



**POLITECHNIKA
GDAŃSKA**

WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI I INFORMATYKI



Imię i nazwisko studenta: Marcin Olszewski
Nr albumu: 137357
Studia pierwszego stopnia
Forma studiów: stacjonarne
Kierunek studiów: Informatyka
Specjalność/profil: -

Imię i nazwisko studenta: Mateusz Pakulski
Nr albumu: 137359
Studia pierwszego stopnia
Forma studiów: stacjonarne
Kierunek studiów: Informatyka
Specjalność/profil: -

Imię i nazwisko studenta: PAWEŁ MAZUREK
Nr albumu: 137342
Studia pierwszego stopnia
Forma studiów: stacjonarne
Kierunek studiów: Informatyka
Specjalność/profil: -

Imię i nazwisko studenta:
SEBASTIAN MIAŁKOWSKI
Nr albumu: 137343
Studia pierwszego stopnia
Forma studiów: stacjonarne
Kierunek studiów: Informatyka
Specjalność/profil: -

PRACA DYPLOMOWA INŻYNIERSKA

Tytuł pracy w języku polskim: Asystent treningu biegowego (RunAnd)

Tytuł pracy w języku angielskim: Personal running assistant (RunAnd)

Potwierdzenie przyjęcia pracy	
Opiekun pracy	Kierownik Katedry/Zakładu
<i>podpis</i>	<i>podpis</i>
dr inż. Krzysztof Bruniecki	

Data oddania pracy do dziekanatu:



**POLITECHNIKA
GDAŃSKA**

WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI I INFORMATYKI



OŚWIADCZENIE

Imię i nazwisko: Marcin Olszewski
Data i miejsce urodzenia: 14.06.1991, Grudziądz
Nr albumu: 137357
Wydział: Wydział Elektroniki, Telekomunikacji i Informatyki
Kierunek: informatyka
Poziom studiów: I stopnia - inżynierskie
Forma studiów: stacjonarne

Ja, niżej podpisany(a), wyrażam zgodę/nie wyrażam zgody* na korzystanie z mojego projektu dyplomowego zatytułowanego: Asystent treningu biegowego (RunAnd) do celów naukowych lub dydaktycznych.¹

Gdańsk, dnia

.....
podpis studenta

Świadomy(a) odpowiedzialności karnej z tytułu naruszenia przepisów ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz. U. z 2006 r., nr 90, poz. 631) i konsekwencji dyscyplinarnych określonych w ustawie Prawo o szkolnictwie wyższym (Dz. U. z 2012 r., poz. 572 z późn. zm.),² a także odpowiedzialności cywilno-prawnej oświadczam, że przedkładany projekt dyplomowy został opracowany przeze mnie samodzielnie.

Niniejszy projekt dyplomowy nie był wcześniej podstawą żadnej innej urzędowej procedury związanej z nadaniem tytułu zawodowego.

Wszystkie informacje umieszczone w ww. projekcie dyplomowym, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami zgodnie z art. 34 ustawy o prawie autorskim i prawach pokrewnych.

Potwierdzam zgodność niniejszej wersji projektu dyplomowego z załączoną wersją elektroniczną.

Gdańsk, dnia

.....
podpis studenta

Upoważniam Politechnikę Gdańską do umieszczenia ww. projektu dyplomowego w wersji elektronicznej w otwartym, cyfrowym repozytorium instytucjonalnym Politechniki Gdańskiej oraz poddawania jego procesom weryfikacji i ochrony przed przywłaszczaniem jego autorstwa.

Gdańsk, dnia

.....
podpis studenta

*) niepotrzebne skreślić

¹ Zarządzenie Rektora Politechniki Gdańskiej nr 34/2009 z 9 listopada 2009 r., załącznik nr 8 do instrukcji archiwalnej PG.

² Ustawa z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym:

Art. 214 ustęp 4. W razie podejrzenia popełnienia przez studenta czynu podlegającego na przypisaniu sobie autorstwa istotnego fragmentu lub innych elementów cudzego utworu rektor niezwłocznie poleca przeprowadzenie postępowania wyjaśniającego.

Art. 214 ustęp 6. Jeżeli w wyniku postępowania wyjaśniającego zebrany materiał potwierdza popełnienie czynu, o którym mowa w ust. 4, rektor wstrzymuje postępowanie o nadanie tytułu zawodowego do czasu wydania orzeczenia przez komisję dyscyplinarną oraz składa zawiadomienie o popełnieniu przestępstwa.

Streszczenie

Głównym celem Projektu jest implementacja systemu wspomagającego trening biegowy, który będzie składał się z trzech autonomicznych podsystemów, które poprzez wzajemną komunikację będą współtworzyły narzędzie do wspomagania treningu biegowego – RunAnd. Aplikacja dla zawodnika, główny moduł systemu, zaprojektowana zostanie z myślą o użytkownikach systemu Android. Będzie umożliwiała trenującemu wybór trasy do treningu spośród udostępnionych, nawigację podczas biegu z wykorzystaniem map z serwisu Google Maps, wykonywanie i zapisywanie zdjęć, czy przeglądanie archiwum ukończonych treningów. Zawodnik będzie na bieżąco widział, m.in. jaki przebył dystans, jak długo biegnie i ile spalił dotychczas kalorii.

Śledzenie postępów zawodników odbywać się będzie poprzez aplikację internetową, opartą o *Framework AngularJS*. Trener, oprócz analizy danych (np. aktualnego położenia zawodnika, jego prędkości oraz dystansu, jaki pokonał), będzie mógł w dowolnym czasie wysłać do niego wiadomość, odczytaną po stronie aplikacji mobilnej dzięki rozwiązaniu *TextToSpeech*.

Komunikację pomiędzy wspomnianymi modułami zapewni aplikacja serwerowa zaimplementowana z wykorzystaniem silnika *NodeJS*, wykorzystująca bazę danych *PostgreSQL* oraz udostępniająca usługi sieciowe typu *REST*. Serwer będzie umożliwiał wymianę wiadomości, archiwizację treningów i tras, będzie także przechowywał materiały multimedialne oraz aktualną prognozę pogody.

Ważnym elementem naszego Projektu będą mapy, na których między innymi będzie odbywało się śledzenie postępów w treningu, nawigacja zawodnika oraz projektowanie nowych tras. Skorzystamy z *Google Maps API*, które daje nam szerokie możliwości wykorzystania map. Dzięki temu rozwiązaniu będziemy mieli dostęp do aktualizowanej bazy map z całego świata.

Dodatkowym efektem naszego projektu będzie załącznik w formie instrukcji laboratoryjnej, który pokaże jak szerokie możliwości daje nam wykorzystanie systemów mobilnych. Na przykładzie systemu Android pokażemy, jak wykorzystać programowanie natywne.

Chcielibyśmy, aby system RunAnd przyczynił się do wzrostu (i tak już dużego) zainteresowania biegami, a co za tym idzie motywował jak największą liczbę użytkowników do uprawiania aktywności fizycznej.

Słowa kluczowe:

nauki techniczne i inżynieryjne, informatyka, Internet, usługi sieciowe, systemy mobilne, geolokalizacja, mapy cyfrowe

Abstract

*Streszczenie w języku angielskim
wraz ze słowami kluczowymi*

Spis treści

1. Wstęp i cel pracy	6
1.1. Cel projektu.....	6
1.2. Motywacje	7
1.3. Organizacja pracy i narzędzia wspomagające	7
2. Przegląd zagadnień z dziedziny wspomagania treningu biegowego	10
2.1. Przegląd istniejących rozwiązań.....	10
2.2. Opis wykorzystanych algorytmów	15
3. Analiza wymagań i projekt funkcjonalny.....	15
3.1. Aplikacja mobilna	15
3.2. Serwer.....	16
3.3. Aplikacja trenera	24
4. Architektura systemu oraz przegląd technologii.....	25
4.1. Ogólna architektura całego systemu.	25
4.2. Aplikacja mobilna	25
4.3. Serwer.....	25
4.4. Aplikacja trenera	25
5. Implementacja projektu	30
5.1. Aplikacji mobilnej	31
5.2. Implementacja serwera.....	32
5.3. Implementacja aplikacji WWW	32
6. Rezultaty projektu.....	36
7. Podsumowanie	36
7.1. Wnioski	36
7.2. Perspektywy.....	36
7.3. Wdrożenie systemu an serwerze katedralnym	36
8. Bibliografia.....	37
9. Załączniki.....	38
9.1. Wykaz obrazów	38
9.2. Wykaz listingów	38
9.3. Autorzy rozdziałów	Błąd! Nie zdefiniowano zakładki.
9.4. Instrukcja instalacji i konfiguracji systemu	38
9.5. Instrukcja laboratoryjna.....	39

1. Wstęp i cel pracy (Marcin Olszewski)

Ważnym elementem życia jest aktywność fizyczna, uprawiana pod wieloma postaciami, na każdym etapie życia. Na temat pozytywnego wpływu aktywności ruchowej napisano już wiele publikacji oraz wykonano szereg badań dotyczących jej wpływu na psychikę człowieka, prawidłowy rozwój fizyczny, badano także wpływ rekreacji na zapadalność na choroby, m.in. miażdżycę, czy infekcje górnych dróg oddechowych. Odnosząc się do jednej z książek na temat aktywności fizycznej, pod tytułem *Sport dla wszystkich*, wydanej przez prof. AWF w Krakowie, dra hab. Ryszarda Winiarskiego, uprawianie aktywności fizycznej pozwala:

- neutralizować stres,
- spowolnić tętno,
- poprawić koordynację nerwowo-mięśniową,
- zwiększyć pojemność życiową płuc,
- zwiększyć nawet 20% objętość krwi, co jednocześnie wpływa na wzrost wydolności naszego organizmu.

Czasem jednak coraz większe tempo życia często powoduje, że rezygnujemy z aktywności fizycznej na rzecz odpoczynku(relaksu biernego). A kiedy mielibyśmy znaleźć czas na planowanie treningów? Wyobraźmy sobie sytuację w której mamy do dyspozycji już gotową bazę treningów przygotowanych przez zawodowych trenerów, na różnych poziomach zaawansowania. Wystarczy zainstalować aplikację i rozpocząć trening. Aplikacja pomoże nam znaleźć właściwą drogę, podpowie jaka może nas spotkać pogoda na trasie, gdzie trenują nasi znajomi, a na koniec pozwoli wysłać innym ciekawe zdjęcie i podzielić się z innymi naszymi podbojami. Nie jest to jednak bujanie w obłokach, tylko rzeczywistość, ponieważ na rynku istnieją już wspomniane rozwiązania. Sytuacja wygląda nieco inaczej, gdybyśmy chcieli trenować nie pod kontrolą aplikacji, ale z udziałem trenera, który na żywo śledzi nasze poczynania i może w każdej chwili przesłać nam wiadomość. Obecność drugiego człowieka, który kontroluje i przypatruje się naszemu treningowi niezwykle pomaga zrealizować cel i motywuje. Aplikację możemy oszukać, a z trenerem spotkamy się twarzą w twarz po zakończonym treningu. Być może stworzymy idealne rozwiązanie dla trenerów biegów przełajowych, rozwiązanie do zdalnego treningu na zróżnicowanym terenie poza miejscem zamieszkania, a może tylko usprawnimy istniejące systemy dla biegaczy. Warto jednak podjąć próbę.

1.1. Cel projektu

Celem projektu jest utworzenie aplikacji mobilnej(Android OS) służącej do wspomagania treningu biegowego za pomocą metod zautomatyzowanych oraz przy udziale trenera monitorującego zdalnie postępy zawodnika.

Monitoring treningu będzie odbywał się za pomocą strony internetowej, poprzez którą, trener otrzyma możliwość przysyłania komunikatów do zawodnika, odczytywanych przez TTS(ang. *Text-to-Speech*). Za pomocą aplikacji Web, trener będzie mógł także zaplanować trening dla swojego zawodnika oraz przeglądać archiwum treningów.

1.2. Motywacje

Aktywność fizyczna to niezbędny element zdrowego stylu życia. Od dawna wiadomo także, że bieganie to najprostsza forma aktywności fizycznej, która, jak wspomniano we wstępie, ma korzystny wpływ na prawidłowe funkcjonowanie naszego organizmu. Wpływa również korzystnie na nasze zdolności umysłowe oraz na samopoczucie. Podejmując się realizacji projektu RunAnd, zakładamy, że przyszli użytkownicy systemu otrzymają proste w obsłudze i intuicyjne narzędzie, które pomoże im zaplanować treningi biegowe. Wprowadzając funkcje publikacji tras biegowych, umożliwimy użytkownikom wymianę doświadczeń i osiągnięć. Z drugiej strony chcemy zmierzyć się z systemem od strony technologicznej, używając popularnych technologii mobilnych oraz webowych, odkrywając ich wady i zalety.

1.3. Organizacja pracy i narzędzia wspomagające

Odwołując się po raz kolejny do celu naszego projektu, którym niewątpliwie jest wytworzenie systemu wspomagania treningu biegowego, należy wspomnieć także o środkach, które pomogą nam w jego realizacji. Naszym zadaniem nie jest jedynie przygotowanie architektury, analiza wymagań oraz późniejsza implementacja. Ważna jest także droga do osiągnięcia celu, czyli przeprowadzenie projektu informatycznego, w naszym wypadku jest to także projekt inżynierski.

