



# POLISH-JAPANESE ACADEMY OF INFORMATION TECHNOLOGY

**Wydział Informatyki**

**Katedra metod programowania**

Programowanie aplikacji biznesowych

**Marcin Wałachowski**

s19541

**ERTS**

E-SPORT RESULTS TRACKING SYSTEM

Praca inżynierska

Krzysztof Barteczko

Warszawa, Luty, 2022r.

# Słowa kluczowe:

E-sport, Aplikacja webowa, C#, React

# Streszczenie:

Praca opisuje projekt ERTS, czyli E-sport results tracking system. Oferuje on aplikację webową, która jest odpowiedzialna za wizualizację statystyk pochodzących z rozgrywek e-sportowych. Dodatkowo w ramach systemu utworzyłem niezależny system, którego zadaniem jest sprowadzanie potrzebnych danych do DB (Database) aplikacji.

Pisemna część pracy dyplomowej jest podzielona na cztery rozdziały.

Pierwszy z nich opisuje wszystkie gry, których wyniki można przeglądać przy pomocy aplikacji. Wspieranymi obecnie tytułami są League of Legends, Counter-Strike Global Offensive, Dota 2, Overwatch oraz Valorant.

Kolejny rozdział opisuje wszystkie funkcjonalności aplikacji webowej utworzonej w ramach projektu. Dla każdej z nich, zawarty jest scenariusz opisujący jak się do niej dostać wewnątrz aplikacji, zrzut ekranu oraz jej opis.

Pozostałe dwa rozdziały zawierają techniczne aspekty aplikacji.

W rozdziale trzecim wymienione zostały technologie użyte podczas procesu tworzenia aplikacji. Każda z nich zawiera krótki opis, oraz jej miejsce w aplikacji.

Ostatni rozdział pokazuje natomiast architekturę systemu, który został podzielony na cztery oddzielne podprojekty. Ta część pracy jest podzielona na podrozdziały, z których każdy opisuje funkcjonalność, oraz zagadnienia implementacyjne dotyczące jednego z projektów utworzonych w ramach solucji aplikacji.

# Spis treści

WSTĘP .....	3
E-Sport .....	3
Cel aplikacji .....	5
1. WSPIERANE GRY .....	7
1.1 League of Legends .....	7
1.2 Counter-Strike: Global Offensive .....	9
1.3 Dota 2.....	10
1.4 Overwatch.....	11
1.5 Valorant .....	12
2. KONCEPCJA I FUNKCJONALNOŚĆ SYSTEMU .....	13
2.1 Przeglądanie listy lig wybranej gry.....	13
2.2 Przeglądanie listy drużyn wybranej gry.....	14
2.3 Wyszukiwanie ligi z listy na podstawie fragmentu jej nazwy .....	16
2.4 Przeglądanie listy serii w lidze .....	17
2.5 Przeglądanie listy turniejów w serii .....	18
2.6 Sprawdzanie tabeli wyników turnieju.....	19
2.7 Sprawdzanie wyników meczów w turnieju .....	20
2.8 Sprawdzanie statystyk zawodników w turnieju .....	22
2.9 Sprawdzanie statystyk drużyn w turnieju .....	24
2.10 Sprawdzanie informacji o wybranej drużynie .....	25
2.11 Sprawdzanie statystyk meczu League of Legends .....	27
2.12 Sprawdzanie statystyk gry League of Legends .....	29
2.13 Zmiana języka aplikacji.....	30

3. WYBRANIE ROZWIĄZANIA TECHNOLOGICZNE .....	31
3.1 C# .....	31
3.2 .Net Core .....	33
3.3 React .....	33
3.4 TypeScript .....	34
3.5 Bootstrap .....	34
3.6 PandaScore api .....	35
3.7 HangFire .....	37
3.8 PostgreSQL .....	37
3.9 Git .....	38
4. ARCHITEKTURA I IMPLEMENTACJA SYSTEMU .....	39
4.1 ErtsModel .....	39
4.2 ErtsApplication .....	51
4.3 ErtsWebApp .....	53
4.4 ErtsApiFetcher .....	59
PODSUMOWANIE .....	67
BIBLIOGRAFIA .....	68

# Wstęp

## E-Sport

Czym jest e-sport?

E-sport to zyskująca popularność w ostatnich czasach forma rywalizacji, w której gracze konkurują ze sobą w różnych dziedzinach (grach) przy pomocy komputerów.

## Rozwój e-sportu

Obecnie rozwój e-sportu przebiega bardzo dynamicznie i turnieje tego rodzaju sportu stają coraz bardziej popularne nawet wśród osób niezwiązanych z grami komputerowymi.

Ta dziedzina rywalizacji swój prężny rozwój zaczęła dopiero w XXI wieku, głównie z powodu rozwinięcia platform streamingowych, które umożliwiają transmitowanie różnego rodzaju rozgrywek na żywo. Wcześniej ze względu na mniejszą dostępność komputerów i ograniczoną możliwość transmitowania zawodów e-sportowych, rozgrywki organizowano raczej na mniejszą skalę i były one przeznaczone bardziej dla amatorów, będą przy tym traktowane bardziej jako forma rozrywki dla uczestników niż widowisko dla obserwatorów. (Wikipedia, 2021)

W tych czasach mistrzostwa e-sportowe są organizowane zdecydowanie na większą skalę i przyciągają setki milionów widzów rocznie. Poza tym e-sport już nie stał się tylko formą rozrywki dla graczy, lecz stał się on całym życiem dla wielu młodych ludzi, którzy poświęcili tysiące godzin, by stać się jak najlepszym. Obecnie z e-sportu utrzymuje się setki tysięcy profesjonalnych zawodników.

Ze względu na szybki i cały czas utrzymujący się rozwój e-sportu w tej branży pracują już nie tylko zawodnicy, oraz ich trenerzy, ale organizacje do swoich drużyn zatrudniają ludzi niezwiązanych z tą dziedziną, takich jak psycholodzy, dietetycy oraz inni ludzie mający za zadanie zapewnić graczom jak najlepsze warunki i zadbać o ich mentalność, co ma duży wpływ na sukces drużyn.

Częstym problemem e-sportu, w ostatnich latach były pieniądze. Spowodowane to było tym, że utrzymanie drużyn oraz organizacja turniejów wymaga dużych nakładów finansowych, a sponsorzy często boją się inwestować w tą branżę, ze względu na to, że jest ona nowa oraz niepewna.

Z tego powodu nie mają oni gwarancji, że ich wkład pieniężny się opłaci. W wyniku tego zjawiska cierpią głównie mniejsze i średnie klasy rozgrywek, w których nagrody nie zawsze są adekwatne do trudności ich zwyciężenia. W wyższych klasach rozgrywkowych ten problem jest mniejszy, a największe turnieje, oraz najlepsze drużyny są sponsorowane, przez ogromne marki takie jak Red bull i Samsung. Na szczęście z ubiegiem czasu ten problem jest coraz mniejszy i ze względu na rosnącą popularność, coraz więcej firm decyduje się włożyć swój kapitał w rozwój tej właśnie dziedziny rywalizacji.

## E-sport, a sport

Kwalifikowanie e-sportu jako formy sportu od lat budzi dużo kontrowersji. Wiele osób uważa, że są to zupełnie inne rodzaje rywalizacji i nie powinno się ich łączyć, a inni twierdzą, że ze względu, na rywalizację, oraz oglądalność przez widzów, stosunkowo nowa dziedzina jest odmianą sportu.

Tak naprawdę bez znaczenia do czego kwalifikuję się nowa forma rywalizacji skoro zarówno ona jak i tradycyjne rozgrywki mają wiele zalet i dają przyjemność i zapewniają rozgrywkę wielu widzom.

Moim zdaniem największą zaletą e-sportu, jest to że turnieje tego typu mogą być rozgrywane zarówno jako LAN, czyli w fizycznym miejscu gdzie gracze zbierają się, by konkurować na przygotowanych przez organizatora komputerach, oraz online, gdzie gracze mogą brać udział w zawodach nawet ze swoich domów. Dzięki tej zaletie e-sport bez problemu przetrwał okres pandemii. Inną powiązaną zaletą jest to, że do treningów gracze nie potrzebują specjalnych obiektów sportowych, a swoje umiejętności mogą szlifować zarówno w specjalnych gaming houses jak i po prostu w domu.

Oczywiście sport ma również zalety, których nie posiada e-sport, których przykładem jest na przykład to, że igrzyska sportowe dostarczają zupełnie innych emocji niż wirtualne mistrzostwa, oraz ich zasady są zrozumiałe dla każdego, a do świadomego oglądania profesjonalnych rozgrywek niektórych gier trzeba mieć o nich niemałą wiedzę.

Ten czynnik powoduje to, że e-sport jest popularny głównie u młodszych ludzi.

#### E-sport w Polsce

W Polsce rozwój e-sportu rozpoczął się w 1997 roku, w którym odbył się pierwszy turniej w wtedy popularną grę Quake. W tamtych czasach komputery były mniej dostępne przez co osoby zainteresowane tematem gier wideo, zamiast w domach grali w kafejkach internetowych.

Najbardziej dynamiczny rozwój tej dziedziny w Polsce przypada natomiast na dopiero 2014 rok, w którym drużyna składająca się z 5 Polaków sensacyjnie wygrała najważniejszy turniej gry Counter-Strike Global Offensive na świecie, którym był Intel Extreme Masters.

Jest to turniej corocznie organizowany w Katowickim Spodku i jest on popularny na całym świecie, ponieważ biorą w nim udział najlepsze drużyny mające siedzibę nawet tysiące kilometrów od Polski. Przez tak ogromny prestiż tego turnieju to miasto nazywane jest stolicą polskiego e-sportu.

Obecnie e-sport w Polsce rozwinął się do takiego poziomu, że mamy wielu utalentowanych reprezentantów grających w drużynach na całym świecie regularnie zdobywających kolejne osiągnięcia.

## Cel aplikacji

#### Ogólny cel

Głównym celem aplikacji jest umożliwienie użytkownikowi śledzenia statystyk e-sportowych dla wspieranych gier. Aplikacja ma zapewnić wygodny, oraz przejrzysty interfejs, pozwalający użytkownikowi odnaleźć pożądane dane.

Obecnie tytułem głównym wspieranym przez aplikację ma być gra League of Legends. Wybrałem tą grę ze względu na jej dużą popularność oraz przez to, że posiadam na jej temat największą wiedzę. Z tego powodu informacje dotyczące meczów tego tytułu w obecnej wersji systemu są bardziej szczegółowe od innych rozgrywek.

W przyszłości jednak planuję stopniowo dodawać wsparcie do innych gier, co sprowadza się głównie do utworzenia nowych typów danych zawierających ich unikalne statystyki, oraz do dodania nowych widoków we front-endzie aplikacji.

Innym powodem stworzenia projektu jest cel edukacyjny, ponieważ podczas realizacji projektu użyłem wielu nowych technologii, z którymi nie miałem wcześniej dużej styczności.

#### Najważniejsze funkcjonalności

Najważniejszymi funkcjonalnościami aplikacji są:

- Sprawdzanie wyników meczy dla wybranych gier
- Przeglądanie wyników turniejów e-sportowych dla wybranych gier
- Wyświetlanie statystyk zawodników i drużyn w turniejach gry League of Legends
- Sprawdzanie szczegółowych statystyk gier League of Legends
- Przeglądanie informacji o drużynach, ligach, seriach i turniejach
- Zmiana języka aplikacji

Podane wyżej funkcjonalności dokładnie opisałem w trzecim rozdziale pracy.

Dodatkową funkcjonalnością aplikacji jest zapewnienie serwisu, odpowiedzialnego za pobieranie danych dotyczących e-sportu i aktualizowanie nimi bazy danych aplikacji w określonych momentach czasowych. Ta funkcjonalność została dokładnie opisana w piątym rozdziale aplikacji we fragmencie dotyczącym projektu ErtsApiFetcher.

#### Ograniczenia

Głównym ograniczeniem aplikacji jest brak dostępności darmowych api służących do pozyskiwania pełnych wyników meczów e-sportowych. Z tego powodu użyłem darmowej wersji PandaScore, która oferuje ograniczoną ilość godzinnych zapytań. Dodatkowo udostępnia ona tylko ogólne dane, przez co nie mogłem uzyskać szczegółowych statystyk poszczególnych gier. Z tego powodu na potrzeby prezentacji projektu, dodałem przykładowe statystyki, służące do umożliwienia zaprezentowania części aplikacji wyświetlającą te dane.

# 1. Wspierane gry

## 1.1 League of Legends

### Opis gry

League of legends to gra typu MOBA (Multiplayer Online Battle Arena, wypuszczona pod koniec 2009 roku, przed firmę Riot Games. Ta gra miała dość trudne początki, ponieważ nie została stworzona przez wielką markę i pracowało nad nią zaledwie kilku programistów. Pomimo tego zdobyła ona ogromną sympatię graczy na całym świecie i obecnie jest to jedna z najbardziej znanych gier na świecie.

Pomimo że LoL został wypuszczona ponad dekadę temu to od wielu lat trzyma się na topowych listach gier i jest ogrywany przez miliony graczy. Riot Games cały czas wypuszcza nowe aktualizacje, przez co mimo wieku gry, jest ona cały czas jest aktualna i daje przyjemność graczom.



1 Oficjalne logo gry League of Legends

Obecnie w grze możemy wcielić się w ponad 150

różnorodnych postaci dzięki czemu każda pojedyncza gra jest inna.

Dodatkowo wszyscy bohaterowie gry mają swoją własną historię fabularną, przez co gracze mogą się utożsamiać ze swoimi bohaterami, co daje im większą przyjemność z gry.

Lore, czyli fabuła gry jest bardzo rozbudowany i developer regularnie wypuszcza nowe opowieści i krótkie animacje przedstawiające losy bohaterów. Ostatnio Riot wypuścił nawet serial „Arcane”, przedstawiający losy bohaterów, w których możemy się wcielić w tej właśnie grze i ku zaskoczeniu wielu ludzi, osiągnął on niesamowitą popularność nie tylko wśród graczy League of Legends, przez co stał się jednym z najbardziej popularnych seriali na platformie Netflix. Dzięki sukcesowi tej animacji gra zyskała wielu nowych graczy.

### Zasady gry

W League of Legends gracze rywalizują, ze sobą w pięcioosobowych drużynach na mapie „Summoners Rift”. Na początku rozgrywki, każdy z zawodników wybiera swojego bohatera, którym następnie steruje do końca meczu. Każda z postaci w grze jest inna i posiada swój własny unikalny zestaw umiejętności. Z tego powodu operator musi dostosować do niej swój styl gry i mało kto potrafi opanować każdą z nich.

Podczas rozgrywki zawodnicy walczą z przeciwnymi postaciami oraz niszczą sterowane przez komputer stwory, zdobywając przy tym doświadczenie i złoto, dzięki któremu mogą kupić lepsze przedmioty, zwiększające statystyki ich postaci. Dodatkowo podczas gry, drużyny



walczą również z potężnymi stworami, takimi jak smoki, których pokonanie daje im przydatne wzmocnienia ułatwiające zwyciężenie rozgrywki. Innymi tego typu stworzeniami w grze są Heraldy oraz Barony, których pokonanie również ma duże znaczenie w przebiegu rozgrywki.

Głównym celem rozgrywki jest natomiast zniszczenie przeciwnej bazy, a konkretniej kryształu położonego w jej sercu nazwanego Nexusem. Pomimo prostoty tego celu gry w League of Legends nie kończą się zbyt szybko i trwają średnio 30 minut, co spowodowane jest tym, że przed zniszczeniem głównej struktury trzeba również zburzyć przeciwne wieże, które są bardzo wytrzymałe i dodatkowo potrafią zadawać obrażenia bohaterom.

### E-sport

Esport w league of legends jest najpopularniejszy ze wszystkich tytułów. Podczas corocznych mistrzostw świata gromadzą się całe stadiony fanów oraz miliony osób ogląda turnieje sprzed komputera bądź telewizora. Finał ostatnich mistrzostw świata pobił wszystkie rekordy oglądalności meczu e-sportowego. Spotkanie między chińskim Edward Gaming i koreańskim Damwon Kia zgromadziło przed komputerami ponad 4 miliony widzów, czego nie osiągnął wcześniej żaden mecz.

E-sport w LoLu jest bardzo popularny również w Polsce i wielu Polaków regularnie odnosi sukcesy w tej dziedzinie na arenie międzynarodowej. Przykładem takiej osoby jest Marcin Jankowski, który wielokrotnie występował w mistrzostwach świata tego tytułu, a w 2019 roku wraz z drużyną G2 Esports zajął nawet 2 miejsce. Dzięki swoim osiągnięciom, został on nagrodzony na gali sportowców roku, jako najlepszy zawodnik e-sportowy w Polsce. Dzięki ogromnej popularności gry w naszym kraju zostały zorganizowane dwie dywizje ligi narodowej, które umożliwiają młodym graczom łatwiejszy rozwój. Organizatorem polskiej Ultraligi jest Polsat, dzięki czemu wszystkie rozgrywki najwyższej półki rozgrywkowej w Polsce są transmitowane na kanale Polsat Games.

### Miejsce w aplikacji

Ze względu na swoją popularność LoL będzie głównym tytułem zawartym w mojej aplikacji i w odróżnieniu od innych gier poza wynikami meczów, będzie można sprawdzać również szczegółowe statystyki gier takie jak uzyskana ilość złota, oraz liczba zniszczonych obiektów.

## 1.2 Counter-Strike: Global Offensive

### Opis gry

Counter-Strike: Global Offensive to gra z gatunku FPS (First-person shooter), wypuszczona w 2012 roku przez firmę Valve Corporation.

Csgo również jest grą wydaną dosyć dawno i mimo to dalej przyciąga ogromną liczbę graczy przynosząc przy tym wydawcy ogromne przychody. Z tego powodu Valve dalej ją rozwija, zamiast wydawać nowe części.

### Zasady gry

W grze rywalizuje 10 graczy rozdzielonych na drużynę terrorystów i antyterrorystów. Gra jest podzielona na rundy, w których celem drużyny terrorystów jest podłożenie bomby i doprowadzenie do jej wybuchu, a druga drużyna ma do tego nie dopuścić.

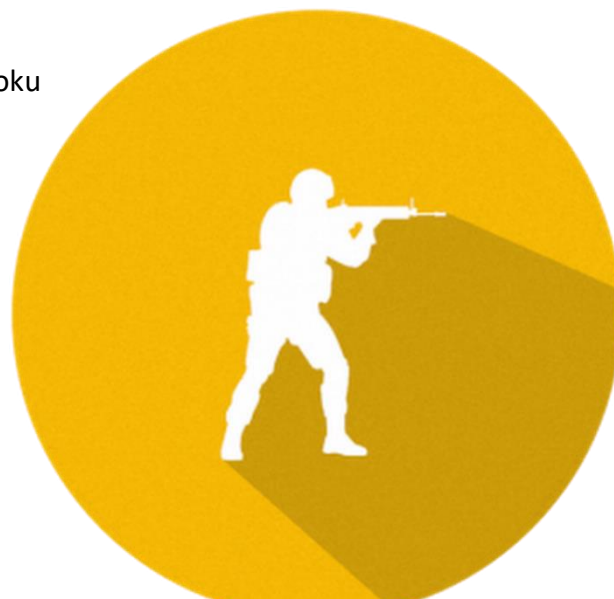
Na początku każdej rundy gracze mogą kupować jedną z wielu broni dostępnych w sklepie, za pomocą których potem rywalizują z przeciwnikiem. Niektóre wyposażenie takie jak karabin snajperski, wymaga od graczy wydania sporej ilości pieniędzy zarobionych podczas poprzednich rund. Z tego powodu dobre zarządzanie ekonomią przez zawodników daje im realną przewagę nad przeciwnikiem.

