

Autorzy:

Drozdowski Hubert, 238328

Mróz Jan, 242440

Inżynieria Systemów Baz Danych

PWr, IZ, III rok, sem. 5, wtorek godz. 13:15

Karta Projektu**Temat:**

Monitorowanie rynku i społeczności kryptowalut.

Nazwa systemu/aplikacji:

CryptoFever

Cel projektu:

Automatyzacja procesu obserwacji kryptowalut przy pomocy aplikacji webowej. Rozwiązanie ma za zadanie pomóc przyszłym inwestorom podejmować decyzje rynkowe na podstawie dodanych przez nich reguł.

Zakres:

Analiza obecnych rozwiązań;
 Projekt i implementacja bazy danych;
 Implementacja analizy rynku i społeczności kryptowalut;
 Implementacja egzekwowania reguł inwestorów;
 Implementacja interfejsu;
 Testowanie jednostkowe, funkcjonalne i E2E;
 Audyt bezpieczeństwa;

Użytkownicy i ich wymagania funkcjonalne – zestaw minimalny:**Inwestor:**

Wymaganie	Ocena realizacji
Możliwość rejestracji i logowania	Must, wymóg prosty w implementacji.
Możliwość dodania/edycji/usuwania grup reguł	Must, wymaga większego nakładu pracy (interfejs, logika, testy)
Możliwość dodania/edycji/usuwania reguł w danej grupie	Must, opiera się na poprzednim wymogu, podobny nakład pracy.
Możliwość wgląd w historie transakcji	Should, polega na systemie reguł, proste w implementacji.

Administrator:

Wymaganie	Ocena realizacji
Możliwość logowania	Must, Można skorzystać z implementacji dla Inwestorów.
Możliwość dodawania/usuwania kryptowalut w panelu admina	Should, django udostępnia przystępny panel admina, nakład pracy minimalny.
Możliwość edycji danych użytkowników	Could, django udostępnia przystępny panel admina, nakład pracy minimalny.

Wymagania funkcjonalne dodatkowe:**Ogólne:**

Wymaganie	Ocena realizacji
Obsługa giełdy kryptowalut np. BitBay API (odczyt)	Must, polega na BitBay API, postępowanie zgodnie z dokumentacją.
Automatyczna analiza rynku	Should, trudne w realizacji, wymaga logiki biznesowej i testów.
Automatyczna analiza społeczności	Should, trudne w realizacji, wymaga logiki biznesowej i testów.
Zapis danych historycznych	Could, proste w realizacji.
Egzekwowanie reguł inwestorów	Must, wymaga dużej ilości testów dla każdej z reguł, każda reguła posiada swoją logikę.

Wymagania niefunkcjonalne:

<i>Wymaganie</i>	<i>Ocena realizacji</i>
Czas reakcji poniżej 1s	Should, początkowo dla małej ilości użytkowników wymaganie będzie spełnione, ale podczas implementacji systemu należy wziąć pod uwagę przyszłą skalowalność.
Czas analizy rynku i społeczności poniżej 1 minuty	Must, trudne w realizacji, ponieważ system polega na zewnętrznych systemach, które mogą odpowiadać dłużej lub w ogóle.
Uptime 99.9%	Should, system powinien stać na przynajmniej dwóch serwerach, kosztowne, ale nie wymaga dużego nakładu pracy.
Powiadomienia do inwestorów są wysyłane w ciągu minuty	Must, wymaga doboru infrastruktury fizycznej na podstawie ilości użytkowników (skalowalność), realizacja podobna jak w przypadku czasu reakcji.

Wykorzystane narzędzia programistyczne:

- baza danych: postgres
- aplikacja: python 3, Django
- dokumentacja: Word, pydoc
- inne: selenium

Charakter projektu:

Nowy

Harmonogram:

<i>Data planowana</i>	<i>Produkt</i>	<i>Data zrealizowania</i>	<i>Ocena realizacji</i>
22.10.19r.	Projekt i implementacja bazy danych		Implementacja bazy przy pomocy Django i Postgres, zadanie o średnim poziomie trudności.
29.10.19r.	Prototyp interfejsu graficznego, operacje bazodanowe CRUD, procedury dostępu do systemu		Użycie narzędzia do projektowania GUI, implementacja operacji bazodanowych w Django, nietrudne, wymaga większego nakładu pracy.
05.11.19r.	Możliwość dodawania/modyfikowania grup reguł i nowych kryptowalut przez administratora		Polega na interfejsie graficznym i operacjach bazodanowych.
12.11.19r.	Możliwość dodawania/modyfikowania podstawowych reguł, testy jednostkowe i funkcjonalne		Polega na interfejsie graficznym i operacjach bazodanowych. Dodatkowo jest potrzeba dużej ilości testów.
19.11.19r.	Implementacja BitBay API i analizy rynku kryptowalut		Zgodnie z dokumentacją BitBay
26.11.19r.	Implementacja analizy społeczności kryptowalut		Użycie zewnętrznej biblioteki Google Trends, proste w realizacji.

03.12.19r.	Egzekwowanie reguł inwestorów, test jednostkowe i funkcjonalne		Trudne w wykonaniu, potrzeba dużej ilości testów.
10.12.19r.	Implementacja przechowywania historii transakcji		Proste w realizacji, polega na interfejsie graficznym oraz analizie rynku.
17.12.19r.	Audyt bezpieczeństwa, testy E2E, poprawa błędów.		W zależności od audytu, istnieje ryzyko dużego nakładu pracy.
07.01.20r.	Ewentualna implementacja kupna/sprzedaży walut		Trudne w realizacji ze względu na kwestie bezpieczeństwa.
14.01.20r.	Finalny release 1.0		

Ocena końcowa:

Ryzyka i ich ocena:

<i>Ryzyko</i>	<i>Ocena początkowa ryzyka</i>	<i>Ocena końcowa ryzyka</i>
Zbyt duża złożoność produktu	Niska	
Dodatkowe wymagania	Niska	

Podpisy studentów:

Data oddania: