Data: 27.11.2017

**NTime**

**Aplikacja do przeliczania wyników**

**Web API**

**Dokumentacja Projektu**

**Autorzy:**

Grzegorz Krzysiak

Tomasz Chudzik

Spis treści

[Specyfikacja 5](#_Toc504086637)

[Opis biznesowy 5](#_Toc504086638)

[Wymagania funkcjonalne 5](#_Toc504086639)

[Wymagania niefunkcjonalne 6](#_Toc504086640)

[Harmonogram projektu 6](#_Toc504086641)

[Architektura aplikacji 6](#_Toc504086642)

[Dokumentacja końcowa (powykonawcza) 7](#_Toc504086643)

[Wymagania systemowe: 8](#_Toc504086644)

[Biblioteki wraz z określeniem licencji 8](#_Toc504086645)

[Instrukcja instalacji 8](#_Toc504086646)

[Instrukcja 9](#_Toc504086647)

[Instrukcja użycia 9](#_Toc504086648)

[Instrukcja utrzymania 11](#_Toc504086649)

[Raport odstępstw od specyfikacji wymagań 11](#_Toc504086650)

[Dokumentacja utworzonego Web API 11](#_Toc504086651)

[Założenia: 12](#_Toc504086652)

[Logowanie: 12](#_Toc504086653)

[POST Token 12](#_Toc504086654)

[Kontroler Account: 13](#_Toc504086655)

[POST api/Account/Logout 13](#_Toc504086656)

[POST api/Account/ChangePassword 13](#_Toc504086657)

[POST api/Account/Register 13](#_Toc504086658)

[GET api/Account/Role 13](#_Toc504086659)

[Kontroler Competition 14](#_Toc504086660)

[Model danych 14](#_Toc504086661)

[GET /api/Competition?ItemsOnPage=10&PageNumber=0 14](#_Toc504086662)

[GET /api/Competition/FromPlayerAccount/1?ItemsOnPage=10&PageNumber=0 14](#_Toc504086663)

[GET /api/Competition/1 14](#_Toc504086664)

[PUT /api/Competition/1 14](#_Toc504086665)

[POST /api/Competition 14](#_Toc504086666)

[POST /api/Competition/1/OrganizerLock/true 14](#_Toc504086667)

[POST /api/Competition/1/OrganizerLock/false 14](#_Toc504086668)

[Kontrolery AgeCategory, Distance, ExtraPlayerInfo 14](#_Toc504086669)

[Model danych AgeCategory 14](#_Toc504086670)

[Model danych Distance 15](#_Toc504086671)

[Model danych ExtraPlayerInfo 15](#_Toc504086672)

[GET api/<attr>/FromCompetition/1 15](#_Toc504086673)

[GET api/<attr>/1 15](#_Toc504086674)

[PUT api/<attr>/1 15](#_Toc504086675)

[POST api/<attr>/IntoCompetition/1 15](#_Toc504086676)

[DELETE api/<attr>/1 15](#_Toc504086677)

[Kontroler Player 15](#_Toc504086678)

[Model danych filtru 15](#_Toc504086679)

[Model danych do publicznej listy (tylko do wglądu) 16](#_Toc504086680)

[Model danych do zapisów 16](#_Toc504086681)

[Model danych pełny 16](#_Toc504086682)

[POST api/Player/TakeSimpleList/FromCompetition/1?ItemsOnPage=10&PageNumber=0 17](#_Toc504086683)

[POST api/Player/TakeFullList/FromCompetition/1?ItemsOnPage=10&PageNumber=0 17](#_Toc504086684)

[GET api/Player/TakeFullList/FromPlayerAccount/1?ItemsOnPage=10&PageNumber=0 17](#_Toc504086685)

[GET api/Player/FromPlayerAccount/1/FromCompetition/1 17](#_Toc504086686)

[GET api/Player/1 17](#_Toc504086687)

[PUT api/Player/1 17](#_Toc504086688)

[DELETE api/Player/1 17](#_Toc504086689)

[GET api/Player/Register/1 18](#_Toc504086690)

[PUT api/Player/Register/1 18](#_Toc504086691)

[POST api/Player/Register/IntoCompetition/1 18](#_Toc504086692)

[Kontroler PlayerAccount 18](#_Toc504086693)

[Model Danych 18](#_Toc504086694)

[GET api/PlayerAccount/Search/Filtr?ItemsOnPage=10&PageNumber=0 18](#_Toc504086695)

[GET api/PlayerAccount/Search?ItemsOnPage=10&PageNumber=0 18](#_Toc504086696)

[GET api/PlayerAccount 18](#_Toc504086697)

[GET api/PlayerAccount/1 18](#_Toc504086698)

[GET api/PlayerAccount/FromCompetition/1?ItemsOnPage=10&PageNumber=0 18](#_Toc504086699)

[PUT api/PlayerAccount/1 18](#_Toc504086700)

[Kontroler OrganizerAccount 19](#_Toc504086701)

[Model Danych 19](#_Toc504086702)

[GET api/OrganizerAccount/Search/Filtr?ItemsOnPage=10&PageNumber=0 19](#_Toc504086703)

