁAŃ FORMULARZY I RAPORTÓW .....	19
6.8	PRZYKŁADOWE FORMULARZE .....	20
6.9	PRZYKŁADOWE RAPORTY .....	23
<b>7.</b>	<b>ZAWARTOŚĆ NOŚNIKA Z SYSTEMEM .....</b>	<b>25</b>
<b>8.</b>	<b>SPOSÓB INSTALACJI / URUCHOMIENIA SYSTEMU .....</b>	<b>26</b>
<b>9.</b>	<b>TESTOWANIE APLIKACJI .....</b>	<b>28</b>
<b>10.</b>	<b>ZAKOŃCZENIE .....</b>	<b>29</b>
<b>11.</b>	<b>ŹRÓDŁA .....</b>	<b>30</b>

# 1. Temat, cel, zakres przedsięwzięcia, użytkownicy systemu.

## 1.1 Temat.

Monitorowanie rynku i społeczności kryptowalut.

## 1.2 Cel projektu.

Automatyzacja procesu obserwacji kursu i statystyk kryptowalut przy pomocy aplikacji webowej. Rozwiązanie ma za zadanie wspomóc przyszłych inwestorów w podejmowaniu decyzji rynkowych na podstawie dodanych przez nich reguł, za pomocą odpowiednich informacji mailowych.

## 1.3 Zakres przedsięwzięcia:

Stworzenie aplikacji umożliwiającej bieżące analizowanie sytuacji na giełdzie kryptowalut BitBay, zapewniającej wysyłanie powiadomień do inwestorów o interesujących ich sytuacjach na rynku. System umożliwiałby definiowanie warunków wysyłania powiadomień oraz uwzględniałby zarówno ceny kryptowalut, jak i dane statystyczne Google Trends. Zakres obejmuje również utworzenie dokumentacji opisującej sposób działania oraz instalację systemu.

## 1.4 Użytkownicy systemu:

- **Gość** - osoba, która nie posiada konta w aplikacji, po zarejestrowaniu staje się Inwestorem.
- **Inwestor** - osoba, posiadająca swoje konto w aplikacji, posiada możliwość obserwacji statystyk i kursu kryptowalut. Może dodawać swoje własne reguły dotyczące kryptowalut, które są następnie przetwarzane przez aplikację, ma również wgląd w historię egzekucji w/w reguł.
- **Administrator** - osoba odpowiedzialna za system, posiada konto z najwyższymi uprawnieniami, ma dostęp do panelu administratora.

## 2. Opis wymagań dla projektowanego systemu.

### 2.1 Wymagania funkcjonalne:

- Obsługa giełdy kryptowalut BitBay API – odczyt cen kryptowalut.
- Automatyczna analiza rynku – analiza procentowej zmiany ceny kryptowalut w czasie na podstawie porównania aktualnej ceny z ceną sprzed godziny, jednego dnia, trzech dni, siedmiu dni oraz trzydziestu dni.
- Automatyczna analiza społeczności – analiza procentowej zmiany popularności wyszukiwania danych o kryptowalucie w wyszukiwarce Google. Porównania wykonywane są tak samo jak przy analizie cen kryptowalut.
- Zapis danych historycznych – zapis co minutę aktualnej ceny kryptowalut oraz popularności wyszukiwania kryptowaluty, zapis egzekwowanych reguł inwestorów.
- Możliwość dodawania reguł oraz grup reguł przez Inwestorów określających interesujące ich warunki na giełdzie kryptowalut.
- Egzekwowanie reguł Inwestorów – wysłanie odpowiedniej informacji mailowej do Inwestora w przypadku gdy wszystkie reguły z danej grupy reguł zostały spełnione.

#### Gość:

- Możliwość rejestracji jako Inwestor;

#### Inwestor:

- Możliwość logowania;
- Możliwość wglądu w aktualne wartości cen oraz statystyk społecznościowych kryptowalut.
- Możliwość wyświetlania, dodawania, edytowania i usuwania swoich grup reguł.
- Możliwość wyświetlania, dodawania, edytowania i usuwania reguł we własnej grupie reguł.
- Możliwość wglądu w historię egzekucji reguł.
- Możliwość wglądu w historię statystyk dotyczących kryptowalut.

#### Admin:

- Możliwość logowania.;
- Dostęp do panelu administratora.
- Możliwość wyświetlania, dodawania, edytowania, usuwania kryptowalut;
- Możliwość edycji danych użytkowników;

### 2.2 Wymagania нефunkcjonalne:

- Czas reakcji systemu poniżej 1s.
- Bezbłędne działanie wszystkich usług w 99.9% czasu pracy.
- Powiadomienia do inwestorów są wysyłane w ciągu minuty.

### 3. Lista transakcji dla poszczególnych grup użytkowników.

#### 3.1 Gość:

##### **TRA/001 Utworzenie konta inwestora.**

**Opis:** Zadaniem transakcji jest pobranie od gościa loginu, adresu mailowego oraz hasła i utworzenie konta inwestora w bazie danych.

#### 3.2 Inwestor:

##### **TRA/002 Przeglądanie danych kryptowaluty.**

**Opis:** Zadaniem transakcji jest wyszukanie oraz wypisanie danych o kryptowalucie.

##### **TRA/003 Utworzenie grupy reguł.**

**Opis:** Zadaniem transakcji jest pobranie danych od inwestora (nazwa grupy reguł) i utworzenie pustej grupy reguł. Utworzona grupa reguł jest automatycznie przypisywana do inwestora wykonującego transakcję.

##### **TRA/004 Przeglądanie grup reguł.**

**Opis:** Zadaniem transakcji jest wyszukanie grupy reguł i wypisanie jej danych. Transakcję może wykonać każdy inwestor, ale tylko do grup reguł utworzonych przez tego inwestora.

##### **TRA/005 Zmiana danych grupy reguł.**

**Opis:** Zadaniem transakcji jest wyszukanie grupy reguł i zmiana jej danych. Transakcję może wykonać każdy inwestor, ale tylko do grup utworzonych przez tego inwestora.

##### **TRA/006 Usunięcie grupy reguł.**

**Opis:** Zadaniem transakcji jest wyszukanie grupy reguł i jej usunięcie. Usunięcie grupy reguł usuwa także dane powiązane z daną grupą reguł, czyli całą historię egzekwowania reguł oraz wszystkie reguły znajdujące się w tej grupie reguł. Transakcję może wykonać każdy inwestor, ale tylko na grupie reguł utworzonych przez tego inwestora.

##### **TRA/007 Utworzenie reguły**

**Opis:** Zadaniem transakcji jest pobranie danych od inwestora (typ reguły, wartość) i utworzenie reguły, przypisując ją do konkretnej grupy reguł. Transakcję może wykonać każdy inwestor.

##### **TRA/008 Przeglądanie reguł.**

**Opis:** Zadaniem transakcji jest wyszukanie reguły i wypisanie jej danych. Transakcję może wykonać każdy inwestor, ale tylko dla swoich reguł.

#### **TRA/009 Zmiana reguły.**

**Opis:** Zadaniem transakcji jest pobranie danych od inwestora, wyszukanie reguły i zmiana jej danych. Transakcję może wykonać każdy inwestor, ale tylko do reguł utworzonych przez tego inwestora.

#### **TRA/010 Usunięcie reguły.**

**Opis:** Zadaniem transakcji jest wyszukanie reguły i usunięcie jej z bazy danych. Transakcję może wykonać każdy inwestor, ale tylko do reguł utworzonych przez tego inwestora.