Redmine (<http://www.redmine.org/>)

dostępny pod adresem <http://runand.greeters.pl>

Na rynku obecnie istnieje kilka czołowych rozwiązań, które oferują nam wsparcie w zarządzaniu projektem(nie tylko, lecz głównie - informatycznym). Należą do nich:

- JIRA – zamknięte oprogramowanie australijskiej firmy Atlassian służące do zarządzania projektem oraz śledzenia błędów(tzw. *issue tracker*). Korzystają z niej m.in. programiści Skype. Umożliwia integrację z innymi produktami tej firmy skierowanymi do programistów, np. Confluence.
- Trac – projekt open-source napisany w języku Python. Oferuje podobny zakres funkcji, jak JIRA, jednak nastawiony jest na prostotę zarządzania projektem.
- Redmine – zaimplementowany w języku Ruby.

Można zauważyć, że przyszłość należy do narzędzi internetowych. Ponadto wszystkie narzędzia oferują podobną rozpiętość funkcji:

- system zgłaszania i wyszukiwania zadań(ang. *ticket*)
- wsparcie dla priorytetów, statusów oraz przypisania zadań do użytkowników
- harmonogramowanie zadań
- wsparcie dla metodyk zwinnych, np. SCRUM
- integracja z repozytoriami SVN, Git
- zarządzanie użytkownikami oraz grupami użytkowników
- fora, dokumenty
- powiadomienia w obrębie systemu jak i drogą email
- wykreślenie wykresów Gantta, wypalania(ang. *Burn down chart*)
- definiowanie przepływów pracy dla zadań

Czasem jednak trzeba doinstalować pluginy, aby móc korzystać z funkcji. Przykładem może być plugin Scrum do Redmine.

#	Typ zagadnienia	Status	Priorytet	Temat	Przypisany do	Data modyfikacji
79	Zadanie	Rozwiązany	Normalny	Działanie serwisu zbierającego informacje o treningu niezależnie od aplikacji.	Sebastian Miąkowski	2014-11-09 00:52
78	Funkcja	Nowy	Niski	Robienie i zapisywanie zdjęć podczas treningu	Sebastian Miąkowski	2014-11-10 15:20
77	Funkcja	Nowy	Normalny	Odczyt wiadomości od trenera TTS	Sebastian Miąkowski	2014-11-09 00:47
76	Funkcja	Nowy	Normalny	Publikacja trasy	Sebastian Miąkowski	2014-11-08 20:15
75	Funkcja	Nowy	Normalny	Akceptacja trenera	Sebastian Miąkowski	2014-11-08 20:14
74	Funkcja	Nowy	Normalny	Wybranie trasy do treningu	Sebastian Miąkowski	2014-11-08 20:12
73	Funkcja	Nowy	Normalny	Wyświetlanie poleceń treningu po zalogowaniu	Sebastian Miąkowski	2014-11-08 20:11
72	Funkcja	Nowy	Wysoki	Opcja treningu dla zalogowanego użytkownika z przypisanym trenerem	Sebastian Miąkowski	2014-11-10 15:20
71	Funkcja	Nowy	Wysoki	Opcja treningu dla zalogowanego użytkownika bez trenera	Sebastian Miąkowski	2014-11-10 15:20
61	Zadanie	Rozwiązany	Normalny	Odczyt głosowy wiadomości od trenera,	Sebastian Miąkowski	2014-11-09 00:47
60	Zadanie	Nowy	Normalny	Wyświetlanie wiadomości od trenera.	Sebastian Miąkowski	2014-11-08 22:04
58	Zadanie	Nowy	Wysoki	Upload bieżącego treningu	Mateusz Pakulski	2014-11-10 15:20
57	Zadanie	Nowy	Normalny	Upload pełnych treningów	Mateusz Pakulski	2014-11-02 20:22
56	Zadanie	Rozwiązany	Normalny	Pominiecie logowania	Mateusz Pakulski	2014-11-02 20:20
50	Zadanie	Rozwiązany	Normalny	Zapis treningu do bazy danych.	Sebastian Miąkowski	2014-11-05 12:15
49	Zadanie	Rozwiązany	Normalny	Dodanie pomiaru czasu przebywanej trasy.	Sebastian Miąkowski	2014-11-05 12:15
48	Zadanie	Rozwiązany	Normalny	Pomiar odległości przebytej przez użytkownika trasy.	Sebastian Miąkowski	2014-11-05 12:15
45	Błąd	Rozwiązany	Normalny	Błąd logowania przy rotacji ekranu	Mateusz Pakulski	2014-10-21 21:02
44	Błąd	Rozwiązany	Wysoki	Nie można zalogować się g+ po wylogowaniu	Mateusz Pakulski	2014-10-19 12:00
43	Zadanie	Rozwiązany	Normalny	Wylogowywanie	Mateusz Pakulski	2014-10-19 00:38

Rys 1. Widok listy zagadnień

Zdecydowaliśmy się na system Redmine, ponieważ został już wcześniej wypróbowany przez nas nie tylko w projektach informatycznych. Jest prosty w obsłudze i w bardzo prosty sposób pozwala przygotowywać tematy(ang. *templates*) poprzez nadpisywanie głównego akusza stylów CSS oraz opcjonalne dodanie kodu JavaScript. Dla ułatwienia wykorzystaliśmy kod CSS do przydzielenia kolorów priorytetom zgłoszeń(rys 1).

Kolejną wprowadzoną przez nas zmianą było wprowadzenie ustalenie nowych typów zagadnień(ang. *issue*) i zdefiniowanie dla nich przepływów pracy(rys 2). Ponadto, system redmine pozwala w skonfigurować na wiele sposobów opcje projektu. Począwszy od wyboru modułów dostępnych w projekcie(m.in. pliki, wiki, fora, kalendarz), szerokiej gamy uprawnień dla poszczególnych użytkowników oraz całych grup, definiowanie niestandardowych grup użytkowników, dodatkowych pól przy tworzeniu zagadnień, czy tworzenia kwerend przechowujących informacje o sposobie wyświetlania listy zagadnień.

Przebieg pracy

 [Kopia](#)  [Podsumowanie](#)

Przebieg między statusami

Uprawnienia do pól

Zaznacz rolę i typ zagadnienia do edycji przepływu pracy:

Rola: Kierownik Typ zagadnienia: Funkcja Edytuj ☒ Wyświetlaj tylko statusy używane przez ten typ zagadnienia

✓ Obecny status	Uprawnione nowe statusy					
	✓ Nowy	✓ W toku	✓ Rozwiązany	✓ Odpowiedź	✓ Zamknięty	✓ Odrzucony
✓ Nowy	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ W toku	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Rozwiązany	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
✓ Odpowiedź	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Zamknięty	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ Odrzucony	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

▶ Additional transitions allowed when the user is the author
 ▶ Additional transitions allowed when the user is the assignee

Zapisz

Rys 2. Definiowanie przepływu pracy

Wspomniane typy zagadnień to:

- **funkcja** – określa funkcję systemu z perspektywy użytkownika, np. *Wysłanie wiadomości od trenera w trakcie treningu*
- **zadanie** – czynność niezbędna do implementacji funkcji, np. *Integracja z Google Maps API*
- **błąd** – zgłoszenie znalezione przy implementacji błędu, np. *Błąd logowania przy rotacji ekranu*
- **rozdział** – zagadnienie związane z dokumentacją, np. *Spis treści*

GitHub(<https://github.com/>)

dostępny pod adresem <https://github.com/RunandPL>

„Nie trzeba nikogo przekonywać, że współczesny twórca nie podejmuje pracy nad żadnym projektem nie mając jakiejś strategii tworzenia kopii zapasowej swojej pracy.” Zgodnie ze zdaniem Pana Joe Loeliger i Matthew McCullough, którzy napisali książkę pt. *Kontrola wersji z systemem Git*, zdecydowaliśmy się na wykorzystanie repozytorium Git(rozproszonego systemu kontroli wersji) w naszej pracy. Zdecydowaliśmy się skorzystać w tym celu z serwisu GitHub, który oferuje programistom darmowy hosting programów open source. Po utworzeniu konta w serwisie oraz utworzenia publicznych projektów(tworzenie prywatnych repozytoriów jest możliwe, ale płatne), wystarczy znać kilka podstawowych komend do pracy z naszym repozytorium.

```
1. touch README.md
2. git init
3. git add README.md
4. git commit -m "first commit"
5. git remote add origin https://github.com/RunandPL/ExampleRunAnd.git
6. git push -u origin master
7.
8. git remote add origin https://github.com/RunandPL/ExampleRunAnd.git
9. git push -u origin master
10.
11. git pull origin master
12.
13. git config user.name „Marcin Olszewski”
14. git config user.email "marolsze@student.pg.gda.pl"
```

Pierwszy sposób(linie 1-5 listingu) polega na utworzeniu repozytorium i jego struktury(folder `.git`) w katalogu projektu(linia 2) oraz powiązaniu go z naszym projektem w serwisie GitHub i wysłaniu pierwszego zapytania typu push, dzięki któremu dodamy nasze zmiany potwierdzone poleceniem `git commit` do naszego projektu. Drugi sposób zakłada, że posiadamy już katalog z projektem, w którym mamy już nasze repozytorium i chcemy np. wysłać je do pustego projektu w serwisie GitHub. Wtedy nie trzeba wykonywać polecenia `git init`.

Jako, że jedną z zalet Gita jest możliwość współdzielenia repozytorium przez wielu użytkowników, powinniśmy zawsze przed wykonaniem komendy `git push`, pobrać aktualną wersję projektu poleceniem `git pull` (linia 10). Jeśli aktualna wersja mocno różni się od naszej i nie będzie możliwe automatyczne połączenie wersji(ang. *merge*), zostaniemy poproszeni o ręczne usunięcie konfliktów.

Użyteczne mogą okazać się komendy z linii 12 i 13 służące do konfiguracji naszej tożsamości w ramach repozytorium.

Aby móc wykonywać powyższe komendy w środowisku Windows, musimy zainstalować np. program `msysGit` lub `Cygwin`. Użytkownicy systemów z rodziny Unix, mają uproszczone zadanie. W dystrybucji systemu Linux Ubuntu, wystarczy w wierszu poleceń wpisać `$ apt-get install git`.

2. Przegląd zagadnień z dziedziny wspomagania treningu biegowego(Sebastian Miałkowski)

2.1.Przegląd istniejących rozwiązań

Endomondo

Jest obecnie jedną z najpopularniejszych lub nawet najpopularniejszą aplikacją wspomagającą treningi. Jest to bardzo rozbudowane narzędzie, umożliwiające rejestrowanie treningów w 58 różnych dyscyplinach. Podczas każdego z nich mamy możliwość rejestrowania podstawowych statystyk

każdego z nich. Endomondo wspiera również funkcję TTS (TextToSpeech). Dzięki temu aplikacja może w trakcie treningu informować nas o jego statystykach. Na popularność tej aplikacji nie wątpliwie wpływa jej uniwersalność, co może również być jej największym minusem. Według mnie autorzy tej aplikacji niepotrzebnie dodawali możliwość rejestracji kilku rodzajów aktywności. Jako przykład mogą posłużyć sztuki walki. Nie trzeba nikogo przekonywać że są one sportem mocno kontaktowym oraz dynamicznym. Z tych powodów nie ma możliwości by podczas ich uprawiania mieć przy sobie telefon. Należy również pamiętać że każdy taki trening może wyglądać inaczej, przez co wskazania aplikacji np. odnośnie spalonych kalorii będą z pewnością mocno przekłamane. Pomimo kilku minusów jest to aplikacja bardzo udana, posiadająca dużo ciekawych i przydatnych funkcji. Pierwszą z nich, obecną w większości dostępnych na rynku aplikacji, jest możliwość podążania wcześniej wytyczoną trasą. Każdy użytkownik aplikacji, ma możliwość poprzez stronę internetową stworzenia trasy, którą może następnie udostępnić. Tak udostępnione trasy stają się dostępne dla wszystkich użytkowników, którzy mogą na nich następnie trenować. Daje to możliwość rywalizacji oraz dzielenia się ciekawymi miejscami do uprawiania sportu. Problemem jest w tym wypadku brak sprawdzania położenia użytkownika względem trasy. W czasie biegu, kiedy nie patrzymy się na ekran telefonu aplikacja nie powie nam kiedy mam skręcić lub czy dobiegliśmy do końca. Mówiąc o rywalizacji, należy wspomnieć o kolejnej ciekawej funkcji. Nazywa się ona po prostu Rywalizacje. Co miesiąc w aplikacji pojawiają się konkurencje polegające osiągnięciu ustalonego celu np. przebiegnięciu największej liczby kilometrów, spaleniu największej liczby kalorii. Każda konkurencja trwa kilka dni. Na koniec osoby z najwyższymi wynikami wygrywają nagrody, wartość jak również ich ilość zależy od konkurencji gdyż niektóre z konkurencji są sponsorowane przez zewnętrzne firmy. Powyższe funkcje dostępne są w wersji darmowej aplikacji, istnieje możliwość kupienia jej wersji premium w której uzyskujemy dostęp do dodatkowych możliwości. Wśród nich znajdują się personalne treningi, strefy tętna, krokomierz, informacje o pogodzie w czasie treningu oraz edytowalny trener audio. Niestety nie ma możliwości przetestowania tych funkcji bez wykupowania płatnej wersji, przydałaby się możliwość kilku dniowego testowania. Sam wygląd aplikacji, przedstawiony na [NUMER RYSUNKU], jest w mojej opinii bardzo dobrze zrobiony, menu jest przejrzyste i czytelne a wszystkie opcje dobrze rozplanowane. Również strona internetowa aplikacji jest intuicyjna oraz wygląda przyjemnie. Nawet odwiedzając ją pierwszy raz bez kłopotu znajdziemy szukane opcje.