[GET api/OrganizerAccount/Search?ItemsOnPage=10&PageNumber=0 19](#_Toc504086704)

[GET api/OrganizerAccount/ByCompetition/1 19](#_Toc504086705)

[GET api/OrganizerAccount/1 19](#_Toc504086706)

[PUT api/OrganizerAccount/1 19](#_Toc504086707)

[POST api/OrganizerAccount 19](#_Toc504086708)

[DELETE api/OrganizerAccount/1 19](#_Toc504086709)

[POST api/OrganizerAccount/1/PasswordReset 19](#_Toc504086710)

[POST api/OrganizerAccount/1/SetCompetition/1 19](#_Toc504086711)

[POST api/OrganizerAccount/1/UnsetCompetition/1 19](#_Toc504086712)

[Kontroler ModeratorAccount 19](#_Toc504086713)

[GET api/ModeratorAccount 19](#_Toc504086714)

[POST api/ModeratorAccount 19](#_Toc504086715)

[DELETE api/ModeratorAccount/aaa 20](#_Toc504086716)

[POST api/ModeratorAccount/aaa/PasswordReset 20](#_Toc504086717)

[POST api/ModeratorAccount/aaa/Bust 20](#_Toc504086718)

[POST api/ModeratorAccount/aaa/Unbust 20](#_Toc504086719)

[Kontroler AdministratorAccount 20](#_Toc504086720)

[GET api/AdministratorAccount 20](#_Toc504086721)

[POST api/AdministratorAccount 20](#_Toc504086722)

[DELETE api/AdministratorAccount/aaa 20](#_Toc504086723)

[POST api/AdministratorAccount/aaa/PasswordReset 20](#_Toc504086724)

[Dokumentacja końcowa (powykonawcza) - punkty wymagane przez prowadzącego zajęcia 20](#_Toc504086725)

[Model danych 21](#_Toc504086726)

# Specyfikacja

Opis biznesowy

Aplikacja ma służyć firmie przeprowadzającej pomiary czasu na zawodach sportowych. Dzięki tej aplikacji użytkownicy z różnych komputerów będą mogli w tym samym czasie zapisywać się na te same zawody. Ponadto każdemu zawodnikowi udostępnione będzie konto użytkowania, co pozwoli mu na edycję własnych danych na zawodach, na które już się zapisał.

## Wymagania funkcjonalne

1. Użytkownik mam możliwość zarejestrowania się i utworzenia swojego własnego konta z hasłem.
2. Po zalogowaniu się użytkownikowi wyświetla się zakładka „Moje konto”
3. W zakładce „Moje konto” użytkownik może definiować swoje szablonowe dane, które będą się samouzupełniały podczas zapisywanie się na zawody
4. Ponadto w zakładce „Moje konto” użytkownik ma możliwość edytowania swoich danych w zawodach, na które już się zapisał. Zawody wybierane są z listy rozwijanej.
5. W zakładce „Zapisy” użytkownik wybiera z listy rozwijanej zawody, których termin zapisów jeszcze nie minął
6. Użytkownik może zapisać się na zawody bez zakładania konta i logowania się.
7. Jeżeli jednak użytkownik się zaloguje, to jego konto przypisuje się do danego zapisu na zawody
8. Przy włączeniu zakładki zapisy dla zalogowanego użytkownika pobierają się szablonowe dane, które zdefiniował w zakładce „Moje konto”
9. W zakładce „Lista zawodników” każdy użytkownik (nawet niezalogowany) ma dostęp do wszystkich zawodów, z których jedne wybiera z listy rozwijanej
10. Po wybraniu konkretnych zawodów wyświetla się pierwsza strona wszystkich zawodników. Jeżeli zawody zostały już przeprowadzone, to przy zawodnikach pojawiają się uzyskane przez nich miejsca.
11. Lista zawodników jest stronicowana i na raz wyświetlanych jest maksymalnie 50 zawodników. Użytkownik naturalnie może wybierać stronę, które w danej chwili go interesuje
12. Zapewniona jest użytkownikowi również możliwość sortowania listy zawodników. Ponadto użytkownik może skorzystać z zaawansowanego filtru, by wedle swoich potrzeb wyfiltrować odpowiednie wyniki, np. dla swojego nazwiska, dystansu, czy zakresu numerów startowych
13. Wszystkie dane w zakładce „Lista zawodników” są wyświetlane w trybie „wyłącznie do odczytu”

## Wymagania niefunkcjonalne

Baza danych: Microsoft SQL Server Express

ORM: Entity Framework

Testy jednostkowe: NUnit

Interfejs graficzny: WPF

Technologia serwera REST API: .NET Web API

Technologia klienta REST API: .NET HttpClient

## Harmonogram projektu

25.11.2017 – zaplanowanie architektury aplikacji

30.11.2017 – stworzenie bazowych architektury do controllerów i klienta http

4.11.2017 – Budowanie podstawowego widoku aplikacji

**6.12.2017 – Prezentacja pierwszego etapu**

15.12.2017 – Większość działających konrollerów i pobieranie danych do widoku użytkownika