#### **TRA/011 Przeglądanie danych historycznych dot. egzekucji grup reguł.**

**Opis:** Zadaniem transakcji jest wyszukanie i wypisanie danych o danych historycznych egzekucji grupy reguł inwestora (t.j. data, kurs, typ transakcji). Operację może wykonać tylko inwestor, który jest właścicielem danej grupy reguł.

#### **TRA/012 Przeglądanie danych historycznych kryptowalut.**

**Opis:** Zadaniem transakcji jest wyszukanie i wypisanie historycznych danych dot. kryptowaluty (t.j. data, statystyki, kurs, współczynnik analizy rynkowej, współczynnik analizy społeczności). Operację może wykonać każdy inwestor.

#### **TRA/013 Przeglądanie danych kryptowaluty.**

**Opis:** Zadaniem transakcji jest wypisanie danych o kryptowalutach (data aktualizacji, kurs, współczynnik rynkowy, społecznościowy).

### **3.3 Administrator:**

#### **TRA/014 Przeglądanie danych użytkowników.**

**Opis:** Zadaniem transakcji wypisanie danych użytkowników za wyjątkiem hasła.

#### **TRA/015 Zmiana danych użytkownika.**

**Opis:** Zadaniem transakcji jest wyszukanie i zmiana danych użytkownika.

#### **TRA/016 Usunięcie użytkownika.**

**Opis:** Zadaniem transakcji jest usunięcie użytkownika z bazy. Dane powiązane z użytkownikiem także zostaną usunięte (grupy reguł, historia egzekucji grup reguł, reguły powiązane).

**TRA/017 Utworzenie modelu kryptowaluty.**

**Opis:** Zadaniem transakcji jest pobranie danych od administratora i utworzenie nowego modelu kryptowaluty w bazie danych.

**TRA/018 Zmiana danych kryptowaluty**

**Opis:** Zadaniem transakcji jest wyszukanie oraz zmiana danych o kryptowalucie.

**TRA/019 Usunięcie modelu kryptowaluty.**

**Opis:** Zadaniem transakcji jest wyszukanie i usunięcie modelu kryptowaluty. Dane powiązane z kryptowalutą także zostaną usunięte (historia cen i statystyk społecznościowych, wszystkie grupy reguł i ich reguły oraz historie egzekucji odnoszące się do tej kryptowaluty).

## 4. Dokumentacja zaimplementowanej bazy danych.

### 4.1 Definicje predykatowe encji.

#### ENC/001

**USER**(IdU, Login, Password, IsAdmin, E-mail, CurrencyAmount)

*Semantyka encji:* Encja zawiera dane na temat użytkownika aplikacji.

#### ENC/002

**RULESET**(IdRS, Name)

*Semantyka encji:* Encja zawiera dane na temat grupy reguł.

#### ENC/003

**RULE**(IdR, Value, TypeOfRule)

*Semantyka encji:* Encja zawiera dane na temat pojedynczej reguły, która musi być spełniona aby wykonać transakcję.

#### ENC/004

**CRYPTO\_MODEL**(IdCM, ShortName, LongName)

*Semantyka encji:* Encja zawiera ogólne dane na temat modelu kryptowaluty.

#### ENC/005

**TRADE**(IdT, Date, Amount, Price)

*Semantyka encji:* Encja zawiera dane sprzedaży oraz kupna kryptowalut.

#### ENC/006

**SOCIAL\_HISTORIC**(IdSH, GtrendsTop7D, Date)

*Semantyka encji:* Encja zawiera dane na temat statystyk społecznościowych o kryptowalucie.

#### ENC/007

**MARKET\_HISTORIC**(IdMH, BidsValue, AsksValue, AvgTransactionValue, Date, Price, ResponseJSON)

*Semantyka encji:* Encja zawiera dane na temat statystyk rynkowych o kryptowalucie.

#### ENC/008

**CRYPTO\_WALLET**(IdCW, Amount, Date)

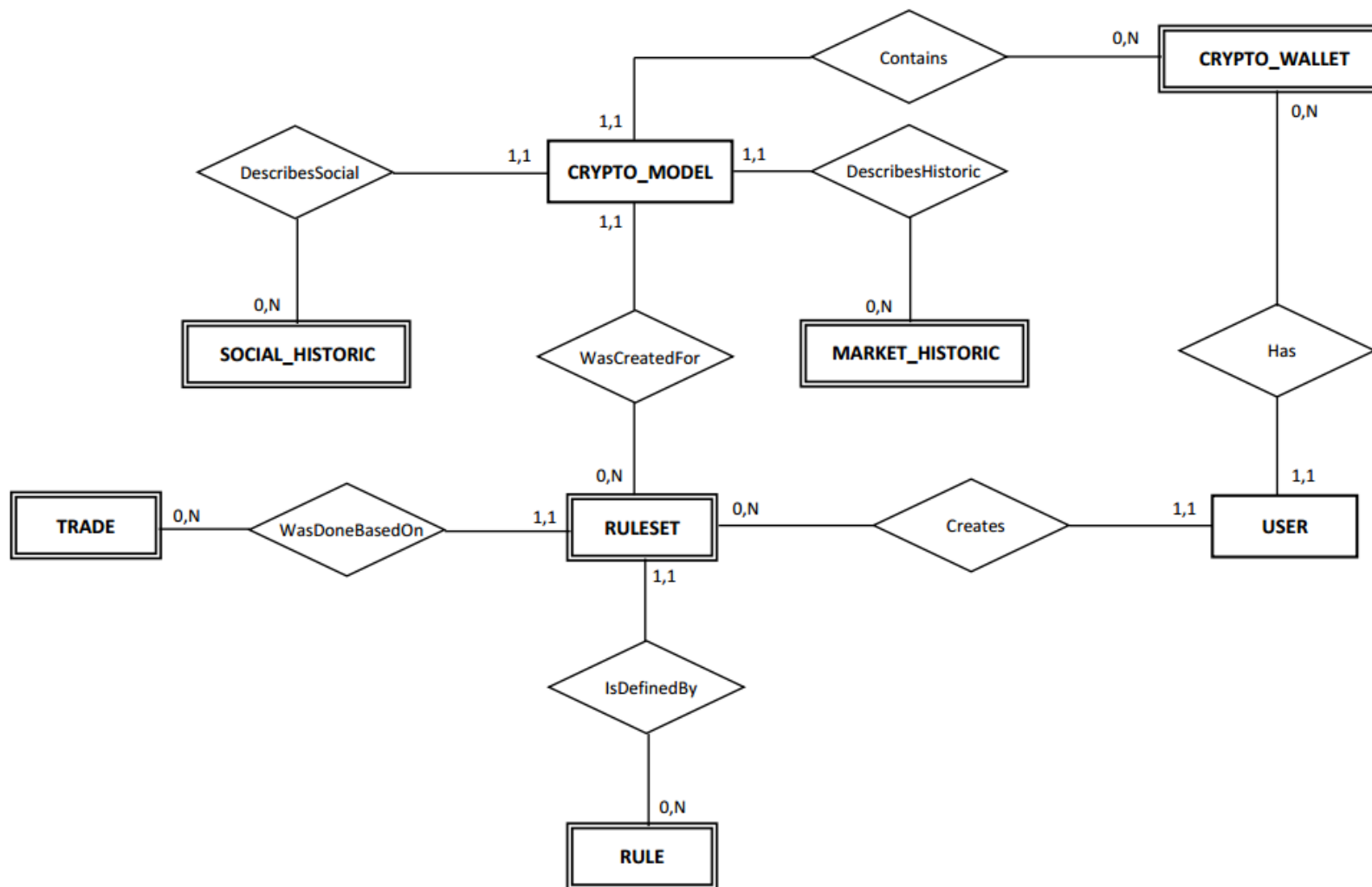
*Semantyka encji:* Encja zawiera dane na temat krypto-portfela użytkownika aplikacji, czyli ilości posiadanej kryptowaluty

### 4.2 Definicje predykatowe związków:



**ZWI/001 IsDefinedBy**(RULESET(1,1):RULE(0,N))  
**ZWI/002 WasDoneBasedOn**(TRADE(0,N):RULESET(1,1))  
**ZWI/003 Creates**(USER(1,1):RULESET(0,N))  
**ZWI/004 DescribesMarket**(MARKET\_HISTORIC(0,N):CRYPTO\_MODEL(1,1))  
**ZWI/005 DescribesSocial**(SOCIAL\_HISTORIC(0,N):CRYPTO\_MODEL(1,1))  
**ZWI/006 Has**(USER(1,1):CRYPTO\_WALLET(0,N))  
**ZWI/007 Contains**(CRYPTO\_WALLET(0,N):CRYPTO\_MODEL(1,1))  
**ZWI/008 WasCreatedFor**(RULESET(0,N):CRYPTO\_MODEL(1,1))

### 4.3 Diagram związków encji.



Rysunek 1. Diagram związków encji.

