Streszczenie

Celem projektu było stworzenie aplikacji, której zadaniem będzie wspomaganie zbierania statystyk z meczów piłki nożnej. Do głównych funkcjonalności należy zaliczyć możliwość przypisania różnorodnych akcji (m. in. strzał, podanie, odbiór piłki, faul) do konkretnego zawodnika. Każda taka informacja zostanie umieszczona w bazie danych. Zaletą tej aplikacji będzie także możliwość tworzenia zestawień w formie tabelarycznej lub też w formie graficznej. Do potencjalnych użytkowników zaliczyć można trenerów, zawodników bądź zwykłych kibiców. Aplikacja będzie również zawierać swój zmodyfikowany odpowiednik w formie mobilnej. Stworzony system ma pozwolić drużynie podnieść poziom sportowy, poprzez nacisk na elementy, w których zespół zawodzi i które są prawdopodobnie przyczyną jego słabszych wyników.

Indywidualny udział w realizacji pracy:

Jakub Hopen: Aplikacja mobilna na system Android, zaprojektowanie i utworzenie lokalnej bazy danych na darmowym serwerze, rozdziały dokumentacji:

Mateusz Szymański: Aplikacja desktopowa na system Windows: implementacja wybranych akcji, zaprojektowanie implementacja interfejsu, kontakt z klientem, sporządzanie notatek ze spotkań z klientem, formatowanie dokumentacji, rozdziały dokumentacji: 1,3, 4,

Wojciech Pasternak: Aplikacja desktopowa na system Windows: (co tam naknociłeś), rozdziały dokumentacji:

Spis treści

[1. Plan projektu 6](#_Toc405682049)

[1.1. Charakterystyka projektu 6](#_Toc405682050)

[1.2. Metodyka i narzędzia 8](#_Toc405682051)

[1.3. Zespół i komunikacja 8](#_Toc405682052)

[1.4. Wstępny harmonogram 9](#_Toc405682053)

[1.5. Zapewnienie jakości 9](#_Toc405682054)

[1.6. Analiza ryzyka 10](#_Toc405682055)

[2. Opis systemu 11](#_Toc405682056)

[2.1. Struktura bazy danych 11](#_Toc405682057)

[2.2. Technologie 16](#_Toc405682058)

[3. Realizacja projektu 18](#_Toc405682059)

[3.1. Product backlog 18](#_Toc405682060)

[3.2. Sprint 1 (12.10.2014- 18.10.2014) 18](#_Toc405682061)

[3.3. Sprint 2 (19.10.2014- 01.11.2014) 20](#_Toc405682062)

[3.4. Sprint 3 (02.11.2014 – 15.11.2014) 23](#_Toc405682063)

[3.5. Sprint 4 (16.11.2014-22.11.2014) 25](#_Toc405682064)

[3.6. Sprint 5 (23.11.2014-29.11.2014) 28](#_Toc405682065)

[3.7. Sprint 6 (30.11.2014 – 06.12.2014) 31](#_Toc405682066)

[4. Raport końcowy 32](#_Toc405682067)

[4.1. Zespół projektowy 32](#_Toc405682068)

[4.2. Temat projektu 32](#_Toc405682069)

[4.3. Kontekst projektu 32](#_Toc405682070)

[4.4. Osiągnięte rezultaty 32](#_Toc405682071)

[4.5. Proces realizacji projektu 33](#_Toc405682072)

[4.6. Dokumentacja 33](#_Toc405682073)

[4.7. Zmiany w trakcie projektu 34](#_Toc405682074)

[4.8. Podział wykonanej pracy między członków grupy projektowej (jutro uzupełnimy) 34](#_Toc405682075)

[4.9. Podsumowanie 35](#_Toc405682076)

[4.10. Opinia klienta 35](#_Toc405682077)

[5. Instrukcja użytkownika aplikacji desktopowej 36](#_Toc405682078)

[5.1. Wymagania 36](#_Toc405682079)

[5.2. Pierwsze uruchomienie, logowanie i rejestracja 36](#_Toc405682080)

[5.3. Zarządzanie drużynami 37](#_Toc405682081)

[5.4. Zbieranie danych statystycznych 39](#_Toc405682082)

[5.5. Analiza danych statystycznych 42](#_Toc405682083)

[6. Poradnik użytkownika dla aplikacji mobilnej ( tymczasowo w osobnym pliku ) 46](#_Toc405682084)

# Plan projektu

## Charakterystyka projektu

### Opis projektu i produktu

Celem naszego projektu jest stworzenie aplikacji, która będzie wspomagać klub piłkarski w zbieraniu statystyk z meczów piłkarskich w/w drużyny. Produkt nie ma konkretnego klienta, a jedynie konsultanta w postaci prezesa w tymże klubie. Będzie on oglądał postęp naszych prac i zgłaszał swoje uwagi, które postaramy się wprowadzić. Produkt jest związany z naszym hobby, którym jest piłka nożna.

### Charakterystyka użytkowników

Aplikacja przeznaczona będzie dla sztabu szkoleniowego TS Mierzeja Wiślana. Produkt przekazany będzie prezesowi, a będzie go używać trener lub osoba przez niego wyznaczona. Jest to młoda osoba zaznajomiona z rozwiązaniami IT, więc nie będzie ona wymagała złożonego szkolenia. Niezbędna będzie natomiast krótka demonstracja działania naszego produktu. Aplikacja stworzona będzie w języku angielskim.

### Zakres produktu

Aplikacja będzie umożliwiała zbieranie statystyk zawodników podczas rozgrywania meczy i umieszczanie ich w bazie danych. Na podstawie tych danych będzie pozwalała ona dokonywania analizy gry zawodników. Aplikacja będzie umożliwiała także zbieranie danych takich jak: udane podania, celne strzały na bramkę, przyjęcia piłki, odebranie piłki przeciwnikowi i wiele innych przydatnych statystyk o zawodnikach, zarówno własnej jak i przeciwnej drużyny. W wersji minimalnej aplikacji przewidujemy wykorzystanie lokalnej bazy danych na urządzeniu do przechowywania danych, w wersji realnej planujemy wykorzystać zdalną bazę danych do której będą przesyłane zapytania za pośrednictwem Internetu, co wymaga połączenia z siecią dla działania. W optymalnej wersji aplikacja będzie posiadała lokalną bazę danych jak i łączność ze zdalną. Przy czym synchronizacja danych będzie następować w chwili, gdy będzie dostępne łącze, co nie będzie wymagało połączenia z Internetem dla działania aplikacji. Oprócz wyświetlania zebranych statystyk, aplikacja będzie mogła porównywać dane dla dwóch graczy porównując ich statystyki w formie liczbowej w wersji minimalnej lub w formie graficznej w wersji optymalnej. Product Backlog znajduje się w 3 plikach, w tym samym folderze co plan projektu.

### Wymagania jako scenariusz użycia

Kazimierz Szczęsny jest asystentem trenera, w lokalnym klubie piłkarskim Zawisza Nowa Wieś. Jego zadaniem jest szczegółowa analiza gry poszczególnych zawodników, sporządzanie uwag dotyczących meczów, notowanie asystentów i strzelców, a także zawodników ukaranych kartkami. Dzisiaj w pracy, dostał od swojego przełożonego nowe zadanie – zapoznać się z innowacyjnym programem, wspomagającym analizę statystyk z meczów piłkarskich.

Po zjedzeniu obiadu, Kazimierz od razu zainstalował go na swoim laptopie i uruchomił. By przetestować testowanie programu, niezbędne okazało się zalogowanie. Zaczął od rejestracji – wpisał login, imię, nazwisko oraz hasło, utworzył konto i zalogował się.

Następnie przeszedł do zakładki zarządzania drużyną - wpisał nazwę i utworzył nowy zespół. Rozpoczął dodawanie zawodników – sięgnął po zeszyt z notatkami, wyjął listę zawodników, by przepisać ją do programu. Dodał każdego z zawodników – wpisał ich dane – imię, nazwisko, numer, wybrał z listy pozycję na boisku, oraz zaznaczył preferowaną nogę. Po kwadransie lista zawodników była już kompletna, ale Kazik przypomniał sobie, że jego notatki nie były aktualne, ponieważ ostatnio kilku zawodników odeszło z drużyny, na skutek czego kilku innych zmieniło numery. Zaczął od zmiany numerów – ale gdy chciał zatwierdzić, dostał komunikat o błędzie, ponieważ w drużynie już byli zawodnicy, występujący z tymi numerami. Usunął zatem zawodników, a dopiero gdy numery były już wolne, przypisał je innym.

Nazajutrz, odbywał się ligowy mecz ze Startem Mierzyno. Kazimierz wziął ze sobą laptopa i nie zasiadł jak zawsze na ławce trenerskiej, lecz na trybunach w najwyższym rzędzie, by mieć, jak najlepszy widok. Uruchomił aplikację i na podstawie protokołu meczowego uzupełnił wyjściową jedenastkę oraz listę piłkarzy rezerwowych. Załadował skład, przeszedł do zakładki zbierania statystyk - wpisał nazwę przeciwnika i był gotowy, aż sędzia rozpocznie mecz. Po chwili sędzia gwizdnął i Kazimierz przystąpił do pracy. Pilnie notował strzały, podania, przechwyty, czy faule, które po zatwierdzeniu wyświetlały się w historii akcji. Składniki aktualnie analizowanej akcji także były wyświetlane i modyfikowane, po dodaniu kolejnych jej elementów, takich jak przypisanie zawodnika, czy rodzaju strzału. Po kwadransie gry, po fatalnym błędzie lewego obrońcy Lewczuka, padł pierwszy gol, opisany komentarzem o wspomnianym błędzie. Po przerwie Kaziu miał mnóstwo roboty - co chwilę napastnicy Zawiszy oddawali kolejne strzały. Po jednym z nich nastąpił rzut rożny – dośrodkowanie, walka o piłkę i faul na Malinowskim – rzut karny - krótka przerwa w grze, którą pan Kazimierz mógł wykorzystać by zanotować niezbędne informacji. Gol z karnego, dał korzystny wynik – trener zdecydował się dokonać dwóch zmian i postawić na grę bardziej defensywną. Pan Szczęsny już zaznaczył zawodników do zmiany, lecz nim do niej doszło Wolski zasygnalizował kontuzję. System umożliwił wycofanie aktualnie dodawanej akcji i wprowadzanie nowej oraz dodanie informacji o kontuzji. W końcówce meczu miała miejsce sprzeczka, po faulu w środku boiska. Nerwy puściły zawodnikom – nie wytrzymał Bykowski, który, za skandaliczne uderzenie przeciwnika w twarz, otrzymał czerwoną kartkę. Po zaznaczeniu go jako wykluczonego, zawodnik zniknął z listy przebywających na boisku – przypisanie go do kolejnych akcji nie było już możliwe. Po chwili słysząc końcowy gwizdek – Kazimierz zakończył mecz – wszystkie kontrolki odpowiadające za tworzenie kolejnych akcji, zostały zablokowane. Pan Kazimierz wyłączył aplikację i udał się do szatni.

Następnego dnia podczas analizy meczu, asystent trenera przedstawił zebrane statystyki. Po porównaniu dwóch napastników, okazało się, że pierwszy ma tendencje do wielu podań. Wykres z kolei pokazał, że drugi zdecydowanie częściej oddaje strzały. Defensywny pomocnik Popek, okazał się niezwykle twardo grającym zawodnikiem – z dwudziestu popełnionych przez drużynę fauli, aż osiem było jego autorstwa. Trenera zaniepokoiła niezwykle niska skuteczność celności podań - 58%. Na podstawie analizy postanowiono, że należy podawać w sposób bardziej przemyślany i preferować krótkie i pewne zagrania, zamiast dalekich wybić, w kierunku wysokiego napastnika. Pozostał także temat nieprzemyślanego zachowania Bykowskiego, ukaranego, już trzeci raz w bieżącym sezonie, czerwoną kartką. Doszło do ostrej wymiany zdań, zawodnik obraził trenera, skutkiem czego został wyrzucony z klubu. Sytuację tę w swojej aplikacji odnotował pan Kazimierz – klikając usunięcie profilu piłkarza. System poinformował go, że zawodnik ten ma na koncie występy w zespole. Aby zachować spójność zebranych dotychczas statystyk w meczach z udziałem tego zawodnika, profil zawodnika nie przestaje istnieć – a jedynie zostaje przez system oznaczony jako nieaktywny. Dzięki temu także, gdyby w przyszłości Bykowski wrócił do drużyny, pan Kazimierz mógłby go bez problemu przywrócić do kadry pierwszego zespołu, bez potrzeby ponownego prowadzania jego danych.

## Metodyka i narzędzia

### Metodyka wytwarzania

Zdecydowaliśmy się bazować na metodyce zwinnej - Scrum. Jednak ze względu na inne obowiązki każdego z nas postanowiliśmy zmodyfikować ją na nasze potrzeby.

Stosujemy różne długości sprintów. Dwa pierwsze sprinty będą trwały jeden tydzień. Dwa kolejne będą dwutygodniowe. Ostatnie dwa sprinty to również jednotygodniowy okres. Dodatkowo zdecydowaliśmy się nie specyfikować dokładnego zakresu każdej zaległości umieszczanej w Sprint Backlog. Uznaliśmy, że zdefiniowane przez nas i naszego konsultanta wymagania funkcjonalne są wystarczające i nie potrzebujemy dodatkowych opisów.

### Wykorzystanie narzędzi

Zdecydowaliśmy się na środowisko NetBeans, gdyż można tworzyć w nim zarówno aplikacje desktopowe oraz mobilne w języku Java, co uprości nam nasze prace. To IDE pozwala również na sprawną synchronizację z repozytorium git’a.

Komunikować się będziemy za pomocą chatu serwisu Facebook oraz poczty elektronicznej. Do komunikacji głosowej wykorzystamy komunikator Skype. Planujemy również spotkania po każdym sprincie w celu podsumowania naszej pracy i uzgodnienia dalszych działań.

Repozytorium utworzyliśmy na stronie GitHub.com.

Nie przewidujemy wykonywania skomplikowanych projektów graficznych. Na wszelki wypadek, gdyby zaszła taka potrzeba będziemy korzystać z systemowego Paint’a oraz z programu Gimp.

### Współdzielenie kodu i dokumentacji

Repozytorium kodu i dokumentacji: http://github.com/sejman92/AplikacjaStatystykiInz

Pomoc w Scrumie: [www.sejmanix.acunote.com](http://www.sejmanix.acunote.com)

### Dokumentacja

Opis architektury.

Schemat ERD i opis bazy danych.

Opis i przebieg sprintów.

Notatki ze spotkań z klientem.

Poradnik dla użytkownika.

## Zespół i komunikacja

### Odpowiedzialność w zespole

Jakub Hopen- programista, projektant, odpowiedzialny będzie za bazę danych i wsparcie programistyczne aplikacji mobilnej i desktopowej.

Wojciech Pasternak- programista, projektant, odpowiedzialny za stworzenie odpowiedniej dokumentacji produktu

Mateusz Szymański – programista, projektant, odpowiedzialny będzie za kontakt z klientem i jego przedstawicielami.

### Komunikacja w zespole

Zespół będzie się komunikował poprzez komunikator Skype, poprzez chat portalu Facebook i droga mailową. Będą też odbywać się spotkania na zakończenie sprintu.

### Komunikacja zewnętrzna

Zespół będzie się komunikował z opiekunem pracy drogą mailowa a także cotygodniowymi spotkaniami, za bezpośredni kontakt z klientem będzie odpowiadał Mateusz Szymański. Będzie on spotykał się z klientem po każdym zakończonym sprincie. Podczas tych spotkań odbywać się będzie prezentacja nowego przyrostu, a także zebranie nowych uwag i wymagań od klienta.

## Wstępny harmonogram

Sprint 1: 12.10.2014 - 18.10.2014 -> 1tydz.

Zadania: GUI wersji desktopowej (dla widoku zbierającego dane), schemat bazy danych, uruchomienie zdalnej bazy danych. Możliwość zarządzania drużynami.

Sprint 2: 19.10.2014 - 1.11.2014 -> 2tyg.

Zadania: dokończenie GUI dla wersji desktopowej i implementacja podstawowych funkcjonalności jak dodanie zawodników i pierwsze zbieranie statystyk ( podstawowe ).

Sprint 3: 02.11.2014 - 15.11.2014 -> 2tyg.

Zadania: uzupełnienie funkcjonalności zbierania statystyk, implementacja podstawowych elementów analizy ( wyświetlanie ).

Sprint 4: 16.11.2014 - 22.12.2014 -> 1tydz

Zadania: rozwijanie “analizatora” poprzez dodanie wykresów.

Sprint 5: 23.11.2014 - 29.12.2014 -> 1tydz

Zadania: ten sprint będzie przeznaczony na implementacje elementów, których nie udało się zaimplementować wcześniej ( wystąpiły problemy i trzeba nadrobić braki)

Sprint 6: 30.11.2014 - 07.12.2014 -> 1tydz

Zadania: Upiększanie aplikacji, prace nad wyglądem GUI.

## Zapewnienie jakości

Podczas wytwarzania aplikacji będą zapewnione systematyczne konsultacje z klientem mające na celu pokrycie wszelkich oczekiwań i wymagań co do produktu końcowego. Po każdym sprincie ( zaczynając od 2 ) będziemy ręcznie testować aplikację.

## Analiza ryzyka

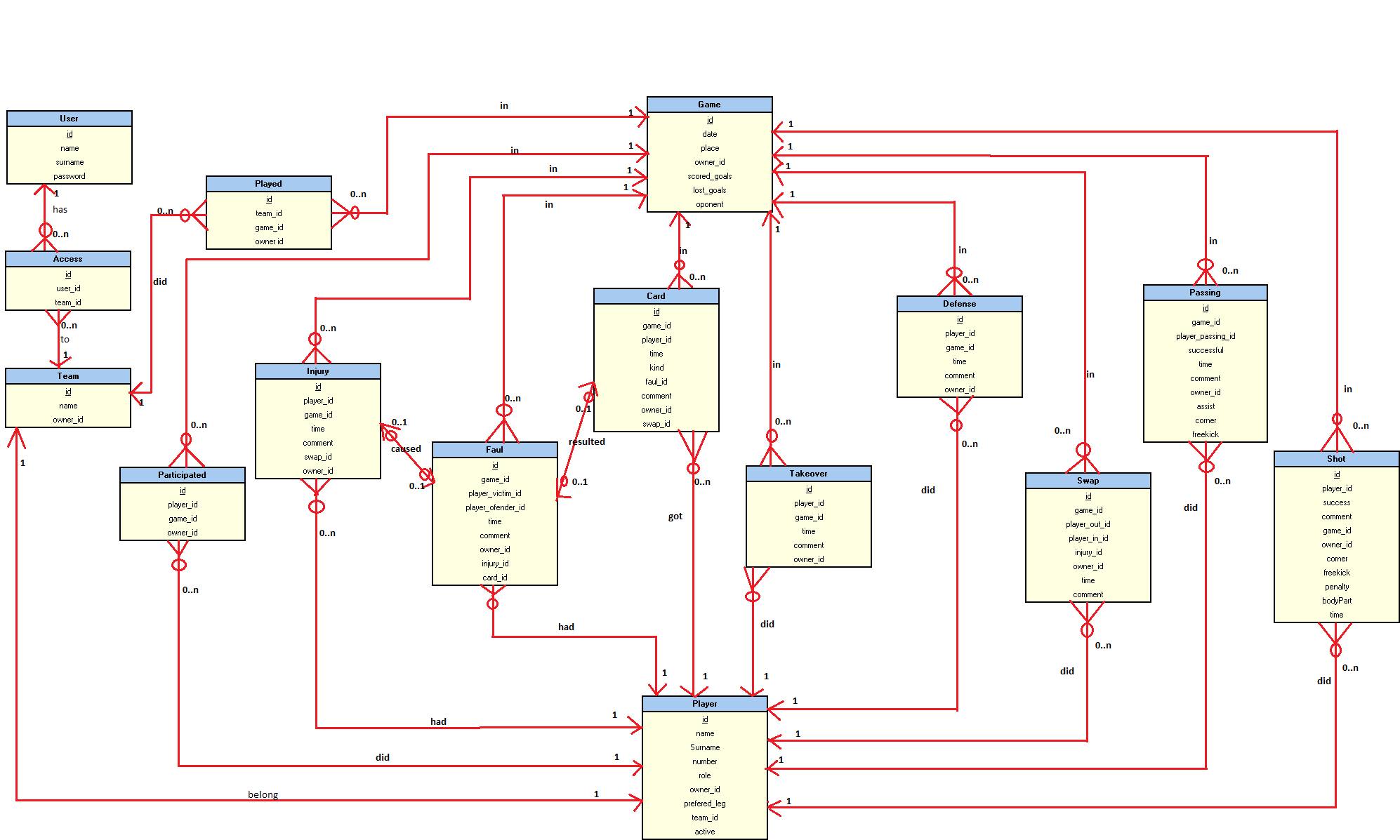
Głównym ryzykiem może być brak możliwości współpracy lokalnej całego zespołu oraz codziennych spotkań. Jednak dzięki komunikatorom i repozytorium ryzyko to jest sprowadzane do minimum.

Innym możliwym ryzykiem jest mały okres czasu w jakim nasz produkt powstanie. Może to wpłynąć na jego jakość i stabilność, jednak przygotowaliśmy 3 wersje Produkt Backlog, co pozwoli nam wykonać produkt, który być może będzie mniej funkcjonalny niż byśmy chcieli, ale będzie produktem działającym.

Niewątpliwie problemem może okazać się kontakt z klientem, jego dostępność i zaangażowanie w projekt. Klient wyraził dużą chęć pomocy, ale ze względu na to, że jest on dość zapracowaną osobą, możemy napotkać problemy. Jednak, aby temu zapobiec będziemy mieli możliwość konsultacji z kompetentną, wyznaczoną przez niego osobą.

# Opis systemu

## Struktura bazy danych

Schemat ERD Relacyjnej bazy danych przedstawia Rys. 2.1 zaprojektowanej dla zdalnej bazy danych wykorzystywanej przez desktopową wersję aplikacji.

Dokładny opis encji znajduje się w tabelach 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 2.10, 2.11, 2.12, 2.13, 2.14, 2.15.

### Opis zbioru encji

Tabela 2.1 User

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Opis** | | | | |
| Zbiór encji przechowujący użytkowników i ich dane | | | | |
| **Atrybuty** | | | | |
| **Nazwa** | **Klucz główny** | **Typ/ dziedzina** | **Opis** | **Null** |
| id | Tak | Int | Unikalny ID użytkownika | Nie |
| name | Nie | varchar | Imię użytkownika | Nie |
| surname | Nie | varchar | Nazwisko użytkownika | Nie |
| password | Nie | varchar | Hasło użytkownika | Nie |

Tabela 2.2Team

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Opis** | | | | |
| Zbiór encji zawierający drużyny i informacje o nich. | | | | |
| **Atrybuty** | | | | |
| **Nazwa** | **Klucz główny** | **Typ/ dziedzina** | **Opis** | **Null** |
| id | Tak | Int | Unikalny id drużyny. | Nie |
| name | Nie | varchar | Nazwa drużyny. | Nie |
| owner\_id | Nie | int | Id właściciela danych. | nie |

Tabela 2.3 Player

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Opis** | | | | |
| Zbiór encji zawierający zawodników i informacje o nich | | | | |
| **Atrybuty** | | | | |
| **Nazwa** | **Klucz główny** | **Typ/ dziedzina** | **Opis** | **Null** |
| id | Tak | int | Unikalne id zawodnika. | Nie |
| name | Nie | varchar | Imię zawodnika. | Nie |
| surname | Nie | varchar | Nazwisko zawodnika. | Nie |
| number | Nie | int | Numer zawodnika. | Nie |
| role | Nie | varchar | Pozycja zawodnika. | Nie |
| owner\_id | Nie | varchar | Id właściciela danych. | Tak |
| prefered\_leg | Nie | char | Preferowana noga zawodnika. Może przyjmować wartości LEWA lub PRAWA | Tak |
| team\_id | Nie | int | Id drużyny, do której należy zawodnik. | Tak |
| active | Nie | varchar | Czy zawodnik jest aktywny. Może przyjmować wartości AKTYWNY lub NIEAKTYWNY | Tak |

Tabela 2.4 Game

| **Opis** | | | | |
| --- | --- | --- | --- | --- |
| Zbiór encji zawierający mecze i dane o nich | | | | |
| **Atrybuty** | | | | |
| **Nazwa** | **Klucz główny** | **Typ/ dziedzina** | **Opis** | **Null** |
| id | Tak | int | Unikalne id meczu. | Nie |
| date | Nie | date | Data meczu. | Tak |
| place | Nie | varchar | Miejsce gry | Tak |
| scored\_goals | Nie | int | Strzelone gole podczas meczu. | Tak |
| lost\_goals | Nie | int | Stracone gole podczas meczu. | Tak |
| oponent | Nie | varchar | Nazwa drużyny przeciwnej. | Tak |
| owner\_id | Nie | varchar | Id właściciela danych. | Tak |

Tabela 2.5 Shot

| **Opis** | | | | |
| --- | --- | --- | --- | --- |
| Zbiór encji zawierający strzały na bramkę w meczy i informacje o nich. | | | | |
| **Atrybuty** | | | | |
| **Nazwa** | **Klucz główny** | **Typ/ dziedzina** | **Opis** | **Null** |
| id | Tak | int | Unikalny id strzału | Nie |
| player\_id | Nie | int | ID zawodnika. | Tak |
| comment | Nie | varchar | Komentarz do strzału. | Tak |
| success | Nie | varchar | Opisuje, jakim efektem zakończył się strzał. Może przyjmować wartości: CELNY, NIECELNY, GOL | Tak |
| game\_id | Nie | int | Id meczu, podczas, którego był strzał. | Tak |
| owner\_id | Nie | int | Id właściciela danych. | Tak |
| corner | Nie | boolean | Czy strzał był rzutem rożnym. | Tak |
| freekick | Nie | boolean | Czy strzał był z rzutu wolnego. | Tak |
| penalty | Nie | boolean | Czy strzał był z rzutu karnego. | Tak |
| bodyPart | Nie | varchar | Część ciała, którą był wykonany strzał. Może przyjmować wartości: LEWA NOGA, PRAWA NOGA, GŁOWA, KLATKA, INNA | Tak |
| time | Nie | int | Czas zajścia akcji w minutach | Tak |

Tabela 2.6 Card

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Opis** | | | | |
| Zbiór encji przechowujący kartki otrzymane przez zawodników podczas meczy. | | | | |
| **Atrybuty** | | | | |
| **Nazwa** | **Klucz główny** | **Typ/ dziedzina** | **Opis** | **Null** |
| id | Tak | int | Unikalny ID kartki | Nie |
| game\_id | Nie | int | Id meczu, podczas którego była kartka | Tak |
| player\_id | Nie | int | ID zawodnika, który dostał kartkę | Tak |
| time | Nie | int | Czas otrzymania kartki w minutach | Tak |
| kind | Nie | varchar | Kolor kartki. Może przyjmować wartości ŻÓŁTA lub CZERWONA | Tak |
| faul\_id | Nie | int | id faulu, za który była nadana kartka | Tak |
| comment | Nie | varchar | komentarz do akcji | Tak |
| owner\_id | Nie | int | id właściciela dane | Tak |
| swap\_id | Nie | int | id zmiany, jeśli miała miejsce | Tak |

Tabela 2.7 Faul

| **Opis** | | | | |
| --- | --- | --- | --- | --- |
| Zbiór encji zawierający faule i informacje o nich | | | | |
| **Atrybuty** | | | | |
| **Nazwa** | **Klucz główny** | **Typ/ dziedzina** | **Opis** | **Null** |
| id | Tak | int | Unikalne id faulu. | Nie |
| game\_id | Nie | int | ID meczu, podczas którego był faul. | Tak |
| player\_victim\_id | Nie | int | ID zawodnika, który był faulowany. | Tak |
| player\_ofender\_id | Nie | int | ID zawodnika, który faulował. | Tak |
| time | Nie | int | Czas faulu w minutach | Tak |
| comment | Nie | varchar | Komentarz do akcji. | Tak |
| owner\_id | Nie | int | id właściciela danych. | Tak |
| injury\_id | Nie | int | id kontuzji, która miała miejsce przez faul. | Tak |
| card\_id | Nie | int | id kartki, którą zawodnik dostał za faul. | Tak |

Tabela 2.8 Injury

| **Opis** | | | | |
| --- | --- | --- | --- | --- |
| Zbiór encji zawierający kontuzje zawodników i informacje o nich. | | | | |
| **Atrybuty** | | | | |
| **Nazwa** | **Klucz główny** | **Typ/ dziedzina** | **Opis** | **Null** |
| id | Tak | int | Unikalne id kontuzji. | Nie |
| player\_id | Nie | int | Id kontuzjowanego zawodnika. | Tak |
| game\_id | Nie | int | Id meczy, podczas, którego zaszła kontuzja. | Tak |
| time | Nie | int | czas zdarzenia w minutach | Tak |
| comment | Nie | varchar | Komentarz do akcji. | Tak |
| swap\_id | Nie | int | Id zmiany która zaszła w wyniku kontuzji | Tak |
| owner\_id | Nie | int | Id właściciela danych. | Tak |

Tabela 2.9 Swap

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Opis** | | | | |
| Zbiór encji zawierający zmiany, które zaszły podczas meczu. | | | | |
| **Atrybuty** | | | | |
| **Nazwa** | **Klucz główny** | **Typ/ dziedzina** | **Opis** | **Null** |
| id | Tak | int | Unikalny id zmiany. | Nie |
| game\_id | Nie | int | Id meczy, podczas, którego zaszła zmiana. | Tak |
| player\_out\_id | Nie | int | Id zawodnika, który schodził. | Tak |
| player\_in\_id | Nie | int | Id zawodnika, który wchodził. | Tak |
| injury\_id | Nie | int | ID kontuzji, jeśli była przyczyną zmiany | Tak |
| owner\_id | Nie | int | Id właściciela danych. | Tak |
| time | Nie | int | Czas zmiany w minutach. | Tak |
| comment | Nie | varchar | Komentarz do zmiany. | Tak |

Tabela 2.10 Defence

| **Opis** | | | | |
| --- | --- | --- | --- | --- |
| Zbiór encji zawierający obrony i informacje o nich | | | | |
| **Atrybuty** | | | | |
| **Nazwa** | **Klucz główny** | **Typ/ dziedzina** | **Opis** | **Null** |
| id | Tak | int | Unikalny id akcji | Nie |
| player\_id | Nie | int | ID zawodnika, który obronił | Tak |
| game\_id | Nie | int | Id meczu, w którym zaszła obrona | Tak |
| time | Nie | int | Czas sytuacji w minutach | Tak |
| comment | Nie | varchar | komentarz do akcji | Tak |
| owner\_id | Nie | int | id właściciela danych | Tak |

Tabela 2.11 Passing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Opis** | | | | |
| Zbiór encji zawierający podania i informacje o nich. | | | | |
| **Atrybuty** | | | | |
| **Nazwa** | **Klucz główny** | **Typ/ dziedzina** | **Opis** | **Null** |
| id | Tak | int | Unikalne id podania | Nie |
| game\_id | Nie | int | ID meczu, podczas, którego zaszło podanie. | Tak |
| player\_passing\_id | Nie | int | ID zawodnika, który podał piłkę | Tak |
| time | Nie | int | Czas zajścia sytuacji w minutach | Tak |
| comment | Nie | varchar | komentarz do akcji | Tak |
| successful | Nie | boolean | Czy podanie było udane. Może przyjmować wartości CELNE lub NIECELNE | Tak |
| owner\_id | Nie | int | Id właściciela danych | Tak |
| assist | Nie | boolean | Czy podanie było asystą. | Tak |
| corner | Nie | boolean | Czy podanie było rzutem rożnym. | Tak |
| freekick | Nie | boolean | Czy podanie było rzutem wolnym. | Tak |

Tabela 2.12 Takeover

| **Opis** | | | | |
| --- | --- | --- | --- | --- |
| Zbiór encji zawierający przejęcia piłki w meczu. | | | | |
| **Atrybuty** | | | | |
| **Nazwa** | **Klucz główny** | **Typ/ dziedzina** | **Opis** | **Null** |
| id | Tak | int | Unikalny id przejęcia. | Nie |
| player\_id | Nie | int | ID zawodnika, który przejął piłkę. | Tak |
| game\_id | Nie | int | Id meczu, w którym nastąpiło przejęcie. | Tak |
| time | Nie | int | Czas przejęcia w minutach | Tak |
| comment | Nie | varchar | Komentarz do akcji. | Tak |
| owner\_id | Nie | int | Id właściciela danych. | Tak |

Tabela 2.13 Access

| **Opis** | | | | |
| --- | --- | --- | --- | --- |
| Zbiór encji służący do określenia, jaki użytkownika ma prawa dostępu, do których drużyn. | | | | |
| **Atrybuty** | | | | |
| **Nazwa** | **Klucz główny** | **Typ/ dziedzina** | **Opis** | **Null** |
| id | Tak | int | Unikalne id praw dostępu. | Nie |
| user\_id | Nie | int | Id użytkownika określanemu prawa | Tak |
| team\_id | Nie | int | ID drużyna, której dotyczą prawa | Tak |

Tabela 2.14 Played

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Opis** | | | | |
| Zbiór encji służący do połączenia drużyny z meczem, który rozegrała. | | | | |
| **Atrybuty** | | | | |
| **Nazwa** | **Klucz główny** | **Typ/ dziedzina** | **Opis** | **Null** |
| id | Tak | int | Unikalne id rozegrania | Nie |
| team\_id | Nie | int | Id drużyny, która rozegrała mecz. | Tak |
| game\_id | Nie | int | Id meczy, który rozegrała drużyna | Tak |
| owner\_id | Nie | int | Id właściciela danych. | Tak |

Tabela 2.15 Participated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Zbiór encji 15: Participated** | | | | |
| **Opis** | | | | |
| Zbiór encji służący do połączenia gracza z meczem, w którym brał udział. | | | | |
| **Atrybuty** | | | | |
| **Nazwa** | **Klucz główny** | **Typ/ dziedzina** | **Opis** | **Null** |
| id | Tak | int | Unikalne id uczestnictwa. | Nie |
| player\_id | Nie | int | Id zawodnika, który brał udział w meczu. | Nie |
| game\_id | Nie | int | Id meczy, w którym zawodnik brał udział | Nie |
| owner\_id | Nie | int | Id właściciela danych | Tak |

### Opis związków

W tabeli 2.16 przedstawiliśmy opis związków encji z uwzględnieniem liczności.

Tabela 2.16 Opis związków encji

| **Nazwa związku** | **Zbiory encji** | | **Liczność związku** |
| --- | --- | --- | --- |
| **Zbiór encji 1** | **Zbiór encji 2** |
| Belong | Team | Player | 1 : n |
| Did | Player | Shot | 1 : 0..n |
| In | Shot | Game | 0..n : 1 |
| In | Game | Card | 1 : 0..n |
| Got | Player | Card | 1 : 0..n |
| In | Game | Faul | 1 : 0..n |
| Had | Player | Faul | 1..2 : 0..n |
| In | Game | Injury | 1 : 0..n |
| Had | Player | Injury | 1 : 0..n |
| Did | Swap | Player | 1 : 2 |
| In | Swap | Game | 0..n : 1 |
| In | Defense | Game | 0..n : 1 |
| Did | Defense | Player | 0..n : 1 |
| In | Game | Passing | 1 : 0..n |
| Did | Player | Passing | 2 : 0..n |
| Did | Player | Takeover | 1 : 0..n |
| In | Game | Takeover | 1 : 0..n |
| Caused | Faul | Injury | 0..1 : 0..1 |
| Has | User | Access | 1 : 0..n |
| To | Access | Team | n : 1 |
| In | Game | Played | 1 : n |
| Did | Team | Played | 1 : 0..n |
| In | Game | Participated | 1 : 0..n |
| Did | Player | Participated | 1 : 0..n |
| Caused | Card | Faul | 0..1 : 0..1 |

## Technologie

Przy tworzeniu wersji desktopowej naszej aplikacji wykorzystaliśmy wzorzec MVC. Wykorzystanie go pozwoliło nam oddzielić logikę biznesową od prezentacji. Jest to główny powód, dlaczego zdecydowaliśmy się na właśnie ten wzorzec. Nasza aplikacja jest przeznaczona dla trenerów drużyn piłkarskich do korzystania podczas meczów do prowadzenia zapisów akcji i statystyk. Aby nasz produkt spełniał swoje zadanie, musiał przede wszystkim mieć czytelny i intuicyjny interfejs użytkownika. Dzięki wykorzystaniu wzorca MVC, gdy tylko klient uznał, iż użycie pewnej funkcjonalności jest dla niego prostsze przy innym ułożeniu elementów na ekranie, mogliśmy szybko wprowadzić wymagane zmiany bez konieczności ingerowania w logikę aplikacji.

Do utworzenia interfejsu użytkownika na platformie Java postanowiliśmy użyć JavaFX, technologia ta doskonale się wpasowała we wzorzec MVC. Wykorzystanie JavaFX pozwoliło na utworzyć przejrzysty i wygodny interfejs graficzny dzięki wbudowanym kontrolkom.

Baza danych została umieszczona na darmowym serwerze [www.db4free.net](http://www.db4free.net). Jest to serwis udostępniający bazy danych zarządzanych w systemie MySQL. Wybraliśmy ten serwis, ponieważ udostępniał on możliwość zarządzania bazą danych poprzez webowe narzędzie phpMyAdmin, które pozwoliło nam w szybki i łatwy sposób modyfikować istniejącą bazę danych poprzez interfejs graficzny. Ponieważ podczas tworzenia aplikacji wielokrotnie doszło do zmian w strukturze bazy danych było to bardzo przydatną cechą przyśpieszającą tworzenie aplikacji. Drugą istotną zaletą tego serwisu był nielimitowany czas istnienia bazy danych na serwerach, dzięki czemu nasz produkt będzie mógł być w dalszym ciągu rozwijany.

Do połączenia aplikacji desktopowej ze zdalną bazą danych użyliśmy standardu Java Persistence API. Zdecydowaliśmy się na korzystanie z niego, ponieważ pozwalał na pracę na automatycznie generowanych obiektach odwzorowujących encje i ich relacje w bazie danych. Pozwoliło to zaoszczędzić wiele czasu gdyż przy każdej zmianie w strukturze bazy danych, wszelkie zmiany były automatycznie wprowadzane do kodu aplikacji.

W wersji aplikacji na urządzenia mobilne z systemem Android także zastosowaliśmy wzorzec MVC, gdyż tak samo uprościło a co za tym idzie przyśpieszyło pracę nad produktem. W wersji mobilnej aplikacji zdecydowaliśmy wykorzystać wbudowaną w system Android bazę danych zarządzaną przez SQLite. Zdecydowaliśmy się na zastosowanie zdalnej bazy danych dla wersji desktopowej a lokalnej dla wersji mobilnej, ze względu na inne docelowe zastosowanie dla obu tych aplikacji.

Wersja mobilna została zaprojektowana dla szybkiego wprowadzania akcji do systemu za pomocą jak najmniejszej liczby kliknięć i w jak najkrótszym czasie, gdyż, jak wiadomo, podczas meczu co chwilę ma miejsce ważna sytuacja. Dzięki skondensowanemu interfejsowi i wbudowanej w system Android możliwości rozpoznawania mowy udało nam się stworzyć aplikację, która w szybki sposób rejestruje zdarzenia i komentarze do nich i zapisuje je do lokalnej bazy danych, dzięki czemu nagły brak dostępu do Internetu podczas meczu nie ma wpływu na zakres funkcjonalności aplikacji.

Wersja desktopowa aplikacji została natomiast zaprojektowana do wykorzystania podczas treningów drużyny. Interfejs został dostosowany do możliwości komputerów osobistych nie kładąc tak dużego nacisku na szybkość wpisywania danych. Jednakże podczas meczy treningowych, zawodnicy obu drużyn są de facto zawodnikami jednej drużyny klubowej, przez co wymagana jest możliwość śledzenia statystyk obu drużyn z osobna. Umożliwi to stworzenie dokładniejszej bazy danych, z której będzie można wywnioskować, jacy zawodnicy grają lepiej ze sobą, a którzy mają większe problemy z dogadaniem się podczas gry. Jednakże wprowadzanie statystyk dla dwóch drużyn jednocześnie przez jedną osobę jest niewykonalne, dlatego też zdecydowaliśmy się na zastosowanie zdalnej bazy danych przechowującej dane z treningów drużyny, do której dostęp będzie mogło mieć wiele urządzeń. Dzięki temu będzie można podzielić zadanie wprowadzania statystyk dla drużyn na dwie i więcej osób, a mimo to wszystkie te dane będą przechowywane w jednym miejscu. Dzięki temu trener będzie mógł dokładnie przestudiować wyniki meczu w późniejszym czasie. Dodatkowym atutem tego rozwiązania jest, iż z racji na większą liczbę meczy treningowych niż rzeczywistych rozgrywek klubowych, dane treningowe będą znacznie obszerniejsze. Dzięki znacznie większym rozmiarom monitorów niż ekranów urządzeń mobilnych będzie możliwe czytelne tworzenie wykresów z treningów. Jednakże oba interfejsy zostały tak zaprojektowane by w przyszłości, jeśli pojawi się potrzeba synchronizacji baz danych z obu wersji aplikacji, możliwa była rozbudowa ich o tę nową funkcjonalność.

# Realizacja projektu

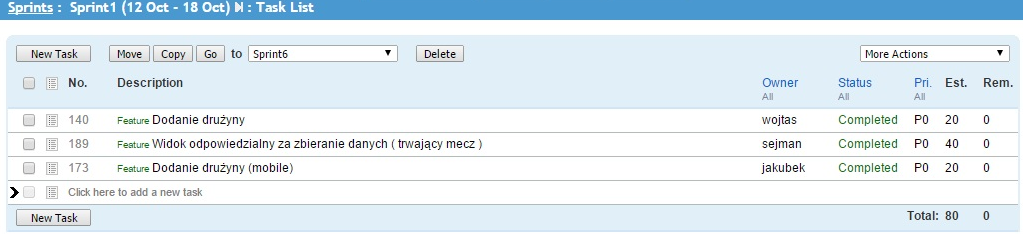
## Product backlog

Tutaj będzie zdjęcie PB

## Sprint 1 (12.10.2014- 18.10.2014)

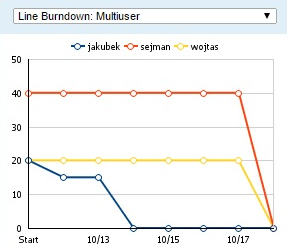
Zadania: GUI wersji desktopowej (dla widoku zbierającego dane), schemat bazy danych, uruchomienie zdalnej bazy danych. Możliwość zarządzania drużynami.

Pierwszy sprint skupiał się przede wszystkim na pracach nad architekturą. Najważniejszymi zadaniami było stworzenie modelu relacyjnej bazy danych i uruchomienie jej w sposób zdalny. W wersji desktopowej została stworzona podstawowa wersja wyglądu interfejsu użytkownika, umożliwiającego zbieranie statystyk. Zarówno w wersji desktopowej oraz mobilnej zostały zaimplementowane pierwsze funkcjonalności – tworzenie drużyny. W trakcie rozmowy za pomocą komunikatora Skype, która odbyła się na zakończenie sprintu, uzgodniliśmy zakres sprintu 2.



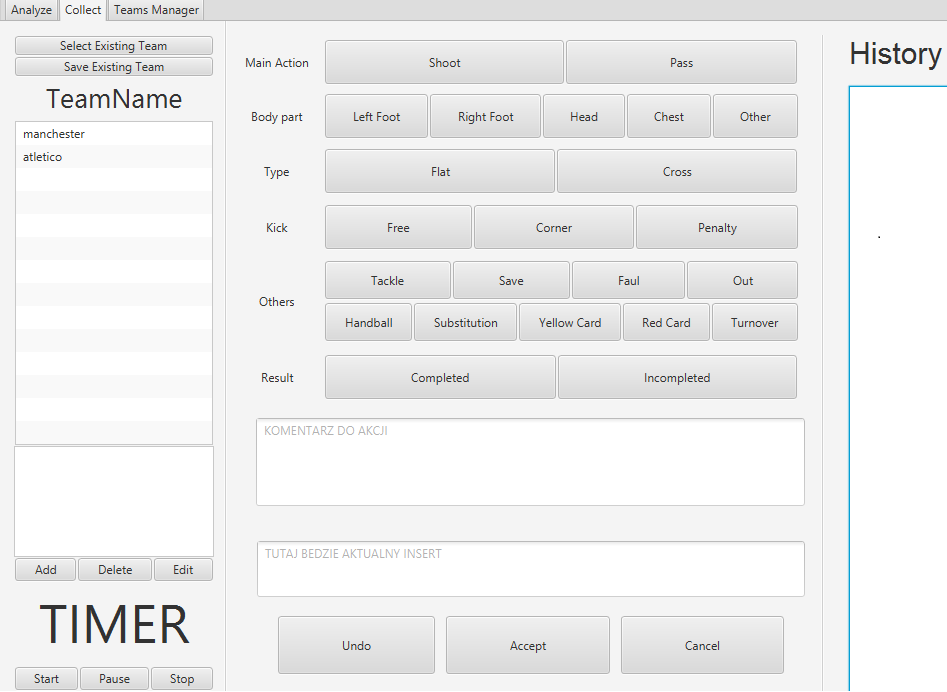
Rysunek . Sprint Backlog- Sprint 1

Na rys 3.1 przedstawiliśmy zadania z pierwszego sprintu. Ich ograniczona liczba spowodowana jest pracami nad architekturą aplikacji, potrzebą poznania technologii, a także krótką długością sprintu – 1 tydzień. Rys. 3.2 przedstawia wykres wypalania, stworzony przy pomocy narzędzia – www.acunote.com. Pokazuje on liczbę Story Points – czyli swojego rodzaju sumę trudności wykonywanych zadań przez poszczególnych członków zespołu w zależności od upływu czasu.

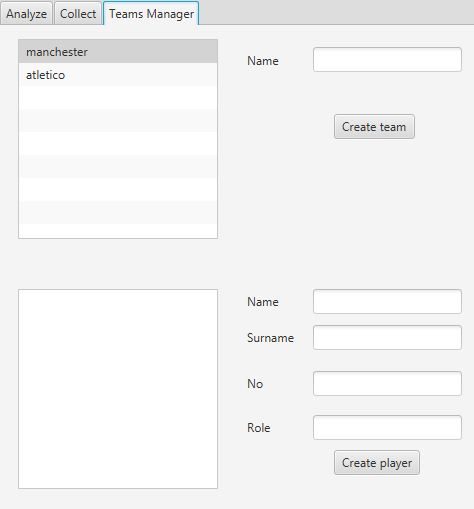


Rysunek . Wykres wypalania -Sprint 1

Poniżej przedstawiliśmy zrzuty ekranu z wyglądem interfejsu użytkownika, dotyczącego zbierania statystyk z meczów (rys 3.3) oraz zarządzania drużynami i ich zawodnikami (rys 3.4). W tym momencie w zakładce zarządzania drużyną dostępne jest tylko prymitywne GUI, umożliwiające tworzenie nowych zespołów i dodawanie do nich nowostworzonych zawodników. Z kolei interfejs zbierania statystyk jest nieco bardziej zaawansowany – posiada większość docelowych elementów, do których jednak, na chwilę obecną, nie są podpięte żadne akcje.



Rysunek . Interfejs dla panelu zbierania statystyk- Sprint 1



Rysunek . Interfejs dla panelu zarządzania drużynami - Sprint 1

## Sprint 2 (19.10.2014- 01.11.2014)

Dokończenie GUI dla wersji desktopowej i implementacja podstawowych funkcjonalności takich jak dodanie zawodników i pierwsze zbieranie statystyk (podstawowe).

Ze względu na długość drugiego sprintu – trwał 2 tygodnie (obok sprintu trzeciego najdłuższy) – głównym założeniem sprintu było wykonanie jak największego zakresu funkcjonalności. Udało się wykonać wszystkie przewidziane na ten czas zadania. Na początku sprintu zrealizowano dodawanie i usuwanie zawodników oraz poprawienie wyglądu GUI. Następnie stworzono dodawanie spotkania i wybierania do niego zawodników – zarówno ze składu wyjściowego oraz rezerwowego. W ostatnich dniach udało się także zrealizować funkcje odpowiedzialne za zbieranie podstawowych statystyk – strzału i podania. Ostatniego dnia odbyła się analiza sprintu przez członków zespołu. Ustalony został zakres prac na kolejny sprint, a postęp w pracach został oceniany przez zespół jako znaczący.

Z kolei przedostatniego dnia sprintu odbyło się spotkanie z klientem.

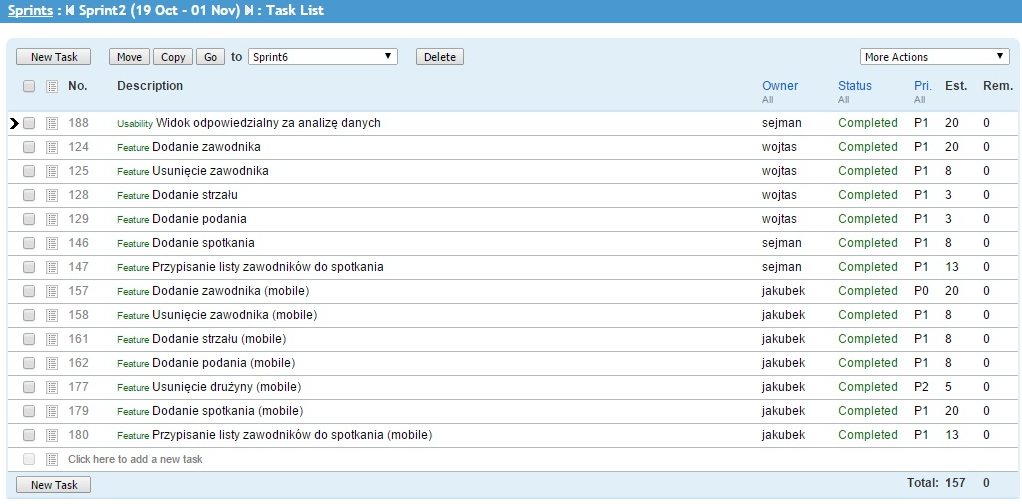
Pierwsze spotkanie z klientem - 31.10.2014:

Obecni: Mateusz Szymański, p. Henryk Kuczma

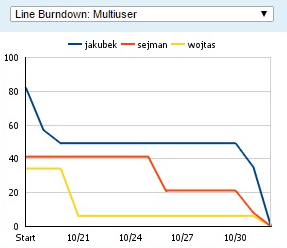
W ww. dniu odbyło się spotkanie przedstawiciela grupy projektowej (Mateusz Szymański) z klientem (p. Henryk Kuczma), podczas którego został zaprezentowany aktualny stan prac. Klient wyraził ogólne zadowolenie z tego, co zobaczył. Stwierdzi, że w widoku "Zarządzaj drużynami" nie potrzebuje tak dużej listy drużyn, gdyż będzie ich zaledwie kilka. Dodatkowo również zaznaczył, że mecz może rozpocząć się już w momencie zapisania 7 zawodników do podstawowego składu (do tej chwili warunkiem było 11 zawodników). Zaznaczył też, że nie zależy mu na wykresie kołowym i wolałby, żebyśmy skupili się na porównywaniu pojedynczych zawodników niż osiągach całej drużyny w różnych meczach. Uzgodniliśmy również, że kolejne spotkanie odbędzie się 14 lub 15 listopada (dokładny termin zostanie uzgodniony telefonicznie).

### Interfejs

Rys. 3.5 przedstawia funkcjonalność systemu wykonaną w czasie drugiej iteracji. Jak widać, zostało zrealizowanych aż kilkanaście funkcjonalności. Dzięki temu, po zakończeniu sprintu aplikacja umożliwiała już zbieranie prostych statystyk dotyczących meczu – strzału i podania. Wykres wypalania ( rys. 3.6) pokazuje, że największą pracę wykonał Jakub Hopen. Wykonał on wszystkie zadania na aplikację mobilną, podczas gdy pozostali członkowie zespołu podzielili pomiędzy siebie zadania, dotyczące aplikacje desktopową.

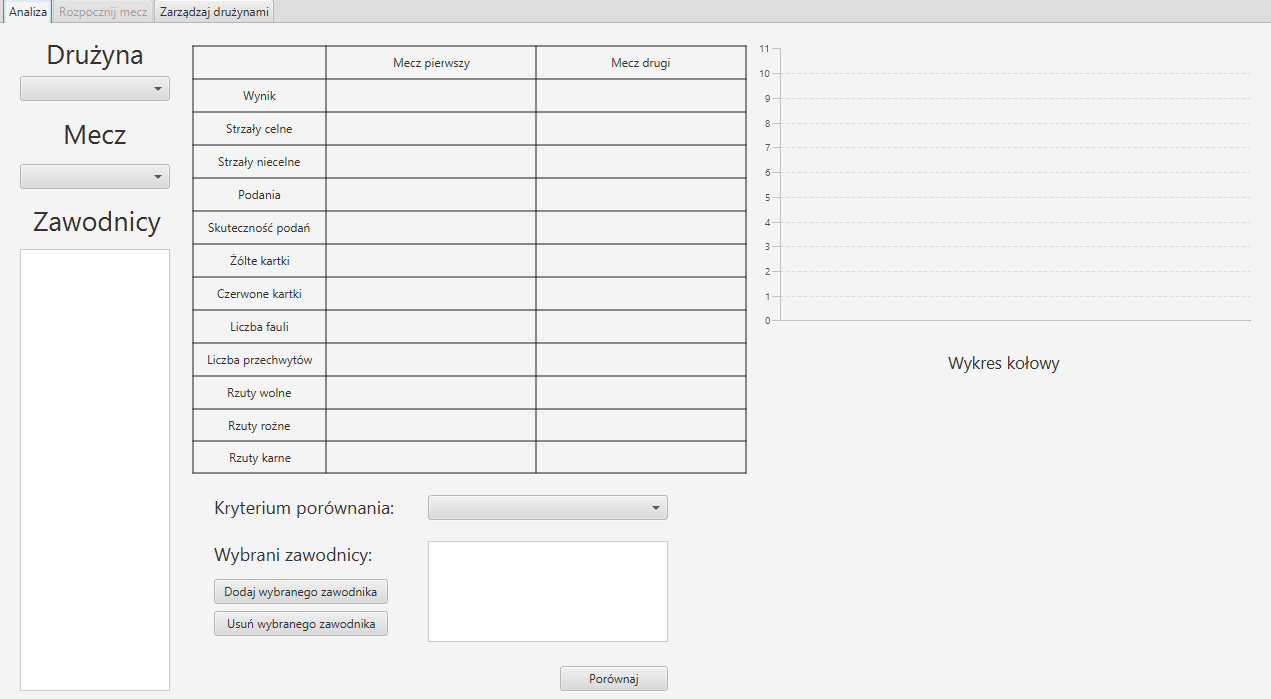


Rysunek . Sprint Backlog - Sprint 2

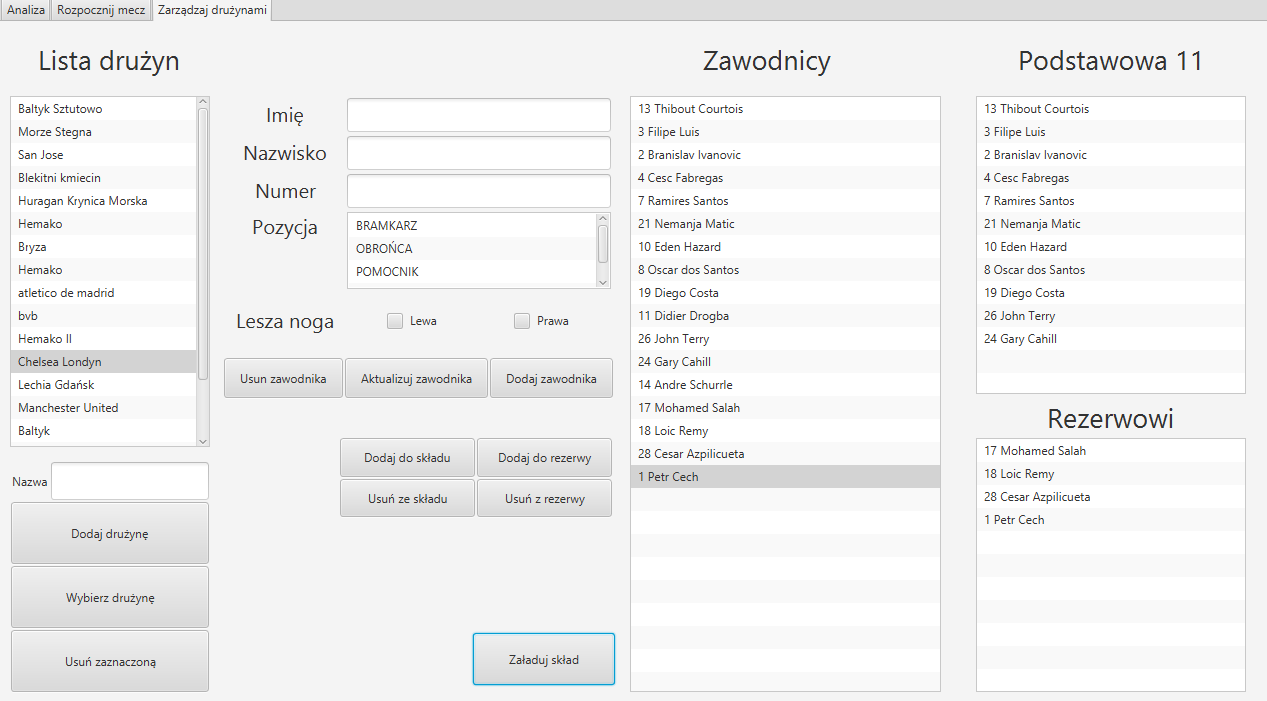


Rysunek . Wykres wypalania - Sprint 2

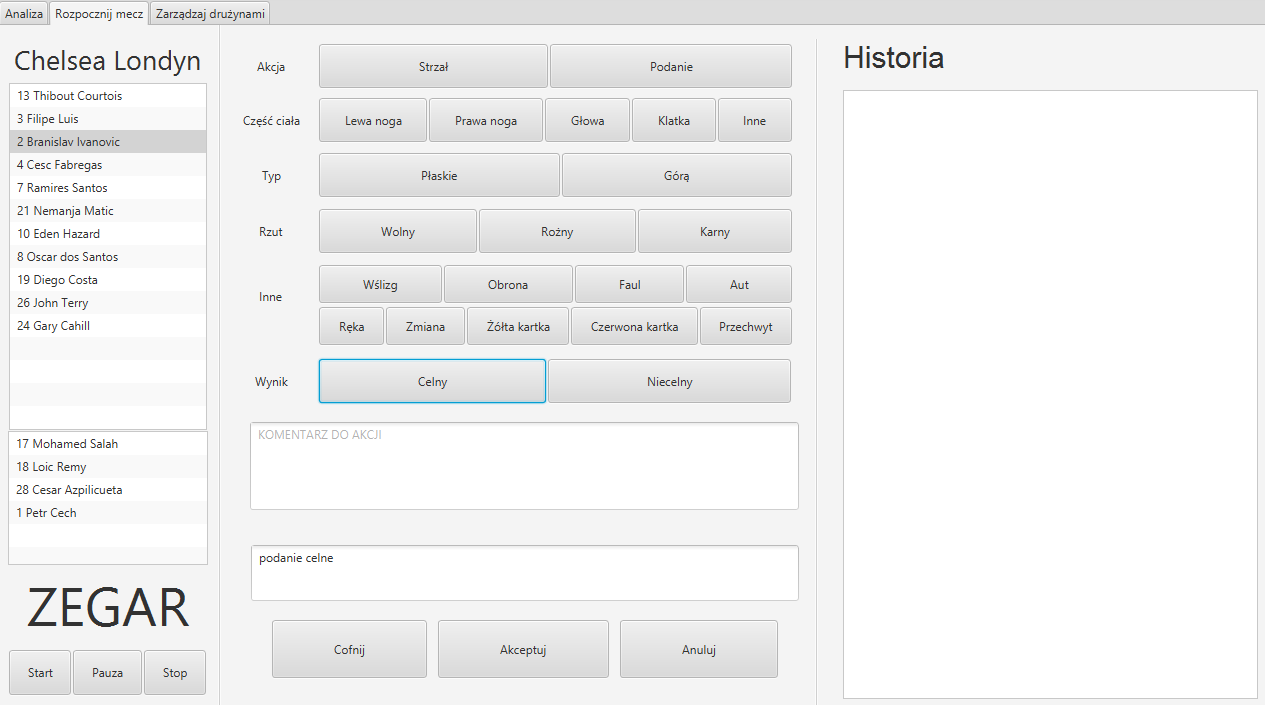
Zrzuty ekranu przedstawiają wygląd interfejsu użytkownika z perspektywy trzech widoków – w różnych zakładkach aplikacji desktopowej. Utworzony został widok analizy statystyk (rys 3.7) – tymczasowo tylko w formie tabeli, docelowo także przy pomocy wykresu. W widoku zarządzania drużynami (rys. 3.8) zostały stworzone przyciski, pozwalające edytować i usuwać drużyny oraz zawodników. Dodatkowo została zaimplementowana także funkcjonalność pozwalająca na dodawanie zawodników, zarówno do pierwszej jedenastki, jak i jako rezerwowych. W widoku analizy (rys. 3.9) , zostały podpięte podstawowe akcje, umożliwiające dodanie strzału i podania.



Rysunek . Interfejs dla panelu analizy danych - Sprint 2



Rysunek . Interfejs dla panelu zarządzania drużynami - Sprint 2



Rysunek . Interfejs dla panelu zbierania statystyk - Sprint 2

## Sprint 3 (02.11.2014 – 15.11.2014)

Zadania: uzupełnienie funkcjonalności zbierania statystyk, implementacja podstawowych elementów analizy (wyświetlanie).

Sprint 3 był jednym z dwóch najdłuższych sprintów, przez co miał okazać się kluczowy ze względu na duży przyrostu funkcjonalności. Zrealizowaliśmy usuwanie drużyn oraz edycję ich i zawodników. Udało się zrealizować dodawanie prawie wszystkich typów akcji. Niestety, poprzez problemy jakie napotkaliśmy, nie udało nam się zaprogramować kilku funkcjonalności, w tym analizy spotkań.

Podczas pracy nad aplikacją mobilną w jednej z aktywności, kontrolki EditText nie chciały zwracać żadnej wartości (zwracały null), przy wykorzystaniu metody EditText.getText().toString(). Nad tym problemem spędziliśmy dwa wieczory. Po wielu próbach udało się to naprawić (utworzenie nowej, analogicznej aktywności i przepisanie kodu). Przez opóźnienia byliśmy zmuszeni przenieść część funkcjonalności do sprintu 4. Mimo to, byliśmy zadowoleni z postępu prac nad projektem.

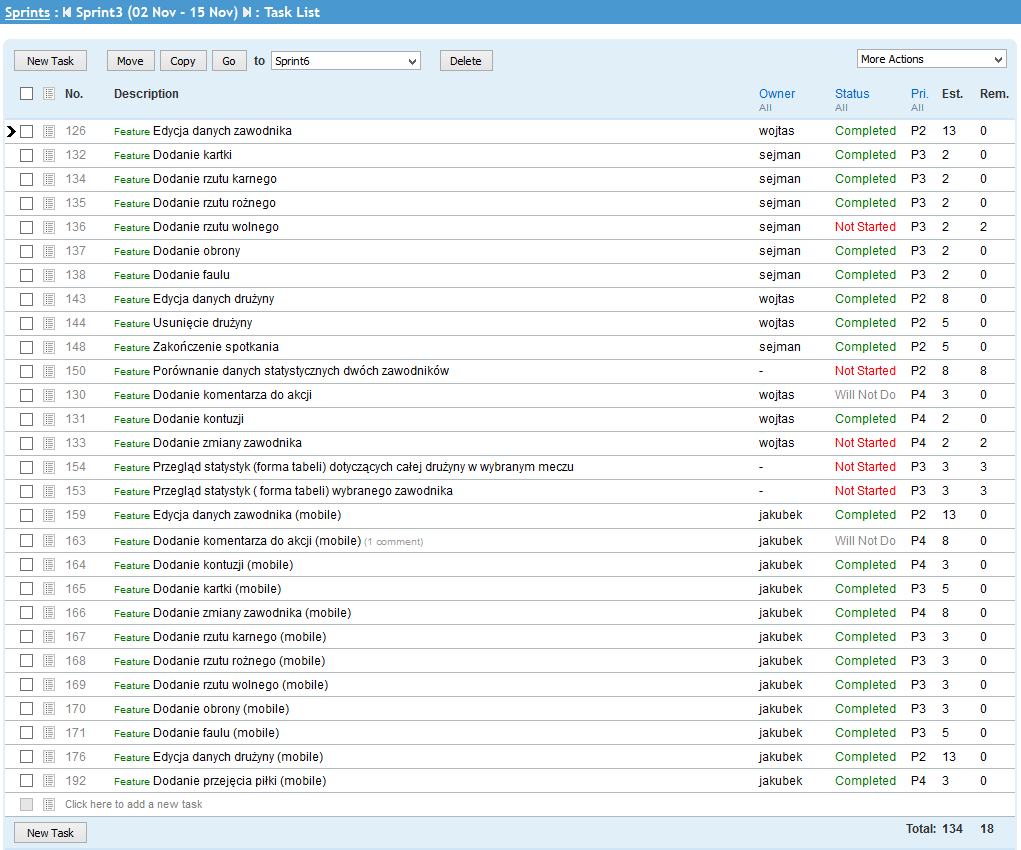
Przedostatniego dnia sprintu odbyło się spotkanie z klientem.

14.11.2014

Obecni: Mateusz Szymański, p. Henryk Kuczma

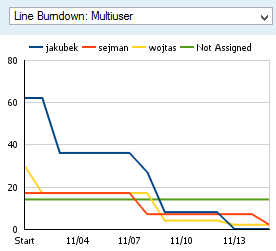
W ww. dniu odbyło się spotkanie przedstawiciela grupy projektowej (Mateusz Szymański) z klientem (p. Henryk Kuczma), podczas którego został zaprezentowany aktualny stan prac. Klient wyraził ogólne zadowolenie z tego, co zobaczył. Poinformowałem klienta o tym, że aplikacja zmierza w dobrą stronę (wtedy jeszcze nie wiedziałem, że pod koniec sprintu będą problemy). Ostatecznie okazało się, że nie zdążyliśmy z pełną założoną, w tym sprincie, funkcjonalnością.

Po uzgodnieniu z Kubą zapytałem też, co trener sądzi o pomyśle z eksportem danych do pliku (w formie tabeli lub jakiegoś podobnego zestawienia). Był zadowolony, nawet bardzo - ze względu na to że prowadzi on stronę internetową klubu, będzie w stanie umieścić informacje z pliku bezpośrednio na stronie, co ułatwi mu też trochę pracę, a w związku z tym, że nie będzie on łączył raczej pracy na telefonie i laptopie, to uznał, że brak synchronizacji jest do zaakceptowania. Kolejne spotkanie zaplanowano na koniec listopada (28-30.11). Dokładny termin uzgodnimy w późniejszym terminie.



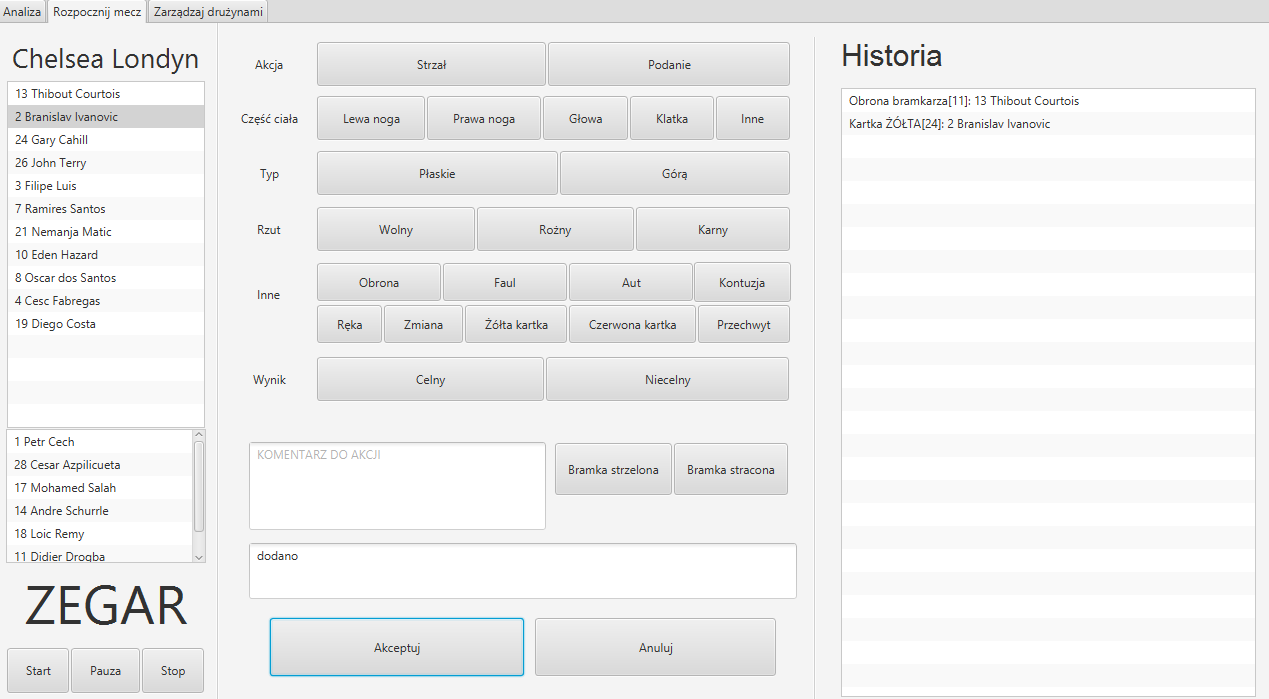
Rysunek . Sprint Backlog - Sprint 3

Rys. 3.10 przedstawia zadania z trzeciego sprintu. Jak widać do wykonania było kilkadziesiąt funkcjonalności, z których zdecydowaną większość udało się wykonać. Po zakończeniu sprintu możliwe było dodawanie większości akcji do meczu. Wykres wypalania(rys. 3.11) pokazuje niewielki postęp w pracach w początkowej części sprintu. Było to spowodowane faktem, iż w tym okresie członkowie zespołu poza pracą nad projektem, musieli także poświęcić czas na naukę na zbliżające się kolokwia.



Rysunek . Wykres wypalania - Sprint 3

Zrzut ekranu (rys. 3.12) przedstawia wygląd interfejsu użytkownika z perspektywy jednego z trzech widoków aplikacji desktopowej – zbierania statystyk ze spotkań. Wygląd pozostałych dwóch zakładek nie uległ zmiany, zostały one jedynie wzbogacone o dodatkowe funkcjonalności.