Pomimo że gra jest strzelanką to wymaga ona mocnej kooperacji pomiędzy sojusznikami i wymaga ona taktycznego podejścia do rozgrywki. Wzajemne ubezpieczanie się przez graczy oraz używanie granatów taktycznych, znacznie ułatwia zwyciężenie rozgrywki.

Zwycięża drużyna, która jako pierwsza wygra 16 rund z przewagą co najmniej 2 rund. W przypadku osiągnięcia 16 punktów z przewagą jednego, następuje dogrywka, w której drużyna musi osiągnąć dwurundowe prowadzenie.

### E-sport

Counter Strike: Global Offensive jest grą nastawioną typowo na e-sport, przez co jest ona oglądana nawet przez ludzi niegrających w ten tytuł. W CS GO odbywa się wiele turniejów, lecz najważniejszymi są turnieje nazywanymi majorami, które odbywają się około 3 razy w roku, a gracze rywalizują na nich o najwyższe nagrody. Ostatni Major był wyjątkowy, ponieważ odbył się po długiej przerwie spowodowanej pandemią, przez co w szczytowym momencie



2 Logo gry Counter-Strike: Global Offensive

zgrupował 2,7 miliona widzów podczas finału między Ukraińskim Natus Vincere oraz Francuskim G2 Esports, ustanawiając przy tym rekord oglądających dla tego tytułu.

W Polsce CS GO jest również bardzo popularny i dla tego tytułu również posiadamy własną oficjalną ligę narodową o nazwie Polska Liga Esportowa (PLE). Dodatkowo ze względu na sukcesy wcześniej wspomnianej drużyny Virtus Pro, przez wiele lat Counter Strike, był najbardziej popularną dziedziną e-sportu w Polsce.

## 1.3 Dota 2

### Opis gry

Dota 2 to podobnie jak League of Legends gra typu MOBA, wydana przez twórców Counter Strike czyli Valve Corporation. Miała ona premierę w 2013 roku i dalej utrzymuje swoją popularność, lecz w przeciwieństwie do pozostałych tytułów, większość jej społeczności pochodzi z Rosji i Chin, a na Zachodzie ma już znacznie mniej zwolenników.

### Zasady gry

Zasady Doty 2, są bardzo podobne do zasad League of Legends, ponieważ również wcielamy się w jedną z wielu postaci, walcząc z przeciwnikami i próbując zniszczyć przeciwną twierdzę. Pomimo podobieństw oba tytuły posiadają jednak sporo różnic, którymi są między innymi mechanika przedmiotów oraz system sterowanych przez sztuczną inteligencję stworów i potworów.

Pomimo tych różnic moim zdaniem największą różnicą jest to, że obie gry zawierają swoje unikalne postacie, które czynią rozgrywki w obu grach zupełnie innymi. Z tego właśnie powodu uważam, że warto wypróbować oba tytuły.

### E-sport

Najważniejszymi turniejami w Dote 2 są Internationale, które odbywają się raz do roku. W przeciwieństwie do np. League Of Legends na te mistrzostwa mogą zakwalifikować się nawet amatorskie drużyny, gdyż kwalifikacje są otwarte dla każdego. Turnieje w Dote 2 słynną z ogromnych nagród pieniężnych, które są wielokrotnie większe niż w innych tytułach. Pula nagród na ostatnim Internationalu wynosiła ponad 40 milionów dolarów.



3 Logo gry Dota 2

## 1.4 Overwatch

### Opis gry

Overwatch to FPS, wydany przez firmę Blizzard Entertainment. Jest on nowszą grą od poprzednich tytułów, gdyż swoją premierę miał w połowie 2016 roku. Ze względu, że został on wydany przez developera mającego na swoim koncie kilka bardzo znanych tytułów takich jak World of Warcraft, gra zyskała ogromną popularność od razu po jej wypuszczeniu. Obecnie niestety gra utraciła mocno na ilości graczy, lecz wciąż ma swoich wiernych graczy.

### Zasady gry

Podczas rozgrywki gracze wcielają się w jedną z ponad 30 postaci i w sześciuosobowych drużynach rywalizują z przeciwnikami w różnych trybach rozgrywki, takich jak Szturm polegający na przejęciu bazy przeciwnika, oraz Eskort, w której jedna z drużyn musi dostarczyć specjalny wózek do wyznaczonego punktu, odpierając przy tym ataki przeciwników.

W Overwatchu, w porównaniu do innych Fpsów, postacie nie posiadają standardowych broni i każdy bohater posiada swoje unikalne narzędzie do walki.

Sprzęt, którym dysponują postacie nie musi być konieczne bronią dystansową i niektórzy bohaterowie walczą za pomocą na przykład miecza. Postacie poza oryginalnymi narzędziami walki, podobnie jak w League of Legends, posiadają swój unikalny zestaw umiejętności, w którego skład wchodzi umiejętności specjalne, które po załadowaniu potrafią zrobić dużą różnicę podczas walki.

Z powodu zróżnicowania bohaterów, podczas rozgrywki gracze podzieleni są na role jakimi są broniący drużyny „Tank”, zadawający obrażenia „Damage”, oraz wspierający drużynę „Support”.

### E-sport

W grze Overwatch również istnieją różne ligi, w których gracze rywalizują ze sobą w celu zakwalifikowania się do najlepszej z nich, czyli „Overwatch League”. Podczas tej ligi gracze rywalizują reprezentując swoje miasto i próbując uczynić je najlepszym na świecie.

Niestety w tej lidze obecnie nie bierze udziału żadna polska drużyna.

Poza rozgrywkami ligowymi co roku odbywają się mistrzostwa świata tego tytułu, w których reprezentacje narodowe rywalizują o miano najlepszej. W tym turnieju polska natomiast ma już swoją reprezentację, lecz największe sukcesy w tej klasie rozgrywek w ostatnich latach osiągała Korea Południowa. (BigBoss, 2020)



4 Logo gry Overwatch

## 1.5 Valorant

### Opis gry

Valorant to gra gatunku FPS, wydana przez firmę Riot Games będącego również twórcą poprzednio omówionego League of Legends.

Valorant jest najnowszym z podanych gier gdyż został wydany w 2020 roku. Przez dużą rozpoznawalność Lola, Valorant szybko również zyskał ogromną popularność, o której świadczy duże zainteresowanie graczy wersją beta gry, którą można było uzyskać w ramach „Twitch Drop”, czyli poprzez dostanie losowo rozdanego prezentu podczas oglądania transmisyj z gry na platformie streamingowej Twitch. Podczas tego okresu setki tysięcy osób oglądało kanały powiązane z tytułem, tylko po to by uzyskać upragniony przedwczesny dostęp do nowo wypuszczonej gry.



5 Logo gry Valorant

### Zasady gry

Podczas rozgrywki Valoranta, gracze rywalizują w pięcio-osobowych drużynach, wcielając się w specjalnych bohaterów, którzy oferują unikalne umiejętności, takie jak stawianie kamer, oraz leczenie sojuszników. W porównaniu do Overwatcha, bohaterowie nie posiadają swoich specjalnych broni, lecz podobnie jak w CS GO, mogą kupić dowolną broń w sklepie na początku rundy. Dzięki temu rozwiązaniu gracze mogą używać dowolnych broni na wszystkich postaciach, co umożliwia im większą elastyczność rozgrywki.

Ogólna idea gry jest podobna jak w Counter Strike, ponieważ gra również jest podzielona na rundy, w których jedna drużyna podkłada bombę, a druga temu zapobiega. Mimo tego podobieństwa, odczucia z tych tytułów są bardzo zróżnicowane, ponieważ obecność różnorodnych postaci w Valorancie, czyni go bardziej dynamicznym, przez co dla wielu jest on ciekawszy.

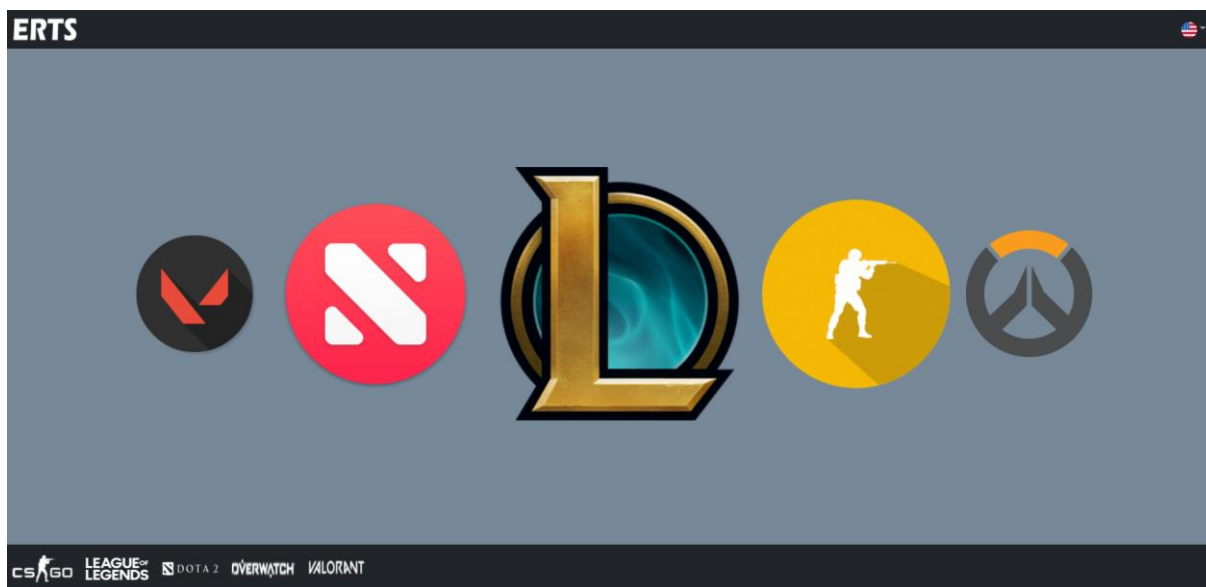
### E-sport

Turnieje Valoranta odbywają się na mniejszą skalę niż turnieje Lola, lecz zyskują one coraz większą popularność, największym turniejem jest Valorant Champions pełniący rolę mistrzostw świata i podczas jego ostatniego finału zgromadził przed ekranem milion widzów.

W Polsce Valorant jest również dość popularny i podobnie jak League of Legends, również posiada własną ligę (Polska liga esportowa). W najbliższym czasie popularność tej gry w Polsce może wzrosnąć z powodu triumfu Aleksandra „zeek” Zygmunata i Patryka „starxo” Kopczyńskiego w finale mistrzostw świata, reprezentując drużynę ACEND.

## 2. Koncepcja i funkcjonalność systemu

W tym rozdziale opisane są wszystkie funkcjonalności oferowane przez utworzony w ramach projektu system. Każda opisana funkcjonalność poza jej opisem posiada również tabelkę zawierającą scenariusz zawierający instrukcje dotyczące uruchomienia podanej funkcjonalności.



6 Ekran zawierający stronę główną aplikacji

Każda z instrukcji dotyczących użycia funkcjonalności systemu, rozpoczyna się od uruchomienia ekranu startowego aplikacji przedstawionego na powyższym zrzucie ekranu.

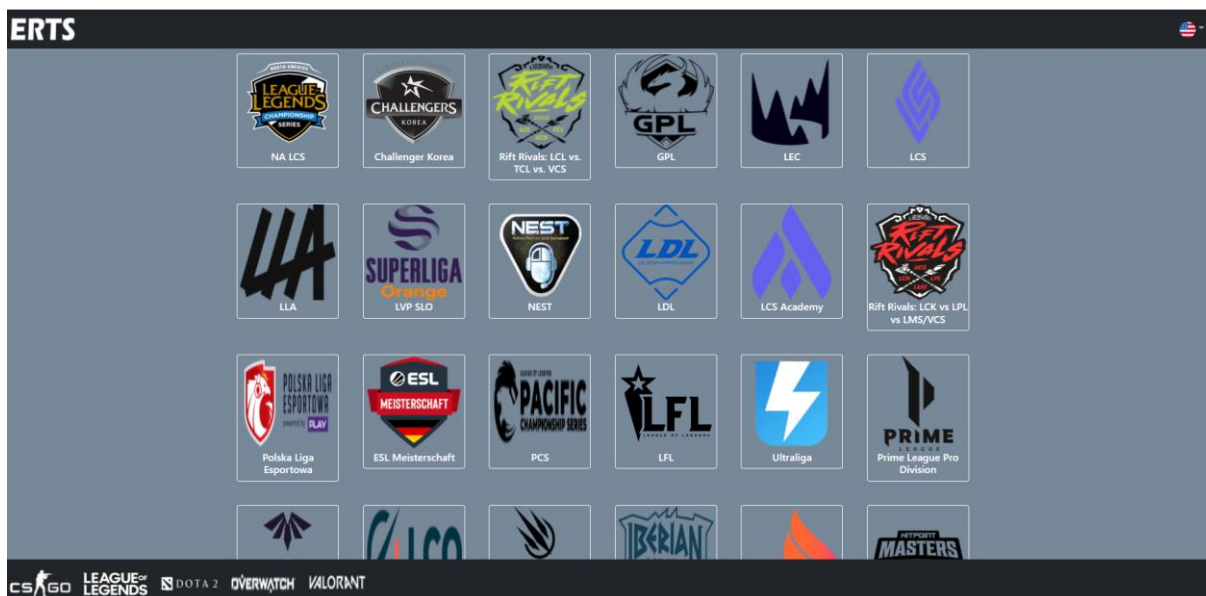
Zawiera on logotypy wszystkich gier obsługiwanych przez aplikację.

Każda z przedstawionych gier jest opisana w rozdziale drugim pracy, w którym zawarte są podstawowe informacje o wspieranych tytułach, zawierające również ich loga wykorzystane w aplikacji.

### 2.1 Przeglądanie listy lig wybranej gry

NAZWA PRZYPADKU UŻYCIA	Przeglądanie listy lig wybranej gry	ID.
		1.
SCENARIUSZ		
KROK	AKCJA	
1. Aktor	Aktor uruchamia ekran główny aplikacji wpisując jej adres w przeglądarkę	
2. System	Wyświetla loga gier wspieranych przez ERTS.	

3. Aktor	Aktor wybiera interesujący go tytuł
4. System	Wyświetla listę zawierającą loga i nazwy wszystkich lig wybranej gry



7 Ekran zawierający listę lig dla gry League of Legends

Ta funkcjonalność polega na wyświetlaniu listy wszystkich lig przechowywanych w bazie danych aplikacji, odbywających się w ramach wybranego przez użytkownika tytułu. W liście każdy obiekt zawiera swoją kartę, zawierającą nazwę oraz logotyp ligi,

Liga to zorganizowana rywalizacja, w ramach której co jakiś czas odbywają się serie, w których rywalizują drużyny. Liga w e-sporcie jest odpowiednikiem lig w rozgrywkach sportowych takich jak Liga mistrzów (piłka nożna) lub NBA (koszykówka).

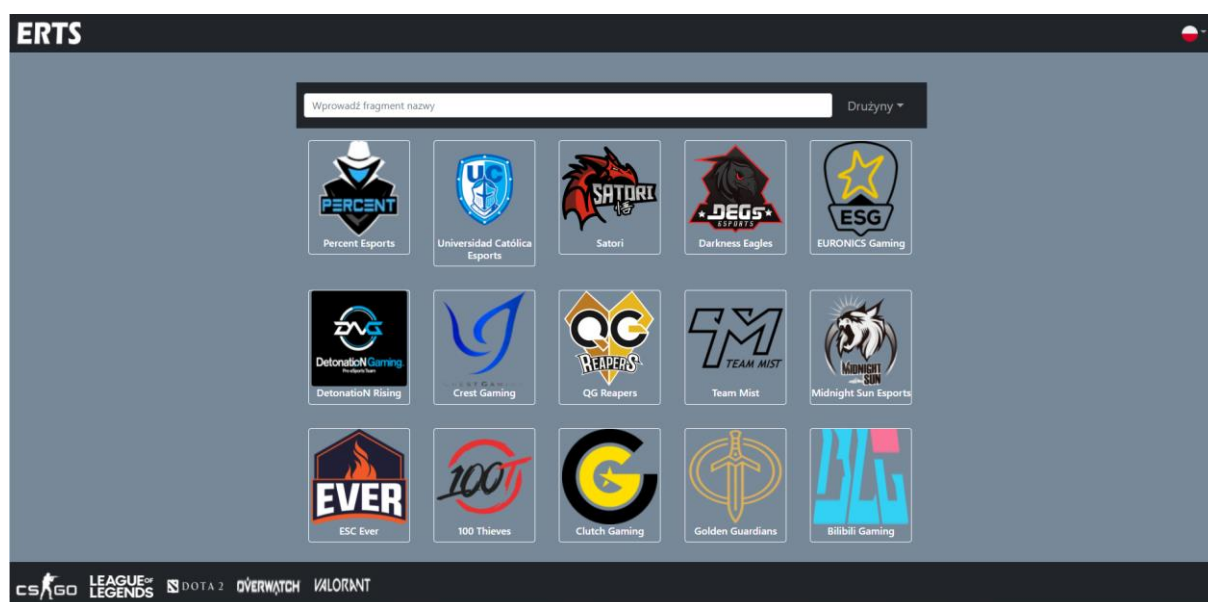
Jak wynika ze scenariusza przedstawionego powyżej, jest to funkcjonalność, do której można dostać się bezpośrednio ze strony głównej aplikacji. Jest to spowodowane tym, że ligi są najogólniejszą strukturą w aplikacji i większość innych obiektów jest w bliskim lub dalszym stopniu z nimi powiązane.

## 2.2 Przeglądanie listy drużyn wybranej gry

NAZWA PRZYPADKU UŻYCIA	Przeglądanie listy drużyn wybranej gry	ID.
		2.
SCENARIUSZ		
KROK	AKCJA	



1. Aktor	Aktor uruchamia ekran główny aplikacji wpisując jej adres w przeglądarkę
2. System	Wyświetla loga gier wspieranych przez ERTS.
3. Aktor	Aktor wybiera interesujący go tytuł
4. System	Wyświetla listę zawierającą loga i nazwy wszystkich lig wybranej gry
5. Aktor	Aktor naciska na napis „Ligi” znajdujący się w nawigacji
6. System	Wyświetla listę z możliwych opcji do wyświetlenia
7. Aktor	Aktor wybiera opcje „Drużyny”
8. System	Wyświetla listę zawierającą loga i nazwy wszystkich drużyn wybranej gry



8 Ekran zawierający listę drużyn rywalizujących w League of Legends

Ta funkcjonalność jest odpowiedzialna za wyświetlanie wszystkich drużyn rywalizujących w ramach wybranego przez użytkownika tytułu. Podobnie jak w przypadku lig zawiera on nazwy oraz herby zespołów.