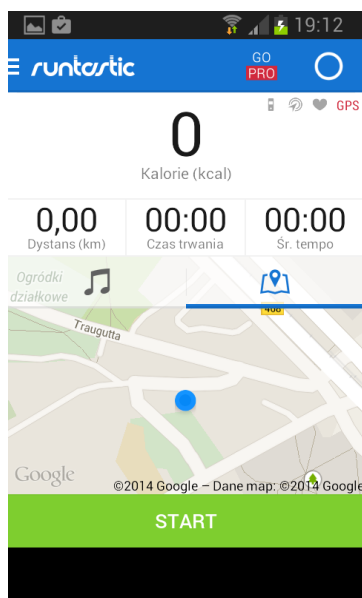
Rys. x Endomondo

Runtastic

Jest kolejną aplikacją treningową dostępną na urządzenia mobilne. Tak samo jak Endomondo, wspiera na wiele typów dyscyplin. Również tutaj mamy możliwość pobrania jej bezpłatnej wersji, jak również zakupienia opcji Pro. W ramach bezpłatnej wersji istnieje możliwość odbycia zwykłego treningu lub z ustalonymi wcześniej przez nas celami, odległością lub czasem. Każda z odbytych przez nas aktywności jest zapisywana, co daje możliwość ich późniejszego podglądu oraz określenia naszych postępów. Runtastic udostępnia również możliwość podglądu statystyk naszych aktywności z ostatnich dwóch miesięcy, w wersji Pro również z ostatniego tygodnia oraz roku. Wykupienie wersji Pro daje oczywiście dostęp do dodatkowych funkcji. Pierwszą z nich są trasy, posiadają one działanie identyczne jak te w Endomondo. Również tutaj powinny być lepiej zaimplementowane. Wybierając trening po trasie nie jest sprawdzane nasze położenie, jak również nie informacji głosowych informujących o kierunku trasy. Kolejną opcją są treningi interwałowe, podczas takiego treningu dostajemy informacje głosowe o kolejnych interwałach. Wersja Pro udostępnia również opcję śledzenia na żywo, dzięki niej nasza pozycja jest na bieżąco aktualizowana i wysyłana na serwer. Za pomocą strony internetowej inni użytkownicy mają potem możliwość podglądania naszej pozycji oraz innych statystyk naszego treningu. Gdy zauważą że nasze tempo spada i będą chcieli nas zmotywować mogą nas o tym powiadomić poprzez wysłanie wiadomości, która zostanie dla nas głosowo odczytana. Jedną z najciekawszych funkcji umilających trening są biegi fabularne, jednak jest to opcja dodatkowo płatna. Podczas biegu fabularnego aplikacja opowiada nam jedną z historii, którą wybraliśmy. Jest ona opowiadana w sposób taki że czujemy się jej częścią co dodatkowo nas motywuje. Dodatkowo wykupić możemy również plany treningowe, ułożone przez prawdziwych

trenerów. Twórcy aplikacji mogliby udostępnić trening przykładowy w celu zapoznania się z działaniem tej funkcji. Runtastic, jak widać na [NUMER RYSUNKU], posiada bardzo dobrze rozplanowany i ładny

wygląd. Dostęp do wszystkich okien aplikacji odbywa się poprzez boczną szufladę akcji. Każdy, nawet początkujący użytkownik tej aplikacji bez problemu się w niej odnajdzie.

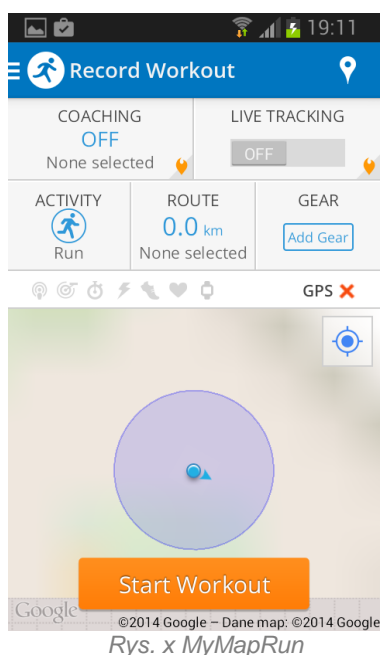


Rys. x Runastic

MapMyRun

Kolejna z aplikacji przeznaczonych dla sportowców, która jest przy tym najbardziej rozbudowana. Do wyboru mamy w niej największą liczbę dyscyplin treningowych, wśród których znajdują się np. wyprowadzanie psa, śpiewanie, skok o linie czy rąbanie drewna. Prócz funkcji związanych z treningiem, aplikacja daje nam dodatkowo możliwość rejestrowania zjedzonych posiłków. To w połączeniu z informacjami o treningu pozwala na kontrolę naszego bilansu kalorycznego i zbilansowanie diety. W informacjach o profilu dostępne są informacje o średnim tygodniowym przebytych dystansie, czasie treningów, ich ilości oraz spalonych kaloriach. Takie same informacje dostępne są również całej historii treningowej. Z funkcji dostępnych w darmowej wersji możemy wybrać tylko dwa rodzaje treningu, zwykły oraz po trasie. Dodatkowo w ich czasie możemy robić zdjęcia. Jest to mocno ograniczony wybór w porównaniu do poprzednich aplikacji gdzie istnieje możliwość ustalenia odległości jaką chcemy pokonać czy też czasu jaki chcemy trenować. Zastępstwem tych opcji może być funkcja udostępniona na stronie internetowej aplikacji. Nosi ona nazwę „Stwórz Cel” i polega na ustaleniu dla siebie zadania do wykonania. Takim zadaniem może być np. przebiegnięcie 100 mil w ciągu 4 tygodni. Tym co może zachęcić do korzystania z tej aplikacji są funkcje niedostępne w innych, lista wydarzeń sportowych oraz informacje o zużyciu obuwia. Pierwsza z nich pozwala nam na łatwe śledzenie informacji o imprezach, które będą miały miejsce w przyszłości w naszej okolicy i przygotowanie się do nich. W treningu biegowym równie ważne jak dobra technika są też dobre buty, które jednak się zużywają. Jednak kontrola kilometrów do czasu wymiany obuwia może być problematyczna, z tego względu funkcja dostępna w MapMyRun jest niezwykle pomocna. W wersji premium, która w wypadku tej aplikacji nosi nazwę MVP dostajemy dostęp do treningów interwałowych, śledzenia na żywo, analizy pracy serca, planów treningowych oraz trenera audio. Należy zwrócić uwagę że liczba funkcji jest ograniczona w porównaniu do innych aplikacji. Na duży plus należy ocenić wygląd, przedstawiony na rysunku [NUMER RYSUNKU], oraz funkcjonalność aplikacji. Jest ona przyjemna dla oka, poszczególne okna są dobrze rozplanowane więc nie ma żadnego problemu z

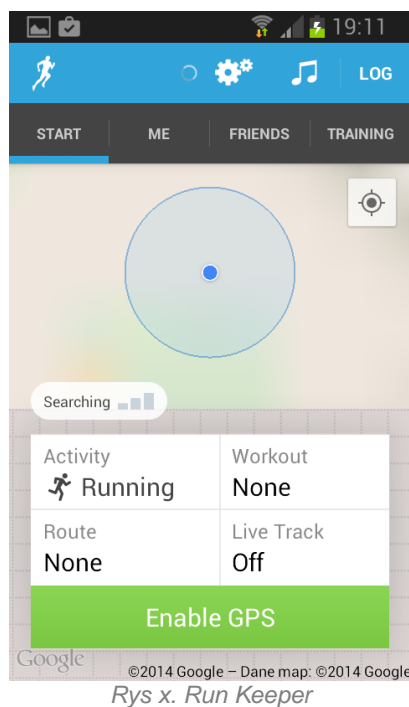
odnalezieniem funkcji jakiej potrzebujemy. Identyczne wrażenie można odnieść korzystając ze strony internetowej, jest ona utrzymana w takiej samej stylistyce i równie przyjemna w użytkowaniu.



RunKeeper

Ostatnią z omawianych przeze mnie aplikacji jest RunKeeper stworzona przez firmę o tej samej nazwie. W porównaniu do wymienianych wcześniej aplikacji charakteryzuje się ona największą liczbą dostępnych opcji w darmowej wersji. Bardzo ciekawie przedstawia się wybór aktywności, które są podzielone na dwie kategorie, GPS oraz stoper. Każda z nich zawiera aktywności wykonywane przy pomocy odpowiedniego narzędzia. Dla przykładu, bieganie zostało przydzielone do obu kategorii a CrossFit do stopera. Wybór przez nas rodzaju treningu ma wpływ na wygląd głównego ekranu aplikacji po jego rozpoczęciu. Podczas treningu z kategorii GPS na głównym ekranie mamy wyświetlone podstawowe informacje np. czas, średnio tempo. Prócz danych znajdują się tam dodatkowo przyciski służące za modyfikację interfejsu np. przełączenie go w tryb nocny, oraz robienie zdjęć. W przypadku aktywności z kategorii stoper na ekranie jest wyświetlana tarcza zegara, ponad nią czas treningu a poniżej przyciski „Stop” oraz „Pauza”. Całość jest bardzo czytelna dzięki czemu czas naszego ćwiczenia można śledzić z odległości kilku metrów. Przeglądając historię ćwiczeń mamy dostęp do wykresów przedstawiających m.in. średni tempo, wysokość na poziomie morza oraz puls. Tej opcji zdecydowanie brakuje w darmowych produktach konkurencji. Dodatkowo można zobaczyć tempo w jakim pokonywaliśmy poszczególne kilometry. Również informacje audio są bardzo rozbudowane, jeżeli tego zachcemy mamy możliwość otrzymywania informacji o prędkości, tempie, odległości czy pulsie. Poprzednie aplikacje miały tą funkcję dużo uboższą. Wersja premium, RunKeeper Elite, daje dostęp do takich samych funkcji jak w przypadku poprzednich aplikacji. Są to plany treningowe, śledzenie na żywo oraz jeszcze bardziej rozbudowane statystyki. Jednak RunKeeper posiada też kilka mniejszych i większych minusów. Tym co może przeszkadzać początkującemu użytkownikowi jest wygląd aplikacji, pokazany na rysunku [NUMER RYSUNKU], oraz rozmieszczenie w niej funkcji.

Sam mając pierwszy raz styczność z tą aplikacją czułem się lekko zagubiony, uczucie to minęło dopiero po kilku minutach obcowania z nią. Duże większym problemem, który ujawnia się dopiero po jakimś czasie jest jej zacinanie się. Wiele razy, podczas normalnego użytku, natrafiałem na chwilowe zawieszenia się lub pojawiania czarnego ekranu. Mimo że nie trwały długo w dużym stopniu psuły przyjemność użytkowania.



2.2. Opis wykorzystanych algorytmów

3. Analiza wymagań i projekt funkcjonalny

3.1. Aplikacja mobilna

ID z systemu redmine	Priorytet	Temat	Uwagi
78	Funkcja	Robienie i zapisywanie zdjęć podczas treningu	
77	Funkcja	Odczyt wiadomości od trenera TTS	
76	Funkcja	Publikacja trasy	
75	Funkcja	Akceptacja trenera	
74	Funkcja	Wybranie trasy do treningu	
73	Funkcja	Wyświetlanie poleceń treningu po zalogowaniu	

72	Funkcja	Opcja treningu dla zalogowanego użytkownika z przypisanym trenerem	
71	Funkcja	Opcja treningu dla zalogowanego użytkownika bez trenera	

3.2. Serwer (Paweł Mazurek)

3.2.1. Terminologia

Termin	Pełna nazwa	Wyjaśnienie
WWW	<i>World Wide Web</i>	Jest to multimedialny system informatyczny, oparty o hipertekst. Jest jedną z usług udostępnianych w ramach Internetu.
HTTP	<i>Hypertext Transfer Protocol</i>	Jest to protokół sieci WWW
API	<i>Application Programming Interface</i>	Jest to ściśle ustalony zestaw reguł w jaki programy komunikują się między sobą.
URL	<i>Uniform Resource Locator</i>	Format adresowania danych wykorzystywany w Internecie i sieciach lokalnych.
POST	-	Metoda z której pomocą przysłane są dane w sieci Internet
GET	-	Sposób w jaki przekazywane są dane pomiędzy dokumentami sieciowymi w sieci HTTP

3.2.2. Cel

Serwer ma na celu udostępnić szereg zapytań – tzw. API, wszystkim zainteresowanym podmiotom, czyli aplikacji mobilnej i webowej. W odpowiedzi na wykonane zapytania(*ang. request*) serwer odpowiada wiadomością(*ang. response*). Zapytania będą wysyłane przez protokół HTTP, więcej informacji na jego temat znajdują się w rozdziale 4.3. Serwer pozwala na komunikację dwóch komponentów opisywanego systemu (aplikacji mobilnej i webowej), zapisywanie danych dostarczanych przez nie, oraz zarządzanie tymi danymi. Zapisuje również dane pogodowe z zewnętrznego serwisu (*forecast.io*) w pamięci podręcznej (proces ten ma swój angielski odpowiednik – *caching*) i udostępnia te dane wewnątrz systemu.

