WOJSKOWA AKADEMIA TECHNICZNA

**im. Jarosława Dąbrowskiego**

#### WYDZIAŁ CYBERNETYKI



## PRACA DYPLOMOWA

### STACJONARNE STUDIA I°

Temat: **Aplikacja wspierająca planowania tras przewozu ładunków w firmie transportowej.**

|  |  |
| --- | --- |
| Autor: | Promotor pracy: |
| **Dawid Zaleski** | **dr inż. Stefan Rozmus**  Opiekun pracy:  **mgr inż. Tomasz Gumkowski** |

**Warszawa 2019**

**OŚWIADCZENIE**

*„Wyrażam zgodę na udostępnianie mojej pracy przez Archiwum WAT”.*

Dnia ........................ .................................

(podpis)

*Pracę przyjąłem*

*promotor pracy*

*dr inż. Stefan Rozmus*

|  |  |  |  |
| --- | --- | --- | --- |
| **WSTĘP** | | **.........................................................................................................................................** | **4** |
| **I.** | **ZAKRES I TECHNOLOGIE……………………………………………......................................................** | | **5** |
|  | 1.1. | Aplikacja wspierająca planowanie tras..................................................................... | 5 |
|  | 1.2. | Aplikacja webowa………….......................................................................................... | 6 |
|  | 1.3 | Wykorzystane technologie…………………………………………………………………………………… | 7 |  |
|  | 1.4. | Uzasadnienie wyboru tematu................................................................................... | 9 |
| **II.** | **PRZEGLĄD ISTNIEJĄCYCH ROZWIĄZAŃ ............................................................................** | | **10** |
|  | 2.1. | Google Maps.……....................................................................................................... | 10 |
|  | 2.2. | Mapbox…….…………………………................................................................................... | 12 |
|  | 2.3 | MapQuest................................................................................................................. | 14 |
|  | 2.4 | Bing Maps……………………………………………………………………………………………………………. | 15 |
|  | 2.5 | Porównanie............................................................................................................... | 16 |
| **III.** | **DOKUEMENTACJA ANALITYCZNA.....................................................................................** | | 18 |
|  | 3.1 | Wymagania funkcjonalne..……………………………………………………................................. | 18 |
|  | 3.2 | Wymagania pozafunkcjonalne…............................................................................... | 21 |
|  | 3.3 | Przypadki użycia…………………………………………………………………………………………………… | 22 |
|  | 3.4 | Diagram przypadków użycia………………………………………………………………..………………. | 23 |
|  | 3.6 | Słownik pojęć………………………………………………………………………………………………………. | 24 |
| **IV.** | **ARCHITEKTURA……………………….......................................................................................** | | **27** |
|  | 4.1. | Diagram klas………………….......................................................................................... | 27 |
|  | 4.2 | Diagram czynności………………………………………………………………………………………………. | 28 |
|  | 4.2. | Diagram pakietów…………………................................................................................. | 29 |
|  | 4.3. | Diagram komponentów……...................................................................................... | 30 |
|  | 4.4 | Diagram wdrożenia……………………………………………………………………………………………… | 31 |
| **V.** | **IMPLEMENTACJA.............................................................................................................** | | 33 |
|  | 6.1. | Baza danych MySQL…….……………………………………………............................................. | 33 |
|  | 6.2. | Backend………………………............................................................................................ | 35 |
|  | 6.2.1 | Pobieranie lokalizacji z wybranego API…………………………………………………………………. | 36 |
|  | 6.2.2 | Wyliczanie odległości między punktami……………………………………………………………….. | 37 |
|  | 6.2.3 | Wyznaczanie tras…………………………………………………………………………………………………. | 37 |
|  | 6.3. | Frontend…………………….............................................................................................. | 37 |
|  | 6.3.1 | Zaprojektowanie strony głównej…………………………………………………………………………… | 38 |
|  | 6.3.2 | Pobieranie danych z backendu……………………………………………………………………………… | 39 |
|  | 6.3.3 | Zaprojektowanie kluczowej strony………………………………………………………………………… | 40 |
|  | 6.3.4 | Wyświetlanie trasy na mapie………………………………………………………………………………… | 41 |
| **VIII.** | | **TESTOWANIE……………………………………………………………………………………………………………..** | 44 |
|  | 8.1. | Rodzaje testów……………........................................................................................... | 44 |
|  | 8.2. | Przeprowadzone testy ............................................................................................. | 45 |
|  |  |  |  |
| **PODSUMOWANIE ........................................................................................................................** | | | 48 |
| **BIBLIOGRAFIA .............................................................................................................................** | | | 49 |
| **SPIS RYSUNKÓW ..........................................................................................................................** | | | 51 |

