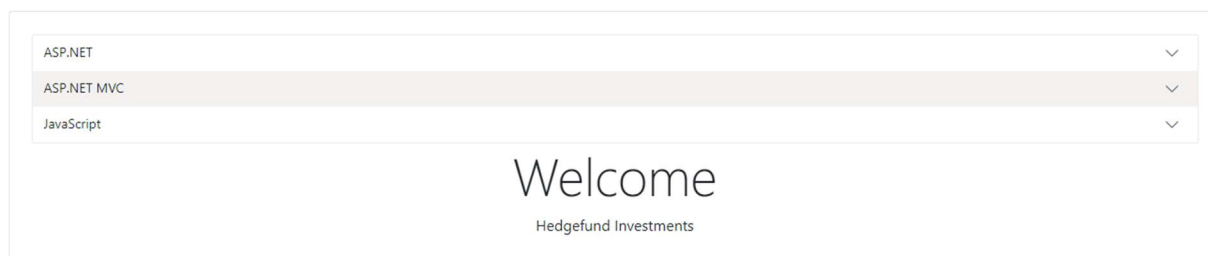


Fundusze Inwestycyjne



Autorzy:

Jakub Witkowski

Tomasz Ustrzycki

Spis Treści:

Spis treści

Spis Treści:	2
Specyfikacja projektu	3
Koncepcja systemu	3
Opis	3
Przepływ danych	4
Zakładanie konta	5
Administracja systemem	6
Wybrana metodyka wytwarzania	6
Słownik pojęć i terminów	7
Przypadki użycia	7
Opis przypadków użycia	8
Założenie konta	8
Logowanie się	8
Przeglądaj dane funduszy	8
Wykonaj symulację ex-ante	8
Zapisz transakcję	8
Modyfikuj dane konta	8
Przeglądaj wykonane transakcje	8
Zarządzanie kontami użytkowników	8
Wymagania	9
Wymagania funkcjonalne	9
Wymagania нефункционалне	9
Wykaz zastosowanych technologii	10
Visual Studio 2022	10
C#	10
ASP.NET Core	10
SyncFusion	10
Diagramy aktywności	11
Wyświetlanie wykresów	11
Zakładanie konta	13
Wyświetlanie wykresu z predykcją stopy zwrotu	14
Zapisanie transakcji	15
Diagram klas	16

Diagramy sekwencji.....	17
Logowanie	17
Tworzenie konta	18
Usuwanie użytkownika.....	18
Przykłady współpracy	18
Testy manualne	20
Refaktoryzacja kodu	20
Wzorce projektowe	21
Instrukcja użytkowania.....	21
Podsumowanie.....	23

Specyfikacja projektu

Koncepcja systemu

Projekt ten miał być pomocą dla początkujących inwestorów, aby pomagać im w rozpoczęciu inwestowania. Dzięki połączeniu z internetem, po uruchomieniu serwera hostującego system oraz wykupieniu domeny, wszyscy użytkownicy będą w stanie korzystać z systemu bez pobierania żadnych aplikacji na swój komputer, jak i również będą mogli korzystać z niego na urządzeniach mobilnych.

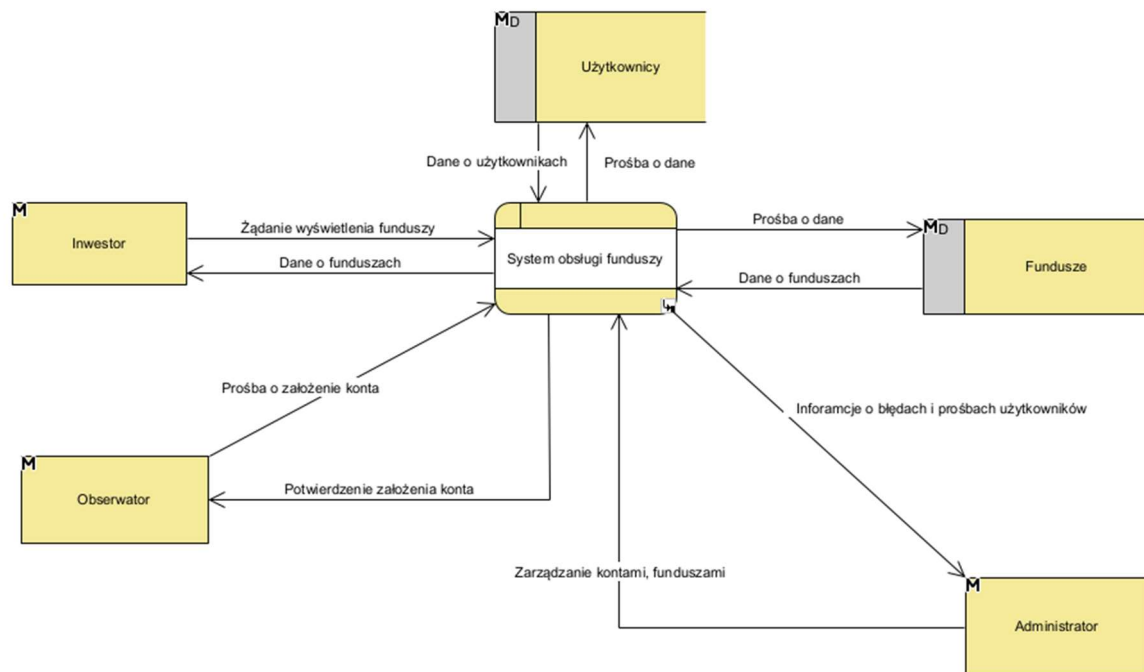
Opis

Hedgefund Investments jest aplikacją internetową napisaną zarówno za pomocą domyślnych narzędzi Microsoft .NET Core, jak i SyncFusion Community, której licencję uzyskaliśmy na potrzeby projektu. Każdy użytkownik posiada swoje własne konto z odpowiednim poziomem uprawnień w zależności od jego funkcji w systemie.

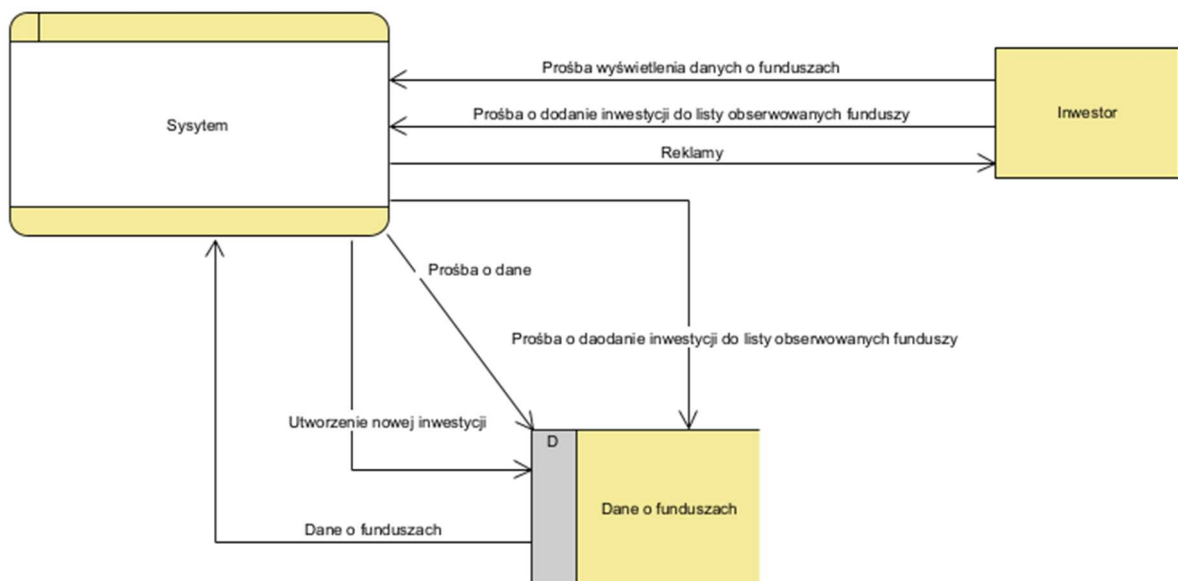
Podstawowy użytkownik, inwestor, może przeglądać listę funduszy wraz z informacjami o nich, jak procentowa stopa zwrotu inwestycji, wymagania funduszu co do miesięcznego dochodu inwestora, jak i może zobaczyć wykres stopy zmiennej w czasie oraz predykcji jej wysokości na następne 5 dni. Może on również zawęzić listę tylko o fundusze, których wymagania spełnia, oraz inwestować w nich za pomocą pieniędzy wprowadzonych w system.

Ma on również dostępne zarządzanie swoim kontem, gdzie zmienia informacje o nim, dostosowuje zabezpieczenia i wykonuje inne czynności z nim związane.