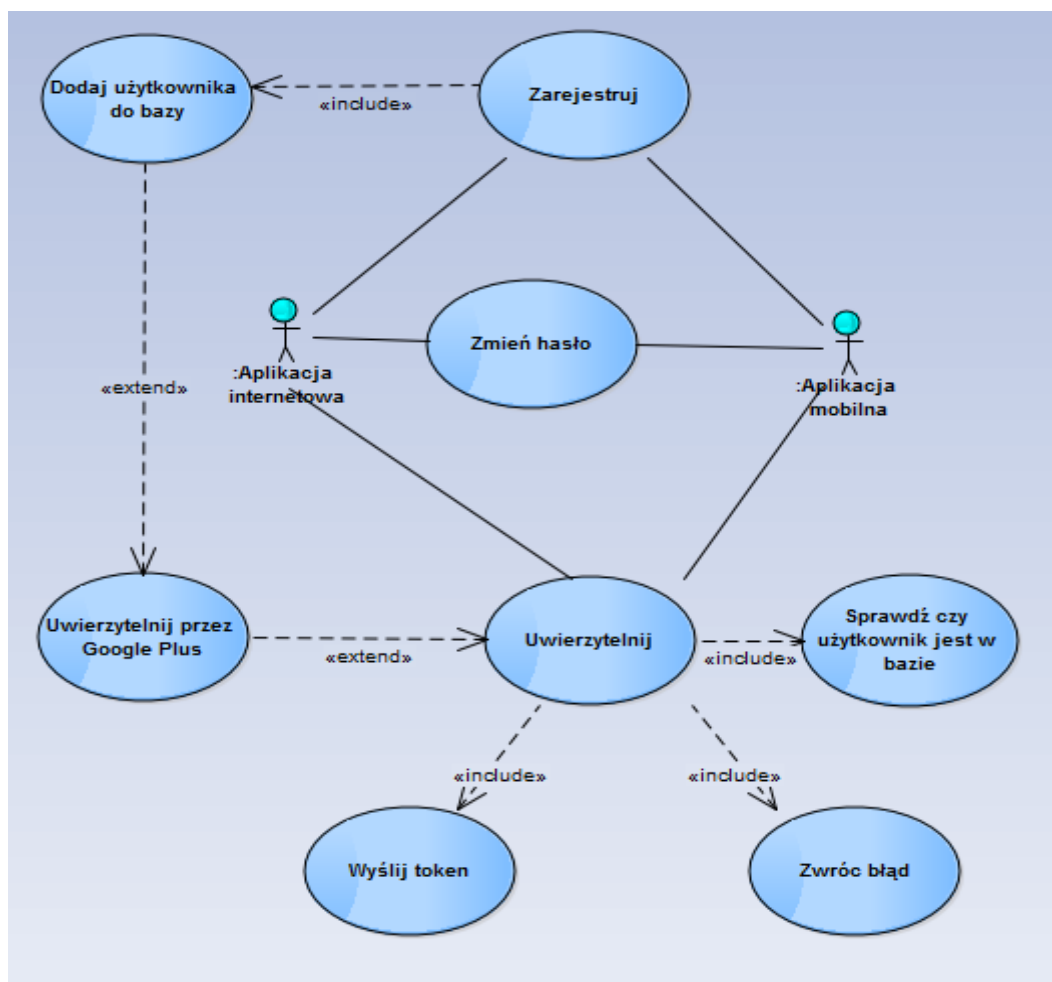
3.2.3. Komponenty

Serwer składa się z kilku współdziałających komponentów, służących do obsługi: wszystkich przychodzących zapytań, użytkowników (w tym ich uwierzytelniania), serwisu przechowującego dane pogodowe z zewnętrznego serwera, treningów 'na żywo', treningów ukończonych, tras, modułu pozwalającego na akceptacje lub odrzucenie zapytań z prośbą o dodanie zawodnika do trenera. Komponenty te reprezentowane są przez osobne pliki ze skryptami obsługującymi daną funkcjonalność.

3.2.4. Wymagania funkcjonalne

ID z systemu redmine	Priorytet	Temat	Uwagi
90	Wysoki	API do podglądu treningu 'na żywo'	Możliwość rozpoczęcia treningu przez biegacza, zakończenie go, i udostępnienie tych danych trenerowi
89	Niski	Zapis tras trenera do swojego własnego konta	Trener ma mieć możliwość stworzenia trasy, oraz zapisania jej (z zachowaniem powiązania trasy z jej twórcą)
88	Normalny	System wysyłania wiadomości podczas treningu od trenera do zawodnika	Podczas gdy trening 'na żywo' jest uruchomiony, należy udostępnić trenerowi możliwość wysyłania wiadomości zawodnikowi
87	Normalny	Udostępnianie tras stworzonych przez trenera biegaczowi	Trasy stworzone przez trenera należy udostępnić biegaczowi
86	Wysoki	Możliwość dodawania zawodnika do trenera	Możliwość dwustronnego wysłania zapytania oraz jego odrzucenie / akceptacja
85	Niski	Zapisywanie zdjęć	Zdjęć trasy
84	Niski	Udostępnianie zdjęć	Zdjęć trasy
127	Pilny	Uwierzytelnianie użytkowników	
128	Niski	Udostępnienie danych pogodowych	Po podaniu współrzędnych, serwer zwraca dla nich pogodę na najbliższe 24 godziny
129	Normalny	Zapisywanie ukończonego treningu	

3.2.5. Przypadki użycia



Rys x. Diagram przypadków użycia odpowiedzialnych za zarządzanie użytkownikami

Uwaga – opisane zostały tylko najważniejsze z nich. Aby praca była czytelniejsza i bardziej przejrzysta, pominąłem opis najbardziej oczywistych.

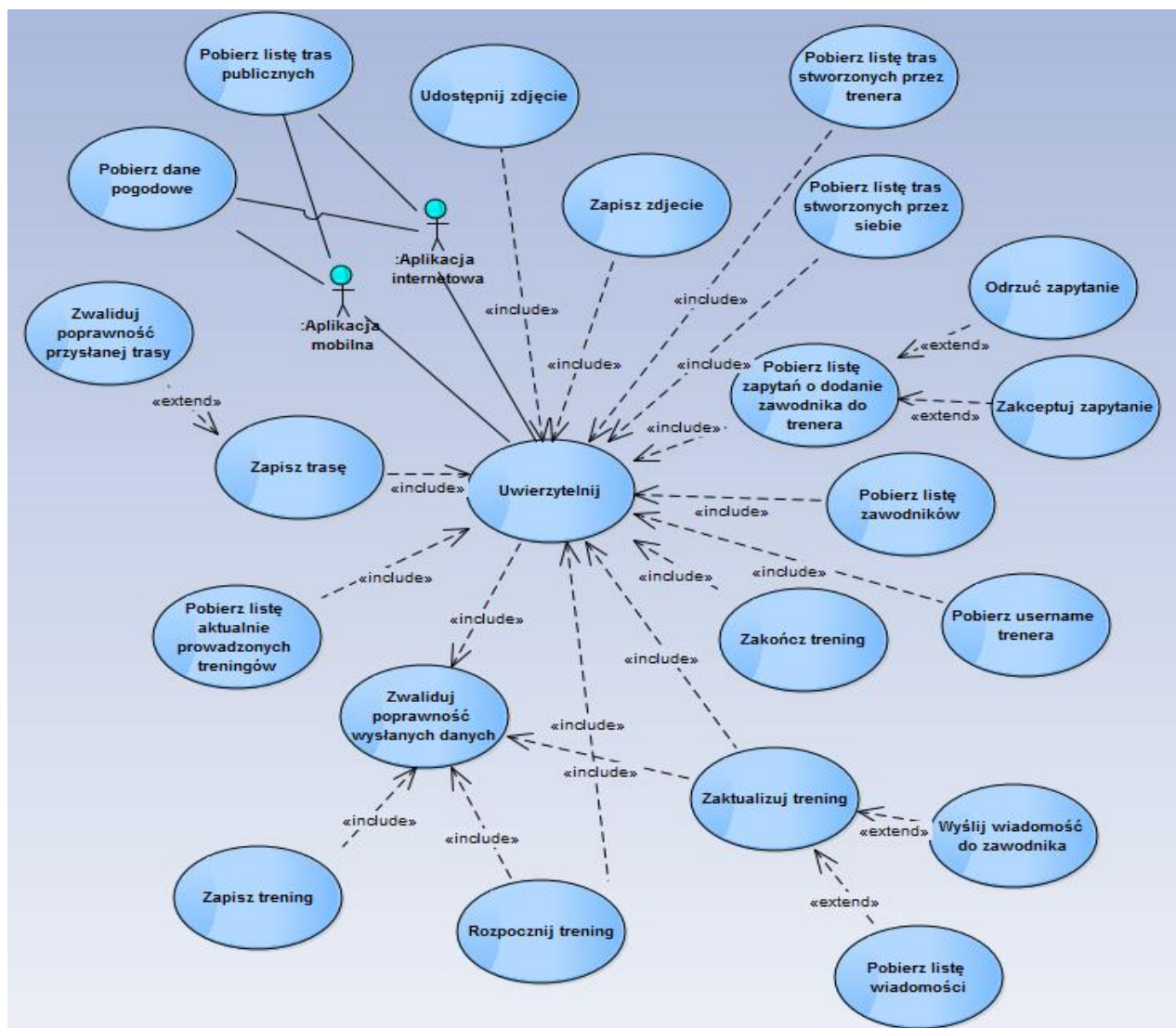
Opisy przypadków użycia

Nazwa przypadku użycia	Zarejestruj
Zdarzenie wywołujące	Użytkownik wysłał odpowiednie zapytanie POST
Opis	Obsługuje rejestrację użytkowników. Serwer po otrzymaniu odpowiednich danych dodaje użytkownika do bazy danych
Aktorzy	Aplikacja mobilna i webowa
Warunki początkowe	Użytkownik wysłał zapytanie do serwera
Przebieg	<ol style="list-style-type: none"> 1. Użytkownik tworzy zapytanie 2. Użytkownik zawiera dane nowego użytkownika w ciele zapytania 3. Użytkownik wysłał zapytanie 4. Serwer weryfikuje poprawność danych (sprawdza czy hasło nie jest puste, i czy nazwa użytkownika jest adresem e-mail) 5. Serwer dodaje użytkownika do bazy danych 6. Serwer wysłał wiadomość potwierdzającą dodanie użytkownika do bazy danych
Warunki końcowe	Użytkownik otrzymuje od serwera wiadomość potwierdzającą dodanie go do bazy, wraz z kodem HTTP 200

Nazwa przypadku użycia	Uwierzytelnij
Zdarzenie wywołujące	Użytkownik wysyła odpowiednie zapytanie POST
Opis	Obsługuje uwierzytelnianie użytkowników, poprzez sprawdzenie czy istnieje w bazie danych, oraz czy hasło jest poprawne. Serwer odpowiada tokenem uwierzytelniającym
Aktorzy	Aplikacja mobilna i webowa
Warunki początkowe	Użytkownik posiada konto w bazie danych, oraz dostarczy poprawne dane
Przebieg	<ol style="list-style-type: none"> 1. Użytkownik tworzy zapytanie 2. Użytkownik zawiera dane użytkownika w ciele zapytania 3. Użytkownik wysyła zapytanie 4. Serwer weryfikuje poprawność danych 5. Serwer wysyła token uwierzytelniający
Warunki końcowe	Użytkownik otrzymuje od serwera token uwierzytelniający, wraz z kodem HTTP 200

Nazwa przypadku użycia	Zmień hasło
Zdarzenie wywołujące	Użytkownik wysyła odpowiednie zapytanie POST
Opis	Obsługuje usługę zmiany hasła przez użytkownika
Aktorzy	Aplikacja mobilna i webowa
Warunki początkowe	Użytkownik wysyła zapytanie do serwera. Użytkownik jest uwierzytelniony.
Przebieg	<ol style="list-style-type: none"> 1. Użytkownik tworzy zapytanie 2. Użytkownik zawiera nowe hasło w ciele zapytania 3. Użytkownik wysyła zapytanie 4. Serwer weryfikuje poprawność danych (sprawdza czy hasło nie jest puste) 5. Serwer zmienia hasło użytkownika w bazie danych 6. Serwer wysyła wiadomość potwierdzającą zmianę hasła
Warunki końcowe	Użytkownik otrzymuje od serwera wiadomość potwierdzającą zmianę hasła, wraz z kodem HTTP 200

Nazwa przypadku użycia	Uwierzytelnij przez Google Plus
Zdarzenie wywołujące	Użytkownik loguje się przez Google Plus
Opis	Obsługuje uwierzytelnianie użytkownika, jeżeli zalogował się przez usługę Google Plus
Aktorzy	Aplikacja mobilna i webowa
Warunki początkowe	Użytkownik zalogował się przez Google Plus, serwery Google zwróciły mu token. Użytkownik wysyła zapytanie do serwera.
Przebieg	<ol style="list-style-type: none"> 1. Użytkownik tworzy zapytanie informujące serwer że loguję się przez Google Plus 2. Użytkownik zawiera swoją nazwę użytkownika w ciele zapytania, 3. Użytkownik wysyła zapytanie 4. Serwer weryfikuje poprawność danych (sprawdza czy użytkownik istnieje) 5. Serwer zwraca użytkownikowi token uwierzytelniający, lub jeżeli nie znalazł go w bazie, to tworzy w niej nowy wpis
Warunki końcowe	Użytkownik otrzymuje od serwera token uwierzytelniający, wraz z kodem HTTP 200



x. Diagram pozostałych przypadków użycia dla serwera

Uwaga – przypadek użycia „Uwierzytelnij” jest tożsamy z przypadkiem użycia o tej samej nazwie z poprzedniego diagramu przedstawionego na rysunku [NUMER RYSUNKU]. Diagramy zostały rozdzielone dla większej czytelności.