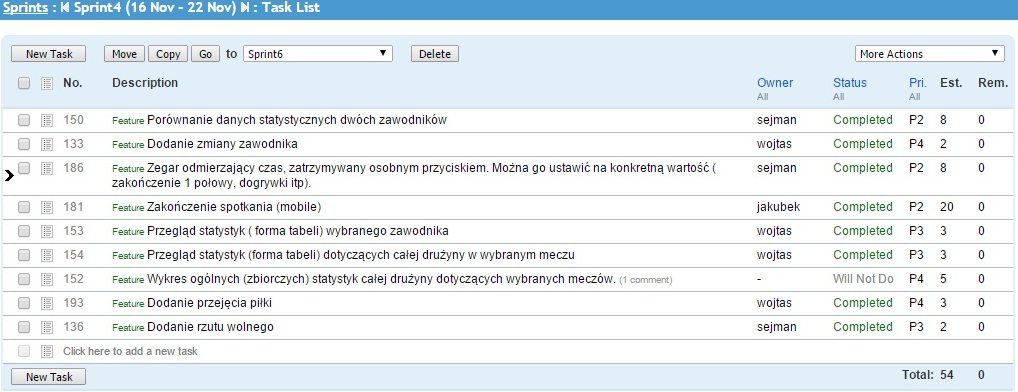
Rysunek . Interfejs dla panelu zbierania statystyk - Sprint 3

## Sprint 4 (16.11.2014-22.11.2014)

Zadania: rozwijanie “analizatora” poprzez dodanie wykresów.

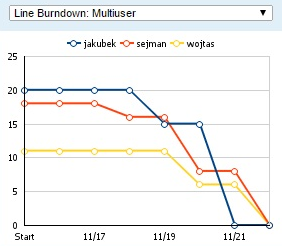
Sprint 4 według początkowych założeń miał być uzupełnieniem dotychczasowej funkcjonalności. Niestety, z powodu problemów w trakcie trzeciego sprintu, założenia te zostały dość mocno zmienione. Uzupełniona została funkcjonalność zbierania statystyk meczu. Dodaliśmy także zegar, pracujący w tle i odmierzający czas meczu. Zaimplementowana została podstawowa wersja analizy statystyk – wyświetlanie ich w tabeli, dla wybranego piłkarza oraz całej drużyny.

Podczas rozmowy zespół był umiarkowanie zadowolony z postępów prac. Udało nam się zrealizować wszystkie wybrane zadania, nie napotykając w tej iteracji większych błędów. Jedyną trudność sprawiła nam implementacja zegara w wersji desktopowej i początkowy problem ze współdzieleniem zasobów, należących do JavaFX. Ustalony został także zakres sprintu piątego.



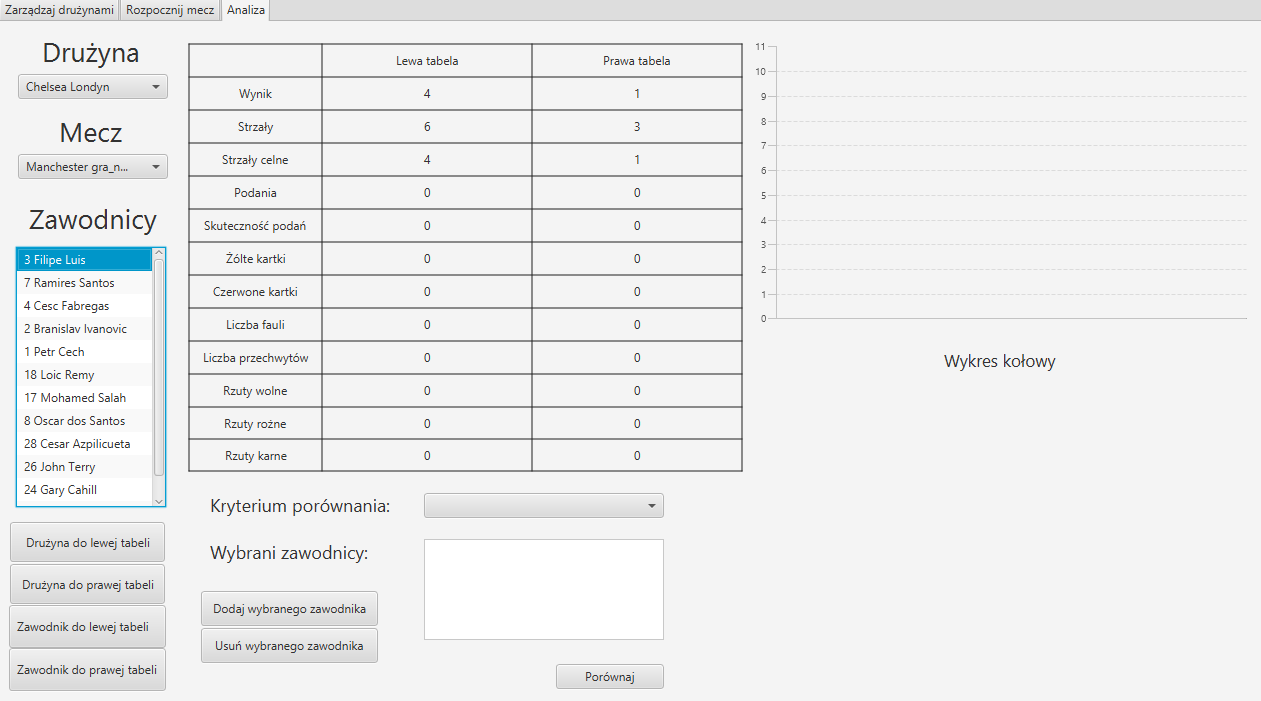
Rysunek . Sprint Backlog - Sprint 4

Rys. 3.13 przedstawia zadania wykonane w czasie czwartej iteracji. Wykonaliśmy wszystkie zaplanowane funkcjonalności. Jedna z nich – wykres zbiorczych statystyk całej drużyny – nie została zaimplementowana. Przyczyną tego była opinia klienta, według którego funkcjonalność ta okazała się zbędna. Dużym nakładem pracy charakteryzowało się zadanie zakończenia meczu w wersji mobilnej. Stało się tak dlatego, że w wersji tej informacje o wszystkich akcjach, które miały miejsce w czasie meczu były przetrzymywane w pamięci, a dopiero po jego zakończeniu zapisywane do bazy danych.

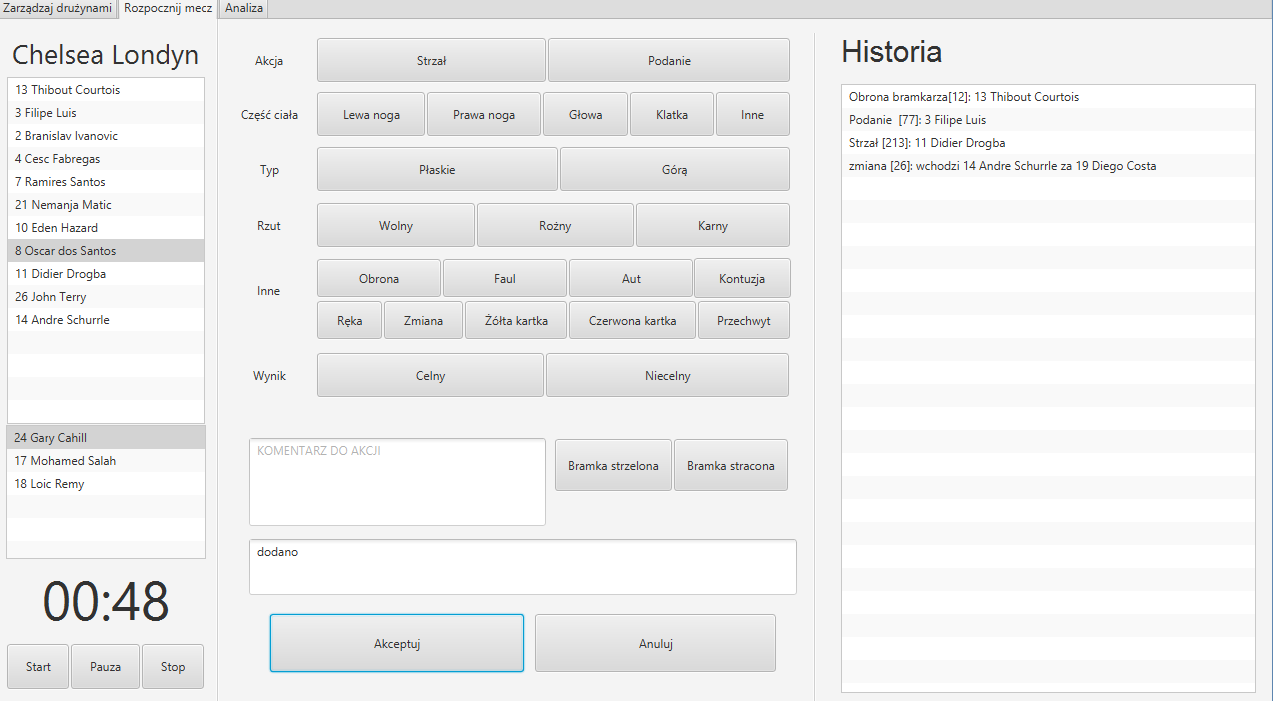


Rysunek . Wykres wypalania - Sprint 4

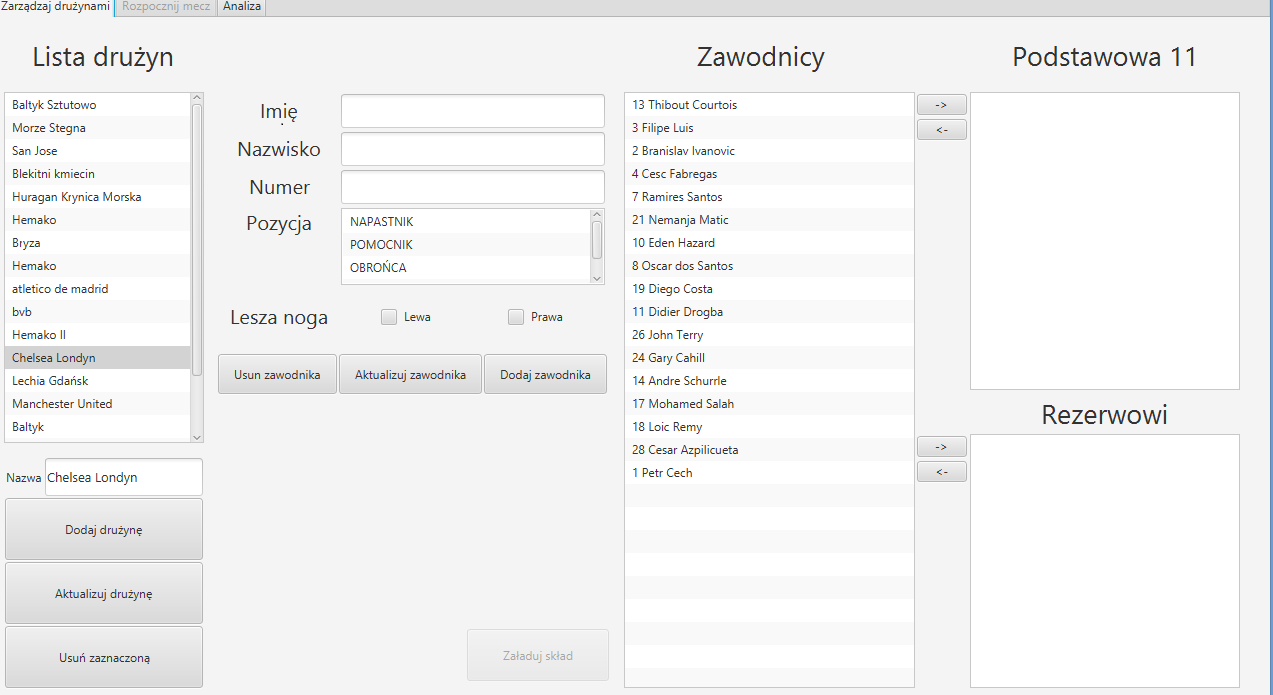
Wygląd poszczególnych zakładek aplikacji desktopowej uległ lekkim zmianom. Spowodowane one były dodaniem funkcjonalności oraz zoptymalizowaniem interfejsu użytkownika. W analizie statystyk (rys. 3.15) zostały dodane przyciski pozwalające załadować statystyki drużyny, jak i zawodnika do tabeli. Z kolei w zbieraniu statystyk (rys. 3.16) został zaimplementowany, działający w tle, zegar. W widoku zarządzania drużynami (rys 3.17) zmienione zostały przyciski, wybierające zawodników do pierwszego składu i jako rezerwowych. Dotychczasowe, duże przyciski z nazwami odpowiednich akcji, zostały zastąpione niewielkimi i bardziej intuicyjnymi przyciskami, umożliwiającymi odpowiednie przenoszenie zawodników.



Rysunek . Interfejs dla panelu analizy - Sprint 4



Rysunek . Interfejs dla panelu zbierania statystyk - Sprint 4



Rysunek . Interfejs dla panelu zarządzania drużynami - Sprint 4

## Sprint 5 (23.11.2014-29.11.2014)

Zadania: sprint przeznaczony na implementacje elementów, których nie udało się zaimplementować wcześniej z powodu problemów z technologią

Sprint 5, tak jak przewidywały założenia początkowe, został przeznaczony na implementację zaległej funkcjonalności. Dla wersji mobilnej została zaimplementowana analiza statystyk. W wersji desktopowej został stworzony podsystem odpowiedzialny za zarządzanie użytkownikami. Zaimplementowane zostały rejestracja i logowanie użytkowników. Pozostałe funkcjonalności systemu były dostępne tylko dla użytkowników zalogowanych. Każdy z nich posiadał własny zestaw analizowanych drużyn – mógł je edytować i usuwać, bez możliwości ingerencji w dane innych.

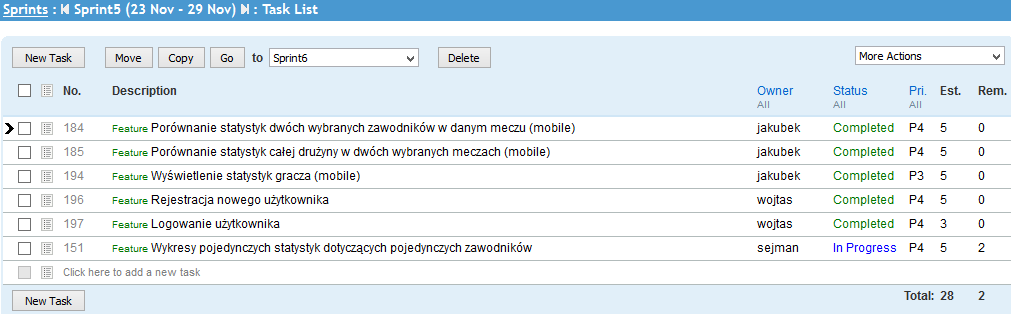
Odbyło się też spotkanie zespołu, na którym Mateusz przedstawił zespołowi opinię klienta i opisał spotkanie. Podczas podsumowania uzgodniliśmy plan działań na kolejny, ostatni sprint. Podzieliliśmy się zadaniami. Uzgodniliśmy, że wersję mobilną będzie testował Wojtek, a desktopową Jakub, który dostał także zadanie - założyć i sformatować dokument, w którym tymczasowo będziemy umieszczać kolejne opisy systemu, a który docelowo stanie się pracą inżynierską. W tym sprincie nie udało się zrealizować w całości jednego zadania – rysowania wykresu, który póki co ograniczał się tylko do wyświetlania dwóch typów akcji – strzału i podania. W czasie implementacji napotkaliśmy na problemy z odświeżaniem wykresu. Szukając rozwiązania, zdecydowaliśmy się przenieść wykres z widoku aplikacji do osobnego okna. Zaletą takiego rozwiązania jest możliwość rysowania w aplikacji kilku wykresów jednocześnie. Ustaliliśmy, że w związku z tym, że w GUI pozostało sporo wolnego miejsca, w następnym sprincie postaramy się rozbudować aplikację o kolejne opcje dotyczące wykresów. Wprowadziliśmy także ostatnie poprawki do bazy danych oraz połączyliśmy gałęzie repozytorium, dotyczące aplikacji desktopowej i mobilnej, by od tego czasu wszystkie zmiany i poprawki pojawiały się w jednym miejscu.

Przedostatniego dnia sprintu odbyło się spotkania z klientem.

28-11-2014

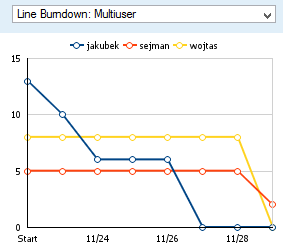
Obecni: Mateusz Szymański, p. Henryk Kuczma

Klientowi został przedstawiony produkt w bardzo zaawansowanej wersji. Jedyne czego nie zawierał to wykresy statystyk. Klient wyraził zadowolenie z postępów. Zapytał się o termin zakończenia prac. Poinformowałem go, że teoretycznie aplikacja jest już niemal gotowa i można byłoby ją używać, jednak chcemy dopracować kilka szczegółów oraz usunąć usterki, o których wiemy, a których z różnych powodów nie rozwiązaliśmy wcześniej. W związku z tym produkt zostanie przekazany klientowi po zakończeniu ostatniego sprintu – około połowy grudnia.



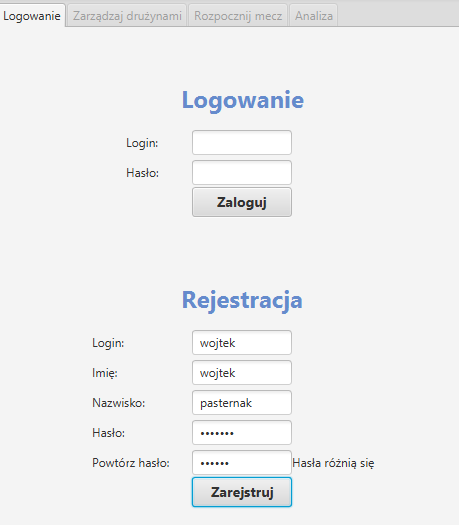
Rysunek . Sprint Backlog - Sprint 5

Rys. 3.18 przedstawia zadania ze sprintu piątego. Prawie wszystkie zadania zostały wykonane w całości. Jedna z funkcjonalności została zrealizowana w sposób niepełny. Jej usprawnienie i działanie w sposób pełny zostało przeniesione do ostatniego sprintu.

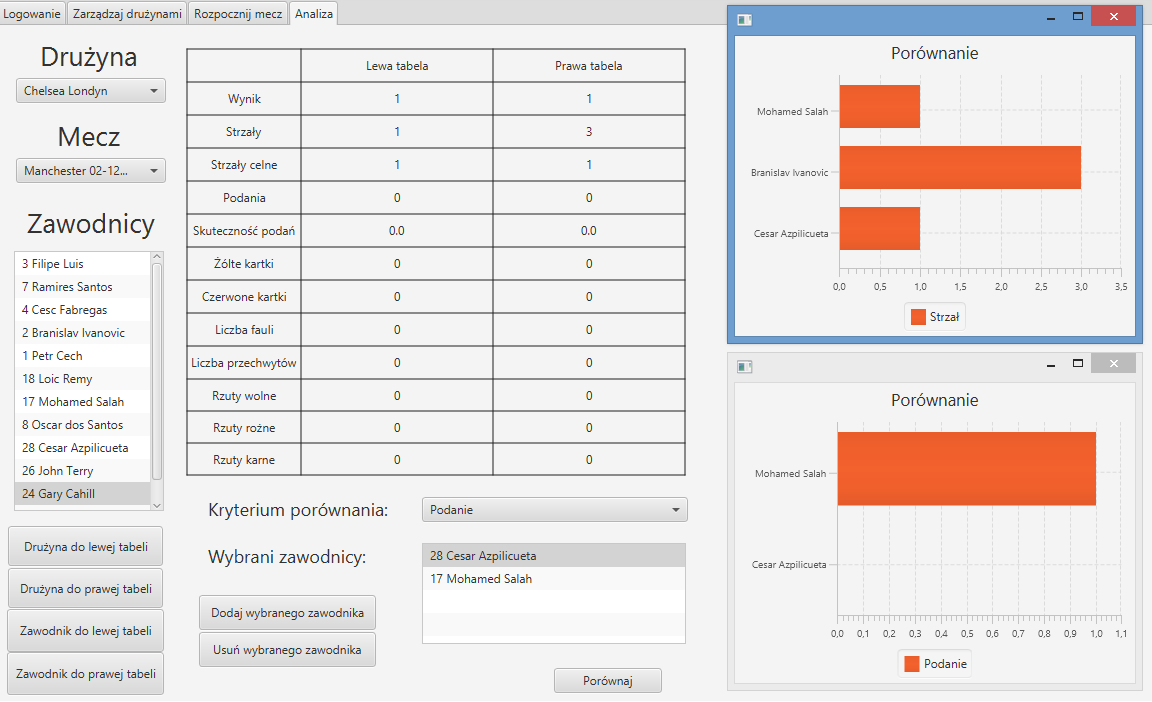


Rysunek . Wykres wypalania - Sprint 5

W czasie sprintu piątego został stworzony podsystem, odpowiadający za logowanie użytkownika (rys. 3.20). Po uruchomieniu aplikacji załadowana była zakładka dotycząca logowania, po dokonaniu którego odblokowywały się pozostałe. Proces rejestracji konta polegał na wypełnieniu pięciu pól tekstowych – loginu, imienia, nazwiska i dwóch dotyczących hasła, a proces logowania na wpisaniu loginu i hasła. Po naciśnięciu na przycisk akcji, wartości wpisane w poszczególne pola, zostały poddawane sprawdzeniu. W przypadku gdy została ona zakończona wykryciem błędnej wartości, na ekranie wyświetlał się odpowiedni komunikat. W zakładce analizy (rys. 3.21) została stworzona pierwsza wersja rysowania wykresów. Możliwe było porównywanie piłkarzy według podstawowych statystyk, takich jak podanie i strzał.



Rysunek . Interfejs dla panelu logowania i rejestracji - Sprint 5



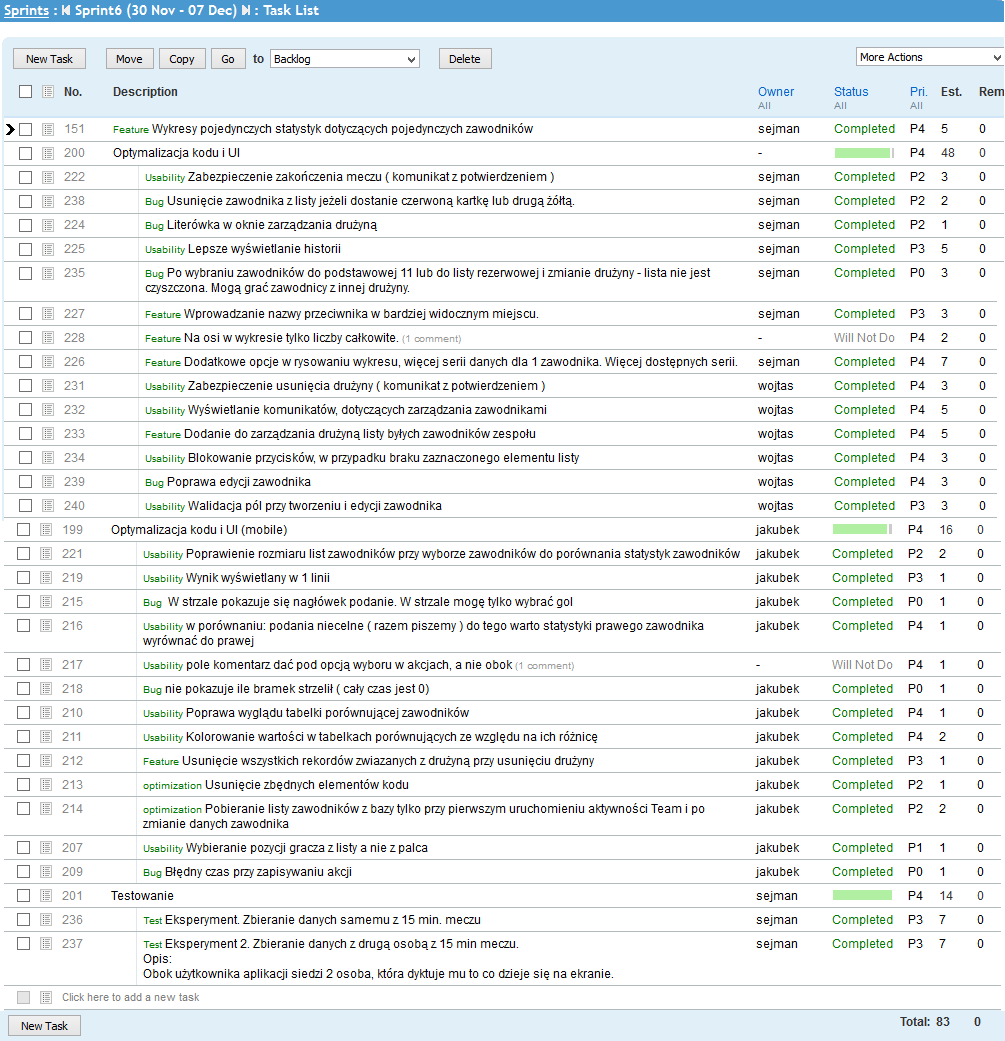
Rysunek . Interfejs dla panelu analizy - Sprint 5

## Sprint 6 (30.11.2014 – 06.12.2014)

Zadania: Upiększanie aplikacji, prace nad wyglądem GUI.

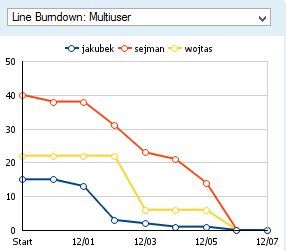
Sprint 6 to przede wszystkim pisanie dokumentu pracy inżynierskiej, a także prace związane z testowaniem aplikacji. Na podstawie tych ostatnich, członkowie zespołu zajmowali się zarówno drobnymi poprawkami i usprawnieniami, jak i usuwaniem błędów. W wersji desktopowej zostały do zadań - usunięcia drużyny i zakończenia spotkania - dodane komunikaty, ostrzegające i wymagające akceptacji w celu potwierdzenia swojej decyzji. Miało to na celu ochronę użytkownika aplikacji od zniszczenia swojej pracy poprzez przypadkowe kliknięcie. Usuwanie zawodnika zostało z kolei rozszerzone do dwóch wariantów. Pierwszy, w przypadku kiedy zawodnik nie miał jeszcze na swoim koncie żadnego występu w meczach danej drużyny, pozwalał na jego całkowite usunięcie z drużyny. Drugi, w celu zapewnienia lepszej integralności danych, zawodnika, który rozegrał już przynajmniej jedno spotkanie, przenosił do listy byłych zawodników danej drużyny. W ten sposób nadal bez problemu można było przeglądać statystyki z meczów z jego udziałem, a dodatkowo w każdej chwili można przywrócić go do drużyny. Dodatkowo funkcjonalność wersji desktopowej została wzbogacona o bardziej zaawansowane rysowanie wykresów. Umożliwiało ono porównywanie wszystkich rodzajów akcji, a także wybór pomiędzy sumą, a średnią liczby akcji danego rodzaju.

Rys. 3.22 przedstawia zadania zrealizowane w ostatniej, szóstej iteracji. Z powodu zbliżającego się terminu oddania projektu, zdecydowano się zrezygnować z prac nad upiększeniem aplikacji. Członkowie zespołu stwierdzili, że wygodny i intuicyjny interfejs użytkownika oraz poprawna funkcjonalność jest najważniejsza. Zadanie dotyczące optymalizacji kodu i UI zostało podzielone na kilkanaście podzadań zarówno dla wersji desktopowej, jak i mobilnej. Dzięki temu członkowie zespołu mogli skupić się nad wykonaniem konkretnych zadań, nie martwiąc się, że jakieś zadanie umknie ich uwadze.

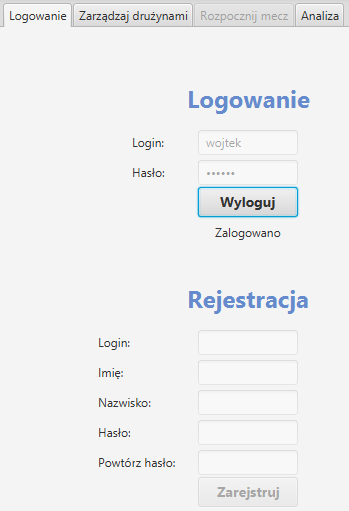


Rysunek . Sprint Backlog - Sprint 6

Na rys. 3.23 przedstawiliśmy wykres wypalania na zakończenie ostatniego Sprintu. Rysunki 3.24, 3.25,3.26 przedstawiają ostateczny wygląd aplikacji.



Rysunek . Wykres wypalania - Sprint 6



Rysunek . Interfejs dla panelu logowania - Sprint 6

### Eksperyment

Po zakończeniu implementacji wersji desktopowej postanowiliśmy przeprowadzić eksperyment, który zweryfikuje działanie aplikacji. Eksperyment ten składa się z 3 części. Podczas pierwszej Mateusz samodzielnie rejestruje dane z fragmentu (15 minut) meczu. Kolejny etap to zbieranie statystyk z pomocą drugiej osoby. Jej zadaniem jest oglądanie meczu i dyktowanie Mateuszowi, co dokładnie dzieje się na boisku. W ten sposób jesteśmy w stanie uniknąć opóźnień i błędów wynikających z konieczności przenoszenia wzroku pomiędzy interfejsem aplikacji oraz drugim monitorem, na którym jest odtwarzany mecz. Trzeci i ostatni etap to spokojne, samodzielne zebranie danych przez Mateusza. Ma on za zadanie zanotować dokładnie każdą akcję z tego samego fragmentu meczu. Jednak w tym przypadku może zatrzymywać czas, aby spokojnie uzupełnić ewentualne akcje. Do eksperymentu wybrano fragment meczu Reprezentacji Polski przeciw Anglii z dnia 17.10.2012 r. W tabeli XXX przedstawiamy wyniki:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Wariant 1 | Wariant 2 | Wariant 3 |
| Podania celne/niecelne | 48/22 | 51/24 | 53/26 |
| Strzały celne/niecelne | 3/0 | 3/0 | 3/0 |
| Faule | 4 | 4 | 4 |
| Przechwyty | 16 | 17 | 18 |
| Żółte kartki | 1 | 1 | 1 |
| Rzuty wolne | 1 | 1 | 1 |
| Obrona | 2 | 2 | 2 |

Jak możemy zauważyć wyniki są różne. Porównując wariant pierwszy i drugi widać, że różnica jest w liczbie zarejestrowanych podań i przechwytów. Dla wariantu pierwszego problemem były przede wszystkim szybsze fragmenty gry. To wtedy właśnie nie udało się zarejestrować wszystkich akcji. Jeżeli spojrzymy na dane z wariantu drugiego i trzeciego, można dojść do wniosku, że zarejestrowaliśmy niemal wszystkie dane. Różnica prawdopodobnie nie wynika ze zbyt dużego tempa gry, ale z różnic w interpretacji przechwytów i następujących po nich podań. Być może w wariancie drugim osoba dyktująca przebieg meczu mogła zinterpretować daną akcję inaczej niż zrobił to tester dla wariantu 3.   
Podsumowując powyższy eksperyment warto zaznaczyć, że aplikacja spełnia swoje zadanie i pozwala zebrać statystyki w sposób szybki i dokładny. Brak zarejestrowania kilku podań w takim okresie gry nie jest dużym błędem. Należy wziąć też pod uwagę, że drużyny klienta grają na niższym poziomie, przez co tempo gry jest dużo wolniejsze. Jednak jeżeli klient ma możliwość wyznaczyć 2 osoby do obsługi aplikacji, naszym zdaniem będzie to bardziej efektywniejsze.

# Raport końcowy

## Zespół projektowy

W skład zespołu weszli: Jakub Hopen, Wojciech Pasternak, Mateusz Szymański.

## Temat projektu

System wspomagający zbieranie statystyk z meczów piłkarskich.

## Kontekst projektu

### Krótka charakterystyka projektu

Aplikacja desktopowa i mobilna usprawniająca pracę sztabu szkoleniowego w małym lokalnym klubie piłkarskim.

### Cele projektu

Projekt ma za zadanie podnieść poziom sportowy w TS Mierzeja Wiślana, poprzez umożliwienie sztabowi szkoleniowemu zebranie statystyk z meczów piłki nożnej drużyn zrzeszonych w tym klubie. Aplikacja ma umożliwiać sprawne zbieranie danych, ich przechowywanie oraz prezentowanie, aby ułatwić analizę. Dzięki temu procesowi sztab szkoleniowy będzie w stanie przekazać zespołowi jak i poszczególnym zawodnikom dokładne wytyczne, wraz z informacjami jakie elementy swojego zachowania i gry powinni poprawić.

### Charakterystyka klienta

Henryk Kuczma – trener, kierownik, prezes TS Mierzeja Wiślana. Jest to człowiek instytucja w tym klubie i nie tylko. Związany ze sportem od dziecka. Początkowo jako piłkarz, a następnie jako trener i prezes. Jest odpowiedzialny za organizację i działanie Towarzystwa. Będzie głównym użytkownikiem aplikacji. Człowiek obyty z technologiami w stopniu podstawowym, krótka demonstracja działania aplikacji wraz z załączonym poradnikiem będzie wystarczająca.

TS Mierzeja Wiślana – klient fikcyjny. Składa się z 3 sekcji piłkarskich: trawiastej, halowej i plażowej. Stworzony projekt ma wspierać drużyny „trawiaste”, jednak w przypadku powodzenia prawdopodobnie będzie chciał również, aby dostosować aplikację do 2 pozostałych rodzajów piłki nożnej.

## Osiągnięte rezultaty

### Produkty

* Aplikacja desktopowa realizująca założone cele wraz ze zdalną bazą danych do przechowywania danych
* Aplikacja mobilna realizująca założone cele wraz z lokalną bazą danych
* Dokumentacja

## Proces realizacji projektu

### Organizacja projektu

Projekt zrealizowany został w trzyosobowym zespole. Podczas pracy wykorzystaliśmy repozytorium git’a w serwisie GitHub. Korzystaliśmy także z serwisu Acunote, które wspierało naszą organizację pracy.

W każdym tygodniu odbywaliśmy spotkania z opiekunem projektu w celu omówienia wykonanych prac, a także zaplanowania tego co wykonamy w następnym tygodniu. Warto zaznaczyć, że 11.11.2014r. spotkanie nie odbyło się ze względu na święto.

Reprezentant zespołu odbywał także spotkania z klientem w celu zademonstrowania postępów prac i uzyskania wskazówek, zaleceń i zastrzeżeń dotyczących wykonanych i dalszych prac.

### Metodologie

Zdecydowaliśmy się na metodykę Scrum, która naszym zdaniem idealnie pasowała do naszego zespołu. Oczywiście na potrzeby tego projektu wprowadziliśmy kilka zmian.

Zmienne długości sprintów to jedna z nich. Ze względu na to, żeby zdyscyplinować się dwa pierwsze sprinty były tygodniowe. Dwa kolejne, które dotyczyły implementacji właściwej funkcjonalności trwały po dwa tygodnie. Ostatnie dwa sprinty to powrót do siedmiodniowego okresu.

Ze względu na rozproszenie zespołu nie byliśmy wstanie wykonywać codziennych spotkań Scruma. Nasze pomysły i plany ogłaszaliśmy na Facebooku. Dodatkowo w celu podsumowania każdego sprintu, a także ustalenia zakresu przyszłego przeprowadzaliśmy rozmowę za pośrednictwem Skype’a. Po każdej takiej konferencji powstawała notatka podsumowująca sprint.

### Wsparcie narzędziowe

* NetBeans IDE 8.0.1
* JavaFX Scene Builder 2.0
* GitHub Shell
* GitHub GUI
* Eclipse JDT 3.8.2
* Eclipse Platform 4.2.1
* Eclipse RCP 4.2.2
* Notepad ++
* phpAdmin na stronie [www.db4free.net](http://www.db4free.net)
* Skype, komunikator Facebooka, TeamViewer

## Dokumentacja

### Techniczna

* Wstępny plan projektu
* Scenariusz użycia
* Diagram ERD bazy danych
* Poradnik użytkownika ( w dwóch egzemplarzach – wersja mobilna i desktopowa)

### Procesowa

* Product Backlog
* Sprint Backlog ( jeden dla każdego sprintu )
* Wykresy wypalania ( jeden dla każdego sprintu )
* Raport końcowy
* Notatki ze spotkań z klientem
* Notatki podsumowujące sprint

## Zmiany w trakcie projektu

### Organizacja projektu i role członków zespołu

Gdy rozpoczynaliśmy pracę założenie było takie, że każdy z nas pracuje na obu platformach, jednak z czasem (po 2 sprincie) podjęliśmy decyzję, że w całości za androida będzie odpowiedzialny tylko Kuba, natomiast Mateusz i Wojtek będą pracować nad wersją desktopową. Dodatkowo Kubie została przekazana opieka nad bazą danych.

### Metodologie i narzędzia

Z początkowo wymienionych narzędzi nie użyliśmy Gimpa. Skorzystaliśmy jednak z Notepad++ w celu sporządzenia notatek. Czasem też przeglądaliśmy w tym programie kod. Zrezygnowaliśmy też z wykorzystywania NetBeans’a do programowania aplikacji mobilnej. Tę część projektu wykonaliśmy w Eclipsie.

### Zakres i harmonogram projektu

Projekt został zrealizowany niemal całkowicie w wersji maksymalnej. Jedynym elementem, którego nie udało się zaimplementować to połączenie obu aplikacji (mobilnej i desktopowej), tak jak zakładaliśmy pierwotnie dla tegoż wariantu.

Harmonogram został zrealizowany zgodnie z założeniami w planie projektu. W kilku sprintach zdarzyły się niezrealizowane zadania, jednak zostały one zaimplementowane w kolejnym.

### Rzeczywiste nakłady pracy w stosunku do zakładanych na początku

Nasze oszacowanie było w dużej mierze prawidłowe i odzwierciedlało włożony przez nas nakład pracy.

## Podział wykonanej pracy między członków grupy projektowej (jutro uzupełnimy)

|  |  |
| --- | --- |
| Członek zespołu | Zakres wykonanej pracy |
| Jakub Hopen | * Zaprojektowanie bazy danych * Implementacja bazy danych * Projektowanie i implementacja interfejsu aplikacji mobilnej * Implementacja aplikacji mobilnej * Tworzenie dokumentacji |
| Wojciech Pasternak | * Implementacja wybranych metod dla aplikacji desktopowej * Tworzenie dokumentacji |
| Mateusz Szymański | * Projektowanie i implementacja interfejsu aplikacji desktopowej * Implementacja wybranych metod aplikacji desktopowej * Kontakt z klientem * Tworzenie dokumentacji |

## Podsumowanie

Naszym celem było stworzenie aplikacji, która będzie wspomagać trenera drużyny piłkarskiej w analizowaniu danych statystycznych, zebranych przez tenże system. Implementowaliśmy system na dwie różne platformy – desktopową, z wykorzystaniem technologii Java oraz mobilną na urządzenia z systemem Android. Każda z aplikacji oferuje zbliżony zestaw funkcjonalności.

Praca przebiegała zgodnie z podstawowymi zasadami metodyki Scrum. Wprowadziliśmy kilka modyfikacji, aby dostosować proces wytwarzania do swoich potrzeb i możliwości. Zastosowaliśmy zmienne długości sprintów, nie definiowaliśmy szczegółów w Sprint Backlogach. Zrezygnowaliśmy z codziennych spotkań, gdyż każdy z nas mieszka w innym miejscu, studiowaliśmy na różnych profilach, więc widywaliśmy się sporadycznie.

Proces wytwarzania przebiegał sprawnie i bez większych problemów. Jeżeli nie udało się nam zrealizować poszczególnych funkcjonalności w danym sprincie, zostały one wykonane w następnym.

## Opinia klienta

Przekazanie systemu odbędzie się 13.12.2014 w związku z tym nie możemy podać opinii końcowej klienta. Jednak z przebiegu spotkań, a także rozmowy telefonicznej klient uważa, że system z pewnością pozwoli jego drużynie podnieść poziom sportowy oraz organizacyjny. Pan Kuczma wyraził też chęć dalszej współpracy po przekazaniu aplikacji i chciałby nam zlecić stworzenie odpowiedników dla piłki halowej i plażowej. Świadczy to o sukcesie naszego projektu.

Jeszcze uboga, ale jutro będę rozmawiał z klientem, żeby uzyskać więcej szczegółów.

# Instrukcja użytkownika aplikacji desktopowej

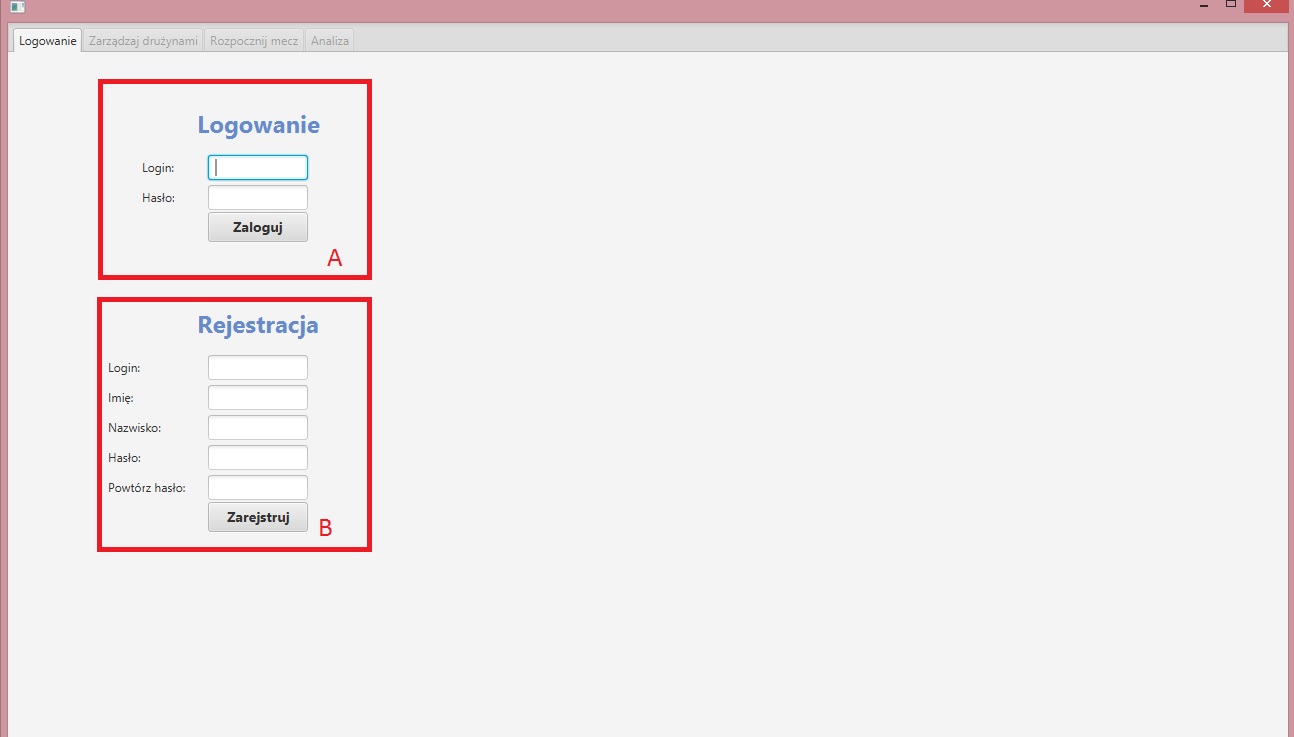
## Wymagania

W tym rozdziale chcielibyśmy omówić dokładnie interfejs użytkownika wraz z uwagami w jaki sposób należy go używać.   
Na początku chciałbym zaznaczyć, że do działania aplikacji wymagane są:

* Java w wersji 8 lub wyższej
* Połączenie z Internetem
* Biblioteka JavaFX w wersji 2.0 lub wyższej (zazwyczaj dołączona do JRE, jednak warto się upewnić)

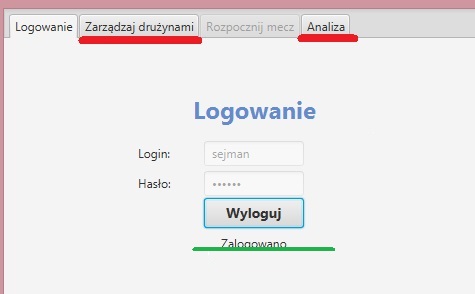
## Pierwsze uruchomienie, logowanie i rejestracja

Po uruchomieniu aplikacji (dwukrotne kliknięcie na plik FootbalStatisticCollector.jar) należy chwilę poczekać, po czym naszym oczom ukazuje się okno aplikacji. Jak widać na rys. 5.1 aplikacja wymusza na nas zalogowanie się (A) , lub jeżeli nie posiadamy jeszcze konta – rejestrację (B).



Rysunek . Interfejs dla panelu logowania i rejestracji – instrukcja

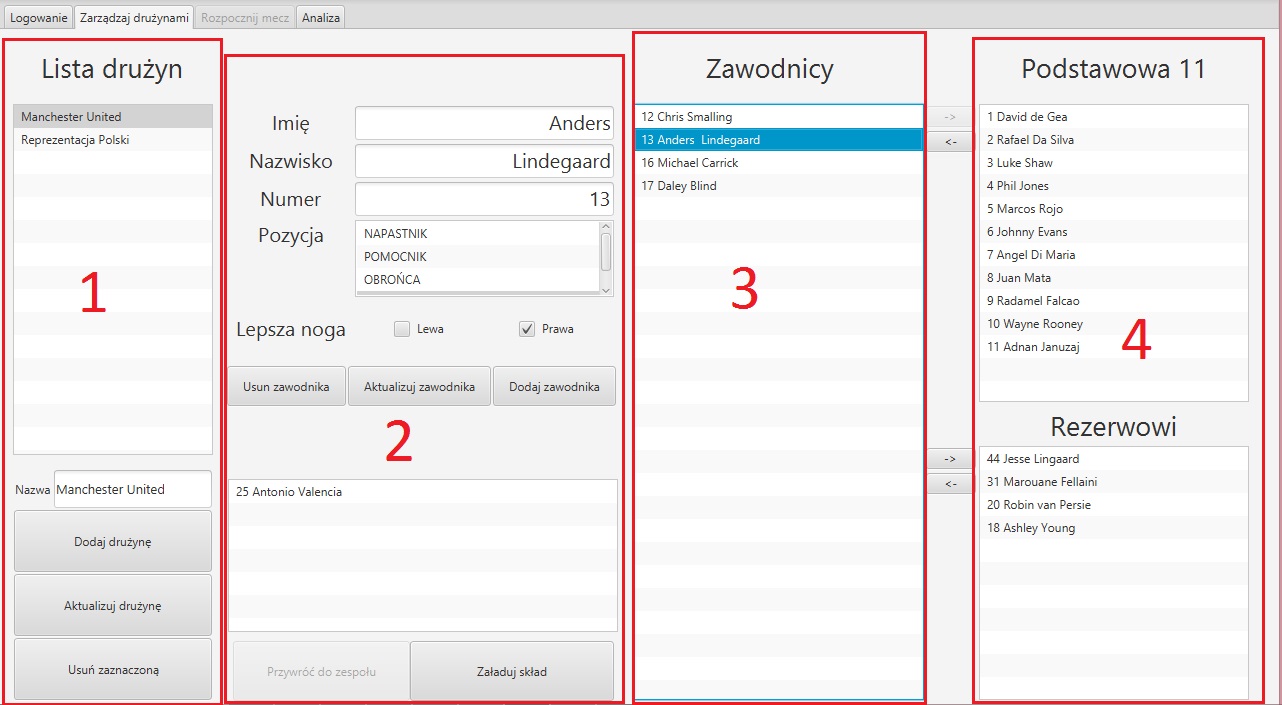
Po udanym logowaniu odblokowują się nam 2 dodatkowe zakładki- „Zarządzaj drużynami” oraz „Analiza”. Zostały one podkreślone kolorem czerwonym, na rys. 5.2. Zaznaczyliśmy też ( kolorem zielonym) komunikat o poprawnym zalogowaniu.



Rysunek . Rezultat poprawnego logowania

## Zarządzanie drużynami

Przejdźmy teraz do zakładki „Zarządzaj drużynami” (rys. 5.3.).

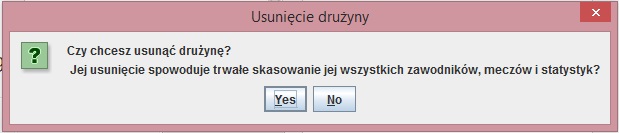


Rysunek . Interfejs dla panelu zarządzania drużynami - instrukcja

Na rys. 5.3. zaznaczyliśmy 4 sekcje, które kolejno omówimy.

### Dodanie, aktualizacja, usunięcie drużyny

Sekcja pierwsza to panel do zarządzania drużynami. Od góry widzimy listę drużyn należących do zalogowanego użytkownika. Po kliknięciu na daną drużynę w polu „Nazwa” umieszczona zostanie jej nazwa. Poprzez jej edycję i kliknięcie na przycisk „Aktualizuj” nazwa zostanie zmieniona. Zmiany zostaną natychmiastowo zaprezentowane w aplikacji. Jeżeli po zmianie nazwy klikniemy przycisk „Dodaj drużynę” – powstanie nowy zespół. Mamy również możliwość usunięcia wybranej drużyny, jednak zanim to nastąpi zostaniemy poproszeni o potwierdzenie (rys.5.4).



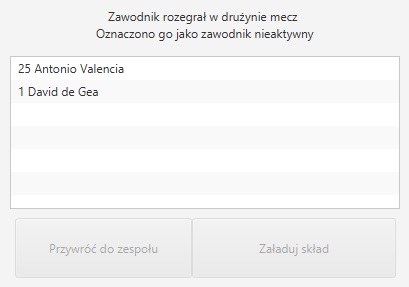
Rysunek . Potwierdzenie usunięcia drużyny – instrukcja

### Dane zawodnika i zawodnicy nieaktywni

W panelu 2 (rys. 5.3) mam wiele elementów. W jego górnej części znajduje się formatka która pozwala na edycję, definiowanie oraz usuwanie zawodników. Wybierając zawodnika z listy wspomniane pola zostaną wypełnione jego danymi. Mamy wówczas możliwość ich edycji. Jeżeli zmienimy jego dane i klikniemy przycisk „Aktualizuj zawodnika” wprowadzone przez nas zmiany zostaną zapisane do bazy. Jeżeli wprowadzimy nowe dane i klikniemy przycisk „Dodaj zawodnika” wówczas nowy gracz zostanie stworzony i przypisany do naszej drużyny.

Inaczej sytuacja wygląda przy usuwaniu zawodnika. Mimo, że zawodnik nie należy już do drużyny to nie chcemy tracić statystyk przypisanych do tego zawodnika ( dotyczą one również całej drużyny ). W związku z tym stworzyliśmy dwa stany w jakich może znajdować się zawodnik – aktywny i nieaktywny. Jeżeli zawodnik, którego chcemy usunąć zagrał przynajmniej jeden mecz zostanie on przeniesiony w stan nieaktywny i nie będzie już wyświetlany na liście zawodników. Użytkownik zostanie też poinformowany o zaistniałej sytuacji ( rys. 5.5). Jeżeli zawodnik nie rozegrał żadnego meczu wówczas zostanie on definitywnie usunięty z bazy. Istnieje możliwość przywrócenia zawodnika do składu. W tym celu należy wybrać zawodnika z listy zaprezentowanej na rys. 5.5 i kliknąć „Przywróć do zespołu”.

W panelu drugim jest jeszcze przycisk „Załaduj skład” służy on, jak wskazuje nazwa, do przypisania wybranych wcześniej zawodników do meczu. Wybór zawodników przedstawimy w kolejnym rozdziale.