#### 4.4 Definicje schematów relacji i przykładowe dane.

##### REL/001 auth\_user/USER

**auth\_user**(id, password, last\_login, is\_superuser, username, first\_name, last\_name, email, is\_staff, is\_active, date\_joined, base\_currency)

##### Objaśnienia:

Schemat relacji *auth\_user* został zaczerpnięty z frameworka Django. (dokumentacja: <https://docs.djangoproject.com/en/3.0/ref/contrib/auth/>), ze względu na używany system autoryzacji, zostało do niego dodane pole *base\_currency*, które zawiera dane o ilości posiadanej przez użytkownika waluty podstawowej, której używa do wykonywania transakcji na giełdzie BitBay, np. polskiego złotego. Aktualnie pole *base\_currency* nie jest w aplikacji używane.

Tabela 1. Przykładowe, najważniejsze dane relacji auth\_user.

id	password	username,	email	is_superuser
1	^invest234	Investo2020	Investor2020@gmail.com	0
3	P@55w0rd	JanKowalski	jan.kowalski1995@gmail.com	1
4	zaq1@wsx	psychonauts56	bigPsycho42@gmail.com	0

##### REL/002 crypto\_ruleset/RULESET

**crypto\_ruleset**(id, name, type\_of\_ruleset, #crypto\_id, #owner\_id)

##### Objaśnienia:

Atrybut *type\_of\_ruleset* przechowuje dane typu varchar(1), czyli jedna litera.

- ‘E’ – Oznacza, że po spełnieniu wszystkich reguł zostanie wysłane powiadomienie mailowe do inwestora.
- ‘S’ – Oznacza sprzedaż kryptowaluty po spełnieniu wszystkich reguł.
- ‘B’ – Oznacza zakup kryptowaluty po spełnieniu wszystkich reguł.

Atrybuty #crypto\_id oraz #owner\_id odwołują się kolejno do relacji *crypto\_cryptomodel*, oraz *auth\_user*.

Tabela 2. Przykładowe, najważniejsze dane relacji crypto\_ruleset.

id	name	type_of_ruleset	crypto_id	owner_id
1	SellEthereum	E	2	1
3	BigSocialBitcoin	E	3	1
4	BuyEthereum	E	2	3

**REL/003** crypto\_rule/RULE**crypto\_rule**(id, value, type\_of\_rule, #rule\_set\_id)**Objasnienia:**

Atrybut *type\_of\_rule* jest typu varchar(3), gdzie każda wartość oznacza następujące reguły:

- BEL – wartość poniżej
- ABO – wartość powyżej
- CNG – zmiana wartości
- CNP – procentowa zmiana wartości
- MVP – maksymalna wartość procentowa
- MVE – maksymalna wartość
- AHS – opóźnienie wykonania akcji grupy reguł
- MBA – bot rynkowy powyżej zadanej wartości
- MBB – bot rynkowy poniżej zadanej wartości
- SBA – trend Google powyżej zadanej wartości
- SBB – trend Google poniżej zadanej wartości

Tabela 3. Przykładowe dane relacji crypto\_rule.

id	value	type_of_rule	rule_set_id
1	350.16	BEL	2
3	0.04	CNP	2
4	43.34	SBA	6

**REL/004** crypto\_cryptomodel/CRYPTO\_MODEL**crypto\_cryptomodel**(id, short\_name, long\_name)

Tabela 4. Przykładowe dane relacji crypto\_cryptomodel.

id	short_name	long_name
1	ETH	Etherium
3	BTC	Bitcoin
4	LTC	Litecoin

**REL/005** crypto\_trade/TRADE**crypto\_trade**(id, date, type\_of\_trade, amount, price, #rule\_set\_id)

Tabela 5. Przykładowe dane relacji crypto\_trade.

id	date	type_of_trade	amount	price	rule_set_id
12	21.12.2019 4:00:00	S	0.004321	54.32	2
34	22.12.2019 4:00:00	M	2.2811	2642.96	4
65	23.12.2019 4:00:00	B	0.02119	95.04	11

**REL/006** crypto\_socialhistoric/SOCIAL\_HISTORIC**crypto\_socialhistoric**(id, date, #crypto\_id, gtrends\_top\_7d)

**Objaśnienia:** Atrybut *gtrends\_top\_7d* zawiera wskaźnik popularności wyszukiwań haseł związanych z daną kryptowalutą wg. Google Trends.

Tabela 6. Przykładowe dane relacji crypto\_socialhistoric.

id	date	crypto_id	gtrends_top_7d
4556	21.12.2019 4:00:00	1	4356.21
4557	22.12.2019 4:00:00	1	4310.28
4558	23.12.2019 4:00:00	2	4198.93

**REL/007** crypto\_markethistoric/MARKET\_HISTORIC**crypto\_markethistoric**(id, price, response\_json, date, #crypto\_id, asks\_value, avg\_trans, bids\_value)

Tabela 7. Przykładowe dane relacji crypto\_markethistoric.

id	price	response_json	date	crypto_id	asks_value	avg_transaction_value	bids_value
6993	572.93	{odpowiedź od BitBay API}	21.12.2019 4:54:03	1	5000.0	600.0	2000.0
6994	572.96	{odpowiedź od BitBay API}	21.12.2019 4:55:04	1	88284.20	1200.0	3572.743
6995	572.99	{odpowiedź od BitBay API}	21.12.2019 4:56:03	2	2242.0	0.0	23122.0

**crypto\_wallet**(id, amount, #crypto\_id, #owner\_id, date)

Relacja *crypto\_cryptowallet* jest relacją, która została utworzona z myślą o możliwości rozszerzenia funkcjonalności aplikacji o automatyczne wykupowanie i sprzedawanie kryptowalut na giełdzie BitBay. Aktualnie relacja nie jest w aplikacji wykorzystywana.

Tabela 8. Przykładowe dane relacji crypto\_cryptowallet.

id	amount	crypto_id	owner_id	date
6	52.21	1	2	21.12.2019 4:54:03
90	52.21	1	2	23.12.2019 6:24:41
456	73.40	4	7	28.12.2019 5:01:10

## 5. Narzędzia implementacyjne.

**Python:** Język w którym została zaimplementowana większa część aplikacji.

**Django:** Wysokopoziomowy framework do języka Python służący do tworzenia aplikacji webowych. Użyty został do zrealizowania mechanizmu logowania i tworzenia kont użytkownika, renderowania stron internetowych, zapewniania komunikacji klient-serwer oraz dostępu do bazy danych. Django dodatkowo zapewnia dostęp do panelu administratora aplikacji, w którym można przeglądać i dowolnie modyfikować dane znajdujące się w bazie danych.