Opisy przypadków użycia

Nazwa przypadku użycia	Pobierz dane pogodowe
Zdarzenie wywołujące	Użytkownik wysła odpowiednie zapytanie GET
Opis	Serwer udostępnia dane pogodowe, przechowywane we własnej pamięci podręcznej.
Aktorzy	Aplikacja mobilna i webowa
Warunki początkowe	Użytkownik wysła zapytanie do serwera
Przebieg	<ol style="list-style-type: none"> 1. Użytkownik tworzy zapytanie, w którym zawiera szerokość i długość geograficzną punktu dla którego pogodę chce sprawdzić 2. Serwer przeszukuje pamięć podręczną w poszukiwaniu odpowiednich danych 3. Serwer odpowiada użytkownikowi danymi pogodowymi z formacie JSON
Warunki końcowe	Użytkownik otrzymuje od serwera dane pogodowe, wraz z kodem HTTP 200

Nazwa przypadku użycia	Pobierz listę zapytań o dodanie zawodnika do trenera
Zdarzenie wywołujące	Użytkownik wysyła odpowiednie zapytanie GET
Opis	Serwer odpowiada użytkownikowi listą wszystkich próśb (wysłanych przez trenerów) o dodanie zawodnika jako własnego biegacza
Aktorzy	Aplikacja mobilna (zawodnik)
Warunki początkowe	Użytkownik wysyła zapytanie do serwera Użytkownik jest uwierzytelniony
Przebieg	1. Użytkownik wysyła do serwera prośbę o zwrócenie listy zapytań wysłanych do jego własnego konta 2. Serwer odpowiada listą zapytań, która w przypadku braku zapytań może być pusta
Warunki końcowe	Użytkownik otrzymuje od serwera listę zapytań, w której znajdują się numery identyfikacyjne zapytań, oraz nazwy użytkownika trenerów którzy wysłali te zapytania, wraz z kodem HTTP 200

Nazwa przypadku użycia	Pobierz listę zawodników
Zdarzenie wywołujące	Użytkownik wysyła odpowiednie zapytanie GET
Opis	Serwer odpowiada użytkownikowi (trenerowi) listą użytkowników obecnie przypisanych do jego konta (tych którzy potwierdzili zapytanie o dodanie zawodnika)
Aktorzy	Aplikacja webowa (trener)
Warunki początkowe	Użytkownik wysyła zapytanie do serwera Użytkownik jest uwierzytelniony
Przebieg	1. Trener wysyła do serwera prośbę o zwrócenie listy zawodników przypisanych do jego konta 2. Serwer odpowiada listą zawodników, w przypadku braku lista jest pusta
Warunki końcowe	Użytkownik otrzymuje od listę zawodników, wraz z kodem HTTP 200

Nazwa przypadku użycia	Pobierz username trenera
Zdarzenie wywołujące	Użytkownik wysyła odpowiednie zapytanie GET
Opis	Użytkownik pobiera nazwę użytkownika trenera który jest obecnie przypisany do jego konta
Aktorzy	Aplikacja mobilna (biegacz)
Warunki początkowe	Użytkownik wysyła zapytanie do serwera Użytkownik jest uwierzytelniony
Przebieg	1. Zawodnik wysyła do serwera zapytanie z prośbą o udostępnienie nazwy trenera obecnie dodanego do jego konta 2. Serwer odpowiada nazwą trenera, lub informuje, że brak przypisanego trenera do konta
Warunki końcowe	Użytkownik otrzymuje od serwera nazwę przypisanego trenera, wraz z kodem HTTP 200

Nazwa przypadku użycia	Zapisz trasę
Zdarzenie wywołujące	Użytkownik wysyła odpowiednie zapytanie POST
Opis	Serwer odbiera trasę wysłaną przez użytkownika i zapisuje ją do bazy danych
Aktorzy	Aplikacja mobilna i webowa
Warunki początkowe	Użytkownik wysyła zapytanie do serwera, w którego ciele umieszcza listę punktów trasy oraz dodatkowe jej parametry Użytkownik jest uwierzytelniony
Przebieg	<ol style="list-style-type: none"> 1. Użytkownik wysyła zapytanie z informacją o całej trasie, czyli jej punktami, tytułem, opisem, czy chce by była widoczna dla wszystkich 2. Serwer waliduje poprawność wysłanych danych 3. Serwer zapisuje trasę do bazy danych
Warunki końcowe	Użytkownik otrzymuje od serwera potwierdzenie zapisania trasy, wraz z kodem HTTP 200

Nazwa przypadku użycia	Zapisz trening
Zdarzenie wywołujące	Użytkownik wysyła odpowiednie zapytanie POST
Opis	Serwer odbiera trening wysłany przez użytkownika i zapisuje go do bazy danych
Aktorzy	Aplikacja mobilna (biegacz)
Warunki początkowe	Użytkownik wysyła zapytanie do serwera Użytkownik jest uwierzytelniony
Przebieg	<ol style="list-style-type: none"> 1. Użytkownik wysyła zapytanie z informacją o całym wykonanym treningu, czyli jego tempie, długości, czasie trwania, liczbie spalonych kalorii, oraz trasą którą przebył 2. Serwer waliduje poprawność wysłanych danych 3. Serwer zapisuje trening do bazy danych
Warunki końcowe	Użytkownik otrzymuje od serwera potwierdzenie zapisania treningu, wraz z kodem HTTP 200

Nazwa przypadku użycia	Rozpocznij trening
Zdarzenie wywołujące	Użytkownik wysyła odpowiednie zapytanie POST
Opis	Użytkownik rozpoczyna trening na żywo, czyli taki który będzie widoczny dla trenera
Aktorzy	Aplikacja mobilna (biegacz)
Warunki początkowe	Użytkownik wysyła zapytanie do serwera Użytkownik jest uwierzytelniony
Przebieg	<ol style="list-style-type: none"> 1. Zawodnik wysyła do serwera zapytanie z prośbą o rozpoczęcie treningu na żywo, zawiera w nim pierwszy punkt w którym się aktualnie znajduje 2. Serwer sprawdza czy dany użytkownik już nie rozpoczął treningu, jeżeli nie, to dodaje trening do pamięci serwera (nie do bazy danych) i zwraca informacje potwierdzającą
Warunki końcowe	Użytkownik otrzymuje od serwera potwierdzenie rozpoczęcia treningu, wraz z kodem HTTP 200

Nazwa przypadku użycia	Zaktualizuj trening
Zdarzenie wywołujące	Użytkownik wysyła odpowiednie zapytanie POST
Opis	Użytkownik wysyła serwerowi swoją aktualną pozycję, która zostanie dopisana do rozpoczętego treningu przez serwer. Zapytanie to jest wysyłane przez użytkownika cyklicznie.
Aktorzy	Aplikacja mobilna (biegacz)
Warunki początkowe	Użytkownik wysyła zapytanie do serwera Użytkownik jest uwierzytelniony
Przebieg	<ol style="list-style-type: none"> 1. Zawodnik wysyła do serwera zapytanie w którym zawiera współrzędne w którym aktualnie się znajduje 2. Serwer sprawdza czy dany użytkownik rozpoczął trening, jeżeli tak, to aktualizuje trening i zwraca informacje potwierdzającą zaktualizowanie treningu 3. Serwer zwraca listę wiadomości wysłanych przez trenera do zawodnika w trakcie treningu
Warunki końcowe	Użytkownik otrzymuje od serwera potwierdzenie zaktualizowania treningu, wraz z kodem HTTP 200

Nazwa przypadku użycia	Zakończ trening
Zdarzenie wywołujące	Użytkownik wysyła odpowiednie zapytanie POST
Opis	Użytkownik kończy wcześniej rozpoczęty trening na żywo
Aktorzy	Aplikacja mobilna (biegacz)
Warunki początkowe	Użytkownik wysyła zapytanie do serwera Użytkownik jest uwierzytelniony
Przebieg	<ol style="list-style-type: none"> 1. Zawodnik wysyła do serwera zapytanie z prośbą o zakończenie treningu na żywo 2. Serwer sprawdza czy dany użytkownik rozpoczął treningu, jeżeli tak, to usuwa go z pamięci i zwraca informację potwierdzającą
Warunki końcowe	Użytkownik otrzymuje od serwera potwierdzenie zakończenia treningu, wraz z kodem HTTP 200

Nazwa przypadku użycia	Pobierz listę aktualnie prowadzonych treningów
Zdarzenie wywołujące	Użytkownik wysyła odpowiednie zapytanie GET
Opis	Trener pobiera listę jego zawodników, którzy są aktualnie podczas treningu na żywo
Aktorzy	Aplikacja webowa (trener)
Warunki początkowe	Użytkownik wysyła zapytanie do serwera Użytkownik jest uwierzytelniony
Przebieg	<ol style="list-style-type: none"> 1. Trener wysyła do serwera zapytanie o udostępnienie listy zawodników którzy aktualnie trenują 2. Serwer odpowiada listą obecnie trenujących zawodników przypisanych do konta trenera, może być pusta w przypadku braku aktywnych treningów.
Warunki końcowe	Użytkownik otrzymuje od serwera listę rozpoczętych treningów przez własnych zawodników, wraz z kodem HTTP 200

Nazwa przypadku użycia	Zapisz zdjęcie
Zdarzenie wywołujące	Użytkownik wysła odpowiednie zapytanie POST
Opis	Zawodnik wysła do serwera zapytanie, w którego ciele umieszcza kod base64 zdjęcia. Serwer odczytuje go, zapisuje we własnym systemie plików, oraz umieszcza adres URL tego zdjęcia w bazie danych, w tabeli tras, ponieważ zdjęcia są powiązane z trasami.
Aktorzy	Aplikacja mobilna (biegacz)
Warunki początkowe	Użytkownik wysła zapytanie do serwera Użytkownik jest uwierzytelniony
Przebieg	1. Użytkownik wysła w zapytaniu kod base64 obrazka, oraz z informację z jaką trasą chce powiązać obrazek 2. Serwer zapisuje obrazek w systemie plików, oraz adres URL w bazie danych.
Warunki końcowe	Użytkownik otrzymuje od serwera potwierdzenie zapisania zdjęcia wraz z kodem HTTP 200

Nazwa przypadku użycia	Pobierz listę tras publicznych
Zdarzenie wywołujące	Użytkownik wysła odpowiednie zapytanie GET
Opis	Zawodnik wysła do serwera zapytanie o udostępnienie listy tras publicznych
Aktorzy	Aplikacja mobilna i webowa
Warunki początkowe	Użytkownik wysła zapytanie do serwera Użytkownik jest uwierzytelniony
Przebieg	1. Użytkownik wysła odpowiednie zapytanie 2. Serwer zapisuje odpowiada użytkownikowi listą tras, łącznie z opcjonalnymi adresami URL zdjęć wykonanych na danej trasie
Warunki końcowe	Użytkownik otrzymuje od serwera listę tras publicznych, łącznie z ich listą punktów, adresami URL zdjęć i kodem HTTP 200

3.3. Aplikacja trenera(Marcin Olszewski)

ID z systemu redmine	Priorytet	Temat	Uwagi
70	Niski	Przeglądanie galerii zdjęć z trasy	
69	Niski	Wykreślanie profilu wysokościowego trasy	
68	Normalny	Listowanie zawodników, którzy obecnie trenują	
67	Normalny	Dodawanie sobie zawodnika	
66	Normalny	Zlecanie treningu przez trenera jego zawodnikowi	

65	Normalny	Listowanie tras treningowych przygotowanych przez trenera	
64	Normalny	Zapis tras treningowych	
63	Wysoki	Kontrola uprawnień - opcje trenera	
62	Normalny	Wysyłanie wiadomości do zawodnika	
10	Normalny	Podgląd trasy zawodnika na żywo	
9	Normalny	Listowanie tras opublikowanych dla wszystkich	

4. Architektura systemu oraz przegląd technologii

4.1. Ogólna architektura całego systemu.

4.2. Aplikacja mobilna (<https://github.com/RunandPL/AndroidApp>)

4.3. Serwer (<https://github.com/RunandPL/Serwer>)

4.4. Aplikacja trenera (Marcin Olszewski)

<https://github.com/RunandPL/Frontend>

Zgodnie z założeniami, dokonanymi podczas analizy wymagań, aplikacja dla trenera będzie modulem webowym. Jej interfejs będzie dostępny z poziomu przeglądarki internetowej. Dzięki zapewnieniu pełnego dostosowania do rozdzielczości przeglądarki (*ang. responsive*), zachowana zostanie wygoda korzystania z aplikacji również na urządzeniach mobilnych. Frameworkiem, którego użyjemy do implementacji aplikacji będzie AngularJS, otwarta biblioteka języka JavaScript, która jest wspierana i firmowana przez Google. Dzięki takiej decyzji, w naturalny sposób będziemy pracować w oparciu o wzorzec architektoniczny MVC (*ang. Model-View-Controller*). Szeroko stosowany od lat 70. minionego stulecia, jednak w programowaniu sieciowym został wprowadzony stosunkowo niedawno. Podstawą MVC jest zastosowanie wyraźnej separacji między logiką aplikacji (kontrolerem), zarządzaniem danymi (model) oraz sposobem prezentacji danych (widok).