Drużyny w e-sporcie to po prostu odpowiednik drużyn w sporcie, czyli zorganizowane zespoły rywalizujące w różnych turniejach. W porównaniu jednak do klasycznej formy rywalizacji, system drużyn jest bardziej dynamiczny i rotacja drużyn w turniejach jest bardzo wysoka i w przeciwieństwie do sportowych drużyn z wieloletnią tradycją, organizacje tutaj często istnieją tylko przez jeden sezon i są zastępowane przez inne.



W aplikacji jest to funkcjonalność umieszczona na tym samym poziomie co lista lig i można się pomiędzy nimi łatwo przełączać zmieniając wartość w wysuwanym menu, znajdującym się w nawigacji tego widoku. Umieściłem te funkcjonalności obok siebie z tego powodu, że obie są zależne tylko od wybranego tytułu i nie są ze sobą ściśle powiązane.

## 2.3 Wyszukiwanie ligi z listy na podstawie fragmentu jej nazwy

NAZWA PRZYPADKU UŻYCIA	Wyszukiwanie ligi z listy na podstawie fragmentu jej nazwy	ID.
		3.
SCENARIUSZ		
KROK	AKCJA	
1. Aktor	Aktor uruchamia ekran główny aplikacji wpisując jej adres w przeglądarkę	
2. System	Wyświetla loga gier wspieranych przez ERTS.	
3. Aktor	Aktor wybiera interesujący go tytuł	
4. System	Wyświetla listę zawierającą loga i nazwy wszystkich lig wybranej gry	
5. Aktor	Aktor wpisuje fragment nazwy ligi w formularzu znajdującym się w nawigacji	
6. System	System wyświetla listę lig spełniających podane kryterium	



9 Ekran zawierający listę lig zawierający w nazwie fragment: „europe”


Jest to funkcjonalność wprowadzająca wyszukiwarkę do widoku lig oraz drużyn. Aby z niej skorzystać użytkownik wpisuje fragment nazwy obiektu w specjalnym formularzu, znajdującym się w nawigacji. Na podstawie wartości wyszukiwanej system zwraca wszystkie ligi lub drużyny, które w swojej nazwie zawierają wprowadzony ciąg znaków.

Ciekawą własnością tej funkcjonalności jest to, że wartość formularza zostaje niezmienna podczas przełączania się pomiędzy widokami lig oraz zespołów.

## 2.4 Przeglądanie listy serii w lidze

NAZWA PRZYPADKU UŻYCIA	Przeglądanie listy serii w lidze	ID.
		4.
SCENARIUSZ		
KROK	AKCJA	
1. Aktor	Aktor uruchamia ekran główny aplikacji wpisując jej adres w przeglądarkę	
2. System	Wyświetla loga gier wspieranych przez ERTS.	
3. Aktor	Aktor wybiera interesujący go tytuł	
4. System	Wyświetla listę zawierającą loga i nazwy wszystkich lig wybranej gry	
5. Aktor	Aktor wybiera kartę ligi, której serie chce wyświetlić	
6. System	System wyświetla listę serii z wybranej ligi	

ERTS



European Masters

Seria	Czas rozpoczęcia	Czas zakończenia
European Masters 2022	16:00 04-04-2022	
European Masters 2021 Summer	14:30 16-08-2021	19:56 19-09-2021
European Masters 2021 Spring	14:30 29-03-2021	19:50 02-05-2021
European Masters 2020 Summer	16:00 17-08-2020	19:00 20-09-2020
European Masters 2020 Spring	16:00 06-04-2020	18:47 10-05-2020
European Masters 2019 Autumn	22:00 08-09-2019	18:10 29-09-2019
European Masters 2019 Spring	15:00 08-04-2019	18:00 30-04-2019
European Masters 2018 Autumn	15:19 10-09-2018	18:01 30-09-2018
European Masters 2018 Spring	16:00 14-04-2018	19:16 29-04-2018

CS:GO

LEAGUE OF LEGENDS

DOTA 2

OVERWATCH

VALORANT

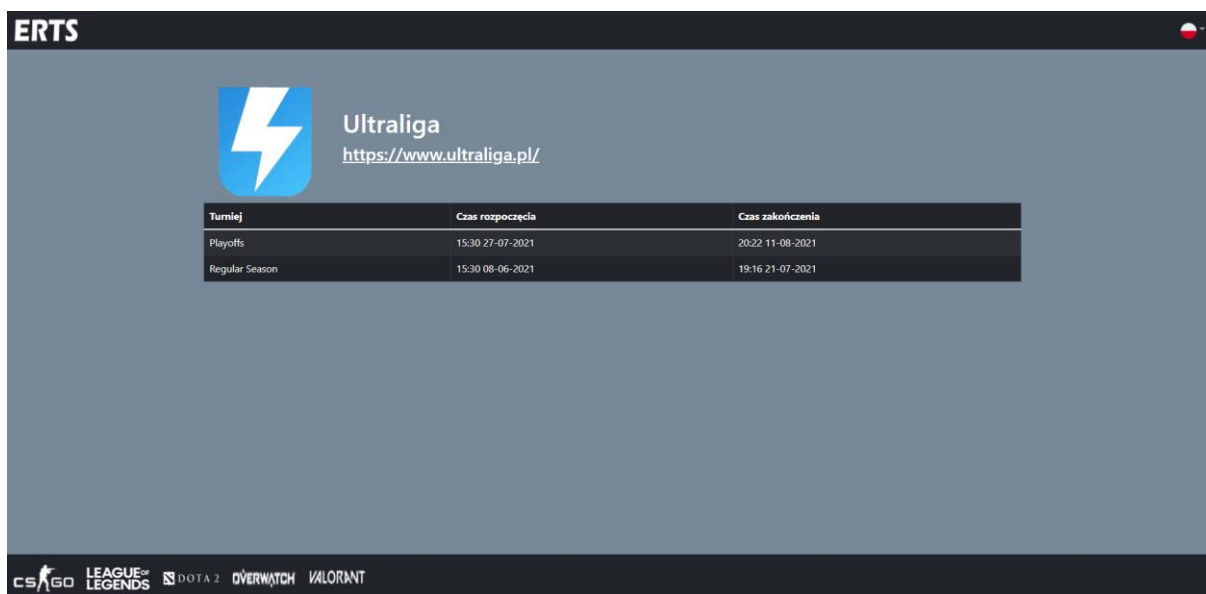
10 Widok z listy serii dla ligi European Masters

Jest to funkcjonalność polegająca na wyświetlaniu użytkownikowi tabeli zawierającą serie odbyte w ramach wybranej przez niego ligi. W tabeli każdy wiersz zawiera nazwę serii oraz datę jej rozpoczęcia oraz zakończenia.

Seria to pojedyncza rywalizacja odbywająca się w ramach określonej ligi, która odbywa się przez określony czas trwający zwykle kilka miesięcy. Przykładem serii są na przykład mistrzostwa świata odbyte w 2021 roku. (Worlds 2021)

## 2.5 Przeglądanie listy turniejów w serii

NAZWA PRZYPADKU UŻYCIA	Przeglądanie listy turniejów w serii	ID.
		5.
SCENARIUSZ		
KROK	AKCJA	
1. Aktor	Aktor uruchamia ekran główny aplikacji wpisując jej adres w przeglądarkę	
2. System	Wyświetla loga gier wspieranych przez ERTS.	
3. Aktor	Aktor wybiera interesujący go tytuł	
4. System	Wyświetla listę zawierającą loga i nazwy wszystkich lig wybranej gry	
5. Aktor	Aktor wybiera kartę ligi, której serie chce wyświetlić	
6. System	System wyświetla listę serii z wybranej ligi	
7. Aktor	Aktor wybiera serie z listy, której turnieje chce wyświetlić	
8. System	System wyświetla listę turniejów z wybranej serii	



1.1 Widok z listy turniejów wchodzących w skład wiosennego sezonu polskiej Ultraligi odbytego w 2021 roku

Jest to funkcjonalność pozwalająca użytkownikowi przeglądać wszystkie turnieje, wchodzące w skład wybranej przez niego serii. Podobnie jak dla poprzedniej tabeli obiekty zawierają informacje o nazwie turnieju oraz jego dacie rozpoczęcia i zakończenia.






W aplikacji turniej reprezentuje fragment serii, będącym jednym z jej etapów. Przykładem turnieju jest na przykład grupa A, bądź faza pucharowa.

## 2.6 Sprawdzanie tabeli wyników turnieju

NAZWA PRZYPADKU UŻYCIA	Sprawdzanie tabeli wyników turnieju	ID.
		6.
SCENARIUSZ		
KROK	AKCJA	
1. Aktor	Aktor uruchamia ekran główny aplikacji wpisując jej adres w przeglądarkę	
2. System	Wyświetla loga gier wspieranych przez ERTS.	
3. Aktor	Aktor wybiera interesujący go tytuł	
4. System	Wyświetla listę zawierającą loga i nazwy wszystkich lig wybranej gry	
5. Aktor	Aktor wybiera kartę ligi, której serie chce wyświetlić	
6. System	System wyświetla listę serii z wybranej ligi	

7. Aktor	Aktor wybiera serie z listy, której turnieje chce wyświetlić
8. System	System wyświetla listę turniejów z wybranej serii
9. Aktor	Aktor wybiera turniej z listy
10. System	System wyświetla tabele wyników wybranego turnieju

ERTS

TABELA WYNIKÓW		MECZE	STATYSTYKI ZAWODNIKÓW	STATYSTYKI DRUŻYN
#	Drużyna	Mecze	Gry	
1	 PDW	3-0	9-3	
2	 Illuminar Gaming	3-1	11-6	
3	 KICKICK	0-1	2-3	
4	 AGO ROGUE	0-2	0-6	
5	 Team ESCA Gaming	0-2	2-6	

CSGO

LEAGUE OF LEGENDS

DOTA 2

OVERWATCH

VALORANT

12 Widok zawierający tabele wyników turnieju

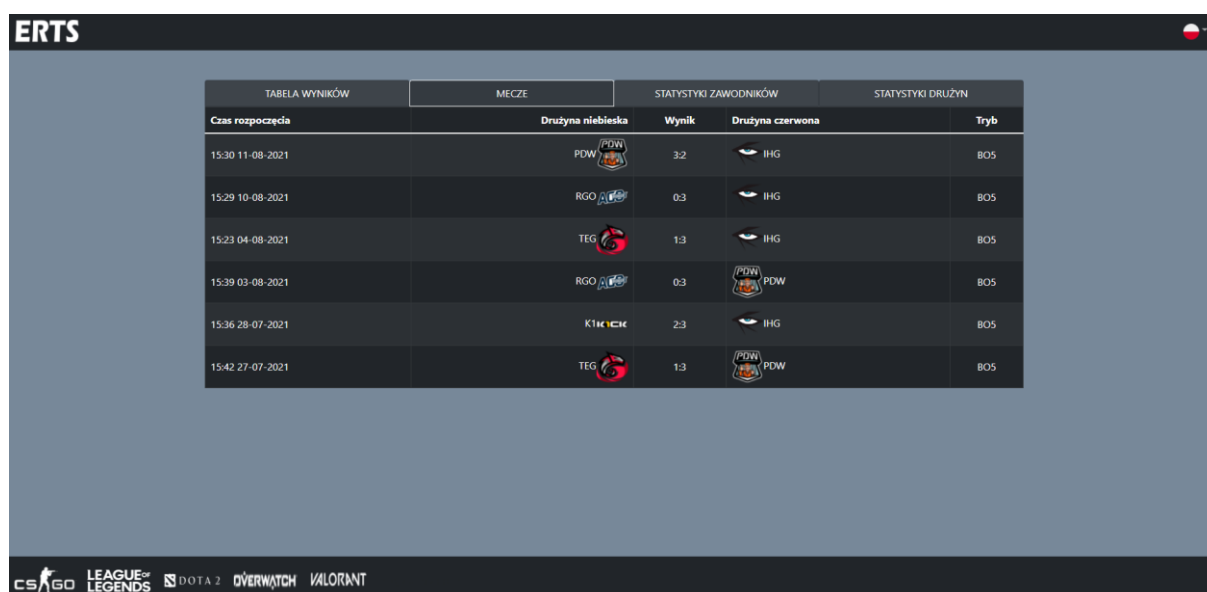
Jest to pierwsza z funkcjonalności dotyczących turnieju, dzięki której możemy zobaczyć tabele wyników dla wybranego wcześniej turnieju. W widoku udostępnianym przez tą funkcjonalność dla każdej drużyny możemy zobaczyć bilans jej meczów oraz gier. Drużyny są posortowane od drużyny prowadzącej do ostatniego miejsca w turnieju.

Z tego widoku możemy przejść bezpośrednio do widoku statystyk wybranej drużyny poprzez naciśnięcie na opisującą ją wiersz tabeli.

## 2.7 Sprawdzanie wyników meczów w turnieju

NAZWA PRZYPADKU UŻYCIA	Sprawdzanie wyników meczów w turnieju	ID.
		7.
SCENARIUSZ		
KROK	AKCJA	
1. Aktor	Aktor uruchamia ekran główny aplikacji wpisując jej adres w przeglądarkę	

2. System	Wyświetla logotypy gier wspieranych przez ERTS.
3. Aktor	Aktor wybiera interesujący go tytuł
4. System	Wyświetla listę zawierającą logotypy i nazwy wszystkich lig wybranej gry
5. Aktor	Aktor wybiera kartę ligi, której serie chce wyświetlić
6. System	System wyświetla listę serii z wybranej ligi
7. Aktor	Aktor wybiera serie z listy, której turnieje chce wyświetlić
8. System	System wyświetla listę turniejów z wybranej serii
9. Aktor	Aktor wybiera turniej z listy
10. System	System wyświetla tabele wyników wybranego turnieju
11. Aktor	Aktor wybiera opcje „MECZE” z nawigacji turnieju
12. System	System wyświetla listę zawierającą wyniki meczów w turnieju



The screenshot shows the ERTS application interface. At the top, there's a header with the ERTS logo and a Polish flag. Below the header, there's a navigation bar with four tabs: "TABELA WYNIKÓW", "MECZE", "STATYSTYKI ZAWODNIKÓW", and "STATYSTYKI DRUŻYN". The "MECZE" tab is selected. The main content area displays a table of match results. The table has five columns: "Czas rozpoczęcia", "Drużyna niebieska", "Wynik", "Drużyna czerwona", and "Tryb". The table lists seven matches with their start times, team names and logos, scores, and the game mode (BOS). At the bottom of the interface, there's a footer with logos for CS:GO, LEAGUE OF LEGENDS, DOTA 2, OVERWATCH, and VALORANT.

Czas rozpoczęcia	Drużyna niebieska	Wynik	Drużyna czerwona	Tryb
15:30 11-08-2021	PDW	3:2	IHG	BOS
15:29 10-08-2021	RGO	0:3	IHG	BOS
15:23 04-08-2021	TEG	1:3	IHG	BOS
15:39 03-08-2021	RGO	0:3	PDW	BOS
15:36 28-07-2021	KIKI	2:3	IHG	BOS
15:42 27-07-2021	TEG	1:3	PDW	BOS

13 Widok zawierający listę z wynikami meczów w turnieju

Przedstawiona na powyższym zrzucie ekranu funkcjonalność pozwala użytkownikowi na sprawdzanie wyników wszystkich meczów odbytych podczas wybranego wcześniej turnieju.

Dla każdej pozycji możemy zobaczyć datę jej rozpoczęcia oraz liczbę gier zwyciężonych przez każdą z drużyn. Dodatkową pozycją opisującą każdy z wierszy jest tryb, czyli czynnik opisujący ilość gier odbytych w ramach meczu.



























Większość meczy w e-sporcie odbywa się w formacie Best of (BO), polegającym na zdefiniowaniu maksymalnej ilości gier w meczu. W tym trybie zwycięża drużyna, która jako pierwsza uzyska większość punktów zdefiniowanych w tym formacie.






Przykładem jest na przykład BO5, czyli format w którym zwycięża drużyna uzyskująca jako pierwsza 3 punkty.

## 2.8 Sprawdzanie statystyk zawodników w turnieju

NAZWA PRZYPADKU UŻYCIA	Sprawdzanie statystyk zawodników w turnieju	ID.
		8.
SCENARIUSZ		
KROK	AKCJA	
1. Aktor	Aktor uruchamia ekran główny aplikacji wpisując jej adres w przeglądarkę	
2. System	Wyświetla loga gier wspieranych przez ERTS.	
3. Aktor	Aktor wybiera interesujący go tytuł (League of Legends)	
4. System	Wyświetla listę zawierającą loga i nazwy wszystkich lig wybranej gry	
5. Aktor	Aktor wybiera kartę ligi, której serie chce wyświetlić	
6. System	System wyświetla listę serii z wybranej ligi	
7. Aktor	Aktor wybiera serie z listy, której turnieje chce wyświetlić	
8. System	System wyświetla listę turniejów z wybranej serii	
9. Aktor	Aktor wybiera turniej z listy	
10. System	System wyświetla tabele wyników wybranego turnieju	
11. Aktor	Aktor wybiera opcje „STATYSTYKI ZAWODNIKÓW” z nawigacji turnieju	
12. System	System wyświetla listę zawierającą statystyki zawodników w turnieju	

ERTS

TABELA WYNIKÓW		MECZE					STATYSTYKI ZAWODNIKÓW					STATYSTYKI DRUŻYN		
Drużyna	Gracz	K	D	A	KDA	CS	CS/M	G	G/M	KP	DMGS	CP	Postacie	
	Armut	2	3	0	0,67	100	2,73	444	12,1	20%	20%	11		
	Kaiser	2	3	0	0,67	100	2,73	444	12,1	20%	20%	11		
	Elyoya	2	3	0	0,67	100	2,73	444	12,1	20%	20%	11		
	UNFORGIVEN	2	3	0	0,67	100	2,73	444	12,1	20%	20%	11		
	RKR	2	3	0	0,67	100	2,73	444	12,1	20%	20%	11		
	NUCLEARINT	2	3	0	0,67	100	2,73	444	12,14	25%	25%	7		
	LIMIT	2	3	0	0,67	100	2,73	444	12,14	25%	25%	6		
	Neon	2	3	0	0,67	100	2,73	444	12,14	25%	25%	7		
	Kirei	2	3	0	0,67	100	2,73	444	12,14	25%	25%	7		
	Hyllsang	2	3	0	0,67	100	2,4	444	10,67	20%	20%	11		
	Upset	2	3	0	0,67	100	2,4	444	10,67	20%	20%	11		
	Wunder	2	3	0	0,67	100	2,4	444	10,67	20%	20%	11		
	Niszy	2	3	0	0,67	100	2,4	444	10,67	20%	20%	11		

14 Widok zawierający statystyki zawodników w turnieju

Jest to kolejna funkcjonalność dotycząca turniejów w aplikacji. W porównaniu do poprzednich przypadków użycia nie jest ona dostępna dla wszystkich tytułów. Powodem tego jest to, że obecna wersja systemu magazynuje dane tylko o szczegółowych statystykach League of Legends.

Ta funkcjonalność oferuje nam przeglądanie statystyk wszystkich zawodników biorących udział w turnieju. Przedstawione są one w formie tabeli, w którym każdy z wierszy zawiera:

- Logo drużyny, po kliknięciu na które system przekierowuje do jej widoku(Drużyna)
- Pseudonim gracza (Gracz)
- Średnia liczba zabójstw na mecz (K)
- Średnia liczba śmierci na mecz (D)
- Średnia liczba asyst na mecz (A)
- Stosunek średniej sumy zabójstw i asyst do liczby śmierci (KDA)
- Średnia liczba pokonanych stworów w meczu (CS)
- Średnia liczba pokonanych stworów na minutę (CS/M)
- Średnia liczba zdobytego złota na mecz (G)
- Średnia liczba zdobytego złota na minutę (G/M)
- Procentowy udział w zabójstwach drużyny (KP)
- Procentowy udział w obrażeniach zadanych, przez drużynę (DMGS)
- Liczba wybranych unikalnych postaci podczas turnieju (CP)
- Miniaturki trzech najczęściej wybieranych w turnieju postaci (Postacie)



## 2.9 Sprawdzanie statystyk drużyn w turnieju

NAZWA PRZYPADKU UŻYCIA	Sprawdzanie statystyk drużyn w turnieju	ID.
		9.
SCENARIUSZ		
KROK	AKCJA	
1. Aktor	Aktor uruchamia ekran główny aplikacji wpisując jej adres w przeglądarkę	
2. System	Wyświetla loga gier wspieranych przez ERTS.	
3. Aktor	Aktor wybiera interesujący go tytuł (League of Legends)	
4. System	Wyświetla listę zawierającą loga i nazwy wszystkich lig wybranej gry	
5. Aktor	Aktor wybiera kartę ligi, której serie chce wyświetlić	
6. System	System wyświetla listę serii z wybranej ligi	
7. Aktor	Aktor wybiera serie z listy, której turnieje chce wyświetlić	
8. System	System wyświetla listę turniejów z wybranej serii	
9. Aktor	Aktor wybiera turniej z listy	
10. System	System wyświetla tabele wyników wybranego turnieju	
11. Aktor	Aktor wybiera opcje „STATYSTYKI DRUŻYN” z nawigacji turnieju	
12. System	System wyświetla listę zawierającą statystyki drużyn w turnieju	



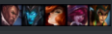

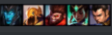

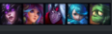

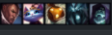

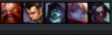

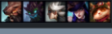





ERTS 

TABELA WYNIKÓW	MECZE		STATYSTYKI ZAWODNIKÓW				STATYSTYKI DRUŻYN		
Drużyna	K	D	A	Złoto	Smoki	Heraldy	Barony	Wieża	Postacie
	10	15	0	2220	3	0	0	3	
	10	15	0	2220	3	0	0	3	
	8	12	0	1776	3	0	0	3	
	10	15	0	2220	3	0	0	3	
	10	15	0	2220	3	0	0	3	
	10	15	0	2220	3	0	0	3	

15 Widok zawierający statystyki drużyn w turnieju

Jest to ostatnia z czterech funkcjonalności dotycząca turniejów w aplikacji. Podobnie jak poprzednia dotyczy ona tylko gry League of Legends, a dla pozostałych tytułów jest ona zablokowana.

Jak wynika ze scenariuszy, możemy dostać się do niej z poziomu dowolnego widoku związanego z turniejem, poprzez wybór odpowiedniej karty z menu znajdującym się na górze każdego z tego typu widoków. Dzięki temu rozwiązaniu można w szybki i wygodny sposób przełączać się pomiędzy scenami związanymi z tą strukturą rozgrywek.


Ta funkcjonalność jest podobna do poprzedniej, lecz zamiast statystyk zawodników, pokazuje ona użytkownikowi dane o drużynach w turnieju. Dla każdej drużyny widok wyświetla:

- Logo drużyny, po kliknięciu na które system przekierowuje do jej widoku(Drużyna)
- Średnia suma liczby zabójstw zawodników drużyny na mecz (K)
- Średnia suma liczby zabójstw zawodników drużyny śmierci na mecz (D)
- Średnia suma liczby zabójstw zawodników drużyny asyst na mecz (A)
- Średnia suma zdobytego złota przez zawodników na mecz (Złoto)
- Średnia liczba pokonanych smoków na mecz (Smoki)
- Średnia liczba pokonanych heraldów na mecz (Heraldy)
- Średnia liczba pokonanych baronów na mecz (Barony)
- Średnia liczba zniszczonych wież na mecz (Wieża)
- Miniaturki pięciu najczęściej wybieranych w turnieju postaci (Postacie)

## 2.10 Sprawdzanie informacji o wybranej drużynie


NAZWA PRZYPADKU UŻYCIA	Sprawdzanie informacji o wybranej drużynie	ID.
		10.
SCENARIUSZ		
KROK	AKCJA	
1. Aktor	Aktor uruchamia ekran główny aplikacji wpisując jej adres w przeglądarkę	
2. System	Wyświetla loga gier wspieranych przez ERTS.	
3. Aktor	Aktor wybiera interesujący go tytuł	
4. System	Wyświetla listę zawierającą loga i nazwy wszystkich lig wybranej gry	
5. Aktor	Aktor naciska na napis „Ligi” znajdujący się w nawigacji	
6. System	Wyświetla listę z możliwych opcji do wyświetlenia	
7. Aktor	Aktor wybiera opcje „Drużyny”	
8. System	Wyświetla listę zawierającą loga i nazwy wszystkich drużyn wybranej gry	
9. Aktor	Aktor wybiera kartę wybranej z drużyny	
10. System	Wyświetla ekran zawierający informacje o wybranej drużynie	


ERTS





Team Vitality  
VIT


Roster


 Milica


 Aljoša Kovandžić


 Perkz


 Luka Perković


 Alphari


 Barney Morris


 Crowshot


 Juš Marušić


 Carzzy


 Matyáš Orság


 Labrov

 Labros Papoutsakis


 Selfmade


 Oskar Boderek


 LIDER


 Adam Byasov


Last matches


 15:02 15-08-2021


 Fnatic 3:2 Team Vitality


 15:57 01-08-2021


 Team Vitality 1:0 Astralis


 17:10 31-07-2021

 Team Vitality 1:0 Rogue


 16:01 30-07-2021


 Team Vitality 1:0 FC Schalke 04 Esports


 15:52 24-07-2021


 Team Vitality 0:1 Excel Esports


Upcoming matches


 17:00 14-01-2022


 Team Vitality vs MAD Lions


 20:00 15-01-2022


 Team Vitality vs Fnatic


 16:00 16-01-2022

 Team Vitality vs Excel Esports

 20:00 21-01-2022

 Team Vitality vs BDS

 20:00 22-01-2022

 Team Vitality vs G2 Esports

CS:GO

LEAGUE OF LEGENDS

DOTA 2

OVERWATCH

VALORANT

16 Widok zawierający informacje o drużynie

26

Jest to funkcjonalność dostępna dla wszystkich tytułów w aplikacji, zawierająca informacje o wybranej przez użytkownika drużynie.

Jest to funkcjonalność, do której możemy się dostać nie tylko za pomocą akcji opisanych w scenariuszu, lecz z poziomu innych funkcjonalności również mamy możliwość przekierowania, właśnie do tego widoku. Z poziomu innych ekranów, do tej opcji możemy się dostać zazwyczaj poprzez naciśnięcie na logo danej drużyny.

Informacjami opisującymi drużynę w tym widoku są jej logo, nazwa oraz akronim, czyli skrócona nazwa drużyny. Poza nimi system wyświetla również listę zawodników, grających aktualnie w wybranej organizacji, zawierającą ich pseudonimy, imiona, nazwiska oraz flagi ich narodowości.

Ta funkcjonalność pokazuje również użytkownikowi pięć ostatnich i nadchodzących meczów drużyny. Dla tych struktur system wyświetla: logotyp ligi, datę rozpoczęcia meczu, oraz nazwy i loga drużyn biorących w nim udział. Dla meczów zakończonych informacja zawiera również ich rezultat w postaci liczby wygranych gier, przez każdego z uczestników spotkania.

## 2.11 Sprawdzanie statystyk meczu League of Legends

NAZWA PRZYPADKU UŻYCIA	Sprawdzanie statystyk meczu League of Legends	ID.
		11.
SCENARIUSZ		
KROK	AKCJA	
1. Aktor	Aktor uruchamia ekran główny aplikacji wpisując jej adres w przeglądarkę	
2. System	Wyświetla loga gier wspieranych przez ERTS.	
3. Aktor	Aktor wybiera interesujący go tytuł (League of Legends)	
4. System	Wyświetla listę zawierającą loga i nazwy wszystkich lig wybranej gry	
5. Aktor	Aktor wybiera kartę ligi, której serie chce wyświetlić	
6. System	System wyświetla listę serii z wybranej ligi	
7. Aktor	Aktor wybiera serie z listy, której turnieje chce wyświetlić	
8. System	System wyświetla listę turniejów z wybranej serii	
9. Aktor	Aktor wybiera turniej z listy	
10. System	System wyświetla tabele wyników wybranego turnieju	

11. Aktor	Aktor wybiera opcje „MECZE” z nawigacji turnieju
12. System	System wyświetla listę zawierającą wyniki meczów w turnieju
13. Aktor	Aktor wybiera mecz z listy, którego statystyki chce sprawdzić
14. System	System wyświetla statystyki meczu

**Game 1**  
Start time: 16:15 29-08-2021 Length : 38:11  
Blue team **Defeat**  
Gold: 10000 Dragons: 2:0 Bans: 3:0

	Nick	KDA	CS	Gold	Items
	Elyoya	2/3/0	100	444	
	Kaiser	2/3/0	100	444	
	RKR	2/3/0	100	444	
	Armut	2/3/0	100	444	
	UNFORGIVEN	2/3/0	100	444	

17 Widok zawierający statystyki meczu

Zadaniem tej funkcjonalności jest wyświetlanie użytkownikowi statystyk wybranego, przez niego meczu. Z powodu braku, przechowywania w aplikacji informacji o szczegółowych statystykach meczy dla wszystkich tytułów, ten widok jest dostępny tylko dla gry League of Legends.

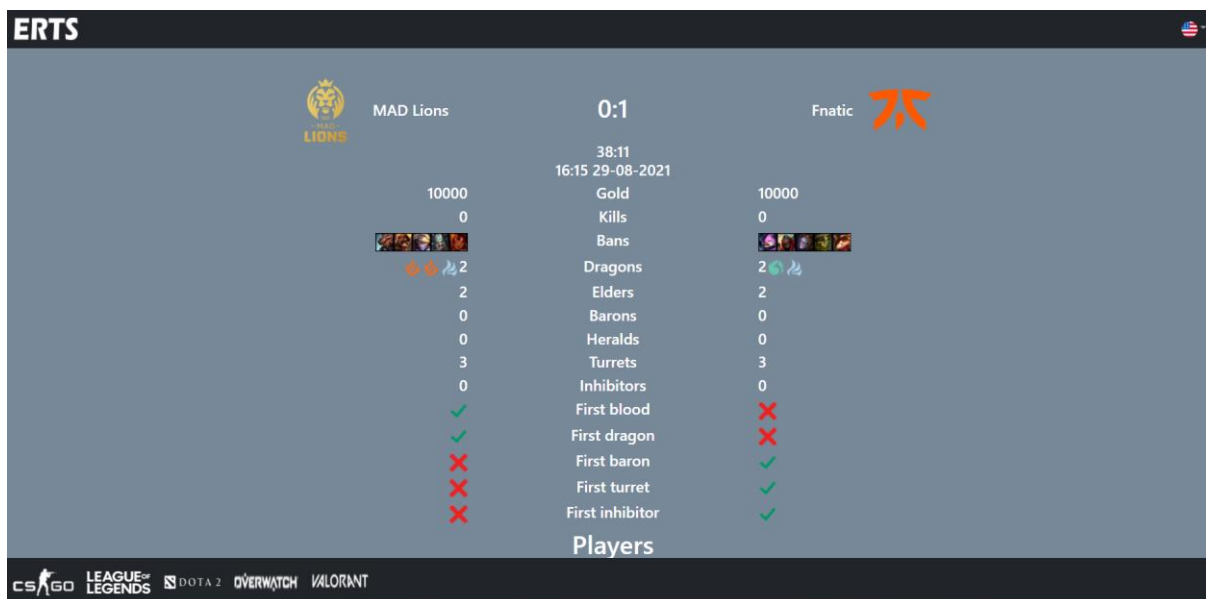
Do tej funkcjonalności możemy dostać na dwa sposoby, z których pierwszy opisany w scenariuszu, a drugim jest naciśnięcie na wybrany mecz z listy ostatnich meczów drużyny w ekranie opisanym w poprzednim punkcie.

Dla każdego z meczu system wyświetla użytkownikowi wynik meczu, w postaci liczby zwyciężonych gier przez drużyny, datę rozpoczęcia meczu oraz link do transmisji, na której mecz był możliwy do oglądania.

Widok zawiera również skrócone statystyki dotyczące każdej z gier odbytej podczas wybranego meczu. Skrócone statystyki gier zawierają między innymi informacje o rodzajach smoków pokonanych, przez drużynę pokazującą ikony zdobytych wzmocnień, oraz miniaturkę wybranego bohatera, zawierającą wybrane przez użytkownika czary przywoływacza, czyli dodatkowe umiejętności wybrane przez użytkownika.

## 2.12 Sprawdzanie statystyk gry League of Legends

NAZWA PRZYPADKU UŻYCIA	Sprawdzanie statystyk gry League of Legends	ID.
		12.
SCENARIUSZ		
KROK	AKCJA	
1. Aktor	Aktor uruchamia ekran główny aplikacji wpisując jej adres w przeglądarkę	
2. System	Wyświetla loga gier wspieranych przez ERTS.	
3. Aktor	Aktor wybiera interesujący go tytuł (League of Legends)	
4. System	Wyświetla listę zawierającą loga i nazwy wszystkich lig wybranej gry	
5. Aktor	Aktor wybiera kartę ligi, której serie chce wyświetlić	
6. System	System wyświetla listę serii z wybranej ligi	
7. Aktor	Aktor wybiera serie z listy, której turnieje chce wyświetlić	
8. System	System wyświetla listę turniejów z wybranej serii	
9. Aktor	Aktor wybiera turniej z listy	
10. System	System wyświetla tabele wyników wybranego turnieju	
11. Aktor	Aktor wybiera opcje „MECZE” z nawigacji turnieju	
12. System	System wyświetla listę zawierającą wyniki meczów w turnieju	
13. Aktor	Aktor wybiera mecz z listy, którego statystyki chce sprawdzić	
14. System	System wyświetla statystyki meczu	
15. Aktor	Aktor wybiera grę, której statystyki chce sprawdzić naciskając na jej nazwę (np. Gra 2)	
16. System	System wyświetla statystyki gry	



18 Widok zawierający statystyki gry

Jest to kolejna funkcjonalność dostępna tylko dla tytułu League of Legends, której zadaniem jest wyświetlanie użytkownikowi rozbudowanych statystyk o wybranej grze.

Jak wynika ze scenariusza możemy się do niej dostać klikając na nazwę gry opisaną w widoku meczu.

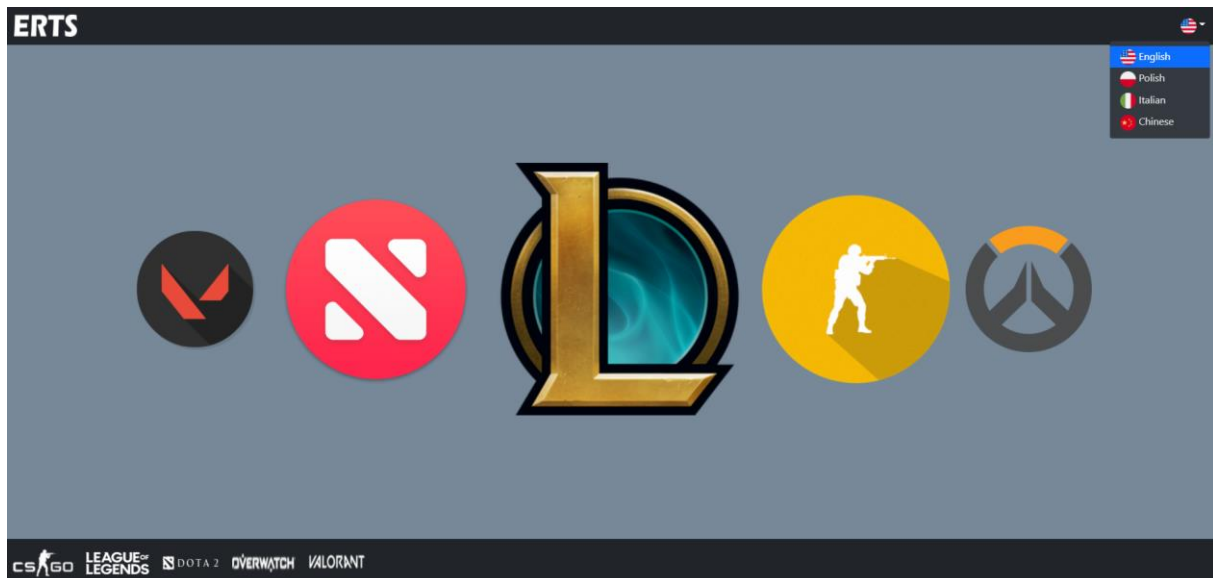
Ten widok zawiera wszystkie właściwości gry wyświetlane w poprzednim widoku rozszerzone o bardziej szczegółowe elementy, takie jak zadane różne typy obrażeń, przez gracza takie jak obrażenia magiczne oraz fizyczne. Jedną z ciekawszych części tego widoku jest fragment zawierający informacje o statystykach wykluczających się dla obu drużyn, takich jak pierwsza zniszczona wieża, która zawsze jest zdobyta tylko przez jedną z drużyn.

Tego typu właściwości pokazywane są poprzez wyświetlanie specjalnych symboli widocznych na rzucie ekranu, które symbolizują dla której z drużyn podana akcja wystąpiła.

## 2.13 Zmiana języka aplikacji

NAZWA PRZYPADKU UŻYCIA	Zmiana języka aplikacji	ID.
		13.
SCENARIUSZ		
KROK	AKCJA	
1. Aktor	Aktor wchodzi na dowolną podstronę aplikacji i naciska na flagę po prawej stronie górnej nawigacji aplikacji	
2. System	System wyświetla języki możliwe do wyboru	

3. Aktor	Aktor wybiera język z listy
4. System	System tłumaczy aplikacje na wybrany język



19 Przykładowa podstrona ze zmienionym językiem na Włoski

Jest to dodatkowa funkcjonalność aplikacji dzięki której możemy zmienić język na inny obsługiwany przez aplikacje. Do tej funkcjonalności możemy dostać się z poziomu każdej podstrony, poprzez wybranie języka z wysuwanego menu, które pojawia się po naciśnięciu na flagę znajdującą się po prawej stronie nagłówka aplikacji.

Obecnie aplikacja oferuje cztery różne języki, jakimi są: Polski, Angielski, Włoski oraz Chiński.

Główną zaletą tej funkcjonalności jest to, że czyni ona aplikacje dostępną również dla ludzi posługujących się innym językiem niż domyślny.

## 3. Wybranie rozwiązania technologiczne

### 3.1 C#

#### Opis

C# to nowoczesny język programowania stworzony przez firmę Microsoft.

Jest to język zorientowany na programowanie obiektowe, co oznacza, że program napisany w tym języku składa się z obiektów komunikujących się pomiędzy sobą podczas działania aplikacji. Dzięki temu podejściu kod jest bardziej czytelny oraz łatwiejszy w konserwacji. (Eckelo, 2006)



Jest to język wysokopoziomowy co oznacza, że składnia i słowa kluczowe mają być zrozumiałe dla człowieka i zbliżone do języka naturalnego, zwiększając przy tym poziom abstrakcji, co dystansuje użytkownika od spraw związanych ze sprzętem. Przez to, że jest to język wysokopoziomowy nie jest on bezpośrednio zrozumiały przez komputer, przez co wymaga on kompilacji, czyli „przetłumaczenia” na język zrozumiały dla komputera, co negatywnie wpływa na jego wydajność. (BBC, 2021)

C# posiada wiele przydatnych udogodnień czyniących programowanie bardziej wygodnym. Jednym z nich jest moduł automatycznie usuwający nieużywane obiekty z pamięci podczas działania aplikacji, który gwarantuje wydajniejszą pracę aplikacji, bez ingerencji programisty. Kolejną przydatną funkcjonalnością jest wbudowany język zapytań o nazwie LINQ, który pozwala użytkownikowi odwoływać się do bazy danych z aplikacji, bez użycia SQL. (Microsoft, 2021)

C# jest językiem stworzonym głównie do tworzenia aplikacji na systemy Windows, lecz z czasem została wprowadzona możliwość programowania w tym języku praktycznie na każdej platformie.

C# jest obecnie jednym z najczęściej używanych języków i zajmuje topowe pozycje w wielu rankingach popularności języków programowania:

Dane aktualne na dzień 10.01.2022r.

- Według rankingu TIOBE index zajmuje on 5 miejsce pod względem popularności (Tiobe, 2022)
- W rankingu Spectrum zajmuje on 6 miejsce (IEEE Spectrum, 2022)
- W serwisie Stack Overflow jest 4 najczęściej tagowanym językiem programowania posiadając ponad półtora miliona zapytań (stack overflow, 2022)

C# jest wciąż rozwijany przez firmę Microsoft, która wydała jego ostatnią wersję (C# 10.0), w Listopadzie 2021r.

#### Zastosowanie w aplikacji

Do stworzenia projektu wybrałem C#, ponieważ jest to idealny język do tworzenia aplikacji webowych, przy pomocy .Net Core.

Przy pomocy tego języka napisałem cały backend aplikacji, oraz serwis odpowiedzialny za pobieranie danych z api Pandascore.

## 3.2 .Net Core

### Opis

.Net Core to framework, czyli platforma programistyczna, definiująca ogólną strukturę aplikacji i mechanizm jej działania. Został on stworzony przez Firmę Microsoft i służy między innymi do tworzenia aplikacji przeznaczonych dla chmury obliczeniowej i IoT, aplikacji klasycznych, oraz aplikacji internetowych z użyciem wzorca MVC. Do tworzenia aplikacji w tym oprogramowaniu najczęściej używa się języka C#. Ważną właściwością .Net Core jest to, że posiada on oficjalny menadżer pakietów NuGet, dzięki któremu możemy pobierać różne przydatne biblioteki stworzone przez Microsoft, oraz przez społeczność. (Microsoft, 2021)

### Zastosowanie w aplikacji

W mojej aplikacji zdecydowałem się użyć tego rozwiązania, ponieważ jest to bardzo dobre i popularne oprogramowanie idealnie nadające się do tworzenia aplikacji webowych. Dodatkowo dzięki tej platformie mogłem użyć wiele przydatnych pakietów takich jak pakiety potrzebne do korzystania z narzędzia Hangfire w aplikacji.

Najważniejszym z użytych pakietów był pakiet Entity framework core, który umożliwił mi między innymi korzystanie z obiektów pochodzących z bazy danych za pomocą obiektów .NET.

## 3.3 React

### Opis

React to bardzo popularna biblioteka języka JavaScript, wykorzystywana podczas tworzenia interfejsu użytkownika w aplikacji. Jest ona zastosowana między innymi, przez aplikacje takie jak: Netflix, Facebook, Instagram i Pinterest. Jest to biblioteka dostępna na zasadzie open source, przez co każdy może korzystać z niej za darmo.

React wykorzystuje nowoczesny sposób generowania stron internetowych, dzięki czemu stają się one bardziej dynamiczne. Takie podejście powoduje to, że interfejs jest wygodny zarówno dla programisty, jak i dla końcowego użytkownika aplikacji. (Nius, 2020)

React ma wiele zalet, ale najważniejszymi z nich są:

- Wygoda i szybkość implementacji
- Strony aktualizują się w czasie rzeczywistym
- Jest wydajny
- Jest popularny, przez co ma dużą społeczność

#### Zastosowanie w aplikacji

React JS zastosowałem do stworzenia interfejsu użytkownika aplikacji. Wybrałem tę bibliotekę, ponieważ pozwala ona na stworzenie szybko działającego UI (User Interface), który pozwala dynamicznie zmieniać ekrany na stronie bez jej odświeżania, które czasami trwa dużo czasu. Dodatkowo programowanie w React JS jest znacznie szybsze i wygodniejsze niż pisanie frontendu w czystym JavaScriptcie, lub HTMLu.

### 3.4 TypeScript

#### Opis

TypeScript to stosunkowo nowy język programowania, gdyż jego pierwsza wersja została wypuszczona w 2012 roku przez firmę Microsoft. Jest on semantycznym nadzbiorem jednego z najbardziej popularnych języków programowania, którym jest JavaScript. Oznacza to, że składnie tych języków są bardzo podobne, lecz TypeScript posiada kilka dodatkowych właściwości niewystępujących w składni jego poprzednika. Główną różnicą między nimi jest to, że TypeScript posiada statyczne typowanie, co czyni kod bardziej przejrzystym. Ciekawą właściwością TS jest to, że jest on kompilowany do czystego JavaScriptu, przez co przeglądarka nie jest w stanie rozróżnić tych dwóch języków. (Ćwik, 2019) (Wikipedia, 2021)

#### Zastosowanie w aplikacji

W aplikacji TypeScripta użyłem podczas pisania front-endu aplikacji, zastępując nim klasyczny JavaScript. Wybrałem go zamiast JS, ponieważ jest on nowocześniejszym językiem, który zyskuje coraz większą popularność i rozwiązuję moim zdaniem duży problem JavaScripta jakim jest brak kontroli typów. Takie rozwiązanie czyni kod bardziej przejrzystym i łatwiejszym w konserwacji.

Dla użytej przeze mnie technologii React domyślnym językiem jest JavaScript, więc w celu instalacji tego języka we front-endzie aplikacji użyłem specjalnego szablonu, tworzącego szkielet aplikacji wykorzystującego TypeScript.

```
npx create-react-app my-app --template typescript
```

*20 Komenda tworząca aplikację reactową z użyciem szablonu TypeScripta*

Szablon wykorzystujący język TypeScript wdrożyłem do aplikacji, poprzez dołączenie argumentu `--template typescript` do komendy tworzącej szkielet Reactowej aplikacji, którą wpisałem w terminalu projektu, zawierającego front-end aplikacji (ErtsWebApp).

### 3.5 Bootstrap

#### Opis

Bootstrap to najbardziej popularny na świecie Open-sourcowy zestaw narzędzi ułatwiający tworzenie widoku aplikacji. Pozwala on znacznie szybciej tworzyć front-end aplikacji, oferując przy tym dobrze wyglądające gotowe komponenty takie jak przyciski i tabele.

Bootstrap również dba o wygląd aplikacji na mniejszych urządzeniach takich jak telefony, tworząc skalowalne fragmenty widoku. (Bootstrap, 2021)

#### Zastosowanie aplikacji

W aplikacji użyłem wersji Bootstrapa przystosowanej do pracy z Reactem (React Bootstrap). Został on użyty w tworzeniu front-endu aplikacji, co znacznie ułatwiło tworzenie części wizualnej aplikacji.

Narzędzie zainstalowałem w projekcie, za pomocą NPM (Node Package Module), czyli systemu paczek dla środowiska Node.

```
npm install react-bootstrap bootstrap@ 5.1 .3
```

*21 Komenda instalująca pakiet bootstrapa w aplikacji*

Dzięki temu narzędziu instalacja Bootstrapa sprowadzała się do wpisania powyższej komendy w terminalu z poziomu projektu zawierającego front-end aplikacji.

Po zainstalowaniu powyższego pakietu mogłem swobodnie korzystać z wszystkich gotowych komponentów oferowanych przez ten zestaw narzędzi, poprzez importowanie ich wewnątrz klas z rozszerzeniem .tsx.

## 3.6 PandaScore api

#### Opis

PandaScore to dostawca statystyk meczów e-sportowych w czasie rzeczywistym. Oferuje on REST API, dzięki któremu za pomocą odpowiednich zapytań możemy uzyskać potrzebne nam dane takie jak wyniki meczów, informacje o drużynach oraz turniejach. Api obsługuje większość popularnych gier e-sportowych, włącznie ze wszystkimi zawartymi w aplikacji. (PandaScore, 2021)

## Zastosowanie w aplikacji

PandaScore został użyty w aplikacji do pozyskiwania wyników meczów, oraz wszelkiego rodzaju statystyk oraz informacji o obiektach takich jak drużyny oraz gracze. W aplikacji użyłem darmowej wersji api, ponieważ płatne wersje kosztują od 150€ za miesięczny abonament. Z tego powodu aplikacja może pozyskać ograniczoną ilość danych na godzinę, przez co pozyskiwanie danych zostało rozłożone w czasie. Dodatkowo w darmowej wersji nie mam dostępu do wszystkich danych, więc dane o meczach i grach League of Legends w celu prezentacji zostały wypełnione przykładowymi danymi.



23 Przykładowe zapytanie

```
1  [
2    {
3      "acronym": null,
4      "current_videogame": "LoL",
5      "id": 26,
6      "name": "Valorant",
7      "slug": "valorant"
8    },
9    {
10     "id": 129908,
11     "image_url": "https://cdn.pandascore.co/images/team/image/129908/600px_kpi_2019_lightmode.png",
12     "location": "RU",
13     "modified_at": "2021-11-26T12:18:46Z",
14     "name": "KPI Gaming",
15     "players": [
16       {
17         "birth_year": 1998,
18         "birthday": "1998-06-13",
19         "first_name": "Aleksandr",
20         "hometown": null,
21         "id": 32852,
22         "image_url": null,
23         "last_name": "Eremin",
24         "name": "hvoya",
25         "nationality": "RU",
26         "role": null,
27         "slug": "hvoya"
28       },
29       {
30         "birth_year": 2001,
31         "birthday": "2001-01-25",
32         "first_name": "Artem",
33         "hometown": null,
34         "id": 32858,
```

22 Przykładowa odpowiedź

Na powyższych fragmentach ekranu, widać przykładowe zapytanie, oraz odpowiedź pozyskaną z api PandaScore. Podane zapytanie zwraca informacje o drużynach gry Valorant.

## 3.7 HangFire

### Opis

Hangfire to darmowa biblioteka, która ułatwia kolejkowanie czasochłonnych operacji i wykonywanie ich w tle. Oferuje ona również dobrze wyglądający panel, który ułatwia śledzenie użytkownikowi zakolejkowane operacje. (Hangfire, 2021)

### Zastosowanie w aplikacji

Hangfire użyłem w aplikacji, do zakolejkowania operacji pobierania danych z api. Dzięki tej bibliotece mogłem łatwo ustawić czasy wykonywania cyklicznych operacji, mając przy tym pełny wgląd w ich status dzięki wygodnemu interfejsowi użytkownika oferowanym przez Hangfire.

Sposób zaimplementowania tego narzędzia w aplikacji opisałem w części opisującej podprojekt ErtsApiFetcher w piątym rozdziale pracy.

## 3.8 PostgreSQL

### Opis

PostgreSQL to obok MySQL i SQLite jeden z trzech najpopularniejszych darmowych systemów zarządzania relacyjnymi bazami danych. Oznacza to, że umożliwia on manipulację danymi zgromadzonymi w DB oraz ich aktualizację. (Wikipedia, 2021)

Najważniejszymi cechami SZBD są:

- Jak najszybsze dostarczenie danych użytkownikowi
- Niezależność danych od aplikacji
- Ochrona danych przed niepowołanym dostępem
- Możliwość odtworzenia bazy danych po awarii
- Uwzględnienie standardu języka zapytań

(Klucz-system, 2020)

### Zastosowanie w aplikacji

W moim projekcie zastosowałem PostgreSQL do zarządzania bazą danych z poziomu aplikacji. Wybrałem to rozwiązanie, ponieważ jest popularne, oraz dobrze zoptymalizowany z .Net Core, oraz ze wszystkimi użytymi modułami.

Ten system zainstalowałem za pomocą technologii Docker, która umożliwia uruchomienie usługi wewnątrz specjalnego kontenera, zamiast bezpośrednio na urządzeniu. Kontener w tej technologii to odizolowane środowisko, które zawiera wszystkie zależności potrzebne do działania programu. Ważną właściwością tej struktury jest to, że nie posiada ona własnej pamięci trwałej i w momencie ich ponownego uruchomienia jest tworzony od nowa. Z tego powodu wyłączone kontenery nie zajmują miejsca na dysku. Mimo tego w momencie restartu nie tracimy danych przechowywanych w bazie danych programu, ponieważ są one zapisywane w specjalnych Volumach, przypisanych do konkretnych kontenerów.

W celu utworzenia kontenera zawierającego PostgreSQL utworzyłem plik docker-compose zawierający konfigurację SZBD i uruchomiłem go komendą docker-compose up.

Dzięki zastosowaniu tej technologii mogłem przyspieszyć proces konfiguracji tego systemu, oraz przez to, że został on utworzony, poza moim komputerem nie jest on od niego zależny i podatny na problemy związanymi ze zmianami w moim PC.

Do kontroli kontenerów w aplikacji użyłem dedykowanego przez twórcę tej technologii, interfejsu graficznego o nazwie Docker Desktop, dzięki któremu mogłem łatwiej zarządzać kontenerem, bez używania specjalnych komend tekstowych takich jak docker container ls, wyświetlającej listę wszystkich utworzonych kontenerów.

Do manualnej kontroli danych w db użyłem programu DBeaver, służącego do administrowania bazą danych i oferującego przejrzysty interfejs, pozwalający na przeglądanie danych systemu.

## 3.9 Git

### Opis

Git to darmowe oprogramowanie, będące systemem kontroli wersji. Pozwala one na śledzenie zmian w kodzie aplikacji, oraz umożliwia ona łączenie zmian w projekcie w przypadku pracy wielu programistów. Korzystanie z tej technologii opiera się głównie na tworzeniu i udostępnianiu commitów projektu, czyli wersji projektu zawierających wprowadzone przez programistę zmiany. (Git, 2021)

### Zastosowanie aplikacji

Gita użyłem do przechowywania poszczególnych wersji aplikacji oraz do zarządzania nimi.

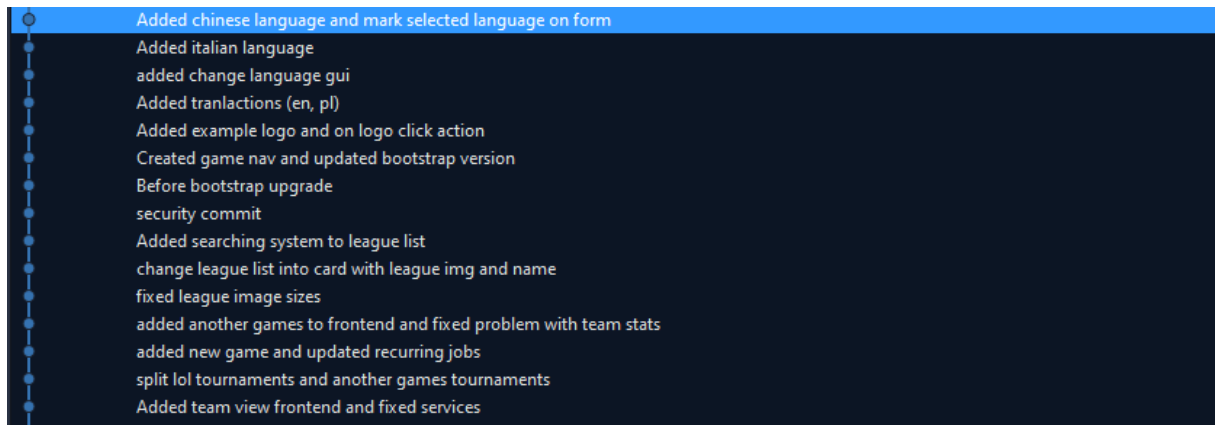
Wraz z tą technologią użyłem również serwisu GitHub, będącego hostingiem wykorzystującym tą właśnie technologię. Dzięki temu mogę bez problemu pobrać projekt na każdym komputerze, co daje gwarancje nie utracenia kodu w przypadku awarii komputera. (Wikipedia, 2021)

Dodatkową ciekawą funkcjonalnością tego serwisu są prowadzone w nim statystyki dotyczące repozytorium projektu. Zawierające one między innymi liczbę commitów, oraz zmian w kodzie w określonych odstępach czasowych.

Aby użyć opisaną powyżej technologię na początku utworzyłem repozytorium w serwisie GitHub, a następnie dodałem do niego utworzony projekt. Podczas procesu tworzenia pracy po dodaniu nowej funkcjonalności lub przed przerwą w pisaniu pracy, dodawałem nowe commity, w celu zabezpieczenia się przed utratą danych w wyniku na przykład problemów z komputerem. Inną zaletą regularnego powtarzania tej czynności było utworzenie struktury,

dzięki której w przypadku nieoczekiwanych problemów, mogłem łatwo znaleźć ich źródło, sprawdzając różnice pomiędzy obecną wersją, a starymi stanami projektu.

W celu ułatwienia korzystania z tego narzędzia użyłem również programu Sourcetree, oferującego wygodny interfejs oferujący przejrzystą wizualizację zmian pomiędzy wersjami systemu, oraz pozwalający na wygodne przeglądanie historii wersji aplikacji.



24 Fragment listy commitów aplikacji wygenerowany przez Sourcetree

Na powyższym zrzucie ekranu widać kilka przykładowych commitów projektu. Ich nazwy zwykle odpowiadają funkcjonalnością w ramach ich dodanym lub celom dla których zostały utworzone.

## 4. Architektura i implementacja systemu

Aplikacja podzielona jest na kilka projektów, z których dwa są niezależnymi od siebie serwisami (ErtsWebApp i ErtsApiFetcher), a pozostałe to biblioteki dostarczające do nich niezbędne funkcjonalności.

### 4.1 ErtsModel

#### Wstęp

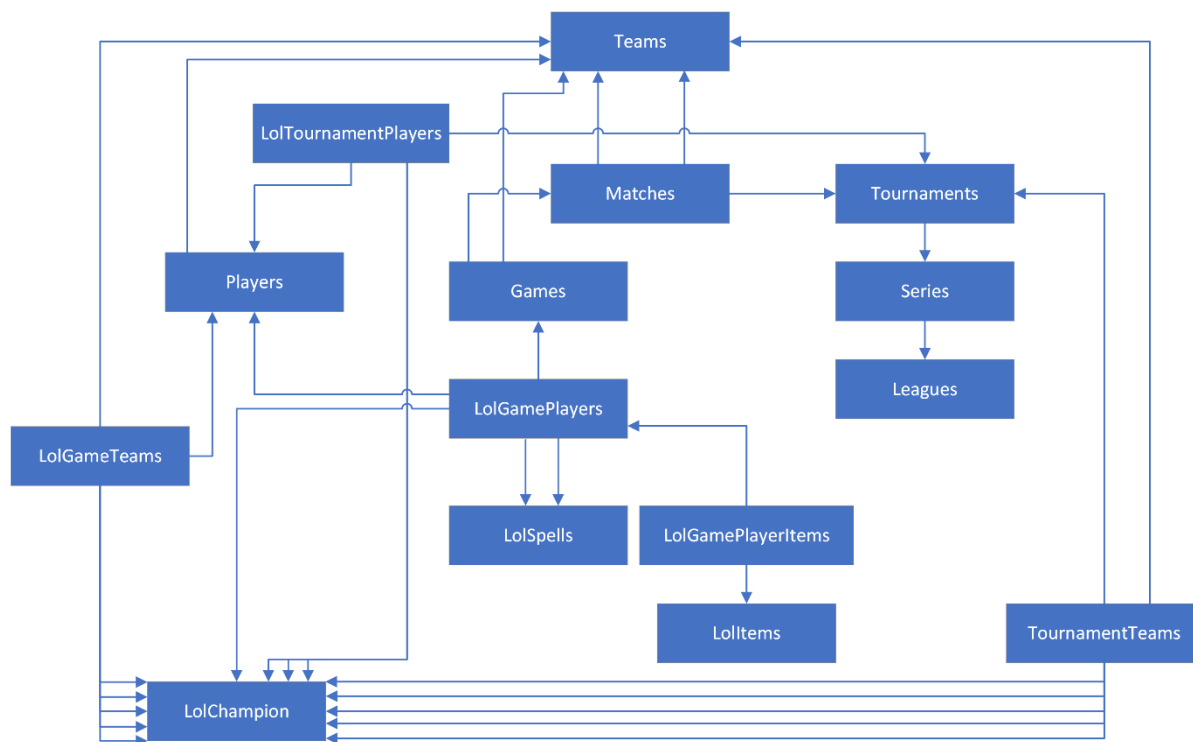
ErtsModel to pierwszy z projektów, mojej aplikacji, który jest biblioteką dostarczającą klasy dotyczące bazy danych aplikacji. Ten projekt jest odpowiedzialny za utworzenie schematu bazy danych aplikacji za pomocą podejścia Code First, które oznacza utworzenie schematu bazy danych przy pomocy kodu aplikacji, nie pisząc przy tym własnego skryptu w s SQL.



## Schemat bazy danych

### Diagram związków encji

Na diagramie poniżej przedstawione są przedstawione wszystkie encje w bazie danych aplikacji, oraz relacje pomiędzy nimi. Na schemacie nie umieściłem atrybutów ze względu na jego czytelność. W dalszej części rozdziału zostaną opisane wszystkie encje wraz z ich atrybutami.



25 Diagram związków encji niezawierający atrybutów

### Drużyny

NAZWA ENCJI		Teams	ID.
			1.
OPIS		Obiekty zawierające informacje o drużynach	
ATRYBUTY			
KLUCZ	NAZWA	Krótki opis	TYP
PK	Id	Identyfikator	Int8
	Name	Nazwa drużyny	Text
	GameType	Typ gry (np. League of Legends)	Text
	ImageUrl	Link do loga drużyny	Text
	Acronym	Skrócona nazwa drużyny	Text
	Apild	Identyfikator w PandaScore api	Int4

### Zawodnicy

NAZWA ENCJI		Players	ID.
			2.
OPIS		Obiekty zawierające informacje o zawodnikach	
ATRYBUTY			
KLUCZ	NAZWA	Krótki opis	TYP
PK	Id	Identyfikator	Int8
	Nick	Pseudonim zawodnika	Text
	Nationality	Narodowość zawodnika	Text
FK	TeamId	Identyfikator drużyny, dla której gra zawodnik	Int8
	FirstName	Imię zawodnika	Text
	LastName	Nazwisko zawodnika	Text
	Apild	Identyfikator w PandaScore api	Int4

### Ligi

NAZWA ENCJI		Leagues	ID.
			3.
OPIS		Obiekty zawierające informacje o ligach	
ATRYBUTY			
KLUCZ	NAZWA	Krótki opis	TYP
PK	Id	Identyfikator	Int8
	Name	Nazwa ligi	Text
	ImageUrl	Link do loga ligi	Text
	GameType	Typ gry (np. League of Legends)	Text
	Url	Link do strony ligi	Text
	Apild	Identyfikator w PandaScore api	Int4

### Serie

NAZWA ENCJI		Series	ID.
			4.
OPIS		Obiekty zawierające informacje o seriach	
ATRYBUTY			
KLUCZ	NAZWA	Krótki opis	TYP
PK	Id	Identyfikator	Int8
	Name	Nazwa serii	Text
	StartTime	Data rozpoczęcia serii	Timestamp
	EndTime	Data zakończenia serii	Timestamp
FK	LeagueId	Identyfikator ligi, w której odbywa się seria	Int8
	ApId	Identyfikator w PandaScore api	Int4

### Turnieje

NAZWA ENCJI		Tournaments	ID.
			5.
OPIS		Obiekty zawierające informacje o turniejach	
ATRYBUTY			
KLUCZ	NAZWA	Krótki opis	TYP
PK	Id	Identyfikator	Int8
	Name	Nazwa turnieju	Text
	StartTime	Data rozpoczęcia turnieju	Timestamp
	EndTime	Data zakończenia turnieju	Timestamp
FK	SeriesId	Identyfikator serii, w której odbywa się turniej	Int8
	Apild	Identyfikator w PandaScore api	Int4

### Mecze

NAZWA ENCJI		Matches	ID.
			6.
OPIS		Obiekty zawierające informacje o meczach	
ATRYBUTY			
KLUCZ	NAZWA	Krótki opis	TYP
PK	Id	Identyfikator	Int8
	StartTime	Data rozpoczęcia meczu	Timestamp
	EndTime	Data zakończenia meczu	Timestamp
FK	Team1Id	Identyfikator pierwszej drużyny, biorącej udział w meczu	Int8
FK	Team2Id	Identyfikator drugiej drużyny, biorącej udział w meczu	Int8
FK	TournamentId	Identyfikator turnieju, w którym odbywa się mecz	Int8
	StreamUrl	Link do transmisji meczu	Text
	NumberOfGames	Liczba gier w meczu	Int4
	ApilId	Identyfikator w PandaScore api	Int4

### Gry

NAZWA ENCJI		Games	ID.
			7.
OPIS		Obiekty zawierające informacje o grach	
ATRYBUTY			
KLUCZ	NAZWA	Krótki opis	TYP
PK	Id	Identyfikator	Int8
	StartTime	Data rozpoczęcia turnieju	Timestamp
	EndTime	Data zakończenia turnieju	Timestamp
FK	MatchId	Identyfikator meczu, w którym odbyła się gra	Int8

FK	WinnerTeamId	Identyfikator zwycięskiej drużyny	Int8
	Apild	Identyfikator w PandaScore api	Int4

### Postacie z League of Legends

NAZWA ENCJI		LolChampions	ID.
			8.
OPIS		Obiekty zawierające informacje o Postaciach z gry League of Legends	
ATRYBUTY			
KLUCZ	NAZWA	Krótki opis	TYP
PK	Id	Identyfikator	Int8
	Name	Nazwa postaci	Text
	ImageUrl	Link do miniaturki postaci	Text

### Czary z League of Legends

NAZWA ENCJI		LolSpells	ID.
			9.
OPIS		Obiekty zawierające informacje o czarach przywoływacza z gry League of Legends	
ATRYBUTY			
KLUCZ	NAZWA	Krótki opis	TYP
PK	Id	Identyfikator	Int8
	Name	Nazwa czaru	Text
	ImageUrl	Link do miniaturki czaru	Text

### Przedmioty z League of Legends

NAZWA ENCJI		LolItems	ID.
			10.
OPIS		Obiekty zawierające informacje o przedmiotach z gry League of Legends	
ATRYBUTY			
KLUCZ	NAZWA	Krótki opis	TYP
PK	Id	Identyfikator	Int8
	Name	Nazwa przedmiotu	Text

	ImageUrl	Link do miniaturki przedmiotu	Text
	IsTrinket	Czy przedmiot jest talizmanem	Bool

### Statystyki drużyn w grach League of Legends

NAZWA ENCJI		LolGameTeams	ID.
			11.
OPIS		Obiekty zawierające informacje o statystykach drużyn w grach League of Legends	
ATRYBUTY			
KLUCZ	NAZWA	Krótki opis	TYP
PK	Id	Identyfikator	Int8
FK	TeamId	Identyfikator drużyny	Int8
FK	Gameld	Identyfikator gry	Int8
	BaronKilled	Liczba pokonanych baronów	Int4
	MountainDrakeKilled	Liczba pokonanych smoków górskich	Int4
	InfernalDrakeKilled	Liczba pokonanych smoków piekielnych	Int4
	OceanDrakeKilled	Liczba pokonanych smoków oceaniczny	Int4
	CloudDrakeKilled	Liczba pokonanych smoków powietrznych	Int4
	ElderDrakeKilled	Liczba pokonanych starszych smoków	Int4
	HeraldKilled	Liczba pokonanych heraldów	Int4
	GoldEarned	Zarobione złoto	Int4
	Kills	Liczba zabójstw	Int4
	TurretDestroyed	Zniszczone wieże	Int4
	InhibitorDestroyed	Zniszczone inhibitory	Int4
	Color	Strona po której gra drużyna (czerwona/niebieska)	Int4
FK	Ban1Id	Identyfikator pierwszej zablokowanej postaci	Int8
FK	Ban2Id	Identyfikator drugiej zablokowanej postaci	Int8
FK	Ban3Id	Identyfikator trzeciej zablokowanej postaci	Int8
FK	Ban4Id	Identyfikator czwartej zablokowanej postaci	Int8

FK	Ban5Id	Identyfikator piątej zablokowanej postaci	Int8
	FirstBaron	Czy drużyna pokonała pierwszego barona	Bool
	FirstBlood	Czy drużyna zdobyła pierwsze zabójstwo	Bool
	FirstDragon	Czy drużyna pokonała pierwszego smoka	Bool
	FirstTurret	Czy drużyna zniszczyła pierwszą wieżę	Bool
	FirstInhibitor	Czy drużyna zniszczyła pierwszy inhibitor	Bool

### Statystyki graczy w grach League of Legends

NAZWA ENCJI		LolGamePlayers	ID.
			12.
OPIS		Obiekty zawierające informacje o statystykach zawodników w grach League of Legends	
ATRYBUTY			
KLUCZ	NAZWA	Krótki opis	TYP
PK	Id	Identyfikator	Int8
FK	Playerd	Identyfikator gracza	Int8
FK	Gameld	Identyfikator gry	Int8
FK	ChampionId	Identyfikator wybranej przez gracza postaci	Int8
	Role	Pozycja zawodnika w meczu (TOP/JUNGLE/MID/BOT/SUPP)	Int4
	Kills	Liczba zabójstw	Int4
	Deaths	Liczba śmierci	Int4
	Assists	Liczba asyst	Int4
	GoldEarned	Zarobione złoto	Int4
	GoldSpent	Wydane złoto	Int4
	LargestCriticalStrike	Największe trafienie krytyczne	Int4
	LargestKillingSpree	Największa seria zabójstw	Int4
	LargestMultiKill	Największe wielobójstwo	Int4
	Level	Poziom gracza w grze	Int4
	DamageDealt	Zadane obrażenia	Int4

	MagicDamageDealt	Zadane obrażenia magiczne	Int4
	PhysicalDamageDealt	Zadane obrażenia fizyczne	Int4
	TrueDamageDealt	Zadane nieuchronne obrażenia	Int4
	DamageDealtToChamps	Zadane obrażenia przeciwnikom	Int4
	PhysicalDamageDealtToChamps	Zadane obrażenia fizyczne przeciwnikom	Int4
	MagicDamageDealtToChamps	Zadane obrażenia magiczne przeciwnikom	Int4
	TrueDamageDealtToChamps	Zadane nieuchronne obrażenia przeciwnikom	Int4
	DamageTaken	Otrzymane obrażenia	Int4
	TotalHeal	Ilość wyleczonych punktów zdrowia	Int4
	TotalTimeCrowdControlDealt	Łączny czas nałożonej kontroli tłumu w sekundach	Int4
	WardsPlaced	Liczba postawionych totemów wizji	Int4
	WardsDestroyed	Zniszczone totemy wizji	Int4
	TurretsDestroyed	Zniszczone wieże	Int4
	InhibitorDestroyed	Zniszczone inhibitory	Int4
	MinionsKilled	Pokonane stwory	Int4
	NeutralMinionsKilled	Pokonane stwory neutralne	Int4
	EnemyNeutralMinionsKilled	Pokonane stwory neutralne po stronie przeciwnika	Int4
	FirstBlood	Czy gracz zdobył pierwsze zabójstwo	Bool
	FirstBloodAssist	Czy gracz uczestniczył w pierwszym zabójstwie	Bool
FK	Spell1Id	Identyfikator pierwszego czaru przywoływacza gracza w grze	Int8
FK	Spell2Id	Identyfikator drugiego czaru przywoływacza gracza w grze	Int8

#### *Przedmioty graczy w grach League of Legends*

NAZWA ENCJI	LolGamePlayerItems	ID.
		13.



OPIS		Obiekty zawierające informacje o przedmiotach posiadanych przez graczy w poszczególnych grach	
ATRYBUTY			
KLUCZ	NAZWA	Krótki opis	TYP
PK	Id	Identyfikator	Int8
FK	ItemId	Identyfikator przedmiotu	Int8
FK	GamePlayerId	Identyfikator obiektu zawierającego statystyki gracza w grze	Int8

### Statystyki drużyn w turniejach

NAZWA ENCJI		TournamentTeams	ID.
			14.
OPIS		Obiekty zawierające informacje o statystykach drużyn w turniejach. Zawiera opcjonalne pola dostępne tylko dla statystyk turniejów League of Legends.	
ATRYBUTY			
KLUCZ	NAZWA	Krótki opis	TYP
PK	Id	Identyfikator	Int8
FK	TeamId	Identyfikator drużyny	Int8
FK	TournamentId	Identyfikator turnieju	Int8
	GamesLost	Liczba przegranych gier	Int4
	GamesWon	Liczba wygranych gier	Int4
	MatchesLost	Liczba przegranych meczów	Int4
	MatchesWon	Liczba wygranych meczów	Int4
Tylko dla League of Legends			
	AverageKills	Średnia ilość zabójstw	Int4
	AverageDeaths	Średnia liczba śmierci	Int4
	AverageAssists	Średnia liczba asyst	Int4
	AverageGoldEarned	Średnia liczba zdobytego złota	Int4
	AverageDragonKilled	Średnia liczba pokonanych smoków	Int4
	AverageHeraldKilled	Średnia liczba pokonanych heraldów	Int4

	AverageBaronKilled	Średnia liczba pokonanych baronów	Int4
	AverageTowerDestroyed	Średnia liczba zniszczonych wież	Int4
FK	Champion1Id	Identyfikator pierwszej najczęściej wybieranej postaci, przez graczy drużyny	Int8
FK	Champion2Id	Identyfikator drugiej najczęściej wybieranej postaci, przez graczy drużyny	Int8
FK	Champion3Id	Identyfikator trzeciej najczęściej wybieranej postaci, przez graczy drużyny	Int8
FK	Champion4Id	Identyfikator czwartej najczęściej wybieranej postaci, przez graczy drużyny	Int8
FK	Champion5Id	Identyfikator piątej najczęściej wybieranej postaci, przez graczy drużyny	Int8

#### Statystyki zawodników w turniejach

NAZWA ENCJI		LolTournamentPlayers	ID.
			15.
OPIS		Obiekty zawierające informacje o statystykach zawodników w turniejach.	
ATRYBUTY			
KLUCZ	NAZWA	Krótki opis	TYP
PK	Id	Identyfikator	Int8
FK	PlayerId	Identyfikator zawodnika	Int8
FK	TournamentId	Identyfikator turnieju	Int8
	AverageKills	Średnia ilość zabójstw	Int4
	AverageDeaths	Średnia liczba śmierci	Int4
	AverageAssists	Średnia liczba asyst	Int4
	AverageMinionsKilled	Średnia liczba pokonanych stworów	Int4
	AverageGoldEarned	Średnia liczba zdobytego złota	Int4
	ChampionsPlayed	Liczba wybranych unikalnych postaci	Int4
	DamageShare	Udział w zadanych obrażeniach drużyny	Int4
	GoldPerMinute	Średnia ilość zdobytego złota na minutę	Int4
	KillParticipation	Udział w zabójstwach drużyny	Int4

	MinionsPerMinute	Średnia liczba pokonanych stworów na minutę	Int4
FK	Champion1Id	Identyfikator pierwszej najczęściej wybieranej postaci, przez gracza	Int8
FK	Champion2Id	Identyfikator drugiej najczęściej wybieranej postaci, przez gracza	Int8
FK	Champion3Id	Identyfikator trzeciej najczęściej wybieranej postaci	Int8

## Implementacja

W implementacji tego projektu zostały utworzone głównie klasy odpowiedzialne za konfigurację bazy danych, przy pomocy Entity framework core.

### ErtsContext

ErtsContext to klasa dziedzicząca po klasie DbContext, która jest odpowiedzialna za komunikację aplikacji z bazą danych. Zawiera ona DbSet-y, które odpowiadają wszystkim typom danych z db, oraz konfigurację na podstawie, których definiowane są właściwości encji. Do utworzenia bazy danych zastosowałem podejście Code first, więc struktura danych jest tworzona na podstawie zawartości tej klasy.