**WSTĘP**

Świat rozwija się coraz szybciej, często wychodzą nowe technologie, które mogą zmienić wiele i sprawić, że konkurencja urośnie szybciej niż by wcześniej się spodziewało. Coraz więcej firm z wielu różnych dziedzin zastanawia się nad informatyzacją swojego biznesu, dzięki czemu szeroko rozumiany segment informatyki może tym prężniej się rozwijać.

Nie inaczej jest w logistyce. Nawet najmniejsze firmy i kurierzy korzystają z ułatwień technologicznych jakimi są aplikacje, dzięki którym nie muszą zastanawiać konkretnie nad trasą, ponieważ optymalny wybór zostanie dokonany za nich. Jednak to jest najniższy szczebel logistyki i im większa firma, tym bardziej rozbudowanych oraz lepiej dostosowanych potrzeba rozwiązań, rozwiązań, które sprawią, że praca będzie nie tylko znacznie prostsza, ale też łatwiej reagować na wszelkie zmiany ze strony konkurencji, ale też technologicznego świata oraz bez problemy kontrolować co się dzieje w poszczególnych etapach.

W mojej pracy inżynierskiej skupię się na jednym z takich etapów w firmie transportowej, którym jest wyznaczanie tras przewozu ładunków. Mogłoby się zdawać, że nie jest to żadnym problemem, by robić to wszystko ręcznie, ale prawda jest taka, że można ten proces znacznie ułatwić, a także sprawić, że będzie można go ukończyć w kilka minut za pomocą kilkunastu kliknięć.

**I ZAKRES I TECHNOLOGIE**

* 1. **Aplikacja wspierająca wyznaczanie trasy**

1. **Problem wyznaczania trasy w przeszłości**

Niemalże codziennie każdy człowiek ma do czynienia z problemem jakim jest wyznaczanie trasy. Jeszcze kilkanaście lat temu każda większa podróż wiązała się z godzinami spędzonymi nad mapą i co jakiś czas pytaniem przechodniów o to czy mogą nam pomóc w znalezieniu konkretnego miejsca.

Na szczęście w ciągu ostatnich kilku lat wiele się zadziało w technologicznym świecie. Najpierw zostały pokazane komórki z dużym ekranem, a kolejne następujące po tym zdarzenia, można powiedzieć, że już wydarzyły się lawinowo. Obecnie ciężko znaleźć człowieka, który nie posiada smartfona i większość z nas prawdopodobnie nie potrafi sobie wyobrazić sobie wyjścia z domu bez tego urządzenia. Nawet portfel stracił na ważności przy tak ważnym kawałku technologii, który może też go całkowicie zastąpić. Oczywiście to wszystko wiąże się też z wieloma niebezpieczeństwami, ale nie o tym będzie w tym rozdziale.

1. **Obecne rozwiązania**

Rozwój sprawił, że męczenie się z wyszukiwaniem miejsc i wyznaczaniem tras przy pomocy fizycznych map stało się w większości przypadków i jedynie wspominkiem naszych rodziców albo naszymi własnymi nostalgicznymi wspomnieniami, do których raczej już nie chciałoby się wracać. Obecnie smartfon pozwala każdemu użytkownikowi znaleźć każde interesujące go miejsce, a także na bieżąco informować go o wszelakich zmianach trasy, a także od samego początku do samego końca wskazywać drogę.

Dla większości z nas ta wbudowana we wszystkie smartfony aplikacja z systemem Android, jaką jest Google Maps, całkowicie starczy niezależnie od miejsca, w które zmierzamy. Jednak problem potrafi się już robić, gdy chcemy wybrać dłuższą trasę, która składa się z większej liczby punktów, które są koniecznymi elementami danej podróży czy też transportu. Tym właśnie będzie zajmowała aplikacja.

1. **Przedstawienie mojej aplikacji**

Moja aplikacja nie pomaga w wyznaczaniu konkretnej trasy między punktami A i B, a w wyznaczaniu znacznie większej trasy złożonej z mnóstwa punktów kluczowych. Większość aplikacji, z którymi się spotkałem, z których miałem przyjemność korzystać, w przypadku właśnie takiego problemu funkcjonalności miała znacznie ograniczone. Na ogół albo można wybrać jedynie dwa punktów, a w przypadku wyboru większej liczby punktów to użytkownik wybiera w jakiej kolejności mają one się znajdować w trasie, a nie aplikacja dokonywała optymalnego wybory, dlatego postanowiłem się tym zająć.