AngularJS (<https://angularjs.org/>)

Możliwości w zakresie tworzenia aplikacji sieciowych są bardzo duże, a świadczy o tym chociażby szeroki przekrój technologii takich jak: PHP, Rails, Java EE, .NET czy Scala. Niestety często możemy się przekonać, że równie wysoki jest stopień skomplikowania, który jest związany z procesem



wytwarzania takich aplikacji. Technologia AngularJS powstała w celu ułatwienia programistom tworzenia aplikacji AJAX(ang. *Asynchronous JavaScript and XML*). Skuteczność AngularJS poparta jest doświadczeniem zespołu programistów Google, którzy zdobywali je pracując przy takich projektach jak Gmail, Mapy oraz Kalendarz. Dzięki uproszczeniu wykonywania niektórych czynności, jak np. wysyłanie żądań http, możemy zwrócić uwagę na decyzje projektowe, ułatwiające testowanie, dalszą rozbudowę aplikacji i jej konserwację. Dzisiaj, projekt AngularJS, rozwijany jest przez społeczność *open source* z całego świata.

W naszym projekcie zdecydowaliśmy się odejść od łączenia kodu HTML z danymi po stronie serwera i przekazywania wygenerowanej strony przeglądarce internetowej. Zamiast tego, korzystamy z szablonów stron, które łączone są z danymi dzięki bibliotece AngularJS, a następnie przekazywane do przeglądarki. W ten sposób, rola serwera ogranicza się do przechowywania i udostępniania zasobów statycznych szablonów oraz przekazywania danych niezbędnych do wypełnienia wspomnianych szablonów. Jest to podejście, które może się kojarzyć z aplikacjami, które według podejścia AJAX, posiadają jedną stronę, która wypełniana jest dynamicznie danymi.

Zgodnie z założeniem implementacji architektury MVC, w aplikacjach AngularJS widokiem jest DOM(ang. *Document Object Model*) – obiektowy model dokumentu, kontrolerami są klasy JavaScript, a dane modelu przechowywane są we właściwościach obiektu. Dzięki tak dużej separacji warstw, możliwe staje się dokładne pokrycie aplikacji testami, co przy innym podejściu, dokładając do tego złą strukturę kodu, było wręcz niemożliwe.

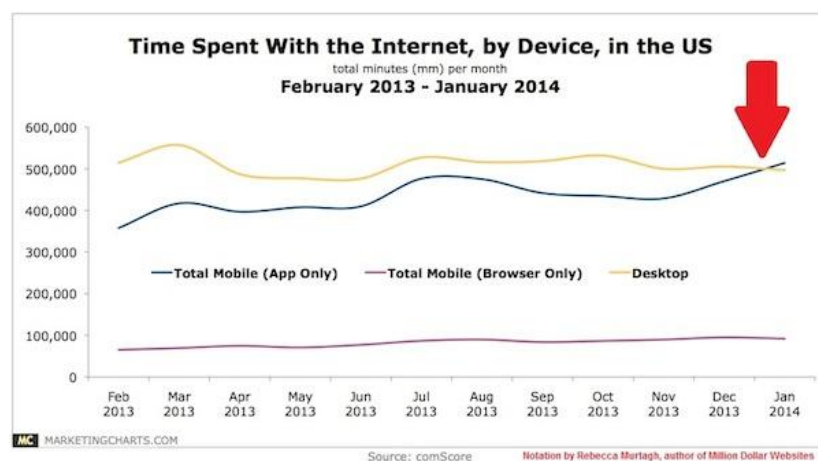
Dużo mówi się o rozszerzaniu możliwości istniejących już rozwiązań. Angular rozszerza możliwości HTML poprzez wprowadzenie dwukierunkowego wiązania danych(ang. *Two Way Data-Binding*). Jak to działa? Stworzone w HTMLu szablony łączone są zgodnie z danymi zawartymi w zakresie(ang. *scope*) zdefiniowanym przez model. Serwis \$scope w Angular identyfikuje zmiany w modelu, po czym modyfikuje HTML w widoku poprzez kontroler. Analogicznie, wszelkie zmiany w widoku są widoczne w modelu. Pozwala to ominąć potrzebę manipulowania na drzewie DOMu i znacznie przyspiesza tworzenie aplikacji internetowych. Biblioteka dostarcza ponadto wiele przydatnych rozwiązań, które dotychczas musieliśmy implementować sami. Są to m.in. mechanizmy routingu, wykorzystanie części stron(ang. *partial*), czy wstrzykiwanie zależności i wykorzystanie dyrektyw. Równie proste staje się wysyłanie żądań http. Poniżej został przedstawiony przykładowy kod najprostszej aplikacji MVC, opartej o bibliotekę AngularJS.

```
1.      <script type="text/javascript">
2.          var myApp = angular.module('myApp', []);
3.          myApp.ExampleController = function ($scope) {
4.              $scope.name = 'RunAnd';
5.          };
6.      </script>
7.
8.      <div ng-app="myApp" ng-controller="ExampleController">
9.          <h1>Aplikacja: {{ name }}!</h1>
10.     </div>
```

Bootstrap (<http://getbootstrap.com/>)

Projektując, a następnie implementując aplikacje dla użytkowników z graficznym interfejsem użytkownika(ang. Graphical User Interface, GUI), musimy zadbać nie tylko o to, aby spełniały swoje podstawowe funkcje, lecz dobrze zastanowić nad tym kto i na jakich urządzeniach będzie korzystał z naszych produktów.

Według firmy *Incisive Media* (dostawcy informacji biznesowych), która porównała łączny czas dostępu do Internetu na urządzeniach mobilnych oraz na komputerach osobistych od lutego 2013 roku do stycznia 2014 roku, nastąpił już moment, kiedy użytkownicy Internetu korzystają z niego więcej za pośrednictwem urządzeń mobilnych. Wyraźny jest też wzrost czasu korzystania z Internetu za pośrednictwem przeglądarek na urządzeniach mobilnych.



<http://cms.searchenginewatch.com/IMG/303/293303/time-spent-on-internet-by-device-in-us.jpg?1404760136>

Zdecydowaliśmy się zatem na wykorzystanie w aplikacji trenerskiej, rozwijanego przez programistów Twittera, Framework CSS o nazwie Bootstrap. Zawiera on zestaw narzędzi, które ułatwiają tworzenie interfejsu graficznego aplikacji internetowych. Bazuje m.in. na gotowych rozwiązaniach HTML oraz CSS. Dzięki wykorzystaniu bootstrapa i jego klas, otrzymamy interfejs dostosowany do urządzenia, nie zależnie czy jest to komputer PC, tablet, czy telefon komórkowy. Wyświetlane moduły dostosują swoją wielkość i wygląd do rozdzielczości urządzenia. Strony i aplikacje posiadające takie właściwości możemy określić mianem responsywnych(ang. *responsive*).



Bootstrap może być stosowany m.in. do stylizacji formularzy, przycisków, menu i wielu innych wyświetlanych. Framework wykorzystuje także język JavaScript.

Aby zintegrować aplikację z Bootstrapem wystarczy pobrać ze strony projektu skompilowany zbiór arkuszy CSS, bibliotekę JavaScript oraz czcionki, a następnie załączyć pobrane zasoby na danej stronie html.

1. `<!-- Bootstrap core CSS -->`
2. `<link href="bootstrap/dist/css/bootstrap.css" rel="stylesheet">`
3. `<!-- Bootstrap theme -->`
4. `<link href="bootstrap/dist/css/bootstrap-theme.css" rel="stylesheet">`

Prosty kod strony html, której wygląd przedstawia rys. 3, może wyglądać następująco:

sample code here

Screen przygotowanej strony demonstracyjnej w bootstrap

Rys 3

Wspierane przeglądarki:

	Chrome	Firefox	Internet Explorer	Opera	Safari
Android	Supported	Supported	N/A	Not Supported	N/A
iOS	Supported	N/A*		Not Supported	Supported
Mac OS X	Supported	Supported		Supported	Supported
Windows	Supported	Supported	Supported	Supported	Not Supported

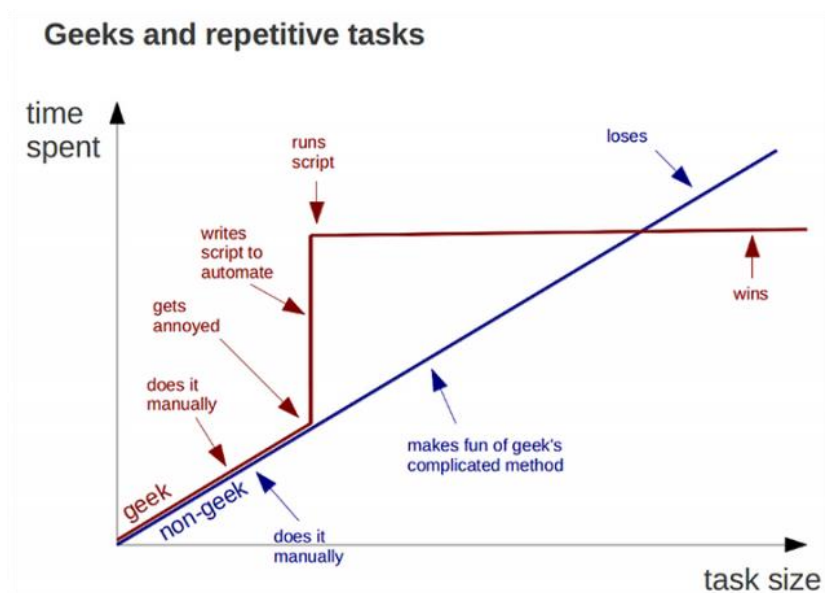
N/A – brak odpowiedzi

Bower (<http://bower.io/>)

Wytwarzanie oprogramowania składa się w dużej mierze z czynności powtarzalnych, takich jak minimalizacja skryptów JS lub kompilacja stylów SASS do CSS. Są to często wykonywane czynności, które zabierają programiście wraz z upływem czasu i rozrastaniem się projektu, coraz więcej czasu.

Zależności te przedstawia poniższy wykres:





Rys.3. <http://webmastah.pl/wp-content/uploads/2013/12/lazy-graph-640x457.png>

W naszym przypadku, najwięcej czasu zajmowałoby wyszukiwanie oraz instalacja bibliotek JS, a w trakcie utrzymania projektu, ich aktualizacja. Warto jednak wykorzystać dostępny manager pakietów Bower, którego zadaniem jest zarządzanie wszystkimi bibliotekami, które będziemy wykorzystywali w przeglądarce. Jest to dojrzały projekt, a świadczy o tym liczba ponad 6000 różnych bibliotek, które posiada. Jego użycie jest bardzo proste i ogranicza się do znajomości 7 podstawowych poleceń.

```
1. npm install -g bower
2. bower init
3. bower search [<name>]
4. bower info [<name>]
5. bower install [<name>]
6. bower update [<name>]
7. bower uninstall [<name>]
```

W pierwszej linii listingu znajduje się polecenie instalacji bowera, następnie jego inicjalizacja. Dostępność interesującej nas biblioteki sprawdzamy poleceniem z linii nr 3, gdzie jako [<name>] należy wprowadzić nazwę biblioteki, np. jquery.

Polecenie z linii nr 5 pozwoli nam zainstalować bibliotekę, a aktualizację wykonamy wydając polecenie z linii 6.