Przepływ danych



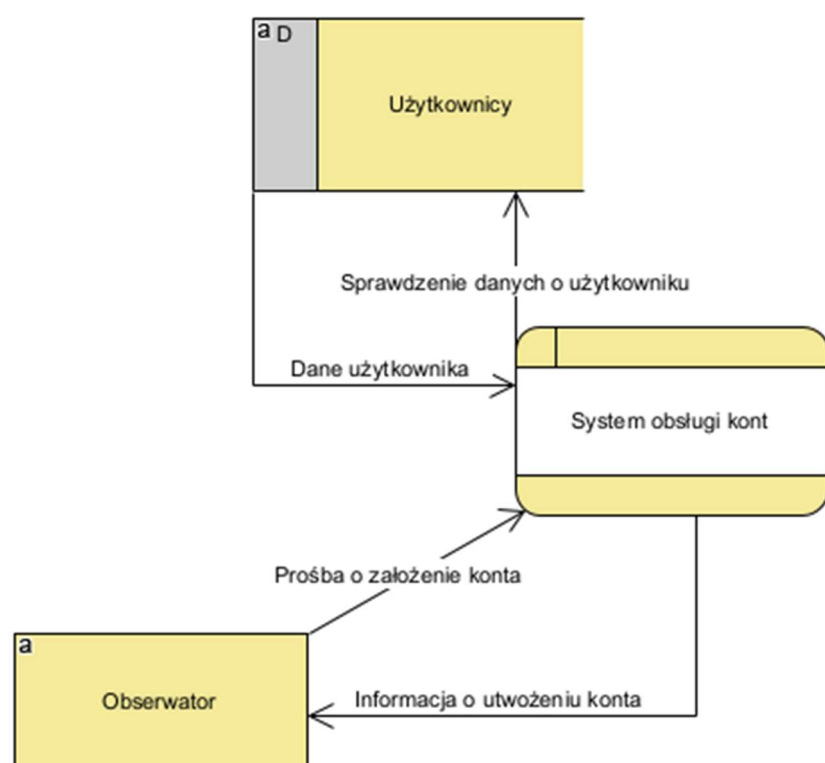
Poziom 0



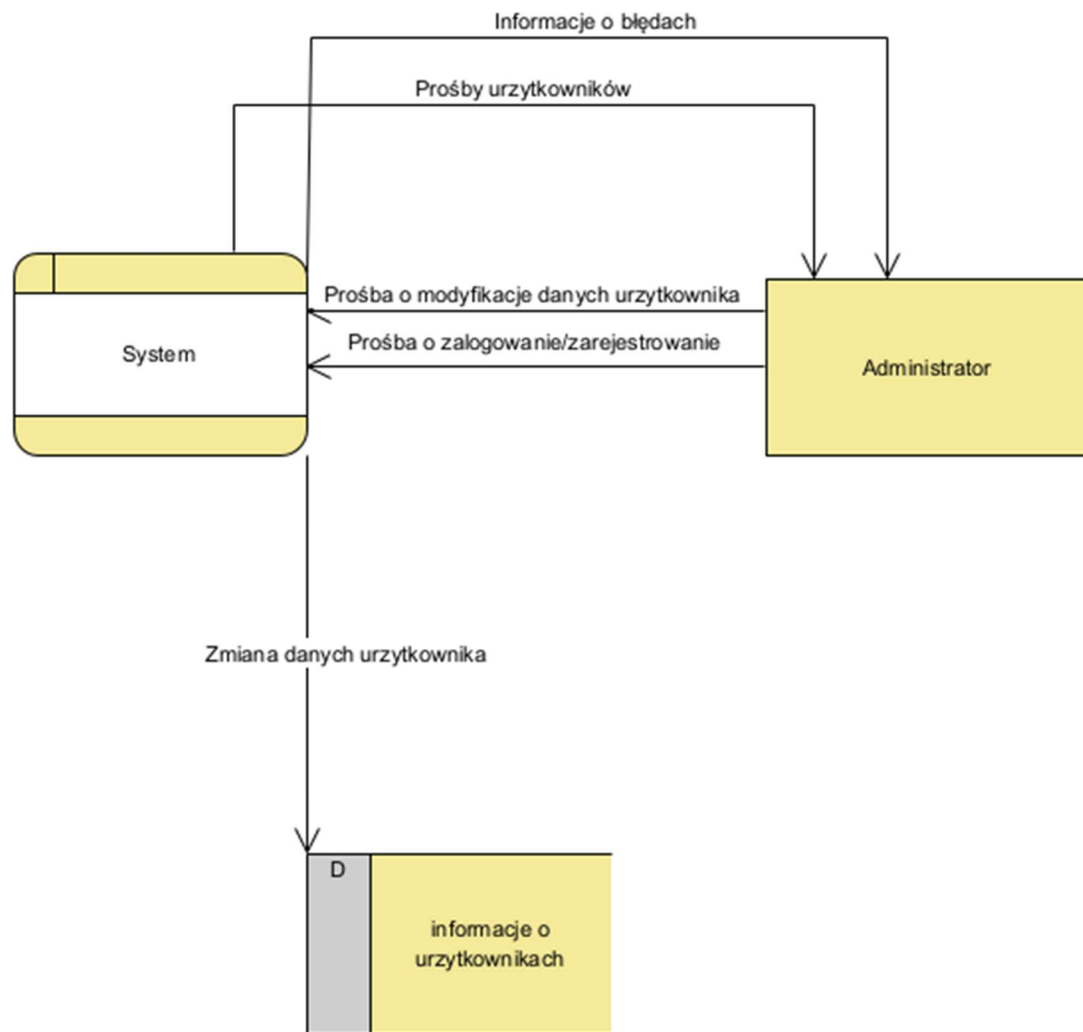
Poziom 1

Zakładanie konta

poziom 1



Administracja systemem



Wybrana metodyka wytwarzania

Dla mojego projektu wybrałem iteracyjny model wytwarzania. Na zakończeniu każdej iteracji dostawałem działającą wersję produktu z kolejnymi funkcjami dodanymi w danej iteracji, aby uzyskać na końcu produkt z wszystkimi zaplanowanymi funkcjami.

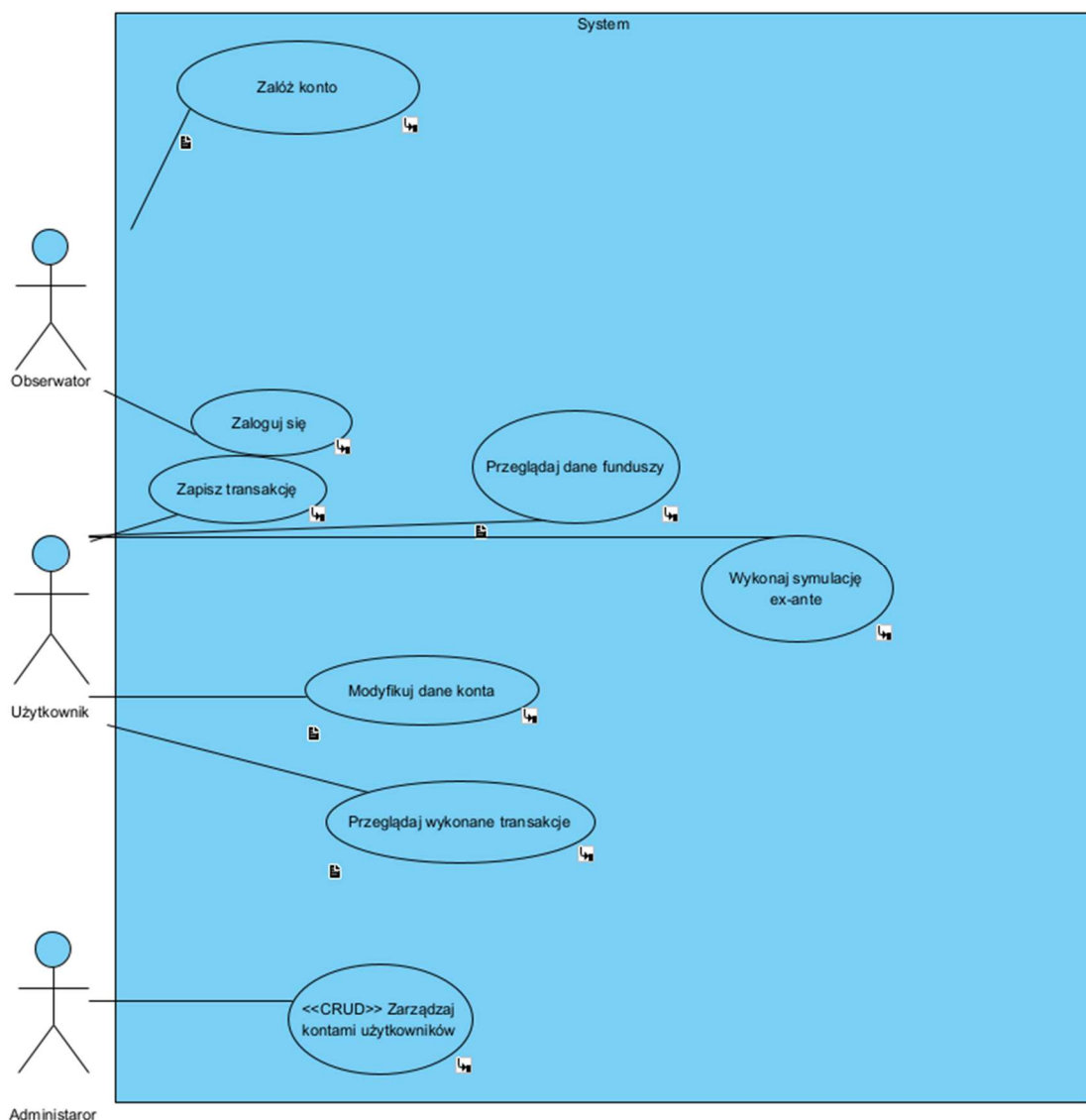
Ten model okazał się być bardzo odpowiedni dla mojego projektu, gdyż zdarzało się, że funkcjonalności były od siebie w pewien sposób zależne, z paroma wyjątkami. W związku z tym unikałem przepisywania kodu na nowo po dodaniu dwóch funkcji tworzonych jednocześnie.

Kolejne iteracje obejmowały: stworzenie podstawowego interfejsu, dodanie historii stopy zwrotu w czasie, historii transakcji, import z pliku, role użytkowników, wykresy stopy zwrotu w czasie, porównywanie wykresów dwóch funduszy, export do .CSV i inne, mniejsze iteracje. Dzięki podzieleniu w ten sposób, mogłem się skupić na skończeniu jednej funkcjonalności na raz, co zwiększało efektywność pisania kodu.

Słownik pojęć i terminów

- Branch - gałąź, w GitHubie: jest to rozgałęzienie umożliwiające paru użytkownikom pracę nad kodem bez konfliktów. Po zakończeniu danej funkcjonalności to rozgałęzienie może zostać połączone z główną gałęzią.
- Projekt - Odniesienie do programu “Fundusze Inwestycyjne” oraz czynności związanych z wytworzeniem oprogramowania oraz dokumentacji do tego programu.
- UI - Interfejs użytkownika
- UX - (ang. User Experience) całość wrażeń, jakich doświadcza użytkownik podczas korzystania z produktu.

Przypadki użycia



Opis przypadków użycia

Założenie konta

Umożliwia ono dodanie użytkownika do systemu, dzięki czemu będzie on w stanie korzystać z systemu. Wymagane jest hasło oraz adres e-mail, na który wysyłana jest wiadomość z potwierdzeniem. Nazwa musi używać wyłącznie znaków UTF-8 oraz być uprzednio niezarejestrowana w systemie, inaczej nie dojdzie do założenia konta.

Logowanie się

Sesja zostaje zautoryzowana do poziomu uprawnień użytkownika, może wtedy korzystać z całości systemu przewidzianej dla jego autoryzacji, jak i może zarządzać swoim kontem.

Przeglądaj dane funduszy

Jest to ogólna funkcja odnosząca się do paru pomniejszych funkcjonalności. Użytkownik może zarówno przeglądać dane wszystkich funduszy, historię w czasie konkretnego, odfiltrować te, których wymagania spełnia, jak i porównać dwa różne na jednym wykresie.

Wykonaj symulację ex-ante

Użytkownik może zobaczyć na wykresie predykcję stopy zwrotu danego funduszu na bazie danych z poprzedniego miesiąca, na następne 5 dni kalendarzowych.

Zapisz transakcję

Użytkownik może zainwestować pieniądze przypisane do jego konta, aby dostać je z powrotem po określonym czasie wraz z należną stopą zwrotu.

Modyfikuj dane konta

Użytkownik jest w stanie zmienić nazwę konta, adres e-mail, hasło, swoją miesięczną pensję oraz parę innych parametrów.

Przeglądaj wykonane transakcje

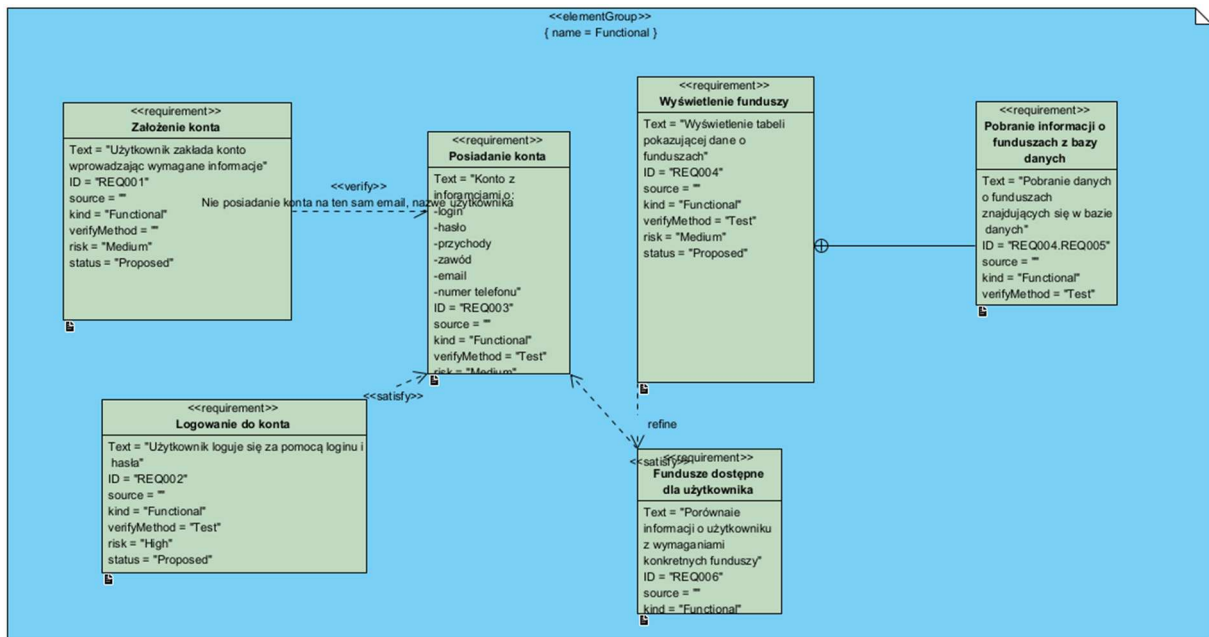
Użytkownik może zobaczyć wszystkie transakcje z nim związane, zarówno przychód na konto od administratora, swoje inwestycje oraz zwroty.

Zarządzanie kontami użytkowników

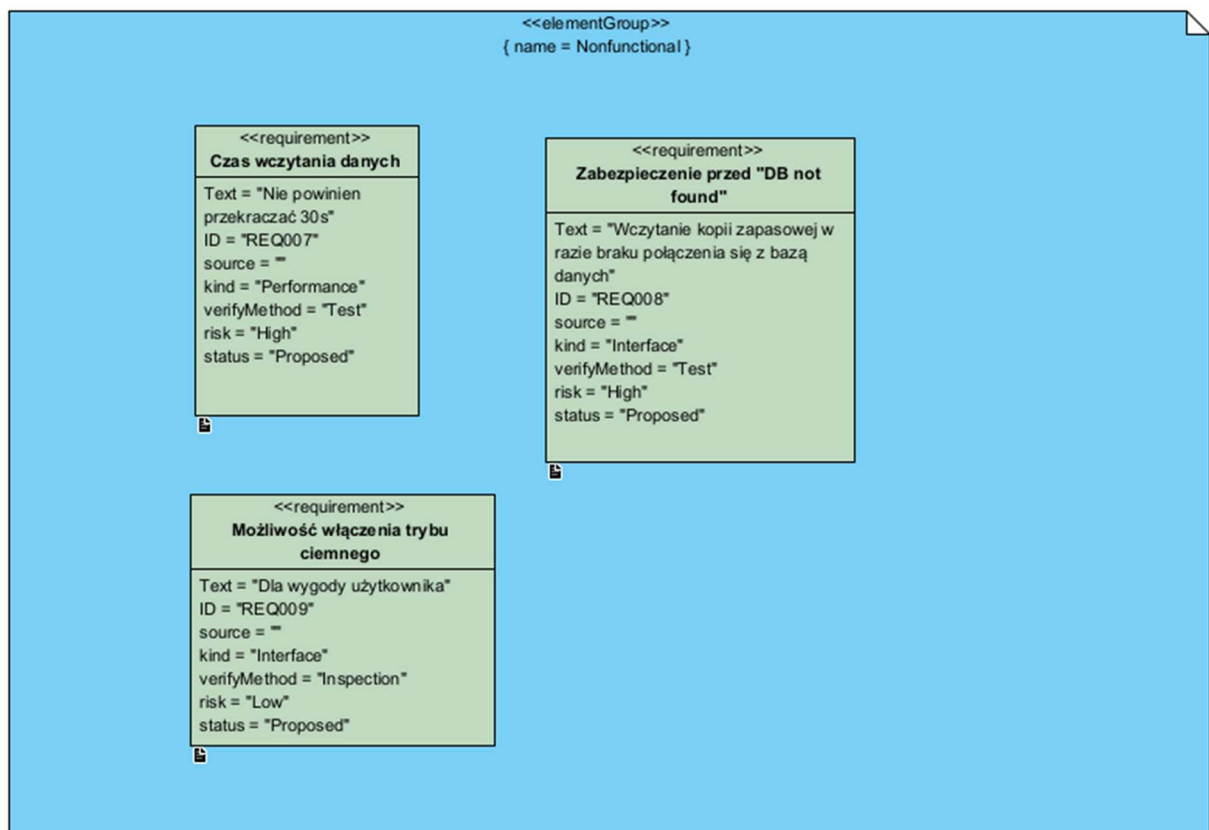
Administrator jest w stanie modyfikować dane użytkowników, dokonywać wpłat na ich konta inwestycyjne oraz usuwać ich konta.

Wymagania

Wymagania funkcjonalne



Wymagania niefunkcjonalne



Wykaz zastosowanych technologii

Visual Studio 2022

Jest to stosowane przeze mnie zintegrowane środowisko programistyczne(IDE). Jest to jeden z najbardziej popularnych programów, jak i prawdopodobnie najbardziej rozbudowany. Wybrałem je ze względu na łatwą obsługę nowych języków oraz dodatków do nich, możliwość zbudowania tymczasowych plików wykonywalnych i testowe odpalenie ich w aplikacji, generowanie kodu z szablonów językowych, oraz zawartą integrację Git, co ułatwiało obsługę różnych branch-y.

C#

Język ten skupia się na programowaniu obiektowym. Podobny do C++, jednakże wymagający znacznie bardziej rozbudowanej wiedzy i umiejętności w używaniu klas. Podstawa ASP.NET Core, który jest fundamentem projektu.

ASP.NET Core

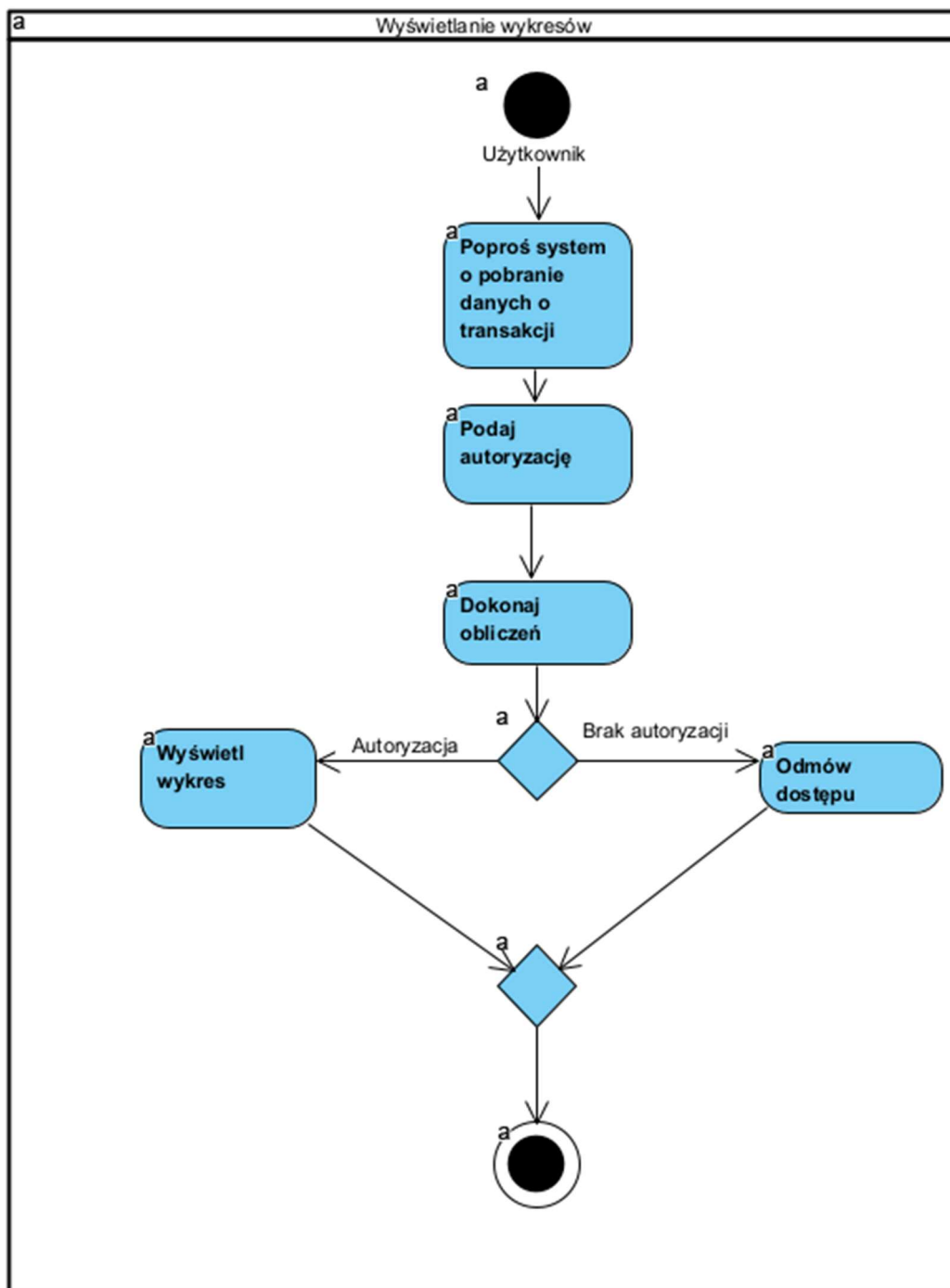
ASP.NET Core to międzyplatformowa, wysokowydajna platforma typu open source do tworzenia nowoczesnych, połączonych z Internetem aplikacji z obsługą chmury. Umożliwia zarówno używanie C#, jak i JavaScript. Okazała się niezastąpiona dla projektów podobnych do mojego, dzięki łatwości nauki oraz dużej ilości źródeł informacji.

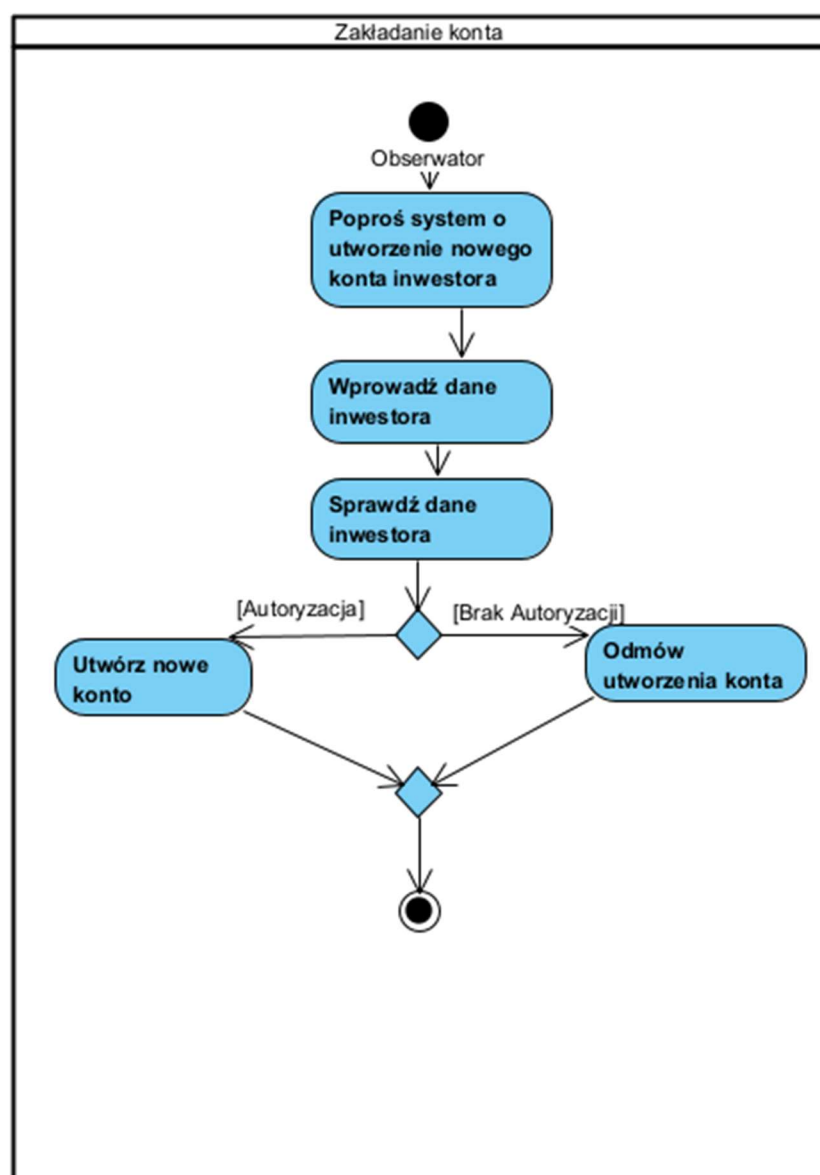
SyncFusion

Pakiet komponentów do modyfikacji UI, jak wykorzystane przeze mnie wykresy, menu i wiele innych.

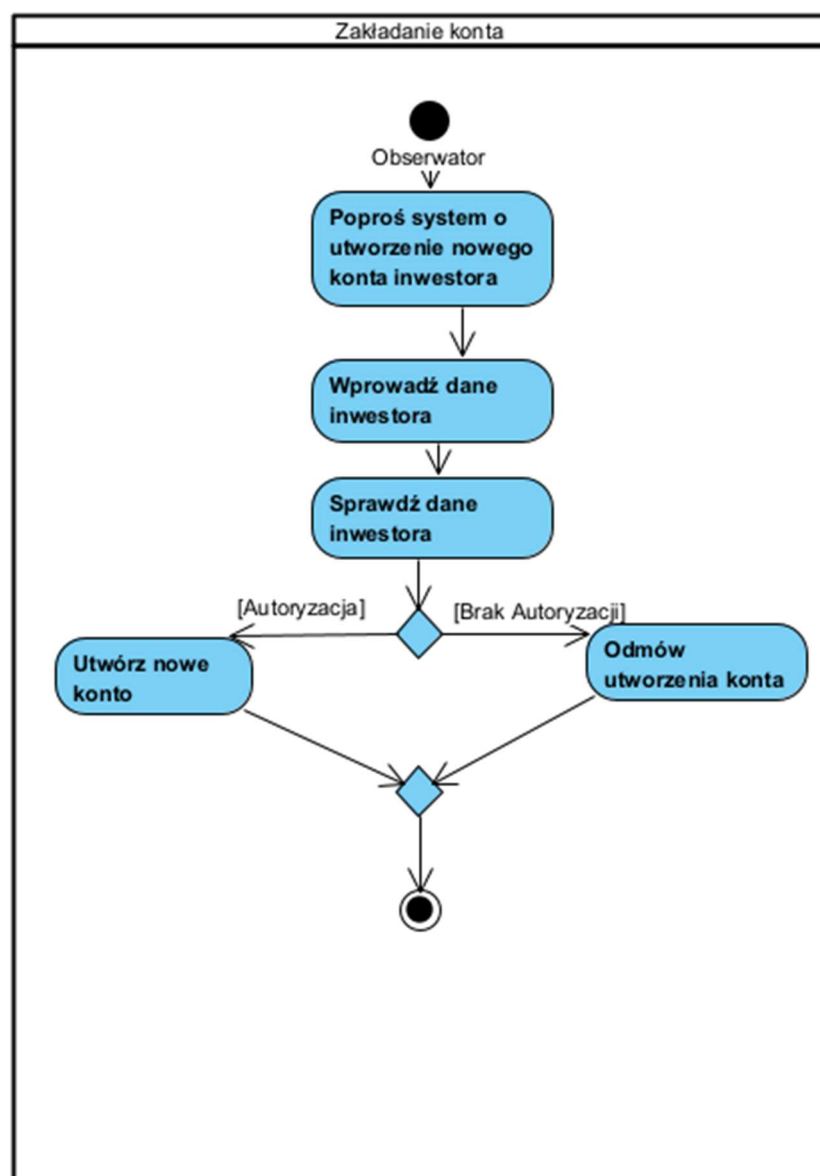
Diagramy aktywności

Wyświetlanie wykresów

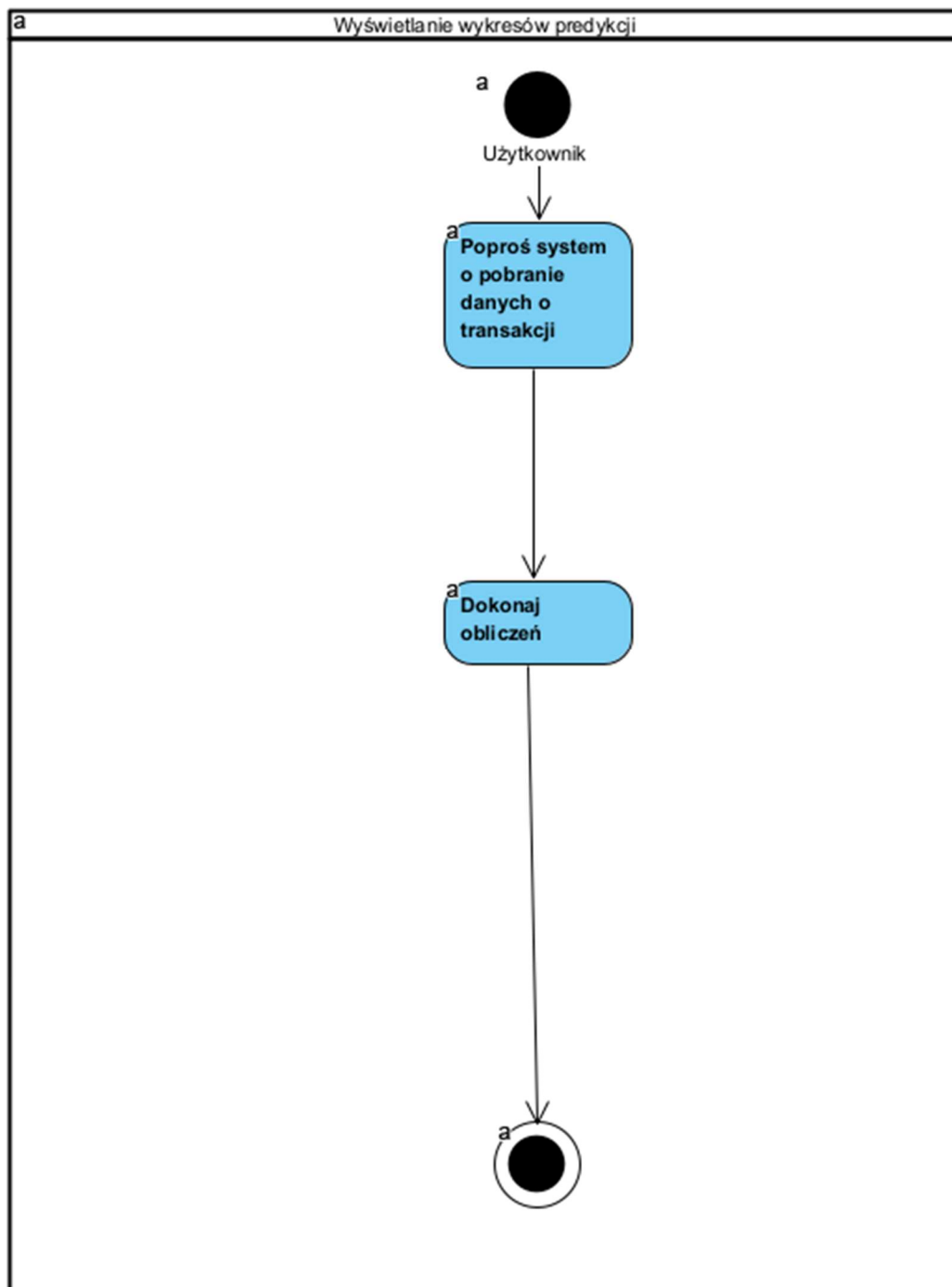




Zakładanie konta



Wyświetlanie wykresu z predykcją stopy zwrotu



Zapisanie transakcji

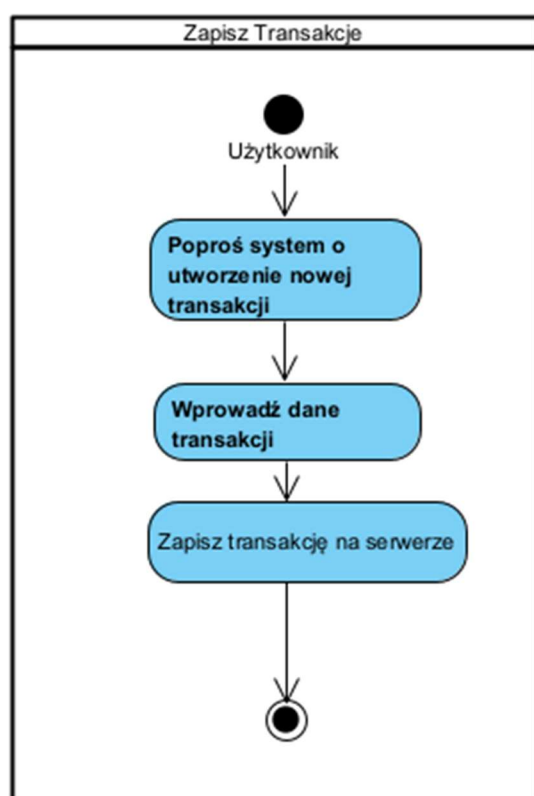
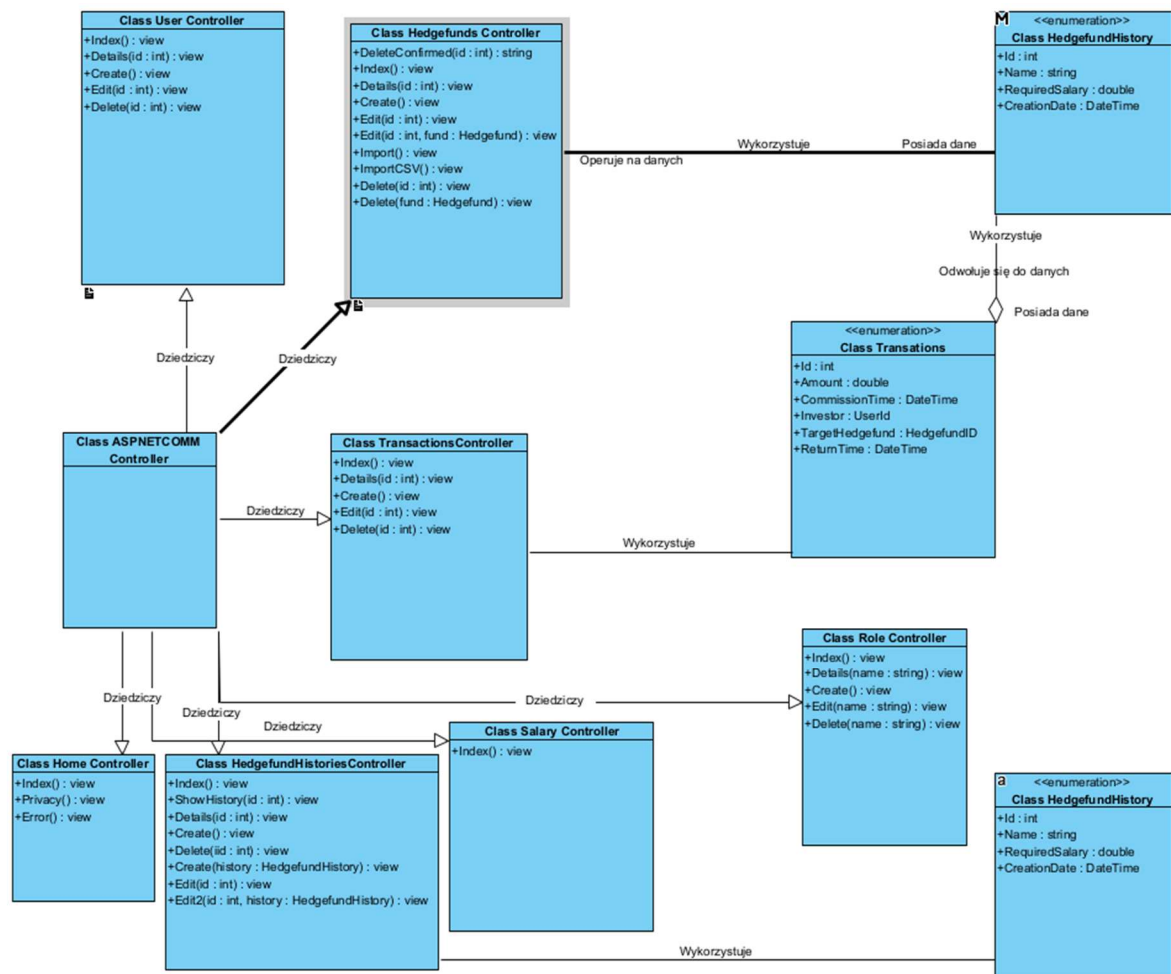
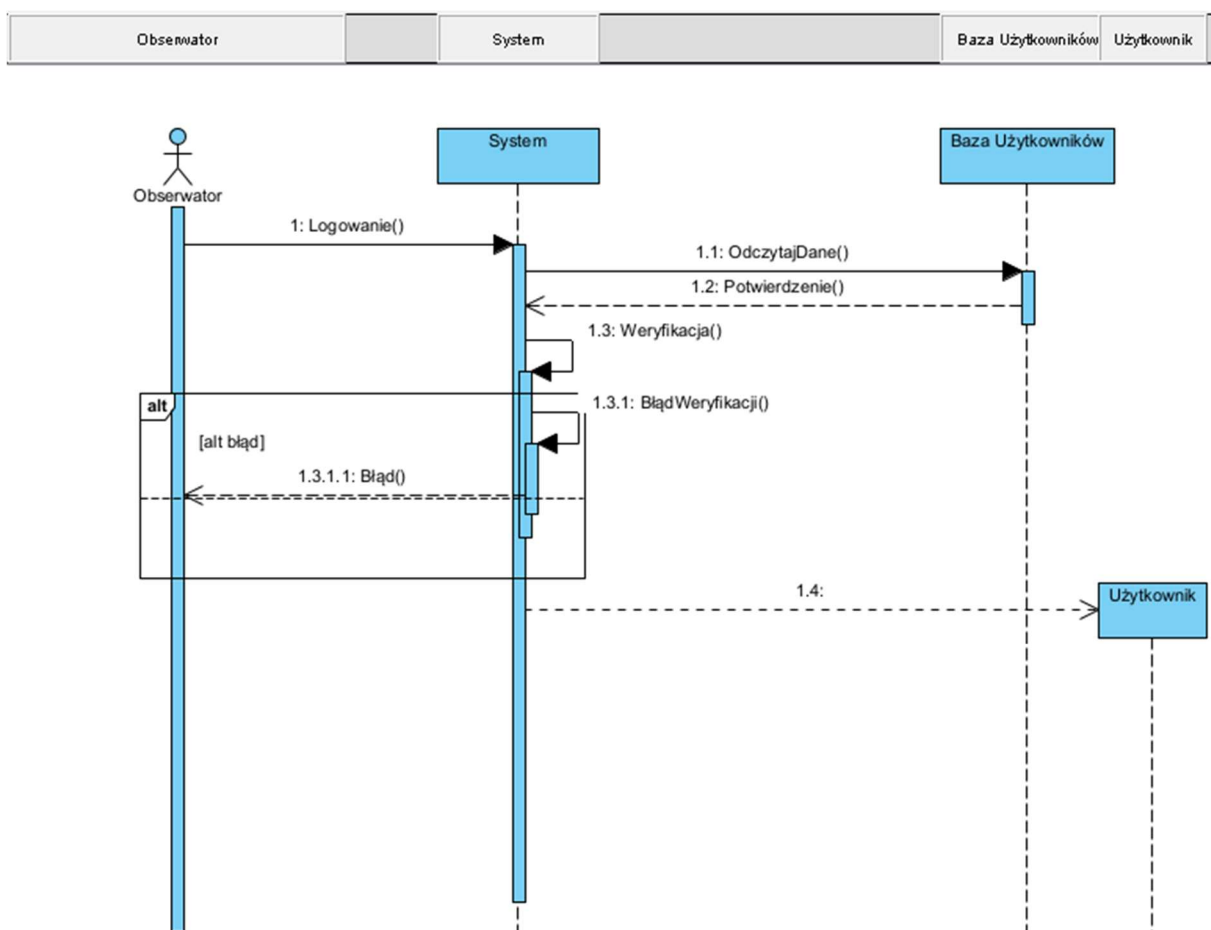


Diagram klas

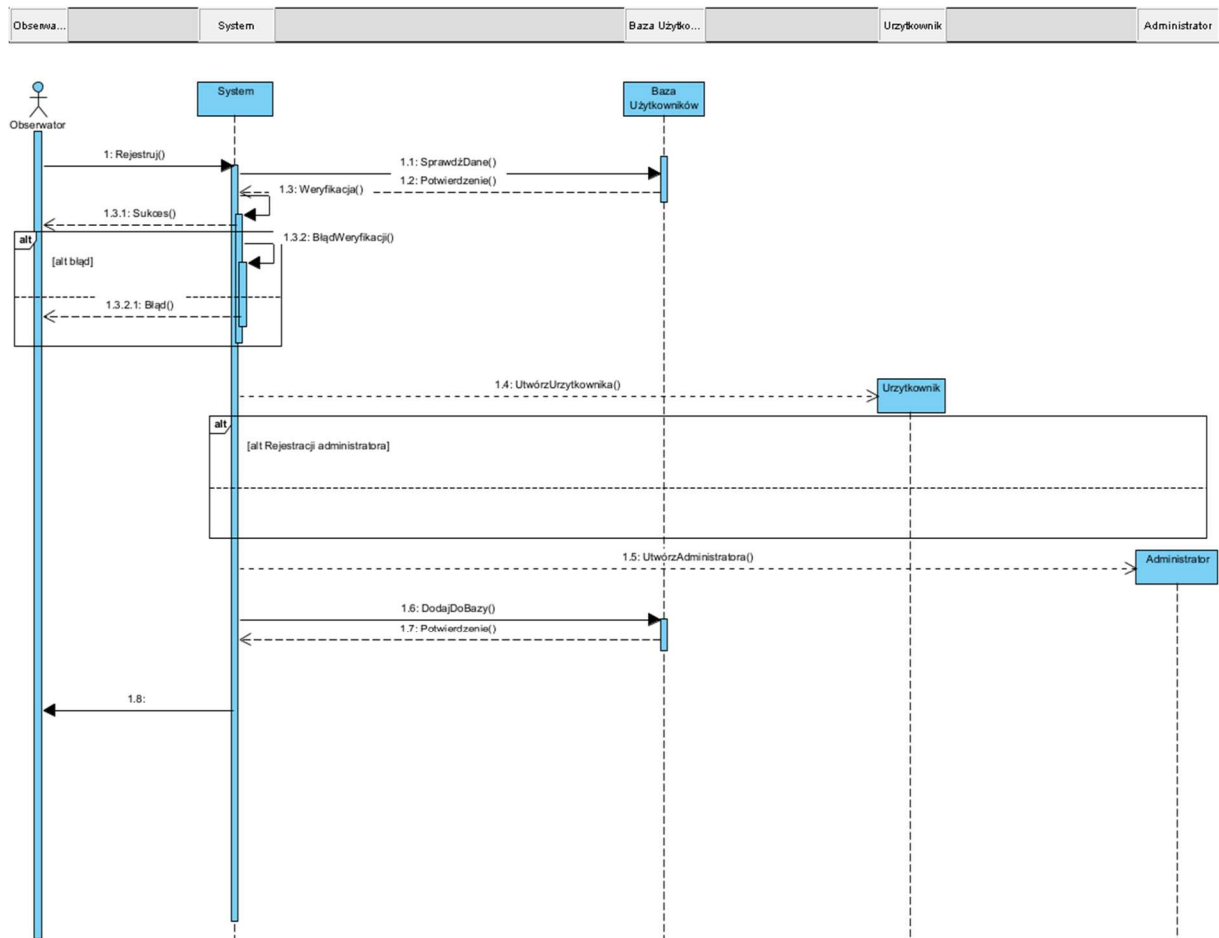


Diagramy sekwencji

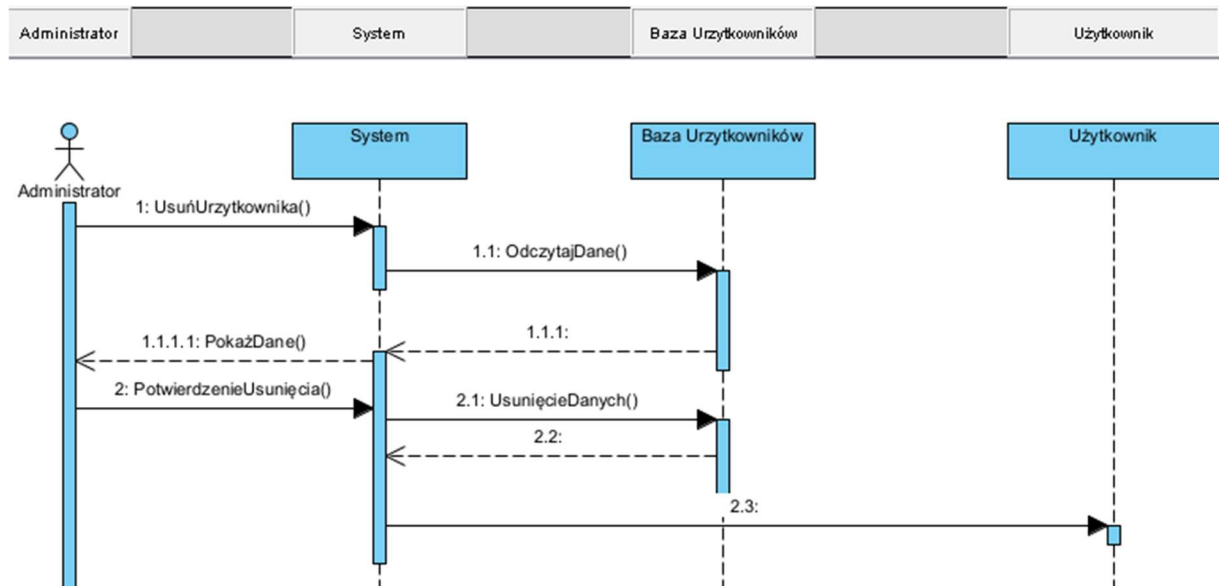
Logowanie



Tworzenie konta



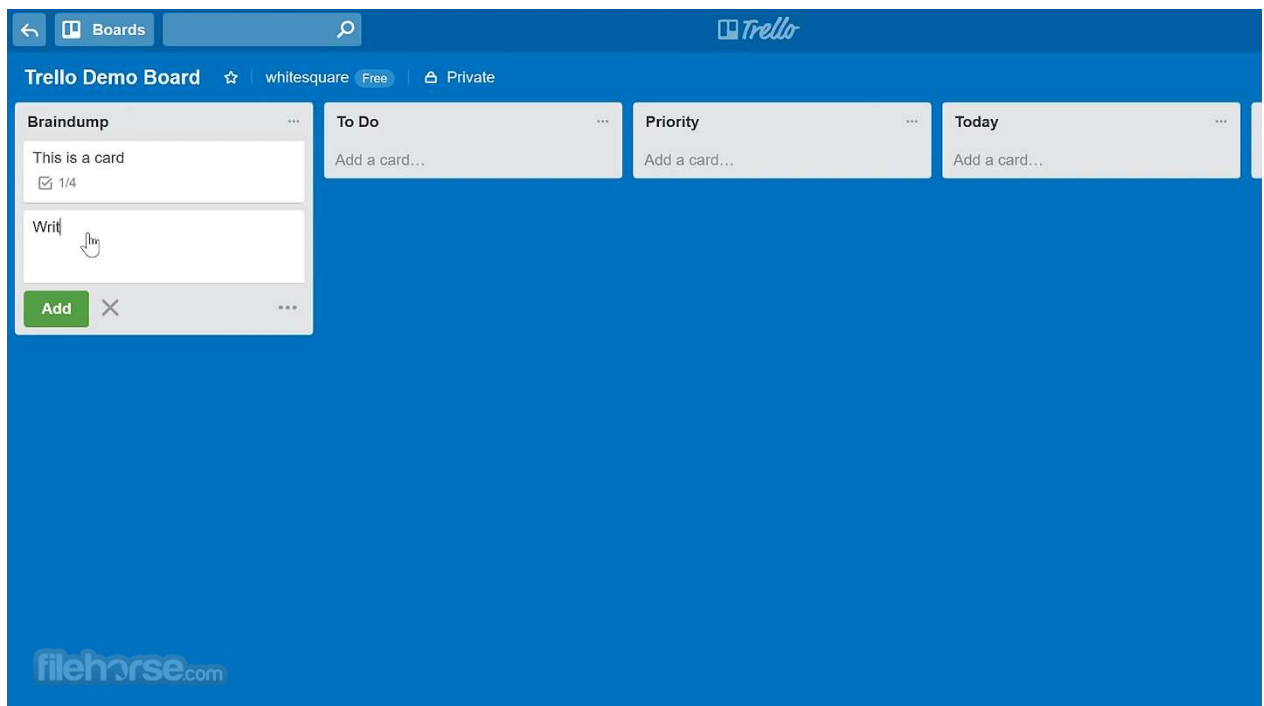
Usuwanie użytkownika



Przykłady współpracy

Trello

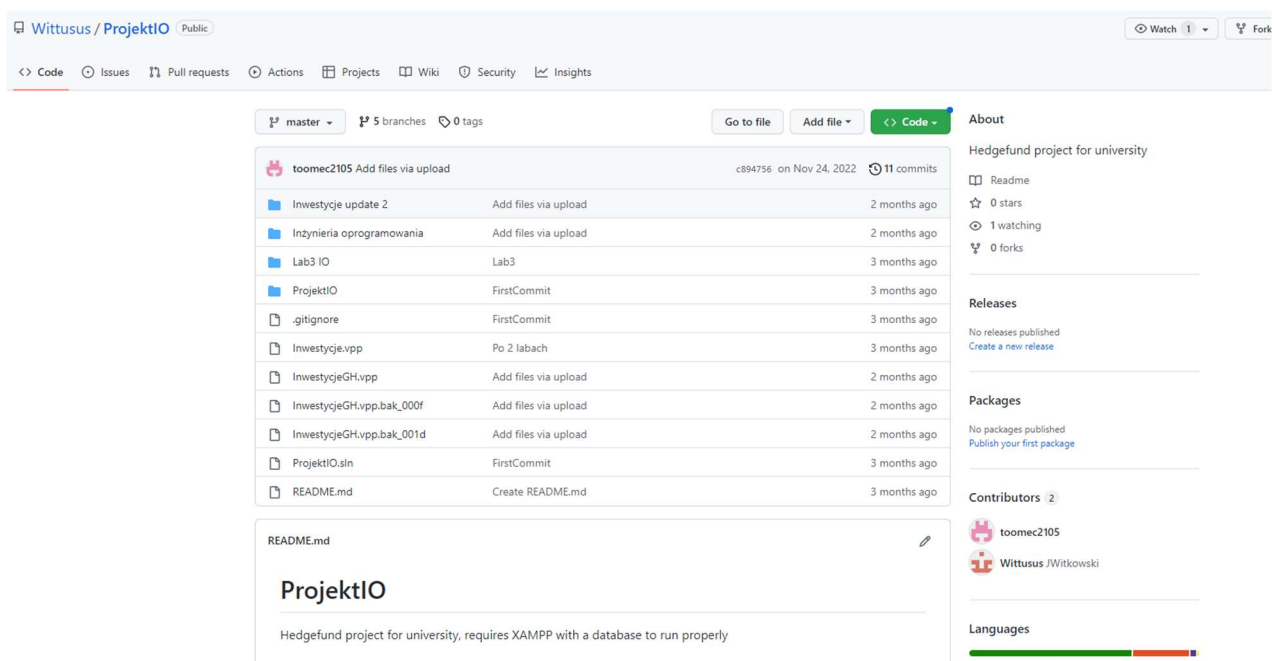
Podczas organizacji prac korzystaliśmy z aplikacji Trello. Dzięki niej mogliśmy łatwo uporządkować pozostałe do zrobienia funkcje, dzieląc zadania na wykonane, w trakcie realizacji oraz w planach albo zaplanowane.



Github

Jest to hostingowy serwis internetowy przeznaczony do projektów programistycznych wykorzystujących system kontroli wersji Git. Dzięki funkcji repozytoriów można łatwo udostępniać kod, co ułatwia pracę nad programem jeżeli tworzy go parę osób.

Dzięki temu narzędziu można szybko wrócić do poprzednich wersji kodu w wypadku, gdy nowe powodują ciężkie do naprawienia błędy.



Ten program oferuje nam również kontrolę wersji - zmiany przechodzą do branch'a main tylko wtedy kiedy zostaną zaakceptowane. Dzięki wizualizacji tego, co zostało zmienione, dodane i usunięte bardzo wyraźnie widać co zostało zrealizowane. Możemy też łatwo naprawiać błędy powstałe w wyniku pracy kilku programistów.

Testy manualne

Z powodu małego obycia się z językiem w trakcie pracy, jak i tego, że znaczna większość pracy była związana z tworzeniem bądź zmienianiem interfejsu użytkownika, testy były przeprowadzane manualnie przez nas, po utworzeniu środowiska tymczasowego. Testy funkcji backend były przeprowadzane za pomocą debuggera Visual Studio.