02.01.2018 – Ostateczny widok aplikacji

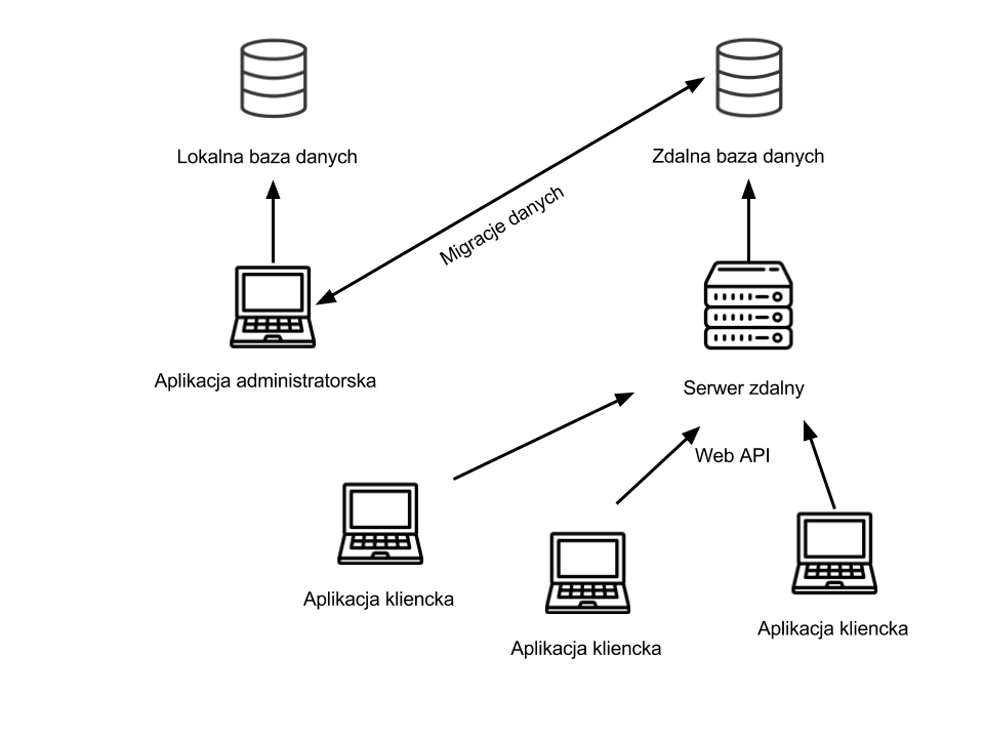
10.01.2018 – Testowanie i poprawianie mniejszych błędów aplikacji

15.01.2018 – Wdrożenie całości aplikacji na serwer

**20.01.2018 – Oddanie całości aplikacji**

## Architektura aplikacji

Na poniższym rysunku przedstawiono architekturę platformy NTime z rozbiciem na poszczególne komponenty



Rysunek 1 Architektura NTime

Na przedstawionym rysunku wyróżniamy następujące elementy:

* **Lokalna baza danych** - Local DB, będąca w pogotowiu, by uruchomić jedną z lokalnych, binarnych kopii bazy danych na wypadek utraty połączenia z siecią
* **Zdalna baza danych** – Główna baza danych znajdująca się na serwerze, w które znajdują się wszystkie dane odnośnie przeprowadzonych zawodów oraz kont użytkowników
* **Aplikacja administratorska** – Aplikacja stworzona w pierwszym etapie projektu pozwalająca administratorowi na edycję wszystkich danych dotyczących zawodów oraz na konfigurację i obliczenie wyników w dniu imprezy
* **Serwer zdalny** – Serwer, na którym znajduje się REST API, z którego korzystają wszystkie aplikacje klienckie i który łączy się ze zdalną bazą danych
* **Aplikacja kliencka** – Aplikacja w technologii WPF, pozwalająca w tym samy czasie różnym użytkownikom na zapisywanie się na zawody i edycję swoich danych

# Dokumentacja końcowa (powykonawcza)

## Wymagania systemowe:

Do instalacji systemy wymagany jest:

Aplikacja użytkownika:

System Operacyjny Windows 7 lub wyższej,

Platforma .net Framework 4.7,

Dostostęp do internetu

Server:

System Opercayjny Windows Server 2012,

Baza danych MS SQL 2012,

IIS 8.0

## Biblioteki wraz z określeniem licencji

|  |  |  |  |
| --- | --- | --- | --- |
| nr | Komponent i wersja | Opis | Licencja |
| 1 | .Net Framework 4.7 | Platforma programistyczna opracowana przez Microsoft, obejmująca środowisko uruchomieniowe (Common Language Runtime – CLR) oraz biblioteki klas dostarczające standardowej funkcjonalności dla aplikacji. | Freeware |
| 2 | Entity Framework 7 | Narzędzie typu ORM (Object Relational Mapping), pozwalającym odwzorować relacyjną bazę danych za pomocą architektury obiektowej. | Apache License |
| 3 | NUnit 3 | Platforma do testów jednostkowych | MIT License for 3.0, BSD-style (modified zlib license) for 2.x |
| 4 | CsvHelper | Biblioteka do wczytania i zapisywania plików typu csv. | Apache License, Version 2.0 |
| 5 | ASP.NET | Zbiór technologii opartych na frameworku zaprojektowanym przez firmę Microsoft. Przeznaczony jest do budowy różnorodnych aplikacji internetowych, a także aplikacji typu XML Web Services. | Apache License |

## Instrukcja instalacji

Aplikacja użytkownika:

Produkt nie wymaga instalacji. Potrzebne jest jedynie spełnienie wymagań opisanych w sekcji Wymagania systemowe

Server:

Wymagane jest wdrożenie aplikacji na serwer IIS. Następnie należy skonfigurować connection stringi o nazwach DefaultConnection i NTime.