**PostgreSQL:** System relacyjnych baz danych w którym została zaimplementowana baza danych aplikacji. Django współpracuje z PostgreSQL dostarczając wygodne funkcje do obsługi bazy danych (w tym pisanie zapytań za pomocą funkcji w języku Python).

**Redis** – Pamięć bazowa używana do kolejkowania zadań.

**Nginx** – Odpowiada za świadczenie usług WWW.

**Celery** – Odpowiada za wykonywanie zadań na serwerze wg ustalonej kolejności.

**Pytest** – Framework do tworzenia testów funkcjonalnych oraz jednostkowych do języka Python.

**Selenium** – Narzędzie do tworzenia testów E2E.

**Docker** – Narzędzie do konteneryzacji aplikacji, umożliwia uruchomienie aplikacji na większości systemów operacyjnych bez zbędnej konfiguracji.

**Bootstrap** – CSS Framework, odpowiedzialny za wygląd stron internetowych.

## 6. Opis aplikacji bazodanowej.

### 6.1 Tytuł.

CryptoFever

### 6.2 Przeznaczenie.

Aplikacja CryptoFever jest przeznaczona dla inwestorów giełdy kryptowalut, którzy chcieliby zautomatyzować obserwację wybranych kryptowalut według własnych preferencji. Inwestor w aplikacji może stworzyć swój zestaw reguł, które definiują interesującą go sytuację na giełdzie kryptowalut BitBay. W przypadku spełnienia reguł zostanie wysłane powiadomienie mailowe do Inwestora. CryptoFever dystrybuowane jest w postaci aplikacji serwerowej, co wymaga komputera z nieprzerywanym czasem pracy oraz ciągłym dostępem do Internetu.

### 6.3 Użytkownicy.

**Gość** – Nie ma jeszcze konta w systemie. Ma jedynie możliwość utworzenia go.

**Inwestor** – Osoba, która posiada konto w aplikacji CryptoFever. Po zalogowaniu ma możliwość obserwacji statystyk i kursu kryptowalut. Może zarządzać grupami reguł oraz regułami, które określają przypadki w których ma zostać wysłane powiadomienie mailowe. Inwestor ma także wgląd w historię wysyłania powiadomień mailowych.

**Administrator** – Osoba odpowiedzialna za prawidłowe funkcjonowanie systemu. Oprócz uprawnień Inwestora posiada dostęp do panelu administratora, w którym może przeglądać i modyfikować wszystkie dane w bazie danych oraz dodawać nowe kryptowaluty do aplikacji.

### 6.4 Zakres.

- Autoryzacja w postaci logowania i rejestracji w systemie.
- Możliwość definiowania reguł oraz grup reguł przez Inwestora.
- Bieżąca analiza giełdy kryptowalut BitBay.
- Zapewnianie powiadomień mailowych dla Inwestora w przypadku gdy zdefiniowane przez niego grupy reguł są spełnione.
- Zapisywanie historii kursu kryptowalut, statystyk społecznościowych oraz historii wykonywania grup reguł.
- Panel administratora z możliwością dodawania kryptowalut oraz modyfikacją danych w bazie danych.

### 6.5 Specyfikacja wymagań funkcjonalnych.

- **Możliwość logowania i zapisu danych Inwestora na serwerze.** Serwer gromadzi dane wprowadzane przez Inwestora, takie jak reguły oraz grupy reguł, zapewniając do nich dostęp niezależnie od urządzenia, na którym Inwestor korzysta z usług.
- **Możliwość wylogowania z konta.**
- **Dostęp do utworzonych przez Inwestora danych jedynie po prawidłowym zalogowaniu na jego konto.** Zabezpieczenie przed niechcianymi wizytami osób trzecich w aplikacji. Posiadanie jedynie adresu internetowego aplikacji nie powinno uprawniać do



przeglądania prywatnych danych użytkowników tej aplikacji, dlatego Inwestor otrzymuje dostęp do swoich danych tylko i wyłącznie po poprawnej autoryzacji.

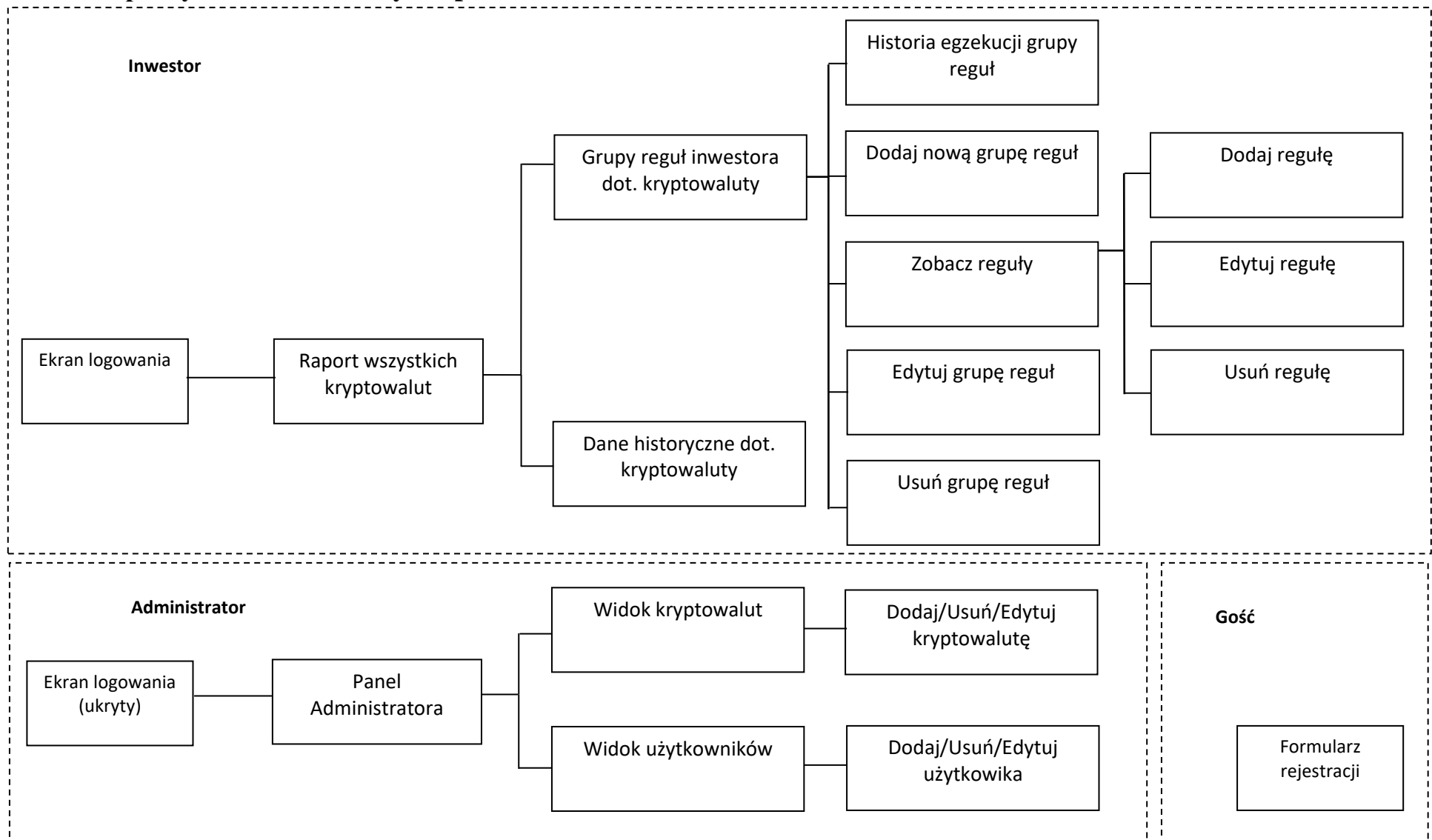
- **Dostęp do poszczególnych funkcji aplikacji następuje dopiero po poprawnym zalogowaniu na konto.** Aplikacja udostępnia swoje funkcjonalności jedynie użytkownikom zalogowanym.
- **Możliwość samodzielnego utworzenia konta w aplikacji przez Gościa.** Do utworzenia konta nie jest potrzebna ingerencja osoby z wyższymi uprawnieniami.
- **Możliwość przeglądania statystyk cen kryptowalut.** Inwestor może zobaczyć procentową zmianę ceny wybranej kryptowaluty w stosunku do wartości sprzed godziny, jednego dnia, trzech dni, tygodnia oraz miesiąca.
- **Możliwość przeglądania historycznych cen kryptowalut.** Inwestor może przeglądać historyczne ceny wybranej kryptowaluty zapisywane co minutę od czasu uruchomienia systemu.
- **Możliwość przeglądania statystyk społecznościowych kryptowalut.** Inwestor może zobaczyć procentową zmianę 7-dniowych trendów zapytań związanych z daną kryptowalutą w wyszukiwarce Google.
- **Możliwość przeglądania historycznych statystyk społecznościowych kryptowalut.** Inwestor może przeglądać historyczne dane odnośnie trendów zapytań zapisywanych codziennie od czasu uruchomienia systemu.
- **Możliwość przeglądania, tworzenia, usuwania i modyfikowania grup reguł.** Inwestor ma możliwość zarządzania grupami reguł utworzonymi w celu obserwacji zachowania konkretnej kryptowaluty na rynku. Każdej grupie można nadać typ, który definiuje co ma się stać po spełnieniu wszystkich reguł w grupie (w pierwszej wersji systemu dostępne jest jedynie powiadomienie mailowe). Każdej grupie można też nadać indywidualną nazwę. Podczas usuwania grupy reguł usuwana jest także cała historia wykonywania usuwanej grupy oraz wszystkie reguły w usuwanej grupie
- **Możliwość przeglądania, tworzenia, usuwania i modyfikowania reguł w grupach.** Inwestor ma możliwość zarządzania regułami znajdującymi się w grupie reguł. Każda reguła w grupie reguł musi mieć inny typ. Każda reguła ma także wartość referencyjną.
- **Egzekwowanie reguł.** Reguły utworzone przez Inwestora są egzekwowane, tzn. jeśli wszystkie warunki określone przez Inwestora za pomocą reguł w jednej grupie zostaną spełnione, następuje akcja określona poprzez typ grupy. Przykładowo, jeśli Inwestor stworzył grupę reguł o typie „powiadomienie mailowe” i umieścił w niej reguły „Cena większa niż 250zł”, „Utrzymuje się przez 10 minut”, to w momencie gdy cena kryptowaluty będzie utrzymywała się na poziomie powyżej 250zł przez dłużej niż 10 minut, zostanie wysłane powiadomienie na adres mailowy Inwestora (podany podczas rejestracji w systemie). Typy egzekwowanych reguł to:
  - Cena powyżej podanej wartości.
  - Cena poniżej podanej wartości.
  - Zmiana kursu waluty o podaną wartość..
  - Procentowa zmiana kursu waluty o podaną wartość.
  - Opóźnienie podejmowania akcji o podany czas.
  - Wskazania rynkowego bota powyżej podanej wartości.
  - Wskazania rynkowego bota poniżej podanej wartości.
  - Wskazania trendów Google powyżej podanej wartości.

- Wskazania trendów Google poniżej podanej wartości.
- **Możliwość przeglądania historii egzekwowanych reguł.** Każde spełnienie wszystkich warunków w grupie reguł powinno być odnotowane do historii w systemie, do której powinien mieć dostęp Inwestor. Każdy wpis w historii powinien zawierać datę,
- **Możliwość sortowania historii egzekwowanych reguł według wybranego kryterium.**
- **Możliwość sortowania danych historycznych o cenach kryptowalut oraz statystykach społecznościowych wg wybranego kryterium.**
- **Brak możliwości usuwania danych historycznych o cenie i statystykach społecznościowych kryptowaluty.** Dane są zapisywane od początku działania aplikacji i są używane do obliczania wartości statystycznych i reguł, dlatego nie powinny być nieumyślnie usuwane. Możliwość usunięcia tych danych ma tylko Administrator z poziomu Panelu Administratora, ręcznie modyfikując dane w bazie danych.
- **Brak możliwości usuwania danych o wykonywaniu grupy reguł.** Inwestor nie ma możliwości usuwania danych o poszczególnych wykonaniach grupy reguł, ale ma możliwość do usunięcia całej grupy reguł, czyli operacji usuwającej całą historię danej grupy oraz wszystkich reguł dotyczących tej grupy.

## 6.6 Specyfikacja wymagań niefunkcjonalnych.

- **Minimalne warunki, które musi spełniać sprzęt na którym zostanie zainstalowana aplikacja:**
  - System: Windows 7/8.1/10, Linux lub macOS.
  - Min. 2GB pamięci RAM
  - Procesor z taktowaniem min 1GHz.
  - Min. 500MB wolnej przestrzeni dyskowej + 7MB wolnej przestrzeni dyskowej na każdą dobę działania systemu. Wynika to z gromadzenia komunikatów JSON otrzymywanych od BitBay API.
  - Stały dostęp do internetu o prędkości minimum 64kb/s.
  - Zainstalowane środowisko Docker.
  - Zainstalowane środowisko Python 3.7+.
- **Dostęp do aplikacji z poziomu przeglądarki internetowej** (np. Mozilla Firefox, Google Chrome, Opera). Ma to zapewnić dostęp do aplikacji bez konieczności instalowania jej na różnych urządzeniach, aby inwestor mógł jak najszybciej zareagować na otrzymywane powiadomienia.
- **Bezawaryjny czas pracy aplikacji na poziomie 99,9% całkowitego czasu pracy.** Aby na systemie można było polegać, musi on działać niezawodnie przez zdecydowaną większość czasu pracy, tak aby Inwestor zawsze był informowany na bieżąco o interesujących go sytuacjach.
- **Średni czas reakcji systemu na akcje użytkownika poniżej 1 sekundy.** Czas reakcji na giełdzie jest bardzo ważny, dlatego reakcja systemu powinna być bardzo szybka, tak aby Inwestor zdążył podjąć akcję w interesujących go warunkach.
- **Maksymalny czas oczekiwania na powiadomienie mailowe to 1 minuta od momentu zaistnienia interesujących Inwestora warunków na giełdzie BitBay.**
- **Aplikacja działająca na platformie Docker.**