Refaktoryzacja kodu

Refaktoryzacja jest techniką polegającą na poprawie ogólnej jakości bądź przejrzystości kodu, który napisaliśmy wcześniej. Ułatwia ona dalsze rozwijanie go, jak dodawanie nowych funkcji. Była ona przeprowadzana za każdym razem, kiedy wykorzystywaliśmy poprzednio użyty fragment kodu, jak widać na przykładzie tego fragmentu, którego wykres zmienialiśmy na ten od SyncFusion:

15	41	<th>
16	-	@Html.DisplayNameFor(model => model.CreationTime)
42	+	Date
17	43	</th>
18	44	<th>
19	-	@Html.DisplayNameFor(model => model.CreationUser)
45	+	From
20	46	</th>
21	47	<th>
22	-	@Html.DisplayNameFor(model => model.TargetUser)
48	+	To
23	49	</th>
24	50	<th>
25	-	@Html.DisplayNameFor(model => model.TransactionType)
51	+	Type
26	52	</th>
27	53	<th>
28	-	@Html.DisplayNameFor(model => model.AmountOfTransaction)
54	+	Amount
29	55	</th>
30	56	<th>
31	-	@Html.DisplayNameFor(model => model.HedgeFund)
57	+	HedgeFund
32	58	</th>
33	59	<th>
34	-	@Html.DisplayNameFor(model => model.ReturnTimeOfInvestment)
60	+	Return time of investment

Oraz kiedy zmienialiśmy technikę porównywania wykresów:

```
61 -         chartData = chartData.OrderByDescending(x => x.xValue).ToList();
62 -         var viewModel = new HedgeFundHistoriesGetViewModel() { HedgefundHistories = histories, ChartDatas = chartData };
63 -         return View(viewModel);

69 +
70 +         return chartData;
71 +     }
72 +
73 +     //GET: Compare two charts
74 +     public async Task<IActionResult> CompareCharts()
75 +     {
76 +         var hedgefunds = await _context.Hedgefunds.ToListAsync();
77 +
78 +         return View(hedgefunds);
79 +     }
80 +
81 +     public async Task<IActionResult> CompareChartsDetails(long hedgefundOneId, long hedgefundTwoId)
82 +     {
83 +         var hedge1 = await _context.Hedgefunds.Where(x => x.Id == hedgefundOneId).FirstOrDefaultAsync();
84 +         var hedge2 = await _context.Hedgefunds.Where(x => x.Id == hedgefundTwoId).FirstOrDefaultAsync();
85 +         var historiesOne = await _context.HedgefundsHistory.Where(x => x.HedgefundId == hedgefundOneId).ToListAsync();
86 +         var historiesTwo = await _context.HedgefundsHistory.Where(x => x.HedgefundId == hedgefundTwoId).ToListAsync();
87 +         List<LineChartData> chartDataForOne = GetChartData(historiesOne).OrderByDescending(x => x.xValue).ToList();
88 +         List<LineChartData> chartDataForTwo = GetChartData(historiesTwo).OrderByDescending(x => x.xValue).ToList();
89 +
90 +         var model = new ChartDetailsViewModel()
91 +         {
92 +             HedgefundOneChartData = chartDataForOne,
93 +             HedgefundTwoChartData = chartDataForTwo,
94 +             HedgefundOneName = hedge1.Name,
95 +             HedgefundTwoName = hedge2.Name
96 +         };
97 +         return View(model);
```

Wzorce projektowe

W tym projekcie użyliśmy framework Microsoft .NET Core, który opiera się na języku C#. Dostarcza on gotowe wzorce, jak klasę Controller, która zawiera większość potrzebnych funkcji z typu CRUD(Create, Read, Update, Delete), jak i wyświetlanie danych. Zastosowanie go znacznie uprościło naszą pracę, dlatego też zdecydowaliśmy się na użycie go.

Drugim narzędziem tego typu było SyncFusion, które jest zbiorem komponentów interfejsu użytkownika. Pochodzą z niego wykresy, jak i tabele wyświetlane na poszczególnych stronach projektu.

Instrukcja użytkownika

Otwieranie aplikacji

1. Do utworzenia programu potrzebny jest program XAMPP oraz baza danych użytkowników.
2. Znajdź miejsce gdzie znajduje się projekt visual studio "ProjektIO.sln". Jeżeli nie wiesz gdzie się znajduje, skorzystaj z wyszukiwarki systemowej.
3. Kliknij dwukrotnie na projekt
4. Otwórz XAMPPa, i kliknij start koło modułów Apache oraz MySql
5. Kliknij Admin koło modułu MySql i zaimportuj bazę danych użytkowników, klikając tabelę projectdb oraz zaimportuj
6. Następnie w projekcie visual studio uruchamiamy projekt i aplikacja powinna otworzyć się w nowym oknie przeglądarki

Rejestracja

7. Po uruchomieniu aplikacji na ekranie pojawi się strona główna
8. Aby używać strony potrzeba się zarejestrować klikając "Register" w prawym górnym rogu
9. W pole "Email" wpisz swój Email.
 - a. Jeżeli Email jest już w użyciu, wyświetli się komunikat.
 - b. W takim wypadku wybierz inny Email.
10. W pole "Password" wpisz hasło, jakie chcesz by twoje konto posiadało.
11. Przepisz hasło z poprzedniego pola, upewniając się, że jest to to samo hasło.
12. Po uzupełnieniu obu pól wybierz opcję "Sign Up".
13. Jeżeli nie wyświetla się żaden komunikat o niepowodzeniu, to twoje konto zostało utworzone.

Logowanie

14. Po uruchomieniu aplikacji na ekranie w prawym górnym rogu pojawi się napis "Login", którym po kliknięciu zabierze nas do ekranu logowania
15. W pole "Email" wpisz swój Email.
 - a. Jeżeli wpiszesz Email, który nie został zarejestrowany, pojawi się komunikat o tym, że użyte dane logowania są niepoprawne
16. W pole "Password" wpisz hasło do swojego konta.
 - b. Jeżeli wpiszesz hasło, które nie pasuje do nazwy użytkownika, pojawi się komunikat o tym, że użyte dane logowania są niepoprawne
17. Po uzupełnieniu obu pól wybierz opcję "Log In"
18. Następnie kliknij w link by potwierdzić swój Email
19. Jeżeli wszystko poszło pomyślnie, zostaniesz przekierowany do menu głównego.

Przeglądanie funduszy

20. Aby przeglądać Fundusze wejdź w zakładkę "Hedge funds" na górze ekranu.
21. Wyświetli się lista funduszy
 - a. aby wyświetlić więcej informacji o danym funduszu kliknij "detalis" koło wybranego funduszu
 - b. aby obejrzeć historię, oraz predykcję zwrotu w czasie, kliknij "Show history" koło danego funduszu

Inwestowanie

22. Aby przejść do ekranu inwestowania w Fundusze wejdź w zakładkę "Transactions" na górze ekranu.
23. Wybierając "create new" tworzysz nową transakcję
24. Wybierając typ inwestycji, ilość oraz rodzaj funduszu tworzymy nową inwestycję.

Podsumowanie

Ogólne wrażenia po projekcie

Podczas realizacji tego projektu zdobyliśmy wiele umiejętności, które będą użyteczne w dalszej karierze. Łatwiej będzie nam tworzyć projekty wymagające użycia XAMPPa oraz ogólnie bazy danych.

Dzięki pisaniu dokumentacji zdobyliśmy dużo wiedzy o tym, jak powinna ona wyglądać. Diagramy pozwoliły w wizualny sposób zaprezentować działanie całego środowiska oraz to jak różne elementy ze sobą oddziałują.

Dodatkowo wzrosły nasze umiejętności pracy w wieloosobowym zespole, zarządzania pracą oraz planowania by zdążyć przed określonym terminem.

Opis prac

Na początku zdecydowaliśmy co powinien zawierać nasz projekt i jakie funkcje chcielibyśmy aby zawierał. Następnie zapisaliśmy jakie funkcjonalności będą potrzebne by projekt działał w pełni. Ostatnim etapem przygotowań było ustalenie co powinno się znaleźć w dokumentacji i jak powinna być zrobiona.

Następnie przeszliśmy do realizacji projektu realizując potrzebne feature-y jeden po drugim.

Część dokumentacji była tworzona równolegle z kodem, dzięki czemu łatwiej było zwizualizować działanie kodu oraz opisanie go. Kiedy osiągnęliśmy stan końcowy projektu dokończyliśmy dokumentację notując zmiany, których nie udokumentowaliśmy wcześniej.

Napotkane problemy

Projekt był dość wyczerpujący, ponieważ był to jeden z pierwszych większych projektów, które musieliśmy zrealizować pracując w grupie.

Tworzenie bazy danych użytkowników okazało się problematyczne, oraz doprowadzenie projektu w łady stan estetyczny okazało się bardzo czasochłonne.

Wystąpiło też dużo niedogodności jeśli chodzi o komunikację w zespole oraz podział pracy.

Z powodu użycia nieznanego nam języka, dopasowanego jednak dość dobrze do typu projektu, początkowe stadia tworzenia go były bardzo problematyczne, jak i nauka tworzenia kolejnych, niewykorzystywanych wcześniej funkcjonalności.

Funkcjonalności wyróżniające rozwiązanie

Nasz projekt wyróżniają wykresy, pokazujące przewidywalny kurs dla danego funduszu, który pomaga w wybraniu najlepszych inwestycji. Łatwy, intuitywny interfejs ułatwia korzystanie z niego. Ma on dodatkowo bardzo małe zużycie zasobów oraz bardzo proste byłoby przeniesienie go na zewnętrzny hosting, aby był dostępny gdziekolwiek, używając połączenia z siecią Internet.