## Instrukcja

Aplikacja użytkownika:

Należy uruchomić plik exe.

Server:

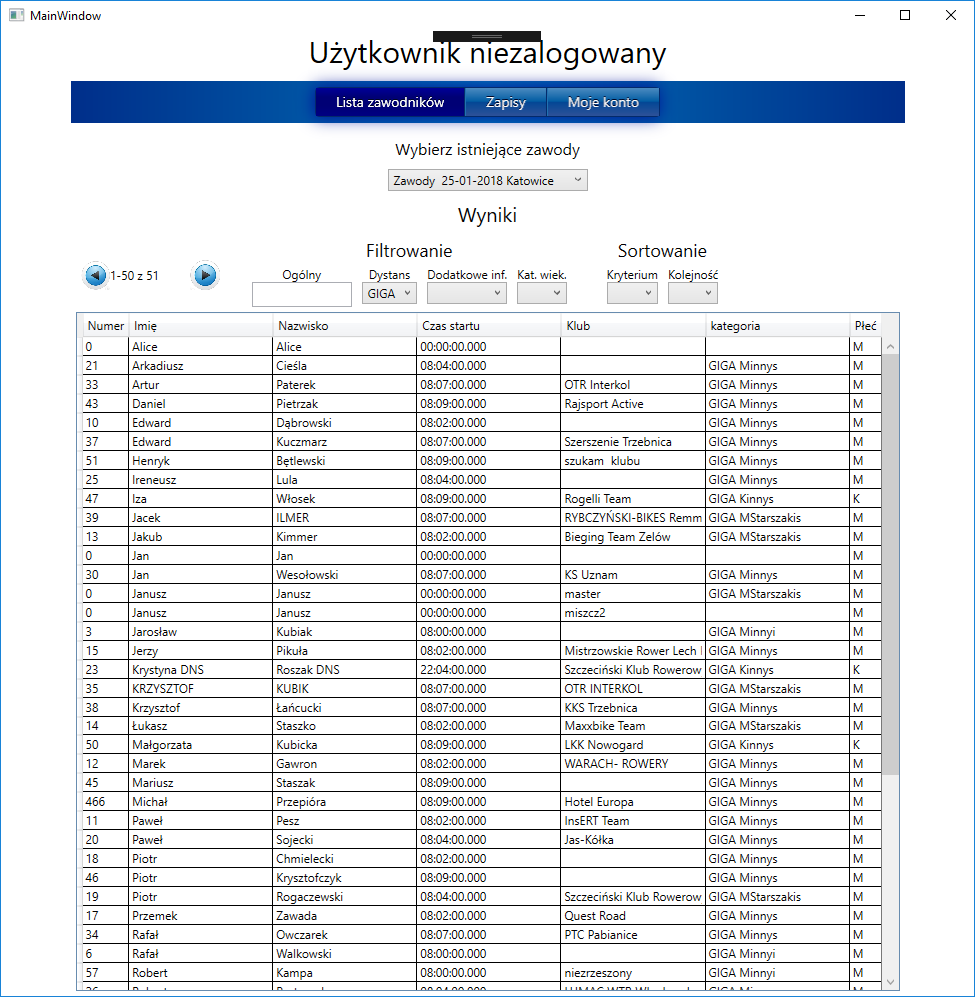
Należy uruchomić aplikacje na serwerze IIS.

## Instrukcja użycia

Aplikacja użytkownika składa się z 3 zakładek:

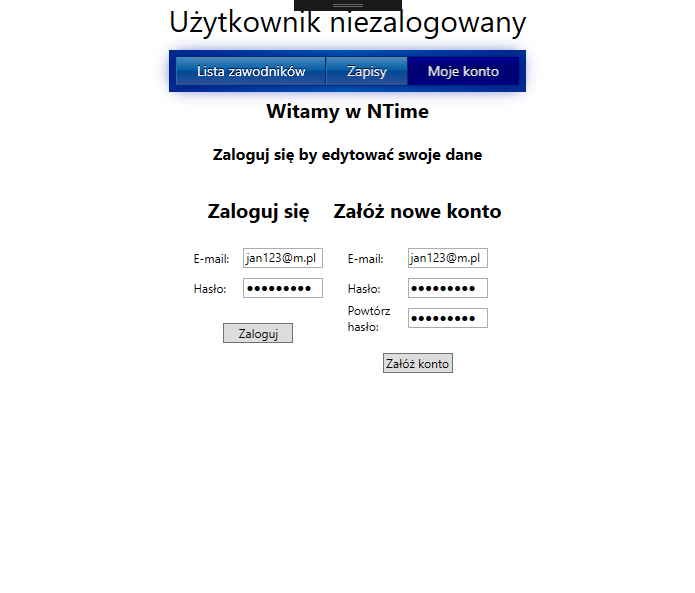
1. Zakładka Lista zawodników

Pozwala na wgląd w dane osób które są już zapisane na dane zawody wybrane za pośrednictwem listy rozwijanej. Wyniki można filtrować po nazwie i po danych składowych pełnej kategorii (dystans, dodatkowe info, kategoria wiekowa). Jest możliwość sortowania wyników przy pomocy odpowiednich opcji w listach rozwijanych.