## 6.7 Mapa wywołań formularzy i raportów.

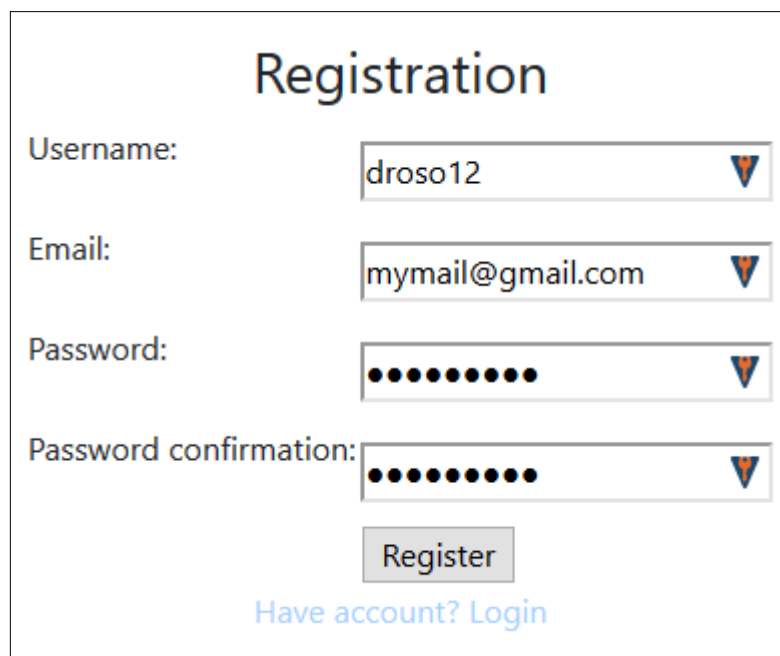


## 6.8 Przykładowe formularze.

### Rejestracja jako inwestor.

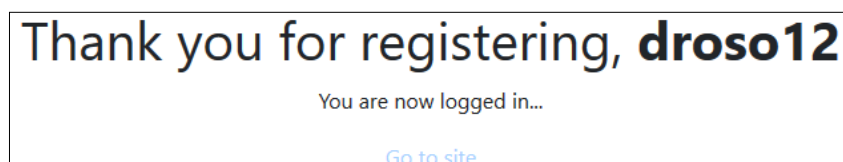
Aby zarejestrować się jako Inwestor w aplikacji należy otworzyć witrynę o adresie <http://127.0.0.1/accounts/register/>, gdzie 127.0.0.1 może być dowolnym adresem internetowym z którym powiązane są usługi serwera CryptoFever.

Formularz rejestracji przedstawiony jest na *rysunku 2*.

A screenshot of a web registration form titled "Registration". It contains four input fields: "Username:" with the value "droso12", "Email:" with the value "mymail@gmail.com", "Password:" with masked characters, and "Password confirmation:" also with masked characters. Each field has a small blue and orange icon on the right. Below the fields is a "Register" button and a link "Have account? Login".

*Rysunek 2. Ekran rejestracji.*

Hasło musi zawierać co najmniej osiem znaków, w tym jeden znak specjalny. Ponadto Inwestor musi posiadać dostęp do podanego podczas rejestracji maila, ponieważ jest on używany w celu wysyłania powiadomień.

A screenshot of a confirmation message box. It says "Thank you for registering, droso12" in large bold text. Below it, in smaller text, it says "You are now logged in...". At the bottom is a blue link "Go to site".

*Rysunek 3. Pomyślne logowanie.*

Po pomyślnym zarejestrowaniu ukaże się takie powiadomienie pomyślnej rejestracji w aplikacji (*Rysunek 3*), inwestor zostanie automatycznie zalogowany. Odnośnik „Go to site” otworzy główny panel aplikacji na którym będzie widoczny raport wszystkich kryptowalut (*Rysunek 4*).

### Dodanie nowej grupy reguł dla inwestora:

Z widoku ekranu głównego (*Rysunek 4*), po wybraniu kryptowaluty (na ten moment dostępne jest tylko Ethereum), należy wybrać opcję „My rulesets”.

	Price	1h	24h	3d	7d	30d	MarketBot	SocialBot	Updated	
Ethereum	547.33PLN	-0.22%	+0.99%	+1.34%	+7.33%	-1.91%	0.69	0.94	Jan. 11, 2020, 4:25 p.m.	History My rulesets

*Rysunek 4. Ekran główny prezentujący dane dot. Kryptowalut.*

W widoku grup reguł inwestora (*Rysunek 4*) należy wybrać opcję „Add ruleset”.

Name	Manage	Rules
<a href="#">Go back</a>		<a href="#">Add ruleset</a>

*Rysunek 5. Widok grup reguł inwestora.*

Ukaże się formularz dodawania nowej grupy reguł (*Rysunek 6*), w polu *Name* należy wpisać nazwę naszej grupy reguł, a w polu *Type* należy wybrać typ grupy. Do wyboru są trzy wartości „Email Only” (wysyłaj tylko wiadomości mail), „Sell” (grupa reguł dotycząca sprzedaży) oraz „Buy” (grupa reguł dotycząca kupna), aktualnie wartość „Email Only” jest zaimplementowana, a pozostałe wartości są wyszarzone. Należy zastosować zmiany wybierając opcję „Submit”.

## Ethereum

### Create new ruleset

Name:

GrupaRegul#1

Type:

Email Only

Back

Submit

*Rysunek 6. Ekran dodawania nowej grupy reguł*

## Dodanie nowej reguły dla grupy reguł.

Z widoku grupy reguł inwestora, należy wybrać wcześniej utworzoną grupę reguł i wybrać opcję „Edit rules”.

Name	Manage	Rules
GrupaRegul#1	<a href="#">Execution Log</a> <a href="#">Remove</a> <a href="#">Edit</a>	<a href="#">Edit rules</a>

Rysunek 7. Widok uzupełnionych grup reguł inwestora

Następnie podobnie jak w widoku dla grup reguł należy wybrać „Add new rule”. Ukaze się formularz dodawania nowej reguły (Rysunek 8), pole *Type of rule* określa typ reguły (wszystkie typy zostały opisane w relacji **REL/003** *crypto\_rule*).

Przykładowo w przypadku „Price is below (PLN)” wpisanie w polu *Value*: 500.0, oznacza (gdy istnieje tylko ta jedna reguła), że do Inwestora zostanie wysłany mail, jeśli cena kryptowaluty będzie wynosić poniżej 500 zł.

Do grupy reguł można dodawać tylko reguły o różnych typach, ale żeby zadanie grupy reguł zostało wykonane (np. wysłanie powiadomienia mailowego), wszystkie reguły muszą zostać spełnione.

Ethereum's GrupaRegul#1

Create new rule

Type of rule: 

Price is below (PLN) ▾

Value: 

500.0

Back

Submit

Rysunek 8. Ekran dodawania nowej reguły dla „GrupaRegul#1”

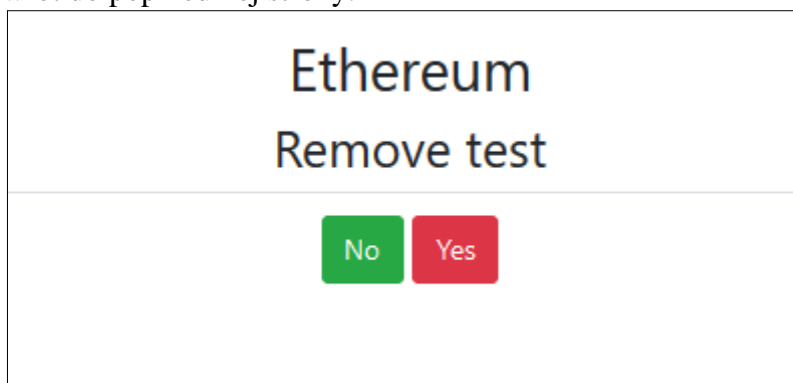
## Usunięcie grupy reguł.

Aby usunąć grupę reguł należy przejść do widoku grupy reguł klikając „My rulesets” przy wybranej przez nas kryptowalucie na stronie głównej. Następnie przy jednej z grup reguł należy kliknąć „Remove”.

Name	Manage	Rules
test	<a href="#">Execution Log</a> <a href="#">Remove</a> <a href="#">Edit</a>	<a href="#">Edit rules</a>

Rysunek 9. Widok grup reguł.

Ostatecznie należy potwierdzić usunięcie grupy reguł wybierając przycisk „Yes”. Wybór „No” oznacza powrót do poprzedniej strony.



Rysunek 10. Potwierdzenie usunięcia grupy reguł

## 6.9 Przykładowe raporty.

### Dane historyczne kryptowaluty.

Interesującą funkcjonalnością aplikacji jest zapisywanie danych historycznych kryptowaluty. Aby mieć dostęp do tych danych należy wybrać opcję „History” z ekranu głównego przy jednej z kryptowalut (Rysunek 10).

	Price	1h	24h	3d	7d	30d	MarketBot	SocialBot	Updated	
Ethereum	547.33PLN	-0.22%	+0.99%	+1.34%	+7.33%	-1.91%	0.69	0.94	Jan. 11, 2020, 4:25 p.m.	History My rulesets

Rysunek 10. Ekran główny, widok na wszystkie obsługiwane kryptowaluty.

Po wyborze ukaże się ekran danych historycznych dla danej kryptowaluty (Rysunek 9), aby przejść do kolejnej strony należy wybrać opcję „Next Page”.

Widok zawiera także możliwość sortowania. Aby posortować dane według wybranego kryterium należy wybrać nagłówek tabeli odpowiadający temu kryterium.

Ethereum				
Date	Price (PLN)	ASKS Value	BIDS Value	Avg. Transaction Value
Jan. 11, 2020, 5:03 p.m.	547.00	563704.23089621	513907.237479228	2866.15578108349
Jan. 11, 2020, 5:02 p.m.	547.00	560306.844539761	513754.513085055	2866.15578108349
Jan. 11, 2020, 5:01 p.m.	549.67	565868.233492306	513410.170015891	2822.79685241049
Jan. 11, 2020, 4:59 p.m.	547.00	565821.103498769	511741.492043927	2823.82825151533
Jan. 11, 2020, 4:57 p.m.	546.50	562395.516068476	509638.632113771	2825.80516618233
Jan. 11, 2020, 4:55 p.m.	546.50	554176.492697681	510568.599396073	2825.80516618233
Jan. 11, 2020, 4:53 p.m.	548.37	565830.781188576	504134.243978568	2840.85329478843
Jan. 11, 2020, 4:51 p.m.	547.95	561446.528708518	503744.761253704	2840.78915475027
Jan. 11, 2020, 4:49 p.m.	547.95	585038.204389867	515278.122318516	2840.78915475027
Jan. 11, 2020, 4:47 p.m.	547.95	569840.93553714	515970.707519582	2840.78915475027
Jan. 11, 2020, 4:45 p.m.	547.70	553020.525771018	515188.810415833	2829.97773275679
Prev. page Next page				

Rysunek 11. Ekran danych historycznych dla Ethereum.

Dane posortowane rosnąco według ceny zostały przedstawione na *rysunku 12*:

Ethereum				
Date	Price (PLN)	ASKS Value	BIDS Value	Avg. Transaction Value
Dec. 18, 2019, 2:21 p.m.	461.00	607547.942222035	346176.726493553	728.478617194338
Dec. 18, 2019, 2:11 p.m.	461.01	609853.016370355	346075.942282811	732.020616211064
Dec. 18, 2019, 2:13 p.m.	461.01	607946.703708505	345644.072145793	732.020616211064
Dec. 18, 2019, 2:15 p.m.	461.01	608303.055241501	344219.190847708	732.020616211064

*Rysunek 12. Ekran danych historycznych dla Ethereum.*

### Dane historyczne egzekucji grupy reguł.

Reguły inwestora są najpierw sprawdzane, a następnie jeżeli wszystkie reguły w danej grupie reguł są spełnione grupa jest egzekwowana. Aby sprawdzić historię egzekucji należy otworzyć panel grup reguł dla danej kryptowaluty (*Rysunek 6*), a następnie wybrać opcję „Execution Log” przy wybranej grupie reguł.

Name	Manage	Rules
GrupaRegul#1	<a href="#">Execution Log</a> <a href="#">Remove</a> <a href="#">Edit</a>	<a href="#">Edit rules</a>

*Rysunek 13. Widok grupy reguł inwestora.*

Na ekranie ukaże się historia wykonywania grupy reguł, czyli wszystkich przypadków w których wszystkie reguły zawarte w danej grupie reguł zostały spełnione. Tabela zawierająca historię wykonywania grupy reguł podlega sortowaniu według kryterium wybieranego z nagłówek tabeli. Dostępne kryteria sortowania to data, cena, ilości kryptowaluty biorącej udział w transakcji oraz typ grupy reguł.















GrupaRegul#1			
Date	Price (PLN)	Amount (ETH)	Type
Jan. 11, 2020, 5:25 p.m.	547.00PLN	0.00000000	Email
Jan. 11, 2020, 5:24 p.m.	547.00PLN	0.00000000	Email
Jan. 11, 2020, 5:23 p.m.	549.32PLN	0.00000000	Email
Jan. 11, 2020, 5:22 p.m.	549.32PLN	0.00000000	Email
Jan. 11, 2020, 5:21 p.m.	549.70PLN	0.00000000	Email
Jan. 11, 2020, 5:20 p.m.	549.70PLN	0.00000000	Email
Jan. 11, 2020, 5:19 p.m.	549.70PLN	0.00000000	Email
Jan. 11, 2020, 5:18 p.m.	549.70PLN	0.00000000	Email
Jan. 11, 2020, 5:17 p.m.	549.70PLN	0.00000000	Email

*Rysunek 14. Historia egzekucji dla grupy reguł GrupaRegul#1.*



## 7. Zawartość nośnika z systemem.

Pliki z całym systemem zajmują łącznie 21,6 MB, przy czym nie zawierają potrzebnych do uruchomienia systemu bibliotek.

	_docs	1/6/2020 12:31 PM	File folder	
	_logs	11/19/2019 11:07 PM	File folder	
	crypto	11/19/2019 11:07 PM	File folder	
	cryptobot	11/19/2019 11:07 PM	File folder	
	registration	11/19/2019 11:07 PM	File folder	
	.gitignore	11/19/2019 11:07 PM	GITIGNORE File	1 KB
	docker-compose.yml	11/19/2019 11:17 PM	YML File	2 KB
	Dockerfile	11/19/2019 11:07 PM	File	1 KB
	LICENSE	11/19/2019 11:07 PM	File	2 KB
	local.env	11/19/2019 11:07 PM	ENV File	1 KB
	manage.py	11/19/2019 11:07 PM	Python File	1 KB
	README.md	11/19/2019 11:07 PM	MD File	1 KB
	requirements.txt	11/19/2019 11:07 PM	Text Document	1 KB
	requirements_dev.txt	11/19/2019 11:07 PM	Text Document	1 KB

Rysunek 14. Zawartość nośnika z systemem.

### Foldery:

- **\_docs** – zawiera kompletną dokumentację aplikacji, w tym tę dokumentację, diagramy ERD, mapę formularzy oraz prototyp aplikacji.
- **\_logs** – zawiera logi, które aplikacja generuje podczas działania.
- **crypto** – zawiera kod odpowiedzialny za poprawne funkcjonowanie grup reguł, reguł, analizę kryptowalut, statystyk społecznościowych oraz wykonywanie zadań grup reguł.
- **cryptobot** – zawiera m.in. ustawienia (settings.py).
- **registration** – zawiera logikę logowania oraz autoryzacji.

### Pliki:

- **Dockerfile** oraz **docker\_compose.yml** – służą do uruchomienia serwera aplikacji przy pomocy darzędzi *Docker* oraz *docker\_compose*.
- **local.env** – przechowuje najważniejsze ustawienia aplikacji.
- Pliki **requirements\*.txt** – zawierają dane o wymaganych modułach i pakietach dla języka Python wymaganych do poprawnego funkcjonowania aplikacji.