### Entities

Są to klasy definiujące poszczególne typy danych w aplikacji, które zawierają ich atrybuty.

```
public class Game : ModelBase{
    public DateTime StartTime { get; set; }
    public DateTime EndTime { get; set; }
    public virtual Team Winner { get; set; }
    public int ApiId { get; set; }
}
```

26 Klasa zawierająca właściwości gry

### Configuration

Są to klasy zawierające konfiguracje poszczególnych encji w bazie danych. Są one wykorzystywane przez Entity Framework podczas tworzenia migracji bazy danych poprzez podejście Code First.

```

class GameConfiguration : IEntityTypeConfiguration<Game> {
    private const string _winnerTeamId = "WinnerTeamId";
    internal const string MatchId = "MatchId";
    public void Configure(EntityTypeBuilder<Game> builder) {
        builder.HasComment("Game");

        builder.HasKey(x => x.Id);
        builder.Property(b => b.Id).HasComment("Id").ValueGeneratedOnAdd();

        builder.Property(b => b.StartTime).HasComment("Start time");
        builder.Property(b => b.EndTime).HasComment("End time");
        builder.HasOne(a => a.Winner)
            .WithMany()
            .HasForeignKey(_winnerTeamId);
        builder.Property(b => b.ApiId).HasComment("ApiId");
    }
}

```

27 Klasa konfigująca encje gry

Wewnątrz klas tego typu ustawia się między innymi klucze główne oraz definiuje się relacje pomiędzy encjami.

#### *FakeSeeds*

Jest to ostatni typ klas zaimplementowanych w projekcie ErtsModel.

Tego typu klasy są odpowiedzialne za generowanie i dodawanie przykładowych obiektów do bazy danych. Zastosowałem klasy tego typu we wczesnej fazie pisania pracy, w celu umożliwienia testowania aplikacji, bez użycia prawdziwych danych.

Dzięki temu podejściu mogłem generować dowolną ilość danych, o wybranych przeze mnie właściwościach, co umożliwiło mi wykonanie dokładniejszych testów.

## 4.2 ErtsApplication

### Wstęp

ErtsApplication to kolejny projekt w aplikacji, w którym zawarta jest logika dotycząca aplikacji webowej. Obecnie jego zadaniem jest dostarczanie do kontrolerów metod pobierających pożądane informacje z bazy danych, przerobione do odpowiedniego formatu.

### Implementacja

ErtsApplication to najkrótszy z projektów w aplikacji i posiada on klasy typu DTO oraz DAL.

#### *DTO*

Klasy typu DTO (Data transfer object), to szablony obiektów, które mają być dostarczone do front-endu aplikacji. Celem tworzenia tego typu klas jest polepszenie wydajności aplikacji, poprzez wysyłanie tylko niezbędnych danych do widoku aplikacji, pomijając przy tym nieużywane właściwości.

```
public class League : ModelBase {
    public string Name { get; set; }
    public string ImageUrl { get; set; }
    public GameType GameType { get; set; }
    public string Url { get; set; }
    public int ApiId { get; set; }
}
```

*28 Klasa w modelu zawierająca informacje o lidze*

```
public class LeagueImageDto {
    public long Id { get; set; }
    public string ImageUrl { get; set; }
    public string Name { get; set; }
}
```

*29 Klasa typu DTO, zawierająca dane potrzebne dla widoku zawierającego listę lig*

Przykładem użyteczności tego podejścia są znajdujące się powyżej bloki kodu. Dla listy lig we front-endzie aplikacji potrzebujemy tylko parametrów Id, ImageUrl, oraz Name, a pozostałe właściwości zdefiniowane w obiekcie ligi są nam niepotrzebne, przez co jeśli byśmy wysyłali do widoku cały obiekt ligi, niektóre parametry byłyby zbędne co nie wpłynęło by pozytywnie na działanie aplikacji. Dzięki stworzeniu klasy LeagueImageDto, możemy wysłać tylko pożądane parametry, a nie całe obiekty przechowywane w bazie danych.

Kolejną zaletą tego podejścia jest to, że użytkownik końcowy otrzymuje dokładne informacje jakie ma dostać i nie ma dostępu do wszystkich atrybutów zawartych w encji bazy danych.

### *DAL*

Kolejnym typem klas zawartych w projekcie są DAL (Data access layer). Oferują one metody odpowiedzialne za pobieranie potrzebnych informacji z baz danych i konwertowanie ich do obiektów typu DTO.

```

public class SerieDbService : ISerieDbService {
    public ErtsContext Context { get; set; }
    public SerieDbService(ErtsContext dbContext) {
        Context = dbContext;
    }
    public ActionResult<IEnumerable<SerieShortDto>> GetSeriesShort(int leagueId)
    {
        List<SerieShortDto> serieShortDtos = new List<SerieShortDto>();
        var series = Context.Series.Where(o => o.League.Id == leagueId).ToList();

        foreach (var serie in series) {
            var serieShortDto = new SerieShortDto() {
                Id = serie.Id,
                Name = serie.Name,
                StartTime = serie.StartTime,
                EndTime = serie.EndTime
            };
            serieShortDtos.Add(serieShortDto);
        }
        return serieShortDtos.OrderByDescending(x => x.StartTime).ToList();
    }
}

```

29 Przykładowa klasa typu DAL, która zawiera metody odpowiedzialne za zwracanie obiektów związanych z seriami

Powyżej przedstawiono przykładową klasę typu DAL, która zawiera metodę, która najpierw pobiera serie z bazy danych, należące do podanej w argumencie ligi, a następnie konwertuje je do obiektów typu SerieShortDto i je zwraca.

## 4.3 ErtsWebApp

### Wstęp

ErtsWebApp to najważniejszy z projektów, zawartych w solucji. Jest on odpowiedzialny za działanie całego front-endu aplikacji, oraz udostępnia rest api za pomocą, którego można uzyskiwać informacje z bazy danych aplikacji. W porównaniu do poprzednich projektów, nie jest on biblioteką klas, lecz jest niezależnym serwisem, który uruchamia się w celu włączenia aplikacji webowej.

### Implementacja

#### Controllers

Są to klasy odpowiedzialne za udostępnianie danych, poprzez wpisywanie odpowiednich zapytań http.

```

[Route("api/[controller]")]
[ApiController]
public class MatchController : ControllerBase {
    private readonly IMatchDbService _dbService;
    public MatchController(IMatchDbService dbService) {
        _dbService = dbService;
    }

    [Route("[action]/{tournamentId}")]
    [HttpGet]
    public ActionResult<IEnumerable<MatchShortDto>> GetMatches(int tournamentId){
        return _dbService.GetMatches(tournamentId);
    }

    [Route("[action]/{matchId}")]
    [HttpGet]
    public ActionResult<MatchDto> GetMatch(int matchId) {
        return _dbService.GetMatch(matchId);
    }
}

```

31 Przykładowy kontroler oferujący endpointy do pobierania informacji o ligach

Powyższy kod to przykładowy kontroler, który umożliwia pobieranie z bazy danych informacje o meczach. Zawiera on dwie końcówki, z których pierwsza (/api/Match/GetMatches/tournamentId), zwraca skrócone informacje o wszystkich meczach w turnieju, a druga (/api/Match/GetMatch/matchId), zwraca bardziej rozbudowaną informację o meczu podanego jako parametr.

<https://localhost:44393/api/League/GetLeague/16>

30 Przykładowe zapytanie do api

Powyższe zapytanie zwraca podstawowe informacje o wybranej lidze. Pierwsza część zapytania (localhost:44393), definiuje adres na którym działa api.

Kolejne części zapytania to nazwa kontrolera (League), nazwa metody w kontrolerze (GetLeague), oraz ligi, o której informacje chce się uzyskać (16).

```

1  {
2    "name": "LFL",
3    "imageUrl": "https://cdn.pandascore.co/images/league/image/4292/1200px-LFL_Logo_2020.png",
4    "url": "http://www.lollfl.com/"
5  }

```

32 Wynik uzyskany z powyższego zapytania w formacie JSON

### NSwag

Do używania danych zwracanych przez zapytania http, we front-endzie użyłem NSwaga. Dzięki temu podejściu nie musiałem pisać ręcznie kodu pobierającego dane na podstawie zapytań http, lecz kod do tego służący został wygenerowany automatycznie w wygenerowanej przez tą usługę pliku GeneratedClient.ts.

Wewnątrz tego pliku znajdują się klasy odpowiadające wszystkim kontrolerom, utworzonym w aplikacji, które umożliwiają łatwy dostęp do danych we front-endzie aplikacji.

Użycie tego typu klas w kodzie widoku aplikacji jest bardzo proste, gdyż wystarczy utworzyć ich obiekt, oraz wywołać na nim pożądaną metodę.

```
new TournamentClient().getTournamentsShort(this.state.serieId)
```

*33 Przykładowe wywołanie metody wygenerowanej, przez NSwag.*



## Pages

Pages to pliki odpowiedzialne za działanie poszczególnych podstron aplikacji. Tego rodzaju pliki posiadają rozszerzenie .tsx, co pozwala im na używanie elementów JSX (format zapisu kodu HTML i XML wewnątrz języka JavaScript), wewnątrz pliku języka TypeScript.

```
<Container
  style={{
    paddingBottom: "10vh",
    paddingTop: "5vh",
    color: "white"
  }}
>
  <TeamInfo team={this.state.team} />
  <Row>
    <Col>
      <PlayerList
        players={this.state.team?.players}
      />
    </Col>
  </Row>
  <Row>
    <Col md="auto">
      <LastMatches
        lastMatches={this.state.team?.lastMatches}
      />
    </Col>
    <Col md="auto">
      <UpcomingMatches
        upcomingMatches={this.state.team?.upcomingMatches}
      />
    </Col>
  </Row>
</Container >
```

34 Kod JSX odpowiedzialny za widok drużyny

Powyższy przykładowy kod w formacie JSX, opisuje podstronę zawierającą informacje o drużynach, przy użyciu funkcji zawartych w aplikacji takich jak LastMatches, która odpowiada za wyświetlenie listy ostatnich meczów, podanej w argumencie.

### Fragments

Fragmenty to części wspólne dla każdej podstrony aplikacji. W moim projekcie są nimi nagłówek (Heading) i stopka (Footer), które zawierają głównie elementy pomagające w nawigacji na stronie.

```
class Footer extends React.Component<RouteComponentProps> {
  private redirectToPage(gameType: string, page: string) {
    this.props.history.push(page);
  }

  render() {
    return (
      <>
        <Navbar bg="dark" variant="dark" fixed="bottom">
          <Navbar.Brand style={{ marginLeft: 10 }}
            onClick={() =>
              this.redirectToPage("csgo", "/csgo/leagues")
            }>
            <Image
              alt=""
              src="/images/csgo-logo.png"
              width="100"
              height="44"
              className="d-inline-block align-top"
            />{" "}
          </Navbar.Brand>
        </>
      </>
    );
  }
}
```

35 Fragment pliku Footer.tsx

Powyższy fragment kodu odpowiedzialny za stopkę strony, który odpowiada za zmianę sceny aplikacji na podstronę CS GO, poprzez kliknięcie w logo gry znajdujące się w nawigacji.

### Tłumaczenie

Tłumaczenie widoku aplikacji zrealizowałem, przy pomocy i18next, dodając tą usługę do projektu przy użyciu npm.

```
npm install i18next --save
```

36 Komenda dodająca i18next do projektu

Kolejnym krokiem było dodanie plików o formacie JSON, zawierających tłumaczenia. Tego rodzaju pliki tworzy się tworząc atrybuty odpowiadające poszczególnym elementom tekstowym w aplikacji i przypisując do nich odpowiedni dla danego języka tekst.

```
{
  "game-type": {
    "nav": {
      "leagues": "Ligi",
      "teams": "Drużyny",
      "placeholder": "Wprowadź fragment nazwy"
    }
  },
  "serie": {
    "serie-name": "Seria",
    "start-time": "Czas rozpoczęcia",
    "end-time": "Czas zakończenia"
  },
  "tournament": {
    "info": {
      "tournament-name": "Turniej",
      "start-time": "Czas rozpoczęcia",
      "end-time": "Czas zakończenia"
    }
  },
  ...
}
```

37 Fragment pliku JSON odpowiedzialnego za tłumaczenie na język polski

Po utworzeniu tego typu pliku trzeba dodać go do pliku konfiguracyjnego `i18n`, aby umożliwić późniejsze użycie go w aplikacji.

```
import translationEnglish from './locales/us';
import translationPolish from './locales/pl';
import translationItalian from './locales/it';

const resources = {
  us: {
    translation: translationEnglish
  },
  pl: {
    translation: translationPolish
  },
  it: {
    translation: translationItalian
  },
};
```

38 Kod dodający zdefiniowane języki do `i18n`

Po skonfigurowaniu i18n, zastąpiłem tekst w aplikacji, odwołaniami do poszczególnych atrybutów, które zawierają pliki tłumaczeń.

```
const { t } = useTranslation();
...
<thead>
  <tr>
    <th>{t("serie.serie-name")}</th>
    <th>{t("serie.start-time")}</th>
    <th>{t("serie.end-time")}</th>
  </tr>
</thead>
```

*39 Przykładowe użycie tłumaczenia na podstronie*

Ostatnią rzeczą związaną z tłumaczeniem było dodanie do nagłówka elementu pozwalającego na zmianę języka w aplikacji.

## 4.4 ErtsApiFetcher

### Wstęp

ErtsApiFetcher to niezależny od aplikacji webowej serwis, którego zadaniem jest pobieranie danych z api PandaScore, oraz zapisywanie ich do bazy danych z której korzysta aplikacja.

Serwis działa na zasadzie prac (jobs), które wykonują się cyklicznie w zależności od ustawionych parametrów w kodzie serwisu. W celu łatwiejszego zarządzania tymi procesami użyłem narzędzia Hangfire, dzięki któremu mogę je łatwo kontrolować.

### Funkcjonalność

Ten serwis tworzy procesy, które są wywoływane automatycznie, w określonym czasie, ale dzięki interfejsowi użytkownika ich wywoływanie może być manipulowane także przez użytkownika.

Serwis obecnie posiada dziewięć rodzajów prac, z których każdy pełni inną funkcję w systemie.

Najważniejszymi Jobami są ShortRecurringJob i LongRecurringJob. Są to procesy, które jako jedyne wykonują się automatycznie, bez ingerencji użytkownika i ich zadaniem jest bieżąca aktualizacja bazy danych aplikacji.

Zadaniem ShortRecurringJob jest pobieranie z api i aktualizacja w bazie danych ostatnich wyników meczy dla każdej gry. Aby wyniki były aktualne cykl tej pracy jest wykonywany w każdej minucie. Ze względu na możliwość lekkich opóźnień w uzyskiwaniu danych z api, w celu uniknięcia potencjalnej utraty danych, cominutowy proces pobiera dane z ostatnich dwóch minut.

LongRecurringJob jest natomiast rozszerzoną wersją poprzedniego procesu. Wywoływany jest on co dziesięć minut i poza wynikami meczów, pobiera on również inne dane zmodyfikowane w api podczas wyznaczonego przedziału czasowego.

Aktualizacja danych w czasie rzeczywistym została rozdzielona na dwa procesy z powodu ograniczonej liczby zapytań narzuconej przez PandaScore, przez którą cominutowa aktualizacja wszystkich danych była niemożliwa. Z powodu tego, że tylko wyniki meczy są często aktualizowane, to zmieniłem częstotliwość pobierania innego rodzaju danych na 10 minut. Takie podejście pozwoliło mi zaoszczędzić dużą ilość punktów oferowanych przez api, dzięki czemu serwis podczas funkcjonowania nie powinien napotykać problemów związanych z odmową wysyłania danych przez api.

Reszta Jobów oferowanych przez serwis nie jest wykonywana automatycznie i mogą być uruchomione tylko w wyniku ingerencji użytkownika. Zadaniem tych Jobów jest aktualizacja historycznych danych.

Te procesy powinny być wywoływane przez administratora, w przypadku resetu bazy danych aplikacji w celu zapełnienia jej danymi. Można również używać tych Jobów w celu uzupełnienia zaległych danych w przypadku awarii, lub tymczasowego wyłączenia serwisu.

Te procesy dzielą się na aktualizujące wszystkie dane z wybranej gry, oraz na pobierające dane z określonego większego przedziału czasowego.

Te procesy powinny być wywoływane pojedynczo, ponieważ pobierają one dużo zasobów z api, przez co ich nałożenie mogłoby skutkować wyczerpaniem godzinnych zasobów, czego rezultatem byłoby nie pobranie zasobów dla żadnego z zadań.

Z powodu przetwarzania dużej ilości danych czas wykonania procesów tego typu jest znacznie dłuższy od czasu wykonywania regularnych Jobów i może on trwać nawet ponad 30 minut.

Kontrola tych procesów jest możliwa dzięki oferowanemu graficznemu interfejsowi użytkownika, który oferuje kilka przydatnych widoków:

#### *Wykres wykonanych prac*

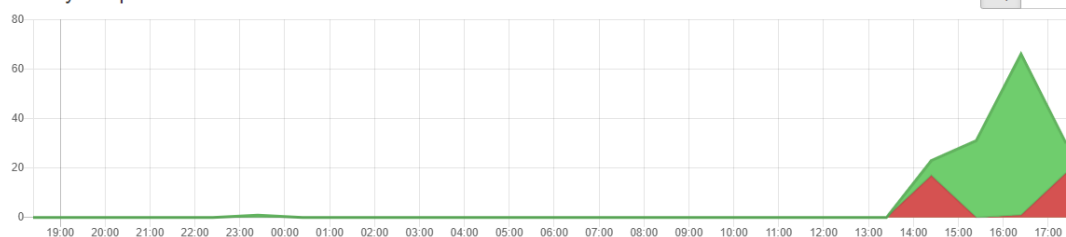
Pierwszym widokiem oferowanym przez GUI jest wykres, pokazujący ilość prac wykonanych pomyślnie, oraz nieudanych w ostatnim czasie.

## Overview

Realtime Graph



History Graph



1 Widok wykresów zawierających wykonane prace w zależności od czasu

### Widok kontroli uruchomionych procesów

Dzięki temu widokowi możemy śledzić status wywołanych wcześniej prac, oraz badać powód prac zakończonych niepowodzeniem z możliwością ich ponownego uruchomienia.

Najczęstszym wyjątkiem zwracanym przez Joby jest `OutOfApiPointsException`, który oznacza, że proces zakończył swoje działanie z powodu wyczerpania ilości zapytań oferowanych przez api. W przypadku tego błędu należy sprawdzić pozostałą liczbę punktów w widoku po zalogowaniu na stronę PandaScore i w przypadku codziennego odnowienia limitu ponownie uruchomić proces.

## API Usage

1014 / 1000 in last hour



2 Pasek zużycia oferowanej ilości zapytań, na stronie api

Enqueued	0 / 1
Scheduled	0
Processing	0
Succeeded	133
Failed	42
Deleted	8
Awaiting	0

```
FromJson<Type>("{\"ErtsApiFetcher.RecurringJobs.ShortRecurringJob, ErtsApiFetcher, Version=1.0.0.0, Cul
ture=neutral, PublicKeyToken=null\"");
```

### Parameters

RecurringJobId	"ShortRecurringJob"
Time	1639852155
CurrentCulture	"pl-PL"
CurrentUICulture	"pl-PL"

### State

Queue

Delete

Failed

kilka sekund temu (+238ms)

An exception occurred during performance of the job.

ErtsApiFetcher.Services.OutOfApiPointsException

Exception of type 'ErtsApiFetcher.Services.OutOfApiPointsException' was thrown.