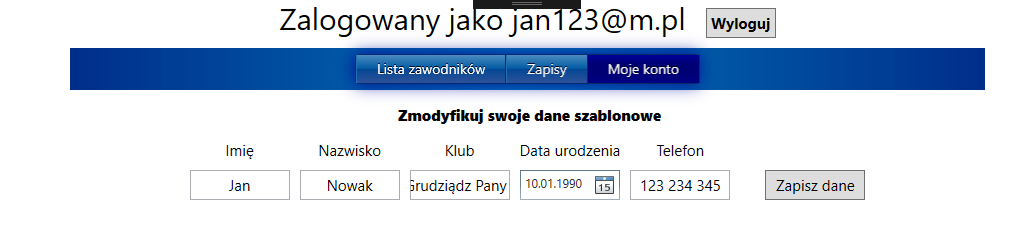


1. Zakładka Moje konto

Będąc niezalogowanym możemy dokonać rejestracji nowego konta podając adres email, hasło i powtórzone hasło, a następnie klikając przycisk załóż konto. Posiadając konto możemy zalogować się na nie podając dane od formularza "Zaloguj się" i klikając przycisk zaloguj.

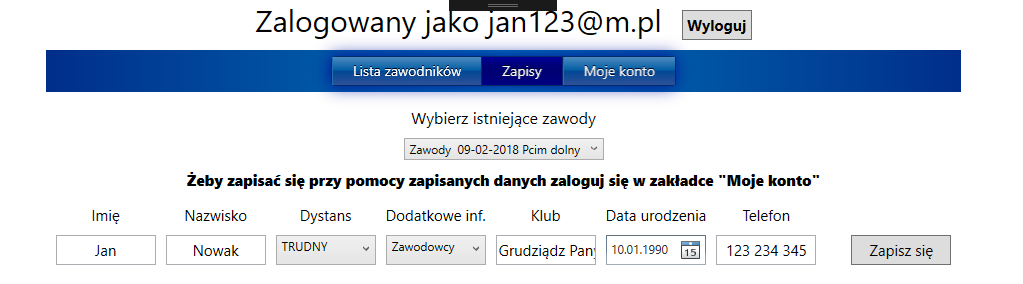


Po zalogowaniu mamy możliwość edycji danych osobowych przypisanych do naszego konta.



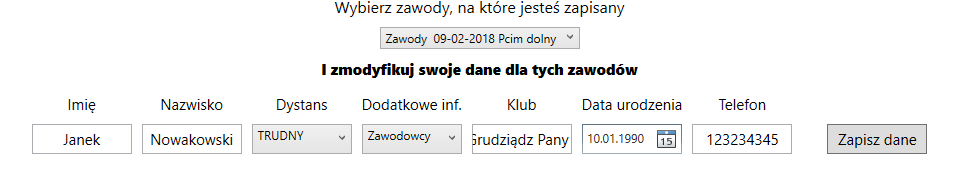
1. Zakładka Zapisy

Po przejściu na zakładki Zapisy będąc zalogowanym, formularz będzie wypełniony danymi osobowymi naszego konta. Wystarczy wybrać dystans i dodatkowe informacje, a następnie kliknąć przycisk zapisz się, żeby dokonać zapisu na zawody. Zawody, na które można się zapisać wybieramy przy pomocy listy rozwijanej. Zapisu można także dokonać będąc niezalogowanym.



1. Edycja zapisanych danych

W przypadku, kiedy chcielibyśmy zmienić dane jakiegoś zapisu na zawody możemy tego dokonać za pośrednictwem zakładki moje konto. Należy wybrać odpowiednie zawody z listy rozwijanej i edytować formularz.



1. Wylogowanie

Po zakończeniu pracy można wylogować się z konta.



## Instrukcja utrzymania

Dane zawodów są przechowywane w bazie danych MS SQL. Do archiwizacji danych można wykorzystać wbudowane mechanizmy tego systemu.

## Raport odstępstw od specyfikacji wymagań

W trakcie realizacji projektu wprowadzonych zostało kilka zmian, które powodują, że część funkcjonalności odbiega od tych zawartych w specyfikacji wstępnej. Zmiany te powstawały podczas bieżących konsultacji z klientem. Często były to sugestie od zamawiającego, który zmieniał swoje wymagania. Niemniej jednak nie były to znaczące modyfikacje. Zdarzało się również, że na wniosek autorów projektu, niektóre zmiany funkcjonalności były zgłaszane klientowi i po jego akceptacji uwzględniane podczas wykonywania projektu. Odstępstwa od specyfikacji wymagań zostało zaprezentowane poniżej:

* Poza samymi controllerami do obsługi kont użytkowników oraz do zapisów powstały również controllery obsługujące konta administratora, organizatora oraz moderatora. Dzięki nim istnieje możliwość stworzenia aplikacji webowej, która pozwoli wykonać część takich samych czynności jak w aplikacji desktopowej stworzonej w pierwszym etapie. Funkcjonalności te nie będą raczej realizowane nawet w trakcie trzeciego etapu, niemniej jednak zostaną one wykorzystane podczas tworzenia ostatecznego produktu dla klienta
* W porównaniu ze specyfikacją wstępną zdecydowano, że podczas zapisów to zawodnik sam decyduje na jakim dystansie będzie odbywał zawody i która z dodatkowych informacji będzie dla niego odpowiednia. Możliwość dokonania tego wyboru wprowadzono podczas zapisów i podczas edycji danych dla zawodów, na które zawodnik już się zapisał

## Dokumentacja utworzonego Web API

Strona: <http://projektnet.mini.pw.edu.pl/NTime>

Administrator:

Login/Email: admin@ntime.pl

Hasło: admin1

## Założenia:

* Są 4 role:
  + Player - P
  + Organizer - O
  + Moderator - M
  + BustModerator – Moderator z odebranymi uprawnieniami
  + Administrator - A
* Organizator ma dostęp do jakiegoś zasobu związanego z zawodami tylko wtedy, gdy jest powiązany z danymi zawodami i może edytować je jak flaga w zawodach "ModeratorEditLock" jest na false
* Player może zapisywać się na zawody i modyfikowac owy wpis o ile operacje te nie są wykonywane po "SignUpEndDate" w obiekcie zawodów
* Player może mieć pełny wgląd tylko na własne wyniki
* Jeśli w ścieżce jest napisane "1" to oznacza, że trzeba tam podać id do danego zasobu (liczba dodatnia, całkowita), chyba że jest poprzedzony przez "/FromCompetition/", "/IntoCompetition/", "/ByCompetition/", "/SetCompetition/" lub "/UnsetCompetition/" wtedy należy podać id do zawodów, a jeśli "/FromPlayerAccount/" to podajemy id do konta zawodnika.
* W przypadku kontrolerów "AdministratorAccount", "ModeratorAccount" id to ciąg znaków
* Api obsługuje Json o ile Content-Type jest nie wskazuje inaczej
* Jeśli adres kończy się na "?ItemsOnPage=10&PageNumber=0" to oznacza, że wynik jest stronicowany, w odpowiedzi dostanie się całkowitą liczbę obiektów i tablicę przeznaczoną do wyświetlenia na wskazanej stronie (Uwaga: strony liczymy od zera). Zwracana jest struktura z polem TotalCount – int i Items – tablica.
* Hasła powinny zawierać co najmniej 8 znaków w tym co najmniej jedną cyfrę i mały znak
* Można ignorować pole id przy tworzeniu nowych zasobów, ale nie można go pominąć, może być dowolne a najlepiej –1
* Jeśli przy polu w wypisanej strukturze danych jest "\*" to pole nie może przyjąć null-a, więc nie może być pominięte
* 404 jeśli id-ki się nie zgadzają
* Czas liczymy w sekundach jako zmienna typu decimal (oznaczone przy niektórych zmiennych)

## Logowanie:

### POST Token

Mogą: wszyscy

Content-Type: application/x-www-form-urlencoded

Body: grant\_type=password&username=admin&password=admin1

Przykładowa odpowiedź:

{

"access\_token": "BMxniHiUjhexqWm1RwexXkAK9Pq70n4-QZNdjRgle1gL15a1mrGLM3lHhjq0DOCaWsGh0T2Nr9KWqStfIQuPxD8iYDDYDKzRbU2Rv\_sXPeEbBDPpvgAVsOnL9aiWZsG2ve811uGBiH0l0Nfko50uQDQcMAuHAbBQjXuwLChO-up5DTLgrHozIlc8han4U-t1nDWeiVd1vPqPcaJi6TvFm6BfNGjZhrm8G2xAoWF2C60894Qjs-VqWJlER9eSBLk6B6TfPFHrEpwHixf5olC2DQjpv1Jd-3PFNu2RQTnlbqMCl5pjJXasYGuzPGWW5n8J58D-tC7lxOYXqIKZTHfGKzAE2oYsN8kMxmC0QCc0tqgnBJvbGiJ-w0t8JktmicFKpJR0zUlC3YLQryPx-HvcvLnaIwXpenVATYosSDoCmvfT6kpy4Sg34qQYYw7dKf8RvJraxbnlqYmJqrZ6L9uD9oQ6CHHAj7Khsfah6KXSSexWZvL4S3vE20-TEIfMiC09UaSzANhDaYXIAMB7\_gJo0g",

"token\_type": "bearer",

"expires\_in": 1209599,

"userName": "admin",

".issued": "Thu, 28 Dec 2017 20:42:00 GMT",

".expires": "Thu, 11 Jan 2018 20:42:00 GMT"

}

W każdym następnym żądaniu należy umieścić w nagłówku:

Authorization: Bearer BMxniHiUjhexqWm1Rwe....