Rysunek . Usunięcie zawodnika – instrukcja

### Lista zawodników

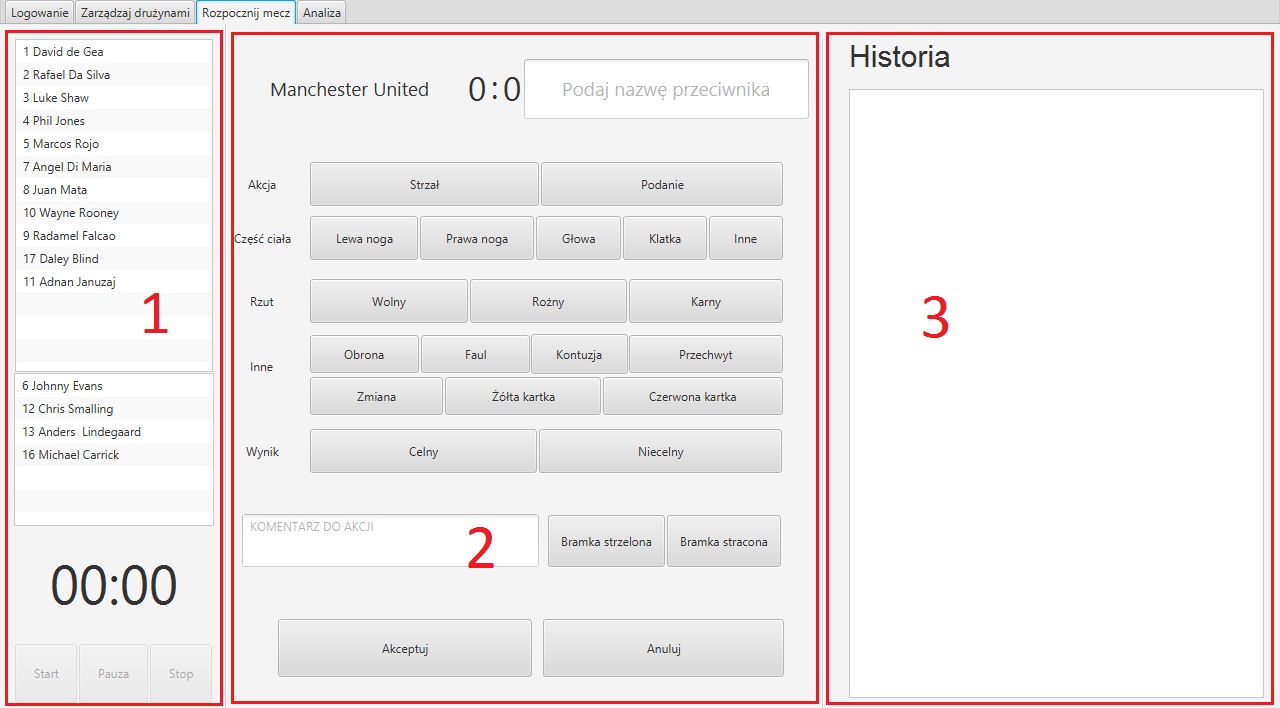
Panel trzeci (rys. 5.3) to lista przypisanych wybranej drużynie zawodników. Klikając na nazwisko zawodnika wybieramy go i przez to jego dane umieszczone zostają w panelu drugim. Tuż obok listy zawodników znajdują się 4 przyciski, które służą do wybierania zawodników do składu podstawowego i rezerwowego. Po wybraniu (kliknięcie) zawodnika z listy, klikamy na jeden z przycisków i zawodnik zostaje przeniesiony do odpowiedniej listy.

### Skład podstawowy i rezerwowy

Panel czwarty (rys. 5.3) to dwie listy odpowiedzialne za wyświetlanie zawodników wybranych do rozegrania meczu. W składzie podstawowym może znaleźć się maksymalnie 11 zawodników, jednak nie mniej niż 7 ( przepisy piłki nożnej). Po spełnieniu tego warunku odblokowany zostanie przycisk „Załaduj skład” z panelu drugiego. Wypełnienie listy rezerwowej jest opcjonalne. Po wybraniu składu możemy przejść do zakładki „Rozpocznij mecz”, która zostanie odblokowana po załadowaniu składu.

## Zbieranie danych statystycznych

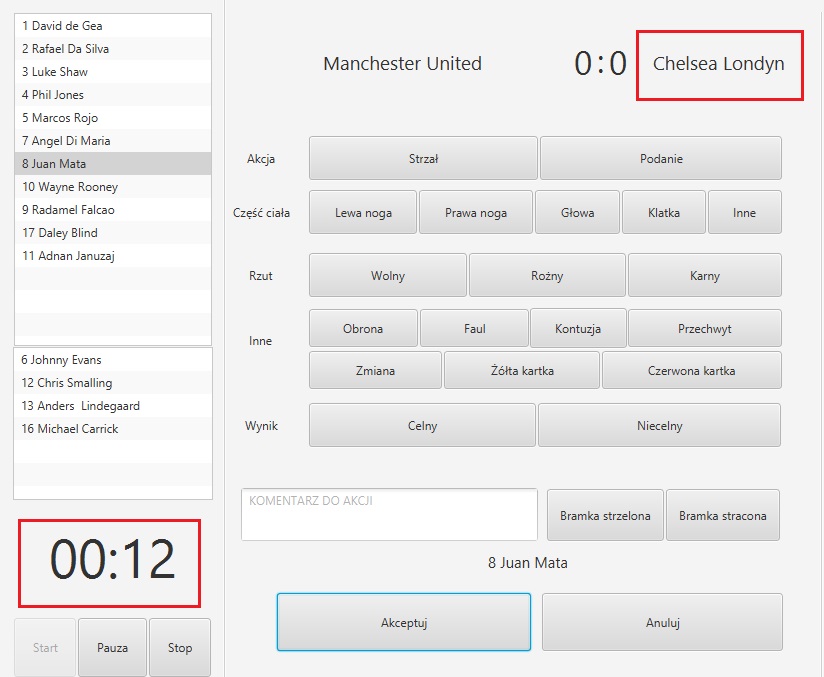
Na rys. 5.6 przedstawiony został interfejs wspomagający zbieranie danych statystycznych w trakcie trwania meczu. W celu dokładnego omówienie, podzieliliśmy go (wzorem poprzedniej zakładki) na trzy panele: listę zawodników wraz z zegarem ( panel 1), przyciski do zbierania statystyk (panel 2) i historię (panel 3). Mecz możemy rozpocząć po wpisaniu nazwy przeciwnika (panel 2) i kliknięciu przycisku start w lewym dolnym rogu okna aplikacji.



Rysunek . Interfejs dla panelu zbierania danych – instrukcja

### Lista zawodników i zegar

W panelu pierwszym (rys. 5.6) znajduje się lista zawodników aktualnie przebywających na boisku, a tuż pod nią – lista rezerwowych. Listy wypełnione są zawodnikami wybranymi w zakładce „Zarządzaj drużynami”. Jeżeli wpisaliśmy już nazwę zawodnika we wskazanym miejscu (rys. 5.7), zostanie odblokowany przycisk pozwalający na rozpoczęcie meczu. Po jego kliknięciu zegar rozpocznie odliczać (rys. 5.7).



Rysunek . Rozpoczęcie meczu – instrukcja

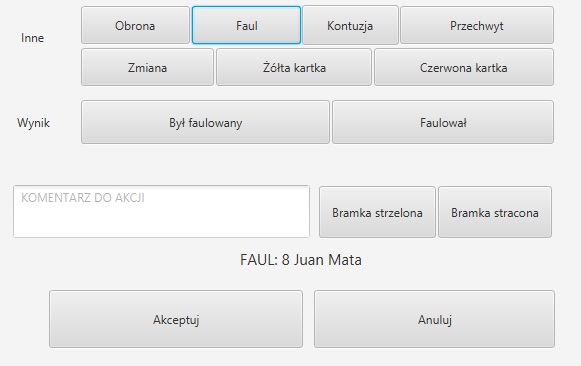
### Przyciski akcji

Na rys. 5.6 wyróżniony został także panel sygnowany numerem 2. Zawiera od wiele przycisków, które są odpowiedzialne za zarejestrowanie zdarzeń z boiska (za pośrednictwem użytkownika). W górnej części tego panelu znajduje się nazwa naszego zespołu, aktualny wynik oraz podana przez nas nazwa przeciwnika.

Kolejną sekcją jest grupa przycisków odpowiedzialnych za rejestrowanie akcji. Opisując tą sekcję chcielibyśmy się skupić bardziej na tym, jakie akcje można ze sobą łączyć, niż opis zawartości, którą doskonale widać na rys. 5.6 i rys. 5.7.

Na początku warto zaznaczyć, że każda akcja wymaga wybrania zawodnika, którego będzie dotyczyła. Do dyspozycji mamy dwie podstawowe akcje: strzał oraz podanie. Dla każdej z nich można przypisać część ciała, którą zostały wykonane. Następnie należy wybrać wynik danej akcji – sukces lub porażka. Jeżeli chcemy zarejestrować wprowadzoną akcję należy kliknąć „Akceptuj”.

Akcje główną możemy zastąpić akcją z sekcji „Inne”. Żółta kartka, czerwona kartka, przechwyt, obrona nie wymagają podania wyniku akcji, ponieważ są one unarne. Inaczej sytuacja wygląda w przypadku faulu. Po wybraniu tej akcji, zawartość przycisków odpowiedzialnych za wynik zmienia się tak jak pokazano na rys. 5.8. W ten sposób możemy rozróżnić, czy to nasz zawodnik faulował, czy przeciwnik przekroczył przepisy. W sekcji „Inne” mamy do dyspozycji także przycisk sygnowany napisem „Zmiana”. Możemy go użyć w połączeniu z wybranymi zawodnikami – jeden z listy podstawowej, jeden z rezerwowej. Po kliknięciu akceptuj zawodnik z listy podstawowej zostaje zmieniony przez rezerwowego.



Rysunek . Faul - instrukcja

Jeżeli drużyna zdobędzie bramkę należy kliknąć przycisk „Bramka strzelona” lub „Bramka stracona”. Po zaakceptowaniu wynik zostanie zaktualizowany.

Pod między przyciskami akcji a przyciskami „Akceptacja” i „Anuluj” wyświetla się bieżąca akcja (rys. 5.9).



Rysunek . Opis akcji – instrukcja

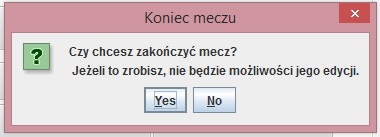
### Historia

Po prawej stronie zakładki przedstawionej na rys. 5.6 znajduje się zapis przebiegu meczu. Są tam umieszczane wszystkie zapisane przez użytkownika akcje. Przykład prezentujemy na rys. 5.10.



Rysunek . Historia – instrukcja

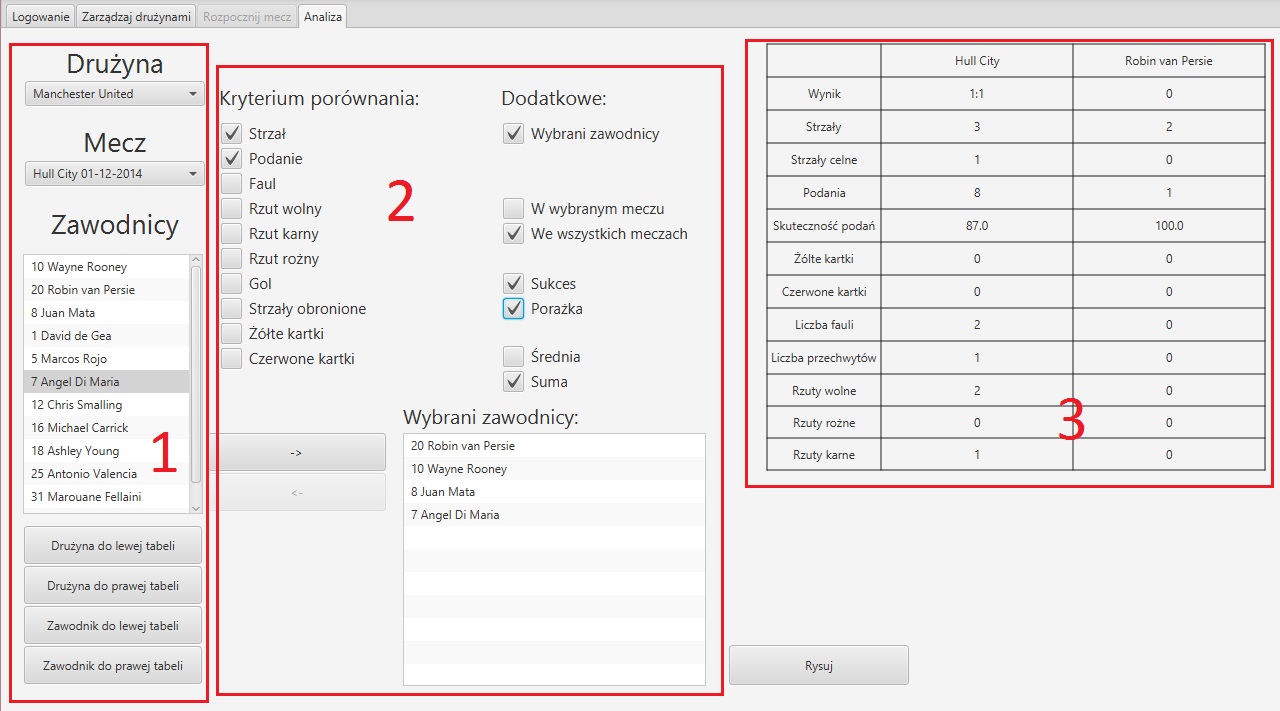
Cały mecz możemy zakończyć wybierając przycisk „Stop” w sekcji 1 pokazanej na rys. 5.6. Po próbie zakończenia meczu aplikacja poprosi nas o potwierdzenie swojej decyzji, tak jak pokazano to na rys. 5.11.



Rysunek . Potwierdzenie zakończenia meczu - instrukcja

## Analiza danych statystycznych

Ostatnią elementem, który chcemy omówić jest zakładka służąca do analizy ( rys. 5.12). Po raz kolejny podzieliliśmy ją na 3 części: listę drużyn, meczy i zawodników (sekcja 1), listę parametrów i danych statystycznych definiujących zawartość wykresu (sekcja 2) oraz tabelę (sekcja 3), którą możemy wypełniać danymi.

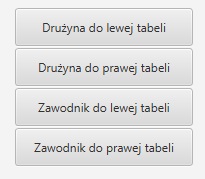


Rysunek . Interfejs analizy statystyk – instrukcja

### Lista drużyn, meczy i zawodników.

Sekcja pierwsza umożliwia nam wybranie drużyny ( na rys. 5.12 jest to Manchester United). W następnej kolejności powinniśmy wybrać mecz jaki nas interesuje. Jeżeli rozwiniemy listę, poprzez kliknięcie na nią, zobaczymy wszystkie rozegrane mecze przez wybraną drużynę. Mecze te są identyfikowane przez nazwę przeciwnika oraz datę rozegrania. Na rys. 5.12 wybraliśmy mecz przeciw Hull City, który rozegrano 01-12-2014 roku.

Poniżej wybranego meczu znajduje się lista zawodników, którzy wystąpili w tym meczu. Obok tej listy są dwa przyciski, na których widnieje strzałka. Przyciski te pozwalają wybrać zawodników, których dane statystyczne chcemy zobaczyć na wykresie. W tym panelu mamy również możliwość wybrania danych do tabeli, poprzez 4 dostępne przyciski zaprezentowane na rys. 5.13.



Rysunek . Przyciski do wypełniania tabeli – instrukcja

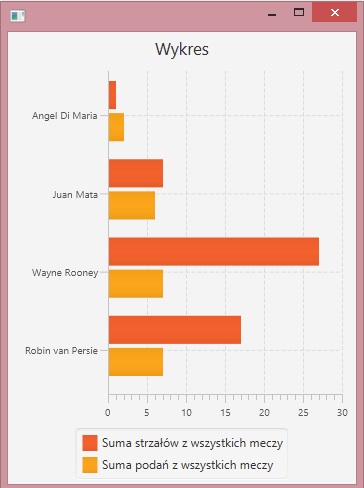
Do działania przycisków powrócimy w rozdziale 5.5.3.

### Lista parametrów definiujących wykres.

Panel sygnowany numerem 2 (rys. 5.12) zawiera zbiór opcji, które możemy wybrać, aby zdefiniować zawartość wykresu. Dostępnych jest 10 kryteriów porównania. Możemy zaznaczyć ich dowolną liczbę. Po prawej stronie tych opcji, są kryteria dodatkowe, lepiej definiujące wykres. Wśród nich domyślnie zaznaczona jest opcja – „Wybrani zawodnicy”, gdyż wykresy mogą dotyczyć tylko graczy.

Kolejną możliwością oferowaną przez aplikację jest wybór liczby meczu. Dokładniej rzecz ujmując można wyświetlać dane dotyczące wybranego meczu lub w odniesieniu do wszystkich meczy wybranej drużyny.

Następnie umieszczone zostały parametry takie jak „Sukces” oraz „Porażka”. Na ich podstawie dobierany jest zakres danych. Jeżeli nie zaznaczymy żadnej z tych opcji, wówczas nie będziemy brali pod uwagę wyniku akcji, którą pokazujemy. Przykładowy wykres znajduje się na rysunku 5.14. Jeżeli zaznaczymy jedną opcję – wtedy brane pod uwagę będą tylko wybrane akcje. W przypadku wyboru obu parametrów wykres będzie prezentował się tak jak na rys 5.15.



Rysunek . Przykładowy wykres – instrukcja

Ostatnią opcją, którą możemy wybrać jest „Suma” i „Średnia”. Tych opcji nie można zaznaczyć jednocześnie.

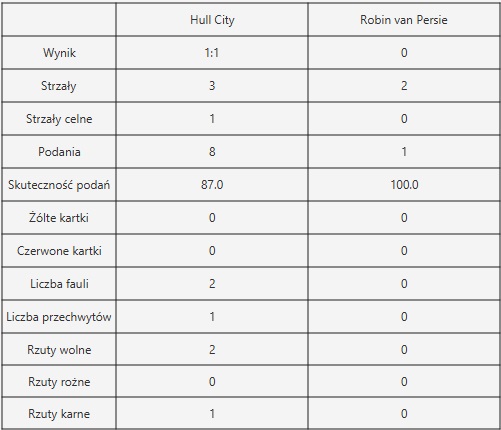
W panelu 2 (rys. 5.12) znajduje się jeszcze lista zawodników, których dane chcemy zobaczyć na wykresie. Wybieramy ich poprzez zaznaczenie zawodnika na liście z panelu 1 (rys . 5.12) i kliknięciu przycisku wskazującego pomiędzy obiema listami.



Rysunek . Przykładowy wykres - instrukcja

### Tabela

Trzeci i ostatni panel (rys. 5.12) zawiera tabelę. Można w niej umieścić statystyki zawodnika lub drużyny. Do tego celu wykorzystujemy przyciski pokazane na rys. 5.13. Dwa z nich pozwalają umieścić w tabeli statystyki naszej drużyny w wybranym meczy, a dwa inne wybranego zawodnika. W każdej parze możemy wybrać pomiędzy lewą i prawą częścią tabeli. W tabeli pokazanej na rys. 5.16 umieściliśmy dane dla meczu przeciw Hull City oraz statystyki Robina van Persiego.



Rysunek . Wypełniona tabela statystyk – instrukcja

Pod tabelką znajduje się przycisk odpowiedzialny za narysowanie wykresu.

# Poradnik użytkownika dla aplikacji mobilnej ( tymczasowo w osobnym pliku )