ErtsApiFetcher.Services.OutOfApiPointsException: Exception of type 'ErtsApiFetcher.Services.OutOfApiPointsException' was thrown.  
at ErtsApiFetcher.Services.DataService.FetchAndCatch[T](Func`2 fetch) in D:\Projekty\ERTS\app\ErtsApiFetcher\Services\DataService.cs:line 22  
at ErtsApiFetcher.Fetchers.DataFetcherBase.FetchMatches(Nullable`1 from) in D:\Projekty\ERTS\app\ErtsApiFetcher\Fetchers\DataFetcherBase.cs:line 165  
at ErtsApiFetcher.ApiDataProcessors.Matches.MatchApiDataProcessor.ProcessInternal(MatchApiDataProcessorParameter parameter) in D:\Projekty\ERTS\app\ErtsApiFetcher\ApiDataProcessors\Matches\MatchApiDataProcessor.cs:line 16  
at ErtsApiFetcher.ApiDataProcessors.ApiDataProcessor`1.Process(TParameter parameter) in D:\Projekty\ERTS\app\ErtsApiFetcher\ApiDataProcessors\ApiDataProcessor.cs:line 16  
at ErtsApiFetcher.Infrastructure.ApiDataProcessors.ApiDataProcessorExecutor.Execute[TParameter](TParameter parameter) in D:\Projekty\ERTS\app\ErtsApiFetcher\Infrastructure\ApiDataProcessors\ApiDataProcessorExecutor.cs:line 15  
at ErtsApiFetcher.RecurringJobs.ShortRecurringJob.Job() in D:\Projekty\ERTS\app\ErtsApiFetcher\RecurringJobs\ShortRecurringJob.cs:line 21  
at ErtsApiFetcher.Infrastructure.RecurringJobs.RecurringJobActivator.CreateScopeAndActivate(Type jobType) in D:\Projekty\ERTS\app\ErtsApiFetcher\Infrastructure\RecurringJobs\RecurringJobActivator.cs:line 16

3 Widok z informacjami o błędzie spowodowanym wyczerpaniem oferowanej ilości zapytań

## Widok kontroli typów procesów

Ostatnim z najważniejszych dla aplikacji widoków jest ekran kontroli typu procesów, który zawiera listę wszystkich rodzajów prac oferowanych przez serwis. Pokazuje on kiedy został uruchomiony ostatni proces danego typu i jaki status aktualnie posiada. Dla Jobów wykonywanych cyklicznie pokazywany jest również czas do kolejnej egzekucji procesu.

Ten widok pozwala również usuwać wybrane prace, oraz ręcznie je uruchamiać. Ten widok przydaje się głównie podczas manualnej manipulacji procesami przez administratora systemu.

62

## Recurring Jobs

Trigger now Delete

Items per page: 10 20 50 100 500 1 000 5 000

<input type="checkbox"/> Id	Cron	Time zone	Job	Next execution	Last execution	Created
<input type="checkbox"/> CsgoInitialRecurringJob	0 0 5 31 2 ?	UTC	RecurringJobActivator.CreateScopeAndActivate	DISABLED	godzinę temu	19 godzin temu
<input type="checkbox"/> Dota2InitialRecurringJob	0 0 5 31 2 ?	UTC	RecurringJobActivator.CreateScopeAndActivate	DISABLED	godzinę temu	19 godzin temu
<input type="checkbox"/> LolInitialRecurringJob	0 0 5 31 2 ?	UTC	RecurringJobActivator.CreateScopeAndActivate	DISABLED	godzinę temu	19 godzin temu
<input type="checkbox"/> OwlInitialRecurringJob	0 0 5 31 2 ?	UTC	RecurringJobActivator.CreateScopeAndActivate	DISABLED	godzinę temu	19 godzin temu
<input type="checkbox"/> ValorantInitialRecurringJob	0 0 5 31 2 ?	UTC	RecurringJobActivator.CreateScopeAndActivate	DISABLED	N/A	19 godzin temu
<input type="checkbox"/> DailyRecurringJob	0 0 5 31 2 ?	UTC	RecurringJobActivator.CreateScopeAndActivate	DISABLED	N/A	4 godziny temu
<input type="checkbox"/> WeeklyRecurringJob	0 0 5 31 2 ?	UTC	RecurringJobActivator.CreateScopeAndActivate	DISABLED	N/A	4 godziny temu
<input type="checkbox"/> LongRecurringJob	* / 10 * * * *	UTC	RecurringJobActivator.CreateScopeAndActivate	za 4 minuty	minutę temu	godzinę temu
<input type="checkbox"/> ShortRecurringJob	* * * * *	UTC	RecurringJobActivator.CreateScopeAndActivate	kilka sekund temu	minutę temu	godzinę temu




Total items: 9

4 Widok zawierający listę prac

### Implementacja

#### Hangfire

Jak wspomniałem wcześniej do zaimplementowania tego serwisu użyłem narzędzia Hangfire. W celu użycia tej technologii w aplikacji zainstalowałem pakiety NuGet, umożliwiające użycie Hangfire w aplikacji ASP .NET Core . Dodatkowo w celu umieszczenia danych pochodzących z tej usługi w bazie danych aplikacji, dodałem kolejny NuGet umożliwiający integrację tego narzędzia z PostgreSQL.

	<b>Hangfire.AspNetCore</b> by Sergey Odínokov	1.7.25
	ASP.NET Core support for Hangfire (background job system for ASP.NET applications).	1.7.28
	<b>Hangfire.Core</b> by Sergey Odínokov	1.7.25
	An easy way to perform fire-and-forget, delayed and recurring tasks inside ASP.NET applications. No Windows Service required.	1.7.28
	<b>Hangfire.PostgreSql</b> by Frank Hommers and others (Burhan Irmikci (barhun), Zachary Sims(zsims), kgamecarter, Stafford Williams (staff	1.8.6
	PostgreSql storage implementation for Hangfire (background job system for ASP.NET and aspnet core applications).	1.9.4

40 Zainstalowane pakiety NuGet do integracji aplikacji z narzędziem Hangfire

Poza dodaniem pakietów, aby umożliwić korzystanie z Hangfire w aplikacji, musiałem również go skonfigurować poprzez wstrzyknięcie wymaganych zależności w startupie w projekcie.



```

var hangfireJobStorage = GetJobStorage(AppConfig.ErtsHangfireConnectionString,
    AppConfig.ErtsHangfireSchemaName, true, 60, 120);

services.AddHangfire(configuration => configuration
    .UseStorage(hangfireJobStorage)
    .SetDataCompatibilityLevel(CompatibilityLevel.Version_170)
    .UseSimpleAssemblyNameTypeSerializer()
    .UseRecommendedSerializerSettings());

services.AddHangfireServer(hangfireJobStorage);

```

42 Dodanie konfiguracji Hangfire do DI

Poza konfiguracją Hangfire z projektem za pomocą powyższego kodu, dodałem również kod umożliwiający korzystanie z UI oferowanego przez wydawcę tego narzędzia. W celu dodania tej funkcjonalności dodałem odpowiednie metody do potoku obsługi żądań projektu, znajdującego się w metodzie „Configure” w startupie projektu.

```

app.UseHangfireDashboard();

app.UseEndpoints(endpoints => {
    endpoints.MapHangfireDashboard();
});

```

41 Dodanie UI do projektu

### *Recurring Jobs*

Podstawowym elementem struktury tego projektu są Recurring Jobs, z których każda odpowiada jednej pracy wyświetlającej się w UI serwisu.

Częścią wspólną dla każdej z tych klas jest metoda Job, w której definiowane jest działanie każdej z prac.

```

public override void Job() {
    context.Database.BeginTransaction();

    executor.Execute(new MatchApiDataProcessorParameter(
        new CsgoDataFetcher(appConfig.PandaScoreApiToken, context), from));
    executor.Execute(new MatchApiDataProcessorParameter(
        new Dota2DataFetcher(appConfig.PandaScoreApiToken, context), from));
    executor.Execute(new MatchApiDataProcessorParameter(
        new ValorantDataFetcher(appConfig.PandaScoreApiToken, context), from));
    executor.Execute(new MatchApiDataProcessorParameter(
        new OwDataFetcher(appConfig.PandaScoreApiToken, context), from));
    executor.Execute(new LolMatchApiDataProcessorParameter(
        new LolDataFetcher(appConfig.PandaScoreApiToken, context), from));

    context.Database.CommitTransaction();
}

```

43 Metoda Job() dla ShortRecurringJob

Przykładem takiej metody jest metoda Job() w klasie ShortRecurringJob przedstawiona powyżej. Wewnątrz niej znajdują się wywołania akcji, aktualizujących wyniki meczów dla wszystkich gier wspieranych przez aplikację, co jest celem tej pracy.

Oprócz metody Job(), każda z tych klas musi posiadać atrybut typu RecurringJobInfo, w którym definiowana jest nazwa, typ oraz częstotliwość wykonywania danej pracy, podana za pomocą Crona.

```
[RecurringJobInfo(typeof(ShortRecurringJob), nameof(ShortRecurringJob),  
    Ertscron.Minutely)]
```

44 Atrybut typu RecurringJobInfo dla klasy ShortRecurringJob

Na podstawie zdefiniowanych w ten sposób atrybutów aplikacja, znajduje wszystkie Joby wstrzyknięte do DI i na podstawie jego atrybutów następnie dodaje je do Hangfire, co pozwala potem bez problemu z nich korzystać.

```
public void AddRecurringJob(RecurringJobInfoAttribute recurringJobInfo) {  
    RecurringJob.AddOrUpdate<RecurringJobActivator>(  
        recurringJobInfo.JobName,  
        (jobActivator) => jobActivator.CreateScopeAndActivate(  
            recurringJobInfo.JobType,  
            recurringJobInfo.CronTime);  
    }  
}
```

45 Metoda w klasie HangfireEnqueuer odpowiedzialna za dodawanie RecurringJoba do Hangfire na podstawie jego atrybutu

### Data Fetchers

Kolejnym ważnym typem klas dla tego serwisu są, klasy typu Data Fetchers. Zadaniem tego rodzaju jest klas pobieranie wybranych danych z PandaScore api, oraz zwrócenie ich w postaci klas zgodnych z modelem aplikacji.

Do pobierania informacji z api, użyłem biblioteki PandaScore.NET, która pozwala generować zapytania zwracające pobrane dane z bazy PandaScore , na podstawie otrzymanych parametrów.

Do tego rozwiązania utworzyłem forka na GitHubie, wprowadzając kilka zmian potrzebnych do poprawnego działania biblioteki w moim projekcie.

Pierwszą zmianą jaką dokonałem w tej bibliotece było rozszerzenie jej funkcjonalności dla wszystkich gier, ponieważ jej pierwotna wersja jest kompatybilna tylko z grą League of Legends.

Poza tym musiałem również zaktualizować strukturę danych wewnątrz PandaScore.NET, ponieważ biblioteka nie była uaktualniana od dwóch lat, przez co była nie zgodna z aktualną wersją wykorzystanego api.

#### *Api Data Processors*

Ostatnim z omawianych elementów struktury projektu ErsApiFetcher są Api Data Processors. Zadaniem tego typu klas jest aktualizacja bazy danych aplikacji danymi uzyskanymi z wcześniej uzyskanych Fetcherów.

```
var apiTournaments = parameter.DataFetcher
    .FetchTournaments(parameter.FromTime);
var newTournaments = apiTournaments.Where(apiTournament => !context.Tournaments
    .Any(contextTournament => contextTournament.ApiId == apiTournament.ApiId));

context.Tournaments.AddRange(newTournaments);
var updatedTournaments = apiTournaments.Where(
    apiTournament => context.Tournaments.Any(
        contextTournament => contextTournament.ApiId == apiTournament.ApiId));

foreach (var updatedTournament in updatedTournaments) {
    var contextTournament = context.Tournaments.Where(
        contextTournament => contextTournament.ApiId == updatedTournament.ApiId)
        .FirstOrDefault();

    contextTournament.Update(
        updatedTournament.Name,
        updatedTournament.StartTime,
        updatedTournament.EndTime,
        updatedTournament.Serie);
}
```

*46 Kod metody Api Data Processora odpowiedzialnego za aktualizacje danych turniejów*

Kod pokazany powyżej jest przykładem elementu tego typu, który na początku pobiera dane z odpowiedniego DataFetchera, a następnie dodaje je, bądź aktualizuje w bazie danych aplikacji.

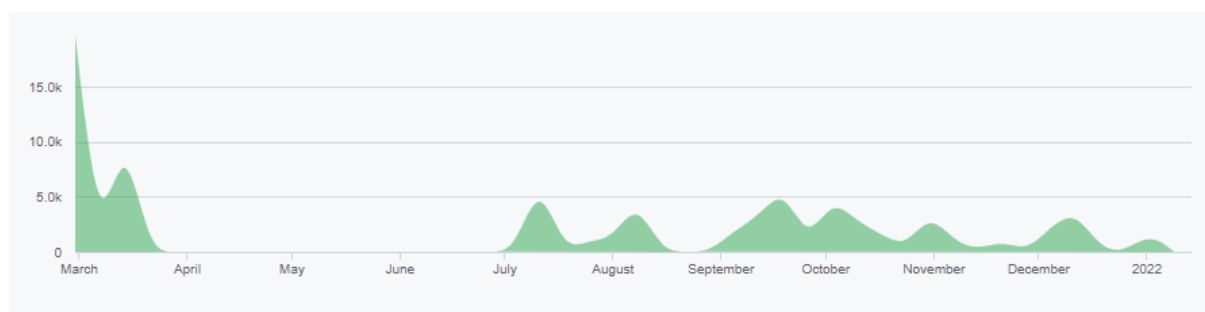
Poza klasycznymi Processorami w projekcie znajdują się również klasy tego typu odpowiedzialne za uzupełnianie bazy danych, informacjami wyliczonymi z danych pobranych z api. Przykładem tego typu jest TournamentTeamStatsApiDataProcessor wyliczający statystyki drużyny w turnieju. Jego kod jest wywoływany każdorazowo podczas dodania nowego meczu do bazy danych, co wymaga aktualizacji statystyk.

# Podsumowanie

W ramach pracy inżynierskiej udało mi się utworzyć nowoczesną aplikację webową, pozwalającą na śledzenie statystyk e-sportowych. Obecna wersja strony, jest gotowa do uruchomienia w środowisku produkcyjnym.

Aplikacja w tym momencie posiada, głównie podstawowe funkcjonalności i widoki dotyczące, wizualizacji wyników rozgrywek, ale ze względu na strukturę i reużywalność jej kodu, umożliwia ona wygodną implementację dodatkowych funkcjonalności.

Proces tworzenia projektu przyniósł mi dużo korzyści edukacyjnych, takich jak poznanie wielu nowych technologii, które obecnie przodują na rynku tego typu aplikacji. Dodatkowo podczas tworzenia projektu poznałem również cykl tworzenia aplikacji webowej od zera.



47 Diagram wygenerowany przez serwis GitHub, przedstawiający ilość dodanego kodu do aplikacji w przedstawionych okresach czasu

Jak wynika z powyższego diagramu, proces tworzenia aplikacji rozpocząłem w Marcu 2021 roku. W tym okresie skonfigurowałem projekt oraz utworzyłem szkielet aplikacji. Dodatkowo pod koniec tego okresu utworzyłem również schemat bazy danych i wypełniłem ją przykładowymi danymi wygenerowanymi, przez stworzone przeze mnie generatory losowych danych.

Przez następne trzy miesiące z powodu braku czasu nie wprowadziłem żadnych zmian w projekcie.

Implementację aplikacji wznowiłem na początku Lipca i przez kolejne kilka miesięcy stopniowo dodawałem wszystkie podstawowe funkcjonalności do systemu.

Proces implementacji podstawowej wersji systemu ukończyłem pod koniec Listopada i po tym okresie wprowadzałem już tylko poprawki dotyczące wydajności aplikacji, oraz jakości jej kodu. Dodatkowo w tym okresie rozpocząłem pisanie teoretycznej części pracy dyplomowej.

# Bibliografia

- BBC. (2021). Types of programming language. Pobrano z lokalizacji <https://www.bbc.co.uk/bitesize/guides/z4cck2p/revision/1>
- BigBoss. (2020, 11 18). *Overwatch esport – co warto wiedzieć na temat tej dyscypliny?* Pobrano z lokalizacji Zapytaj Bukmachera.PL: <https://zapytajbukmachera.pl/overwatch-esport/>
- Bootstrap. (2021). *Bootstarp docs*. Pobrano z lokalizacji get bootstrap: <https://getbootstrap.com/>
- Ćwik, P. (2019, Lipiec 4). *Co to TypeScript*. Pobrano z lokalizacji clockwork java: <https://clockworkjava.pl/2019/07/co-to-typescript/>
- Eckelo, B. (2006). *Thinking in Java*.
- Git. (2021). Pobrano z lokalizacji Git: <https://git-scm.com/>
- Hangfire. (2021). *Przegląd*. Pobrano z lokalizacji Hangfire : <https://www.hangfire.io/overview.html>
- IEEE Spectrum. (2022, Styczeń). *Top Programming Languages 2021*. Pobrano z lokalizacji spectrum.ieee: <https://spectrum.ieee.org/top-programming-languages/>
- Klucz-system. (2020, Sierpień 26). *CZYM JEST I JAK DZIAŁA SYSTEM ZARZĄDZANIA BAZĄ DANYCH?* Pobrano z lokalizacji Klucz-system: <https://www.kluczsystem-sklep.pl/czym-jest-i-jak-dziala-system-zarzadzania-baza-danych-n-19.html>
- Microsoft. (2021). *docs.microsoft*. Pobrano z lokalizacji Dokumentacja C#: <https://docs.microsoft.com/>
- Microsoft. (2021, Wrzesień 24). *Wprowadzenie do platformy ASP.NET Core*. Pobrano z lokalizacji docs.microsoft: <https://docs.microsoft.com/pl-pl/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-6.0>
- Nius, B. (2020, Sierpień 27). *Czym jest React i jakie ma zalety?* Pobrano z lokalizacji global4net: <https://global4net.com/ecommerce/czym-jest-react-i-jakie-ma-zalety/>
- PandaScore. (2021). *PandaScore dokumentacja*. Pobrano z lokalizacji developers.pandascore: <https://developers.pandascore.co/docs>
- stack overflow. (2022, Styczeń). *Tags*. Pobrano z lokalizacji stackoverflow: <https://stackoverflow.com/tags>
- Tiobe. (2022, Styczeń). *TIOBE Index for January 2022*. Pobrano z lokalizacji Tiobe: <https://www.tiobe.com/tiobe-index/>

Wikipedia. (2021). *GitHub*. Pobrano z lokalizacji Wikipedia:  
<https://pl.wikipedia.org/wiki/GitHub>

Wikipedia. (2021). *PostgreSQL*. Pobrano z lokalizacji Wikipedia:  
<https://pl.wikipedia.org/wiki/PostgreSQL>

Wikipedia. (2021). *Sport elektroniczny*. Pobrano z lokalizacji Wikipedia:  
[https://pl.wikipedia.org/wiki/Sport\\_elektroniczny](https://pl.wikipedia.org/wiki/Sport_elektroniczny)

Wikipedia. (2021). *TypeScript*. Pobrano z lokalizacji Wikipedia:  
<https://pl.wikipedia.org/wiki/TypeScript>