## 8. Sposób instalacji / uruchomienia systemu.

System nie jest scentralizowany i nie istnieje żaden publiczny serwer który udostępnia usługi CryptoFever. Zostało to zrobione z powodów bezpieczeństwa, gdyż na serwerze przechowywane są prywatne informacje inwestorów o ich inwestycjach na rynku kryptowalut. Zamiast scentralizowanego serwera aplikacja jest dystrybuowana za pomocą prostej aplikacji umieszczonej w kontenerze Dockera. Każda osoba chcąc skorzystać z aplikacji może zainstalować ją w kontenerze Dockera u siebie na lokalnym komputerze, który zacznie wtedy działać jako serwer dostępny jedynie dla użytkownika. Dzięki temu na jednym serwerze CryptoFever znajdzie się o wiele mniej informacji o użytkownikach systemu.

Instrukcja została utworzona z myślą o użytkownikach systemu Windows 10, jednak kroki dla systemu macOS oraz Linux są analogiczne.

1. Przekopiować zawartość nośnika z systemem do wybranego folderu na komputerze.
2. Zainstalować aplikację *Docker* z witryny internetowej <https://docs.docker.com/docker-for-windows/install/>
3. Przejsć do folderu w którym znajduje się przekopowana wcześniej aplikacja.
4. Edytować plik *local.env*. Uzupełnić następujące ustawienia:
  - EMAIL\_HOST – adres serwera świadczącego usługi poczty elektronicznej.
  - EMAIL\_HOST\_USER – adres mailowy z którego mają być wysyłane powiadomienia aplikacji.
  - EMAIL\_HOST\_PASSWORD – hasło do podanego adresu mailowego.
  - EMAIL\_PORT – port na którym świadczona jest usługa poczty elektronicznej.

Uwaga: Należy wcześniej skonfigurować adres pocztowy umożliwiając usługom zewnętrznym wysyłanie wiadomości mailowych.
5. Uruchomić okno konsoli w folderze aplikacji i wywołać następującą komendę, wywołującą kompilację oraz uruchomienie aplikacji:  
`docker-compose up`  
Okna konsoli nie należy zamykać, w jego ramach odbywa się świadczenie usług CryptoFever.
6. Przywołać kolejne okno konsoli w folderze aplikacji i wywołać następujące komendy:  
`docker-compose exec web python manage.py collectstatic --noinput`  
`docker-compose exec web python manage.py migrate`
7. Aplikacja CryptoFever jest gotowa do użycia. W przeglądarce na urządzeniu wprowadź adres `http://127.0.0.1/` aby sprawdzić poprawność działania.

Domyślnie w aplikacji nie istnieje konto Administratora. Aby je utworzyć należy wykonać następujące kroki:

1. Uruchomić okno konsoli w folderze aplikacji i wywołać komendę:  
`docker-compose exec web python manage.py createsuperuser`
2. Postępować zgodnie z instrukcjami w konsoli, czyli podać login oraz hasło dla Administratora.
3. Sprawdzić poprawność działania konta Administratora logując się według podanych wcześniej danych pod adresem <http://127.0.0.1/admin/>. Po poprawnym zalogowaniu zostanie wyświetlony panel administratora.
4. O panelu admina można poczytać w *literaturze 7*.

Przykładowa baza danych została spakowana do pliku dbbackup.zip, aby jej dane zaimportować do aplikacji należy postępować zgodnie z *literaturą 6*. Każdą komendę musi poprzedzać `docker-compose exec db`

## 9. Testowanie aplikacji.

1. Aby przetestować aplikację należy najpierw zainstalować język Python 3.7+.  
<https://www.python.org/>
2. Następnie należy w oknie konsoli w folderze aplikacji wywołać następujące komendy:  

```
python -m ensurepip  
python -m pip install -r requirements.txt  
python -m pip install -r requirements_dev.txt
```
3. A następnie żeby przetestować aplikację należy wywołać komendę:  

```
DJANGO_SETTINGS_MODULE=cryptobot.settings pytest
```

Aplikacja posiada testy E2E, funkcjonalne oraz jednostkowe. Podczas testów E2E zostanie otworzona przeglądarka. Projekt zawiera 20 testów.

## 10. Zakończenie.

Celem projektu była automatyzacja procesu obserwacji kursu i statystyk kryptowalut przy pomocy aplikacji webowej. Rozwiązanie miało za zadanie wspomóc przyszłych inwestorów w podejmowaniu decyzji rynkowych na podstawie dodawanych przez nich reguł.

Cel został osiągnięty, aplikacja spełnia swoje zadanie, jednak nie zostały zaimplementowane dodatkowe wymagania takie jak np. automatyczna sprzedaż i kupno kryptowalut. Ponadto więcej czasu podczas implementacji należałoby poświęcić wyglądowi interfejsu, ponieważ jest on zbyt mało nowoczesny porównując ze współczesnymi standardami. Aplikacja została napisana modularnie tzn. dodawanie nowych kryptowalut i giełd nie stoi na przeszkodzie. Podczas implementacji zabrakło czasu na realizację dodatkowych funkcjonalności, takich jak automatyczna sprzedaż oraz kupno kryptowalut. Przyczyną był głównie spory nakład pracy wymagany do zrealizowania tych funkcjonalności.

Użyte narzędzia sprawdziły się, Django oraz PostgreSQL wspierają nowoczesne technologie bazodanowe, które mogą pomóc w przyszłym rozwoju aplikacji (np. skalowanie). Język Python pozwolił na szybką implementację logiki biznesowej, natomiast Bootstrap pozwolił na szybkie stworzenie szaty graficznej, którą można dalej rozwijać. Narzędzia testujące Pytest oraz Selenium także zdały swój egzamin. Dzięki konteneryzacji aplikacji w dockerze aplikację można uruchomić na każdym popularnym systemie operacyjnym. Użycie innych narzędzi do implementacji systemu prawdopodobnie nie zwiększyłoby znacznie wygody oraz czasu pracy.

Mówiąc o rzeczach, które można by było poprawić, formularze mogłyby zostać napisane przy pomocy Django forms (*literatura 5*), zwiększyłoby to bezpieczeństwo przesyłanych danych. Aplikacja potrzebuje także nowocześniejszej szaty graficznej, a wszystko to jest możliwe do zaimplementowania z aktualnie używanymi narzędziami. Projekt bazy danych umożliwia także rozwój systemu o dalsze funkcjonalności jak np. handel na giełdzie (patrz *REL/008 crypto\_cryptowallet*).

Podsumowując, projekt się powiódł, jednak porusza on temat na tyle obszerny i tak szybko zmieniający się, że należałoby go dalej rozwijać. Jednym z pomysłów jest wypuszczenie aplikacji na licencji Open Source tak aby także społeczność mogła pomóc w rozwoju CryptoFever.

## 11. Źródła.

1. <https://docs.djangoproject.com/en/3.0/> - dokumentacja Django
2. <https://www.postgresql.org/docs/> - dokumentacja PostgreSQL
3. <https://www.livecoinwatch.com/> - strona ze statystykami kryptowalut
4. <https://docs.docker.com/> - dokumentacja Dockera
5. <https://docs.djangoproject.com/en/3.0/topics/forms/> - Django forms
6. <https://www.linode.com/docs/databases/postgresql/how-to-back-up-your-postgresql-database/> - Tworzenie kopii zapasowej i importowanie bazy danych
7. <https://docs.djangoproject.com/en/3.0/ref/contrib/admin/> - panel admina