Domyślną lokalizacją dla bibliotek Bowera jest folder /bower_components, jednak można to zmienić w pliku konfiguracyjnym .bowerrc. Kolejnym ważnym plikiem jest bower.json, który przechowuje informację o potrzebnych bibliotekach oraz ich wersji. Plik bower.json można wygenerować poleceniem z linii nr 2 lub można stworzyć go ręcznie.

```
1. {
2.   "name": "runand-frontend",
3.   "description": "A RunAnd AngularJS Project",
4.   "version": "0.2.0",
5.   "homepage": "https://github.com/RunandPL",
```

```
6.     "license": "MIT",
7.     "private": true,
8.     "dependencies": {
9.         "angular": "1.2.x",
10.        "angular-route": "1.2.x",
11.        "angular-loader": "1.2.x",
12.        "angular-mocks": "~1.2.x",
13.        "html5-boilerplate": "~4.3.0",
14.        "bootstrap": ">= 3.0.0",
15.        "angular-google-maps": "~2.0.6"
16.    }
17. }
```

Angular Google Maps (<http://angular-ui.github.io/angular-google-maps/>)

W naszej pracy inżynierskiej, zarówno aplikacji mobilnej oraz webowej, korzystamy z dostępu do map. Zdecydowaliśmy się na usługi oferowane przez firmę Google, ze względu na jej renomę oraz jakość świadczonych usług. W dalszej części rozdziału opiszę sposób rejestracji, uwierzytelniania oraz komunikacji z serwisami Google. Są one dostępne dla każdego programisty, który zechce z nich skorzystać. Jedyne co musi zrobić to utworzyć konto Google a następnie zarejestrować tworzoną przez siebie aplikację. Kiedy to zrobimy dostaniemy dostęp do wersji bezpłatnych serwisów, w tej wersji większość z nich daje nam możliwość wykonania określonej, dziennej liczby zapytań. Na czas rozwoju oraz testów naszej aplikacji jest to wystarczająca opcja, gdyż liczba zapytań sięga zazwyczaj kilku tysięcy. Gdy jednak przestanie nam to wystarczać, pozostaje nam konieczność ponoszenia opłat dodatkowych opłat za dodatkowe zapytania.

Proces rejestracji jest niezwykle prosty i szybki, zajmuje dosłownie kilka minut. Jedyne co nam będzie potrzebne to nazwa pakietu, w którym znajduje się nasza aplikacja oraz wygenerowany dla niej (odcisk?)certyfikatu. Kiedy wszystko będzie gotowe należy wejść na stronę Google Developer Console. Z jej poziomu mamy możliwość zarządzania zarejestrowanymi aplikacjami, włączonymi dla naszych aplikacji serwisami, przeglądania płatności oraz statystyki. Rejestrację aplikacji rozpoczynamy od rozwinięcia zakładki „APIs & auth” a następnie kliknięcia pozycji „Cridentials”. W nowym oknie wybieramy opcję „Create new Key”. Dalej kierujemy się instrukcjami podanymi na ekranie, ich wynikiem będzie rejestracja naszej aplikacji. Następnie należy stworzyć identyfikator protokołu OAuth 2.0. Google używa go do uwierzytelniania oraz autoryzacji usług i aplikacji. Proces tworzenia identyfikatora jest równie prosty jak rejestracji aplikacji.

Google udostępnia do naszego użytku, odpłatnego lub darmowego, ponad 70 usług. Domyślnie wszystkie są dla nas wyłączone przez co nasza aplikacja nie będzie mogła z nich skorzystać. W zakładce „APIs” znajdziemy listę wszystkich interfejsów, wybrane z nich mamy możliwość aktywować. Na liście prócz nazw interfejsów znajduje się też limit zapytań jaki możemy wykorzystać, a po aktywacji interfejsu mamy możliwość śledzenia ich wykorzystania.

5. Implementacja projektu

Uwierzytelnianie z tokenem

Uwierzytelnianie Google+

Implementacja Google Maps API

5.1. Aplikacji mobilnej (Sebastian Miałkowski)

Zgodnie z założeniami projektu, aplikacja mobilna została zaimplementowana na telefony z systemem Android. Zdecydowaliśmy się na to z dwóch powodów. Pierwszym była chęć lepszego poznania systemu oraz kontynuowanie nauki pisania aplikacji dla niego przeznaczonych, którą rozpoczęliśmy na przedmiocie Programowanie aplikacji mobilnych. Kolejnym, chęć dotarcia z naszą aplikacją do jak największej liczby odbiorców. Biorąc pod uwagę te dwa założenia, wybór systemu Android był wręcz oczywisty, gdyż jak wynika z tabeli [NR TABELI] jest to obecnie najpopularniejszy system na urządzenia mobilne, a jego udział w tym rynku stale rośnie.

Okres	Android	IOS	Windows Phone	BlackBerry OS	Inne
Połowa roku 2014	84.4%	11.7%	2.9%	0.5%	0.6%
Połowa roku 2013	81.2%	12.8%	3.6%	1.7%	0.6%
Połowa roku 2012	74.9%	14.4%	2.0%	4.1%	4.5%

Tabela [NR] Udział systemów mobilnych w rynku

(źródło: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>)

Dodatkowo zdecydowaliśmy że najniższą wersją systemu wspieraną przez naszą aplikację będzie Android 4.1.2(Jelly Bean). Na podstawie tabeli [NR TABELI] widać że nasza aplikacja będzie wspierać 82.6% urządzeń z systemem firmy Google.

4.1.x	Jelly Bean	16	21.3%
4.2.x	Jelly Bean	17	20.4%
4.3	Jelly Bean	18	7.0%
4.4	KitKat	19	33.9%

Tabela [NR] Udział rynkowy poszczególnych wersji systemu Android

(źródło: developer.android.com/about/dashboards/index.html)

Podczas implementacji aplikacji korzystaliśmy z dwóch frameworków: Butterknife oraz Picasso. Niezbędna również była możliwość wyświetlania map, by to uzyskać zintegrowaliśmy aplikację z serwisem Google Maps.

Picasso(<https://square.github.io/picasso>)

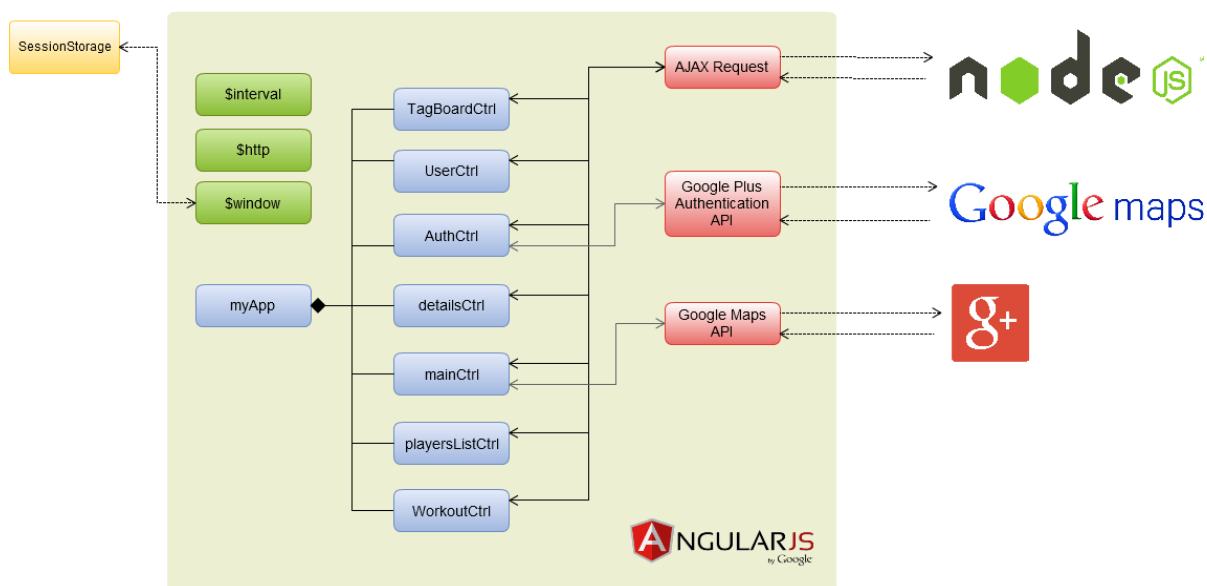
W czasach gdy serwisy społecznościowe cieszą się dużą popularnością, ludzie robią oraz wysyłają na nie coraz więcej zdjęć. Skutkuje to również tym że duża liczba aplikacji musi mieć

zaimplementowaną możliwość ich obsługi. Poza tak oczywistymi przykładami jak *Facebook*, czy *Instagram* robienie oraz podgląd zdjęć udostępniają aplikacje treningowe. W takich właśnie aplikacjach zastosowanie znajduje *framework Picasso*. Zdjęcia czy obrazki często mają rozmiary po kilka megabajtów, ich wczytanie więc może powodować blokowanie interfejsu użytkownika. By temu zapobiec takie funkcje można zaimplementować na osobnych wątkach, jednak są to często operacje kłopotliwe oraz występujące w wielu miejscach. By ułatwić sobie pracę można skorzystać z Picasso, który wszystkie operacje wykona za nas a dodatkowo zajmie się obsługą błędów. Framework pozwala wyświetlenie obrazków pobranych z Internetu, wczytanych z dysku oraz pamięci. Prócz samego odczytu plików graficznych została nam udostępniona również możliwość ich prostej edycji np. zmiana rozmiaru, by lepiej mogły się dopasować wyglądu naszej aplikacji. Na wypadek gdyby dostępne możliwości edycji były dla nas niewystarczające, możemy zaimplementować własne i przekazać do wykorzystania dla Picasso. W naszej aplikacji używamy go do pobrania obrazka trasy, który jest generowany przez Google Static Map. Poniżej zamieszczam przykładowy fragment kodu powodujący pobranie obrazka z danego adresu internetowego, a następnie załadowanie go do widok *imageView*.

```
1. Picasso.with(context).load("http://i.imgur.com/Dvpvklm").into(imageView);
```

5.2.Implementacja serwera

5.3.Implementacja aplikacji WWW (Marcin Olszewski)



Powyższy diagram przedstawia strukturę modułów aplikacji trenerskiej. Aplikacja AngularJS (na diagramie *myApp*), zawiera kontrolery, które komunikują się asynchronicznie poprzez żądania AJAX z serwerem, udostępniającym usługi REST. AngulaJS umożliwia wysyłanie żądań poprzez wykonywanie metod obiektu *\$http* (metody *.post()* oraz *.get()*), które poprzez mechanizm *Dependency Injections*, wstrzykujemy do kontrolera. Wyróżnione zostały także obiekty *\$interval* – odpowiadający za cykliczne wywoływanie funkcji

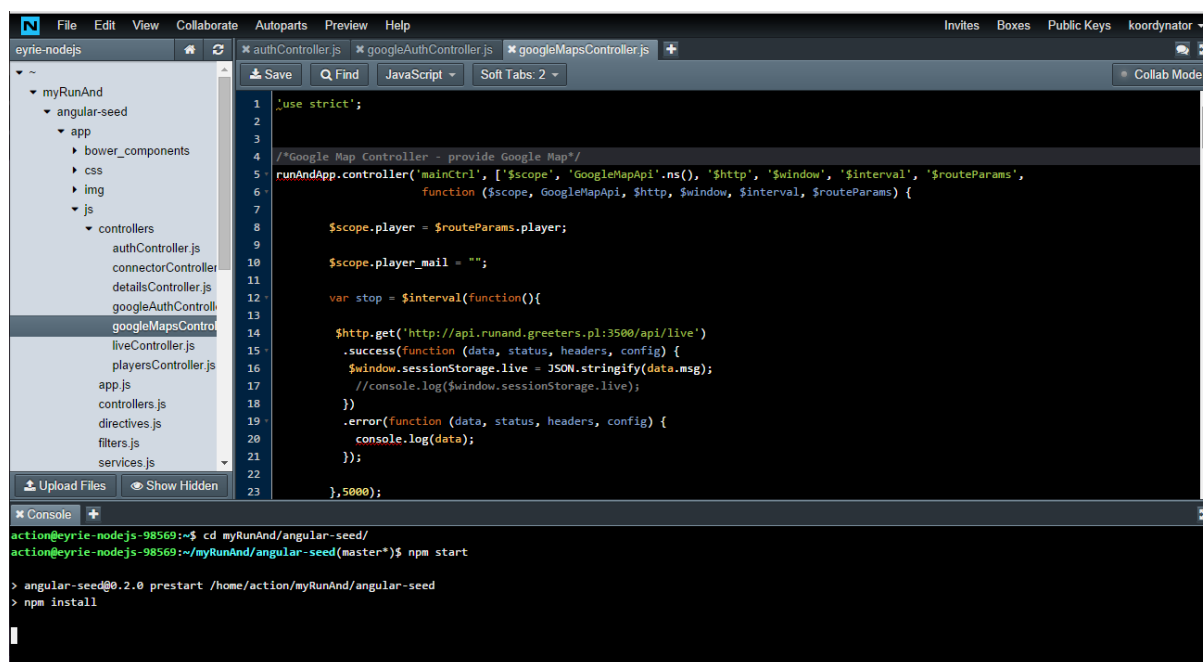
oraz `$window` – dzięki niemu uzyskujemy dostęp do obiektu `window` języka JavaScript i możemy zapisywać i odczytywać dane z `SessionStorage`. Aplikacja składa się z 7 kontrolerów:

- TagBoardCtrl
- UserCtrl
- AuthCtrl
- detailsCtrl
- mainCtrl
- playersListCtrl
- WorkoutCtrl

W projekcie użyte zostały także niestandardowe moduły do obsługi Google Maps API oraz uwierzytelniania Google Plus.

