



**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: PAWEŁ MAZUREK
Nr albumu: 137342
Studia pierwszego stopnia
Forma studiów: stacjonarne
Kierunek studiów: Informatyka
Specjalność/profil: -

Imię i nazwisko studenta: PAWEŁ CICHOWSKI
Nr albumu: 137259
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: -

PROJEKT DYPLOMOWY INŻYNIERSKI

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

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

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

Data oddania projektu 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 przywłaszczeniem 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.

Śledzenie postępów zawodników, odbywać się będzie poprzez aplikację Web, opartą o *Framework AngularJS*. Trener, oprócz analizy danych z treningu takich jak, aktualne położenie zawodnika, czy jego prędkość oraz dystans, 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* i wykorzystująca bazę danych *PostgreSQL*. To właśnie 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 zatem z *Google Maps API*, które daje nam szerokie możliwości wykorzystania map. Dzięki temu rozwiązaniu będziemy mieli dostęp do aktualizowanej bazy map z całego świata.

Chcielibyśmy, aby efekt naszej pracy, czyli 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.

Abstract

*Streszczenie w języku angielskim
wraz ze słowami kluczowymi
oraz określoną dziedziną nauki i techniką.*

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 istniejących rozwiązań wspomagania treningu biegowego	9
3. Analiza wymagań i projekt funkcjonalny	9
3.1. Aplikacja mobilna	9
3.2. Serwer.....	10
3.3. Aplikacja trenera	10
4. Architektura systemu oraz przegląd technologii	11
4.1. Ogólna architektura całego systemu.....	11
4.2. Aplikacja mobilna	11
4.3. Serwer.....	11
4.4. Aplikacja trenera	11
5. Implementacja aplikacji mobilnej	14
6. Implementacja serwera	14
7. Implementacja aplikacji WWW	14
8. Opis wykorzystanych algorytmów	14
9. Wdrożenie systemu na serwerze katedralnym	14
10.Podsumowanie	14
10.1.Wnioski	14
10.2.Perspektywy	14
11. Bibliografia	15

1. Wstęp i cel pracy

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 przesyłania 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
- wykreślnie 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		2014-11-08 20:15
75	Funkcja	Nowy	Normalny	Akceptacja trenera		2014-11-08 20:14
74	Funkcja	Nowy	Normalny	Wybranie trasy do treningu		2014-11-08 20:12
73	Funkcja	Nowy	Normalny	Wyświetlanie poleceń treningu po zalogowaniu		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	Pominięcie 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).

Przepływ pracy

Kopia Podsumowanie

Przejścia między statusami

Uprawnienia do pól

Zaznacz rolę i typ zadania do edycji przepływu pracy:

Rola: Kierownik

Typ zadania: Funkcja

Edytuj

☒ Wyświetlaj tylko statusy używane przez ten typ zadania

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 zadań 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 – zadanie związane z dokumentacją, np. *Spis treści*

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

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

2. Przegląd istniejących rozwiązań wspomagania treningu biegowego

Opis istniejących rozwiązań z wyraźnym zaznaczeniem ich zalet i wad. Trzeba zwrócić uwagę na to, jakie wady istniejącego oprogramowania rozwiążemy w naszym produkcie.

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

3.3. Aplikacja trenera

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

Zgodnie z założeniami, wykonanymi 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.

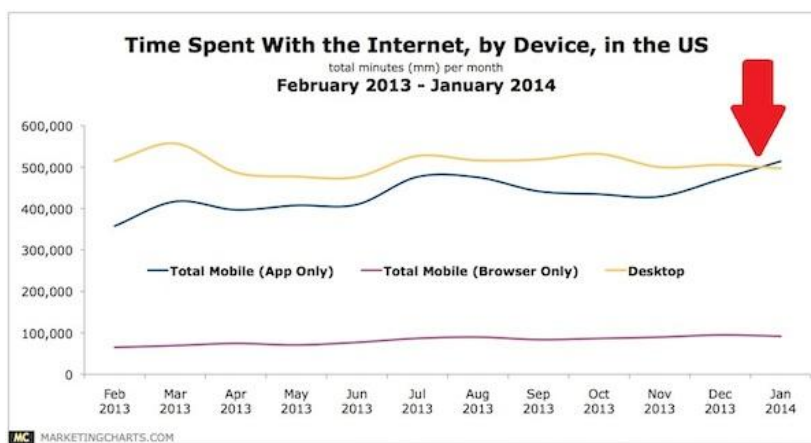
Zgodnie z założeniem implementacji architektury MVC, w aplikacjach AngularJS widokiem jest DOM(*ang. Document Object Model*) – obiektowy model dokumentu, kontrolerami są klasy JavaScript, a dane modelu przechowywane są we właściwościach obiektu. Dzięki tak dużej separacji warstw, możliwe staje się dokładne 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/>)

Przygotowując aplikacje dla użytkowników, musimy zadbać nie tylko o to, aby spełniały swoje funkcje. Powinniśmy się także dobrze zastanowić nad tym kto i na jakich urządzeniach będzie korzystał z naszych stron, serwisów czy systemów.

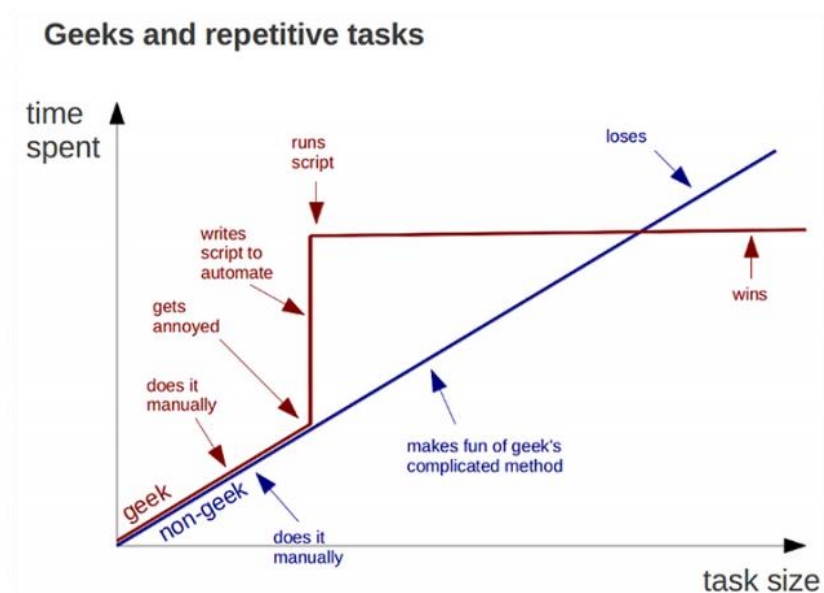


<http://www.engeniusweb.com/wp-content/uploads/2014/07/time-spent-on-internet-by-device-in-us.jpg>

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/>)

5. Implementacja aplikacji mobilnej

6. Implementacja serwera

7. Implementacja aplikacji WWW

NitrousIO i struktura projektu

Uwierzytelnianie z tokenem

Uwierzytelnianie Google+

Szablony i dyrektywy

Routing

Komunikacja z serwerem, interceptory, pobieranie danych

Funkcje dla wszystkich użytkowników

Funkcje dla trenera

Google Maps API



8. Opis wykorzystanych algorytmów

9. Wdrożenie systemu na serwerze katedralnym

10. Podsumowanie

10.1. Wnioski

10.2. Perspektywy

11. 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