## Kontroler Account:

### POST api/Account/Logout

Mogą: wszyscy zalogowani

### POST api/Account/ChangePassword

Mogą: wszyscy zalogowani

Body:

* \*OldPassword
* \*NewPassword
* \*ConfirmPassword

### POST api/Account/Register

Mogą: wszyscy

Body:

* \*Email
* \*Password
* \*ConfirmPassword

### GET api/Account/Role

Mogą: wszyscy zalogowani

Zwraca id konta (string), email i przypisaną role

## Kontroler Competition

### Model danych

* \*Id
* \*Name
* \*EventDate
* SignUpEndDate - jeśli null to nie ma deadline-u na zapisy
* Description
* Link
* Organizer - string
* City
* \*OrganizerEditLock – nie da się zmodyfikować to poprzez PUT i POST (przy tworzeniu), po stworzeniu nowych zawodów ustawiony jest na false

### GET /api/Competition?ItemsOnPage=10&PageNumber=0

Mogą: wszyscy

Pobiera stronicowaną listę zawodów sortowaną po czasie rozpoczęcia

### GET /api/Competition/FromPlayerAccount/1?ItemsOnPage=10&PageNumber=0

Mogą: A, M, P

Pobiera stronicowaną listę zawodów, na które dany użytkownik jest zapisany.

### GET /api/Competition/1

Mogą: wszyscy

Pobiera dane jednych zawodów

### PUT /api/Competition/1

Mogą: A, O

Modyfikuje zawody

### POST /api/Competition

Mogą: A

Tworzy nowe zawody

### POST /api/Competition/1/OrganizerLock/true

### POST /api/Competition/1/OrganizerLock/false

Mogą: A

Nic nie trzeba podawać nic w body

Ustawia flagę OrganizerEditLock

## Kontrolery AgeCategory, Distance, ExtraPlayerInfo

W miejsce "<attr>" wstaw AgeCategory, Distance lub ExtraPlayerInfo

### Model danych AgeCategory

* \*Id
* \*Name
* \*YearFrom
* \*YearTo

### Model danych Distance

* \*Id
* \*Name
* \*Length
* \*DistanceTypeId – int: może przyjąć
  + 0 – na dystans
  + 1 – na okrążenia - wtedy LapsCunt jest istotne
  + 2 – na czas w pętli - wtedy TimeLimit jest istotne
* \*LapsCount
* \*TimeLimit - decimal

### Model danych ExtraPlayerInfo

* \*Id
* \*Name
* \*ShortName

### GET api/<attr>/FromCompetition/1

Mogą: wszyscy

Wylistuje dane z danych zawodów

### GET api/<attr>/1

Mogą: wszyscy

Pobiera obiekt o podanym id

### PUT api/<attr>/1

Mogą: A, O

Modyfikuje obiekt o podanym id

### POST api/<attr>/IntoCompetition/1

Mogą: A, O

Tworzy nowy obiekt w danych zawodach

### DELETE api/<attr>/1

Usuwa obiekt o podanym id

Mogą: A

## Kontroler Player

Dotyczy "wpisu" na konkretne zawody i ich ewentualne wyniki, a nie konta zawodników

### Model danych filtru

Jeśli jest nullem to filtr nie jest brany pod uwagę

* \*PlayerSort - int
  + 0 - imię
  + 1 – nazwisko
  + 2 – klub
  + 3 – numer startowy
  + 4 – czas startu
  + 5 – kategoria
  + 6 – data urodzenia
* \*DescendingSort - bool
* Query – string, zaawansowane wyszukiwanie po napisie
  + "30 - 50" wybierze numery startowe z przedziału
  + "30, 40, 45, 50" wybierze podane numery startowe
  + "jan kow MINI" wybierze wpisy, w których zawierają się podane po spacji selektory np. Jan Kowalski z wystansu MINI i Jan Kowalik z wystansu MINI+
* Men – bool
* WithoutStartTime - bool
* Invalid - bool
* CompleatedCompetition - bool
* HasVoid - bool
* Distance – id
* AgeCategory – id
* ExtraPlayerInfo – id

### Model danych do publicznej listy (tylko do wglądu)

* \*Id
* \*FirstName
* \*LastName
* \*IsMale - bool
* Team - string
* \*StartNumber - int
* StartTime – decimal (liczone od godziny 00:00)
* FullCategory - string

### Model danych do zapisów

* \*Id
* \*FirstName
* \*LastName
* \*BirthDate
* \*IsMale
* Team
* \*ExtraPlayerInfoId - id
* \*DistanceId - id
* \*CompetitionId -id

### Model danych pełny

* \*Id
* \*FirstName
* \*LastName
* \*BirthDate
* \*IsMale
* Team
* \*StartNumber
* StartTime – decimal (liczone od godziny 00:00)
* \*IsStartTimeFromReader - być może zostanie usunięte
* FullCategory – aktualizuje się automatycznie
* \*LapsCount
* \*Time - decimal
* \*DistancePlaceNumber
* \*CategoryPlaceNumber
* \*CompetitionCompleted - bool
* ExtraPlayerInfoId - id
* DistanceId - id
* CompetitionId -i d
* PlayerAccountId - id