NitrousIO i struktura projektu

Jedną z głównych decyzji projektowych był wybór środowiska implementacji aplikacji trenera. AngularJS wydał nam się najlepszym rozwiązaniem, ponieważ dostarcza on najpotrzebniejszych mechanizmów, takich, jak: tworzenie szablonów widoków, routing URL, wiązanie danych do kontrolki widoku oraz składowanie danych, czyli wsparcie dla technologii AJAX, która była niezbędna do realizacji komunikacji z serwerem poprzez usługę REST. Aby umożliwić hosting aplikacji trenerskiej, wykorzystaliśmy środowisko *Nitrous* udostępniające maszynę wirtualną w chmurze, która wspiera takie platformy, jak Ruby, Python oraz Node.js. Tworząc darmowe konto w serwisie Nitrous, otrzymaliśmy dostęp do maszyny wirtualnej posiadającej 300 MB pamięci RAM oraz 1000 MB pamięci dyskowej. Dodatkowo, zyskaliśmy dostęp do narzędzi programistycznych – IDE z poziomu przeglądarki internetowej, dzięki czemu, w tym samym czasie kilku użytkowników może edytować ten sam kod. IDE posiada także konsolę, która przez SSH łączy się z serwerem, oraz chat do komunikacji programistów.



```
File Edit View Collaborate Autoparts Preview Help
eyrie-nodejs
myRunAnd
  angular-seed
    app
      bower_components
      css
      img
      js
      controllers
        authController.js
        connectorController.js
        detailsController.js
        googleAuthController.js
        googleMapsController.js
        liveController.js
        playersController.js
        app.js
        controllers.js
        directives.js
        filters.js
        services.js
    Upload Files Show Hidden
  Save Find JavaScript Soft Tabs: 2 Collab Mode
1 'use strict';
2
3
4 /*Google Map Controller - provide Google Map*/
5 runAndApp.controller('mainCtrl', ['$scope', 'GoogleMapApi', '$http', '$window', '$interval', '$routeParams',
6   function ($scope, GoogleMapApi, $http, $window, $interval, $routeParams) {
7
8     $scope.player = $routeParams.player;
9
10    $scope.player_mail = "";
11
12    var stop = $interval(function(){
13
14      $http.get('http://api.runand.greeters.pl:3500/api/live')
15        .success(function (data, status, headers, config) {
16          $window.sessionStorage.live = JSON.stringify(data.msg);
17          //console.log($window.sessionStorage.live);
18        })
19        .error(function (data, status, headers, config) {
20          console.log(data);
21        });
22    },5000);
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
25
```

Kolejnym krokiem, było utworzenie struktury projektu. Jako, że aplikacja oparta o Framework AngularJS umożliwia korzystanie z kontrolerów, szablonów widoku, dyrektyw, serwisów oraz zawiera zasoby statyczne(np. pliki json, arkusze stylów, pliki graficzne), należało utworzyć strukturę projektu, która w intuicyjny sposób zorganizuje projekt.

app/	-> główny folder aplikacji
bower_components/	-> dodatkowe biblioteki menedżera pakietów bower
css/	-> arkusze stylów css
img/	-> pliki graficzne
js/	
controllers/	-> kontrolery komunikujące się z serwerem
controllers.js	-> zbiór kontrolerów związanych z wyświetlaniem layoutu
app.js	-> punkt wejścia do aplikacji AngularJS
directives.js	-> dyrektywy dostarczające zawartość szablonów
services.js	-> usługi, np. interceptory
navbar_provider/	-> pliki konfiguracyjne json, dostępne opcje w menu
partials/	-> pliki widoków aplikacji
tpl/	-> pliki szablonów dla widoków
google_map.html	-> główny widok mapy
tag_board.html	-> widok listy opublikowanych treningów
workouts.html	-> widok listy aktualnie trenujących zawodników
routes/	
dummy_route.json	-> zastępcza trasa testowa
slider/	
slides.json	-> plik konfiguracyjny slidera
workouts_provider/	
dummy_workouts.json	-> dane testowe, lista opublikowanych tras
index.html	-> strona główna aplikacji

Widok - szablony i dyrektywy

Koncepcję wzorca MVC we frameworku AngularJS realizuje się poprzez składanie części widoku w jeden, który widzi użytkownik. W aplikacji trenerskiej zaimplementowane są tylko trzy główne partiale – widok mapy, listy opublikowanych tras oraz treningów. Żeby uprościć i uporządkować kod źródłowy partiali, można zastosować szablony. Tą funkcję Framework uzyskujemy poprzez mechanizm dyrektyw, który jest przez jego twórców nazwany poszerzeniem funkcji HTML.

Dyrektywy w kodzie HTML, to miejsca, w które Angular w fazie kompilacji, podczas skanowania drzewa DOM, podcina nowe funkcje, które mają zastosowanie dla danego węzła oraz jego podwęzłów(tzw. dzieci). Można dzięki temu uzyskać komponenty, które pozwalają manipulować drzewem DOM oraz dotaczać mu nowe funkcje. Przykładami predefiniowanych dyrektyw Framework są ng-app, ng-controller lub ng-repeat, która pozwala budować dynamiczną listę w oparciu o kolekcję obiektów.

Dyrektywy można tworzyć na kilka różnych sposobów, posiadają także szereg parametrów konfiguracyjnych. W poniższym przykładzie zaprezentowany jest przykład prostej dyrektywy, dołączającej plik szablonu do drzewa DOM.

```
myDirectives.directive('carouselDirective', function () {  
  return{  
    templateUrl: 'partials/tpl/carousel.tpl.html'
```

```

    }
  });

  <!-- użycie dyrektywy -->
  <div class="wrap" navbar-directive></div>

```

Zastosowany w powyższym listingu parametr `templateUrl` pozwala na określenie ścieżki do szablonu – osobnego pliku HTML. Szablon taki zostanie załadowany asynchronicznie oraz będzie zapisany w pamięci podręcznej(ang. *cache*).

Routing URL

W odniesieniu do aplikacji internetowych, trasowanie(ang. *routing*) to mechanizm dynamicznego określania ścieżek URL(ścieżek dostępu, które są zawarte w URL żądania HTTP) odwzorowywanych dla różnych obszarów aplikacji. Routing w aplikacjach internetowych pozwala utrzymać porządek po stronie programisty, ale z drugiej strony powoduje, że nasza aplikacja jest bardziej czytelna po stronie użytkownika. Pozbywamy się nieczytelnych i często długich ścieżek, jak np. `index.php?article_id=12&category=2` na rzecz czytelniejszego adresu, np. `/articles/run-and`.

Dostarczenie mechanizmów routingu jest jedną z zalet korzystania z frameworków. Oprócz Symfony, Zend Framework, czy Backbone, również AngularJS je udostępnia. Należy jednak dołączyć do pliku `index.html` bibliotekę *angular-route.js* i w prosty sposób skonfigurować dostawcę routingu(ang. *routing provider*). Poniżej znajduje się pełna konfiguracja routingu dla frontendu aplikacji trenerskiej.

```

1.  /*deklaracja modułów wykorzystywanych w aplikacji AngularJS*/
2.  var myApp = angular.module('myApp', [
3.      'ngRoute', /*dołączenie modułu routingu do aplikacji myApp*/
4.      'myApp.filters',
5.      'myApp.services',
6.      'myApp.directives',
7.      'myApp.controllers',
8.      'google-maps'.ns(),
9.      'googleplus'
10.  ]);
11.
12.  /*konfiguracja route providera - dodawanie reguł routingu*/
13.  myApp.config(['$routeProvider', function ($routeProvider) {
14.      $routeProvider.when('/workouts_board', {
15.          templateUrl: 'partials/workouts_board.html',
16.          controller: 'WorkoutsBoardCtrl'
17.      });
18.      $routeProvider.when('/main_map', {
19.          templateUrl: 'partials/google_map.html',
20.          controller: 'WorkoutCtrl'
21.      });
22.      $routeProvider.otherwise({
23.          redirectTo: '/tag_board'
24.      });
25.  }]);

```

Z powyższego listingu pochodzącego z pliku `app/js/app.js` wynika, że korzystaliśmy z dwóch reguł obsługujących wyświetlanie tablicy z opublikowanymi treningami powiązanej z kontrolerem `WorkoutsBoardCtrl` oraz mapy, na której wykonywane są operacje właściwe dla trenera oraz śledzonego treningu (np. wyświetlanie trasy biegacza na żywo lub wysyłanie zlecenia treningu i listowanie zawodników trenowanych przez zalogowanego trenera), obsługiwanej przez `WorkoutCtrl`. Określiliśmy także routing domyślny, gdy adresu url nie będzie można dopasować do żadnej z reguł (wyświetli nam się tablica opublikowanych treningów) i szablony html, które zostaną wczytane w miejsce dyrektywy `ng-view` w pliku `index.html`.

1. `<div ng-view></div>`

Komunikacja z serwerem, interceptory, pobieranie danych

Funkcje dla wszystkich użytkowników

Funkcje dla trenera

6. Rezultaty projektu

7. Podsumowanie

7.1. Wnioski

7.2. Perspektywy

7.3. Wdrożenie systemu na serwerze produkcyjnym

8. Bibliografia

1. <https://angularjs.org/>
2. <http://bower.io/>
3. <http://angular-ui.github.io/angular-google-maps>
4. *AngularJS*, Brad Green, Shyam Seshadri, tłum. Robert Górczyński, wyd. Helion 2014
5. Winiarski R., Przewęda R., Wit B., Jegier A.: Sport dla wszystkich, TKKF Warszawa 1995
6. <http://www.redmine.org/>
7. Version Control with Git, 2nd Edition, Joe Loeliger, Matthew McCullough, tłum. Zdzisław Płoski, wyd. Helion 2012
8. Rebecca Murtagh, Mobile Now Exceeds PC: The Biggest Shift Since the Internet Began,
<http://searchenginewatch.com/article/2353616/Mobile-Now-Exceeds-PC-The-Biggest-Shift-Since-the-Internet-Began>

9. Załączniki

9.1. Wykaz ilustracji

9.2. Wykaz listingów

9.3. Podział prac

Marcin Olszewski

Odpowiedzialny za konfigurację i administrację systemu zarządzania projektem Redmine, pełnił rolę koordynatora zespołu. Administrator repozytorium GitHub projektu. Przygotował projekt architektury, dokonał wyboru technologii oraz zaimplementował aplikację dla trenera.

Sebastian Miałkowski

Mateusz Pakulski

Paweł Mazurek

9.4. Instrukcja instalacji i konfiguracji systemu

Programowanie natywne w systemie Android

Marcin Olszewski

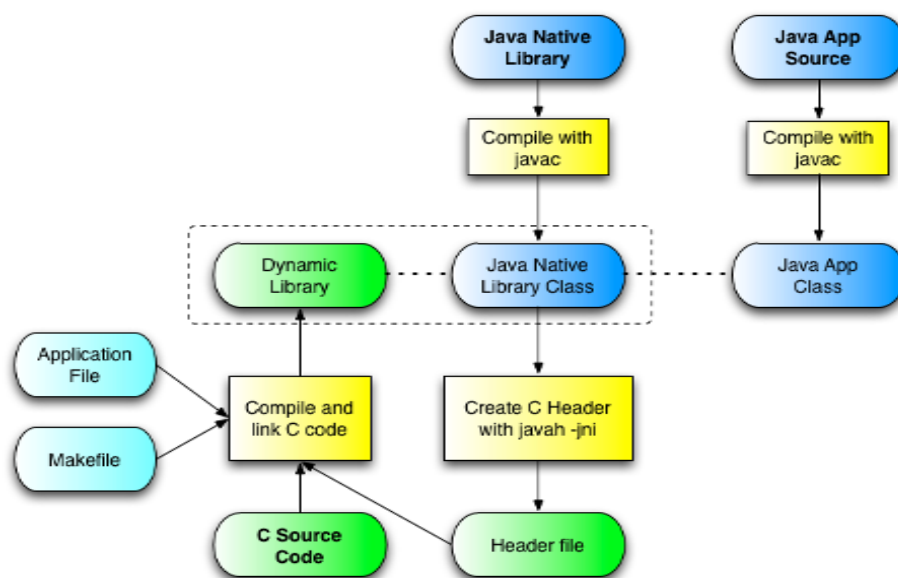
1. Cel ćwiczenia

2. NDK(Native Development Kit)

3. Kiedy stosować NDK

4. JNDI

5. Kompilacja kodu źródłowego C dla systemu Android



6. Zadania do samodzielnego wykonania

1. Kompilacja przykładowego projektu HelloWorld dostarczonego z NDK.
2. Implementacja własnej biblioteki języka C i wywołanej z poziomu aplikacji Android funkcji wyliczającej n-tą wartość ciągu fibonachiego.
3. Implementacja własnej biblioteki języka C jako wrappera dla istniejącej natywnej biblioteki języka C.