Co prawda jednocześnie moja aplikacja nie będzie oferowała szczegółowych tras, ale za to ogólnie będzie lepsza do wyznaczania większych tras. Co prawda mam w zamiarze wykonać rozbudowaną wersję tej aplikacji, w której nie tylko kolejność punktów będzie się zgadzała, ale także trasy między nimi będą wyznaczane nie w linii prostej, a będą konkretnie wyznaczane, ale jestem zdania, że na początek należy skupić się na podstawach i dopiero gdy te elementy zostaną wykonane, to można zmienić swój priorytet.

1. **Wybór punktów kluczowych trasy**

Dla użytkownika końcowego w mojej aplikacji będzie widoczny prosty interfejs, gdzie będzie mógł wybrać interesujące go punkty kluczowe. W tej pracy inżynierskiej mowa konkretnie o firmie transportowej, więc tymi punktami będą magazyny należące do tej firmy. Wybór będzie można dokonywać przy pomocy pola kombi, gdzie będzie można wpisać konkretne miejsce albo wybrać element dostępny na liście.

Wybór dwóch magazynów jest obligatoryjny, ponieważ pierwszy z nich jest tym skąd ma się zacząć transport, a drugi oznacza gdzie on się kończy. Punkt początkowy oraz końcowy mogą być tym samym magazynem. Poza tym można też wybrać więcej punktów kluczowych, które znacznie rozszerzą trasę i póki co nie określiłem żadnego limitu, ale ostatecznie prawdopodobnie będzie trzeba ze względu na to, że jest to problem wykładniczy i przy kilkudziesięciu punktach kluczowych obliczenia mogą się wykonywać znacznie zbyt długo, by użytkownik był zadowolony z czasu czekania.

1. **Wyznaczanie trasy**

Gdy już zostaną wybrane miejsca można procedować do samego wyznaczania trasy, tutaj pojawia się tzw. Problem komiwojażera. Jest to problem wykładniczy, więc od chwili temat został zatwierdzony myślałem o tym jak uniknąć porównywania odległości między każdym z wybranych punktów. Zdecydowałem, że oprócz punktów kluczowych, będą też pomocnicze, które będą wspomagały wyznaczanie tras. Wszystkie te punkty będą odpowiednio opisane w bazie danych. Pozwolą mi one na dokładniejsze podzielenie mapy na węzły i zamiast porównywać wszystkie punkty kluczowe ze wszystkimi, mógłbym w bazie danych oznaczyć, który z węzłów łączy się z innymi, co by znacznie skróciło pracę algorytmu.

https://cs.stackexchange.com/questions/57860/travelling-salesman-problem-with-unknown-shortest-paths-between-nodes

Opis wygenerowany automatycznie

Rysunek 1. Problem komiwojażera - cs.stackexchange.com

Gdy już zostanie wyznaczona optymalna trasa, to na stronie pojawi lista ze wszystkimi wybranymi nazwami magazynów w kolejności w jakiej powinna się odbyć trasa. Poza tym dla ułatwienia wyobrażenia sobie tego użytkownik zauważy też na stronie mapę, na której będą oznaczone wszystkie wybrane magazyny wraz z nazwą oraz liczbą, która będzie oznaczała, który w kolejności jest ten magazyn na trasie.

1. **Punkty kluczowe i pomysł na rozszerzenie**

Należy jeszcze wspomnieć w trochę bardziej rozbudowany sposób o punktach kluczowych wybieranych przez użytkownika i samym transporcie. Każdy z punktów będzie w bazie danych miał kolumny, które będą opisywały czy można się do niego dostać dzięki pojazdom lądowym, morskim lub powietrznym. W sposób będzie podzielony transport i związane z tym atrybuty – prędkość i koszt. Są one ważne ze względu na to, że wybór trasy chciałbym oprzeć nie tylko na wyznaczaniu najmniejszej odległości, a też na daniu wybory użytkownikowi, ponieważ może akurat ważniejsze jest dla niego, by koszt transportu był jak najniższy albo czas w jakim transport przebędzie trasę będzie najmniejszy.

Poza tym ten projekt można by było rozszerzyć w taki sposób, że w magazynach też będą dane surowce i wybierałoby się jedynie miejsce, do którego mają dotrzeć dane surowce i byłaby wyznaczona trasa do najbliższego miejsca, w którym znajdują się te surowce, która są potrzebne. Jednak to jedynie dodatkowy pomysł na przyszłość, więc najpierw będę musiał wykonać podstawowe funkcjonalności na stronie, a dopiero następnie myślał o dodatkach.

* 1. **Aplikacja webowa**
* Czym jest aplikacja webowa?

Aplikacja internetowa (ang. web application), zwana również aplikacją webową – program komputerowy, który pracuje na serwerze i komunikuje się poprzez sieć komputerową z hostem użytkownika komputera z wykorzystaniem przeglądarki internetowej użytkownika, będącego w takim przypadku interaktywnym klientem aplikacji internetowej.

W pracy aplikacji internetowej musi pośredniczyć serwer WWW. Do przygotowania samej aplikacji używa się różnych mechanizmów (np. CGI, JSP, ASP.NET) i języków (np. PHP, Java, C#), jak również serwerów aplikacji. Mechanizm prezentacji danych w przeglądarce określa się czasem mianem cienkiego klienta.

Przykładem aplikacji internetowej jest mechanizm edycji treści encyklopedii Wikipedia. Inne witryny WWW należące do tej kategorii to np. serwis aukcyjny Allegro czy księgarnia internetowa Merlin.

* Aplikacja webowa, a strona internetowa

Jaka jest różnica między tymi dwoma pojęciami? Często wydaje się ona niewielka, wiele zależy od naszej perspektywy czy samej semantyki. Wielu z nas użyje słowa strona internetowa dla określenia wszystkich tworów posiadających domenę i istniejących w sieci. Jeśli natomiast zapytasz kogoś kto pracuje w IT, z pewnością tłumaczenia kilku osób będą się od siebie różniły, jednak ich główna myśli sprowadza się do dwóch zdań:

1. Strona internetowa ma charakter informacyjny.
2. Aplikacja webowa ma charakter interaktywny.

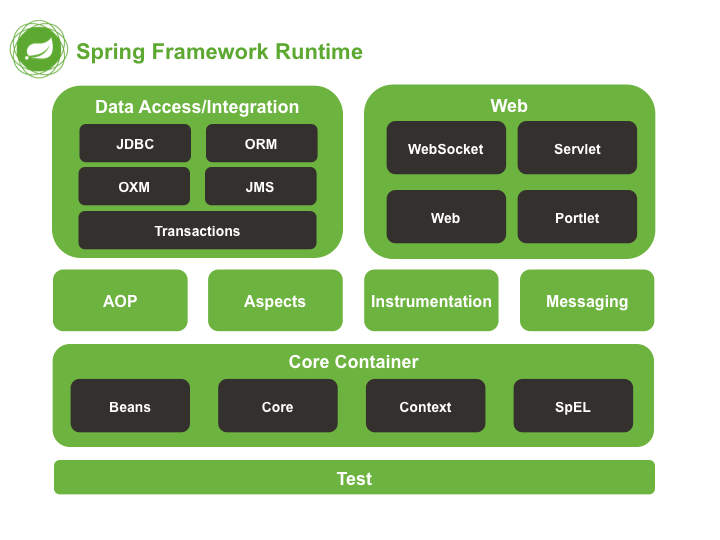
Mając na uwadze dwa powyższe zdania dokonajmy przykładu. Powiedzmy, że wchodzisz na stronę internetową teatru, jeśli znajdujesz na niej statyczne informacje o godzinach działaniu teatru, aktualnych spektaklach i ich godzinach to masz do czynienia ze stroną internetową, jeśli natomiast będziesz mógł dodatkowo zarezerwować/kupić bilet, wybrać dla siebie odpowiednie miejsce, to wiedz, że jesteś w aplikacji internetowej. Aby wytłumaczyć to jeszcze jaśniej, aplikacja internetowa, to często taka strona internetowa, którą użytkownik może kontrolować.

Technologicznie natomiast stronę internetową określa się jako coś, co jest na statycznym HTML-u i CSS-ie. Przy wykorzystaniu innych technologii mamy już do czynienia z aplikacją internetową. W takim wypadku dużego problemu semantycznego mogą dostarczyć nam strony internetowe o charakterze informacyjnym, stworzone na popularnych CMS-ach, które wykorzystują także inne technologie, niż tylko HTML i CSS, dlatego warto zwracać również uwagę na wyżej opisane kryterium użyteczności.

* 1. **Wykorzystane technologie**
* Spring Framework

Jest to szkielet tworzenia aplikacji (ang. application framework) w języku Java dla platformy Java Platform, Enterprise Edition (aczkolwiek istnieje też wersja dla środowiska .NET).

Spring Framework powstał na bazie kodu opublikowanego w książce Roda Johnsona Design and Development (​ISBN 0-7645-4385-7​). Pozytywny odzew czytelników sprawił, że był on dalej rozwijany przez autorów (m.in. Johnsona, Juergena Hoellera), którzy postanowili założyć firmę Interface21, świadczącą usługi konsultingowe związane z tym szablonem.

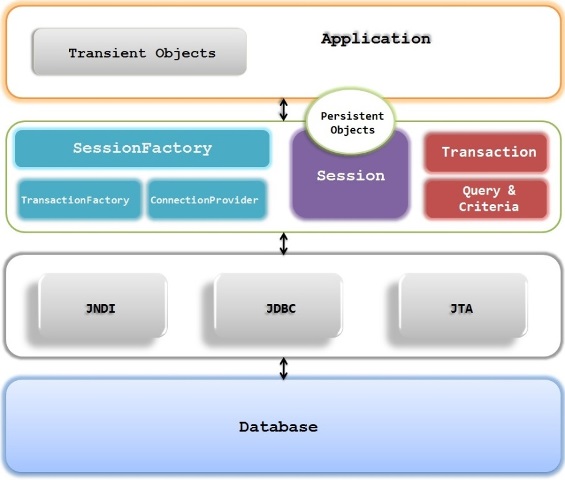


Rysunek 2: Architektura Spring Framework, źródło: docs.spring.io

Spring Framework może być rozważany jako zbiór pomniejszonych szablonów lub "szablon zbudowany z szablonów". Większość z tych szablonów została zaprojektowana aby pracować niezależnie, aczkolwiek użycie ich razem zapewnia większą funkcjonalność. Szablony te można podzielić ze względu na podstawowe komponenty przedstawione na rysunku powyżej.

* Hibernate

Jest to framework do realizacji warstwy dostępu do danych (ang. persistence layer). Zapewnia on przede wszystkim translację danych pomiędzy relacyjną bazą danych a światem obiektowym (ang. O/R mapping). Opiera się na wykorzystaniu opisu struktury danych za pomocą języka XML, dzięki czemu można rzutować obiekty, stosowane w obiektowych językach programowania, takich jak Java bezpośrednio na istniejące tabele bazy danych. Dodatkowo Hibernate zwiększa wydajność operacji na bazie danych dzięki buforowaniu i minimalizacji liczby przesyłanych zapytań. Jest to projekt rozwijany jako open source. Głównym inicjatorem oraz liderem projektu jest Gavin King.



Rysunek 3. Architektura Hibernate, źródło : HowToDoInJava

* Angular

Angular – otwarty framework i platforma do tworzenia SPA, napisany w języku TypeScript i wspierany oraz rozwijany przez Google. Angular początkowo miał być wersją 2 popularnego frameworku AngularJS, jednak decyzje projektowe sprawiły, że Google zdecydował się wydać go jako osobny byt, m.in. ze względu na brak kompatybilności wstecznej oraz prostej ścieżki aktualizacji aplikacji napisanych w AngularJS do Angular 2. Angular wydany jest na licencji MIT.

* Oracle baza danych

MySQL rozwijany jest przez firmę Oracle. Wcześniej przez większość czasu jego tworzeniem zajmowała się szwedzka firma MySQL AB. MySQL AB została kupiona 16 stycznia 2008 roku przez Sun Microsystems, a ten 27 stycznia 2010 roku przez Oracle. W międzyczasie Monty Widenius (współtwórca MySQL) stworzył MariaDB – forka (alternatywną wersję) opartego na licencji GPL. MariaDB jest oparta na tym samym kodzie bazowym co MySQL i dąży do utrzymania kompatybilności z jej poprzednimi wersjami.

MySQL był pisany raczej z myślą o szybkości niż kompatybilności ze standardem SQL – przez dłuższy czas MySQL nie obsługiwał nawet transakcji, co było zresztą głównym argumentem przeciwników tego projektu. MySQL obsługuje większą część obecnego standardu ANSI/ISO SQL (tj. SQL:2003). Wprowadza również swoje rozszerzenia i nowe elementy języka[4].