### POST api/Player/TakeSimpleList/FromCompetition/1?ItemsOnPage=10&PageNumber=0

Mogą: wszyscy

W body podać obiekt filtru

Zwraca dane publicznej listy

### POST api/Player/TakeFullList/FromCompetition/1?ItemsOnPage=10&PageNumber=0

Mogą: A, O, M

W body podać obiekt filtru

Zwraca pełne dane

### GET api/Player/TakeFullList/FromPlayerAccount/1?ItemsOnPage=10&PageNumber=0

Mogą: A, M, P

Zwraca zapisy (pełne dane) z odbytych zawodów i zawodów na które się zapisało podanego zawodnika

### GET api/Player/FromPlayerAccount/1/FromCompetition/1

Mogą: A, M, P

Zwraca pełne dane zarejestrowane użytkownika dla wybranych zawodów

### GET api/Player/1

Mogą: A, O, M, P

Zwraca pełne dane o podanym id

### PUT api/Player/1

Mogą: A, O, M

Modyfikuje pełne dane

### DELETE api/Player/1

Mogą: A, O, M, P

Kasuje zapis

### GET api/Player/Register/1

Mogą: A, O, M, P

Pobiera zapis w o podanym id

### PUT api/Player/Register/1

Mogą: P

Modyfikuje zapis

### POST api/Player/Register/IntoCompetition/1

Mogą: wszyscy

Zapisuje się na dane zawody

## Kontroler PlayerAccount

### Model Danych

* \*Id
* \*FirstName
* \*LastName
* \*BirthDate
* \*IsMale
* Team
* PhoneNumber
* \*EMail

### GET api/PlayerAccount/Search/Filtr?ItemsOnPage=10&PageNumber=0

### GET api/PlayerAccount/Search?ItemsOnPage=10&PageNumber=0

Może: A

Zwraca stronicowaną listę kont zawodników z lub bez filtru ("Filtr") który jest stringiem

### GET api/PlayerAccount

Może: P

Zwraca dane aktualnie zalogowanego zawodnika

### GET api/PlayerAccount/1

Może: A, P

Zwraca dane zawodnika o podanym id

### GET api/PlayerAccount/FromCompetition/1?ItemsOnPage=10&PageNumber=0

Może: A

Zwraca listę kont zawodników z których zapisano się na konkretne zawody

### PUT api/PlayerAccount/1

Może: A, P

Edycja

## Kontroler OrganizerAccount

Wszystkie operacje może tylko administrator

### Model Danych

* \*Id
* \*FirstName
* \*LastName
* \*PhoneNumber - string
* \*EMail
* CompetitionDtos – tablica zawodów do których jest przypisanych (model danych zawodów), przy wylistowaniu null, pole pomijane przy dodawaniu i modyfikacjach

### GET api/OrganizerAccount/Search/Filtr?ItemsOnPage=10&PageNumber=0

### GET api/OrganizerAccount/Search?ItemsOnPage=10&PageNumber=0

Zwraca stronicowaną listę organizatorów z lub bez filtru ("Filtr") który jest stringiem

### GET api/OrganizerAccount/ByCompetition/1

Zwraca organizatorów przypisanych do danych zawodów

### GET api/OrganizerAccount/1

Pojedynczy organizator o podanym id, z polem CompetitionDtos

### PUT api/OrganizerAccount/1

Edycja

### POST api/OrganizerAccount

Stworzenie nowego organizatora z losowym hasłem. Hasło nie jest zwracane.

### DELETE api/OrganizerAccount/1

Usunięcie organizatora

### POST api/OrganizerAccount/1/PasswordReset

Reset hasła. Nowe, losowe hasło jest zwracane.

### POST api/OrganizerAccount/1/SetCompetition/1

Ustawia organizatorowi o podanym id dostęp do zawodów o podanym id

### POST api/OrganizerAccount/1/UnsetCompetition/1

Analogicznie

## Kontroler ModeratorAccount

Wszystkie operacje może tylko administrator

Model danych taki jak w akcji "Role"

Id to string, (w prototypach oznaczony jako "aaa")

### GET api/ModeratorAccount

Pobiera listę

### POST api/ModeratorAccount

Tworzy nowe kont,

W body to samo jak przy rejestracji

### DELETE api/ModeratorAccount/aaa

Usuwa

### POST api/ModeratorAccount/aaa/PasswordReset

Zmienia hasło na losowe i je zwraca

### POST api/ModeratorAccount/aaa/Bust

Zmienia role na "BustModerator"

### POST api/ModeratorAccount/aaa/Unbust

Analogicznie

## Kontroler AdministratorAccount

Wszystkie operacje może tylko administrator

Analogicznie jak w Moderatorze

### GET api/AdministratorAccount

### POST api/AdministratorAccount

### DELETE api/AdministratorAccount/aaa

### POST api/AdministratorAccount/aaa/PasswordReset

# Dokumentacja końcowa (powykonawcza) - punkty wymagane przez prowadzącego zajęcia

## Model danych