W wersji 5 dodano m.in.:

* procedury składowane (ang. stored procedures) – obecne od wersji 5.0,
* wyzwalacze (ang. triggers) – obecne od wersji 5.0.2
* widoki
* kursory – obecne od wersji 5.1
* partycjonowanie tabel – od wersji 5.1
* harmonogram zadań – od wersji 5.2
* co zbliża najnowsze wersje MySQL do PostgreSQL pod względem funkcjonalności.
  1. **Uzasadnienie wyboru tematu**

Jeśli chodzi o wybór tematu pracy inżynierskiej to pierwsza myśl związana była przede wszystkim z narzędziami, których przede wszystkim chciałbym się nauczyć. Dzięki studiowaniu informatyki na Wojskowej Akademii Technicznej nauczyłem się chociażby podstaw C/C++ i Javy i jeśli chodzi o backend, to właśnie ten język bardzo mi się spodobał. Jest znacznie przyjemniejszy od C++ i wciąż się prężnie rozwija, a także raczej nie traci na popularności. Chciałbym się nauczyć przy okazji tego tematu korzystania ze Springa, a także Hibernate’a, z którymi mam już doświadczenie jest ono niezbyt bogate.

Poza tym chciałbym też rozwijać się, jeśli chodzi o frontendową część aplikacji, która w moim przypadku ma być wykonana w Angularze, którego chciałbym poznać znacznie lepiej. Już miałem z nim styczność przy jednym projekcie i chciałbym wciąż poszerzać tę wiedzę, poznawać biblioteki z nim związane (chociażby Material i Grid), a także mieć styczność z TypeScript, HTML i CSS większą niż jedynie wiedzę, że takie narzędzia istnieją.

Poza chęcią nauczenia się podanych wyżej narzędzi interesuje mnie ogółem temat przemieszczania się ludzi i towarów, a przy tym widzę jak niewiele jest stron, które mogłyby to odpowiednio ułatwić. Istnieje taka strona jak jakDojadę, która na pierwszy rzut oka spełnia wszystkie wymagania, ale przy niewielkich trasach potrafi przedstawić trasę daleką od optymalnej. Jest jeszcze chociażby Google Maps, które działa coraz lepiej i obecnie dodawanie do niego jest mnóstwo funkcji, ale na niższym poziomie szczegółowości, przy wyborze więcej niż kilku punktów, to nie pomaga wybrać optymalnej trasy.

Na koniec, ale nie najmniej ważny jest fakt, że ten temat pozwoli zmierzyć mi się z dużym wyzwaniem, do którego powinny mnie przygotować trzy lata studiów. Nie będzie to łatwe, ale na pewno dużo się nauczę wykonując pracę inżynierską i mam nadzieję, że uda mi się ją zrobić nie tylko w podstawowej wersji, ale ostatecznie będę w stanie rozszerzyć w sposób przedstawiony w opisie sposobu działania aplikacji, a może nawet więcej pomysłów przyjdzie mi do głowy.

**BIBLIOGRAFIA**

* <https://pl.wikipedia.org/wiki/Aplikacja_internetowa>
* <https://smartbees.pl/blog/aplikacje-webowe-strony-internetowe-czym-sie-roznia>
* <https://cs.stackexchange.com/questions/57860/travelling-salesman-problem-with-unknown-shortest-paths-between-nodes>
* <https://docs.spring.io/spring/docs/4.2.x/spring-framework-reference/html/overview.html>
* <https://pl.wikipedia.org/wiki/Angular_(framework)>
* <https://javastart.pl/baza-wiedzy/frameworki/hibernate>
* ,,Inżynieria oprogramowania” Krzysztof Sacha
* ,,Angular’’ dokumentacja
* „R. Wilson: Wprowadzenie do teorii grafów, Warszawa 1998
* B. Korzan: Elementy teorii grafów i sieci. Metody i zastosowania, WNT, Warszawa 1978
* M. Chudy: Wybrane algorytmy optymalizacji, Warszawa 2001
* https://pl.wikipedia.org/wiki/Mapa
* „Nowoczesny język JavaScript” - Larry Ullman
* „Responsive Web Design. Projektowanie elastycznych witryn w HTML5 i CSS3” - Ben Frai
* Thinking in Java. Edycja polska. Wydanie IV – Bruce Eckel