POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: INFORMATYKA
SPECJALNOŚĆ: SYSTEMY I SIECI KOMPUTEROWE

PRACA DYPLOMOWA INŻYNIERSKA

Projekt strony internetowej do recenzji albumów muzycznych w technologii ASP.NET Core oraz grafowej bazie danych

Design of an internet website for music albums reviewing based on ASP.NET Core and graph database

AUTOR:
Marcin Kotas

PROWADZĄCY PRACĘ:
dr inż. Mariusz Topolski, W4/K2

OCENA PRACY:

WROCŁAW, 2019

Spis treści

Spis rysunków

Spis tabel

Skróty

OGC (ang. *Open Geospatial Consortium*)
XML (ang. *eXtensible Markup Language*)
SOAP (ang. *Simple Object Access Protocol*)
WSDL (ang. *Web Services Description Language*)
UDDI (ang. *Universal Description Discovery and Integration*)
GIS (ang. *Geographical Information System*)
SDI (ang. *Spatial Data Infrastructure*)
ISO (ang. *International Standards Organization*)
WMS (ang. *Web Map Service*)
WFS (ang. *Web Feature Service*)
WPS (ang. *Web Processing Service*)
GML (ang. *Geography Markup Language*)
SRG (ang. *Seeded Region Growing*)
SOA (ang. *Service Oriented Architecture*)
IT (ang. *Information Technology*)

Rozdział 1

Wprowadzenie

1.1. Cel i zakres pracy

Celem pracy jest implementacja strony internetowej pozwalającej dodawać recenzje albumów muzycznych, wykorzystując najnowsze technologie z tej dziedziny zgodne z obecnymi trendami.

Główny nacisk nałożony zostanie na stworzenie architektury stanowiącej solidną bazę do dalszego rozwijania aplikacji. Projekt wykorzystywać będzie tylko oprogramowanie open source.

Zakres pracy obejmuje:

- analiza
- opracowanie struktury aplikacji
- implementacja aplikacji
- testy

1.2. Układ pracy

Rozdział 2

Analiza wymagań projektowych

2.1. Przegląd istniejących rozwiązań

Na rynku obecnych jest wiele stron oferujących podobną funkcjonalność:

- Rate Your Music
- AllMusic
- Discogs

2.2. Wymagania funkcjonalne

1. System zawiera katalog albumów
2. Każdy album można oceniać oraz dodawać recenzję
3. Istnieje możliwość dodawania nowych albumów
4. Wyszukiwanie albumów korzysta z bazy zewnętrznego serwisu
5. Możliwość rejestracji i logowania

2.3. Wymagania niefunkcjonalne

1. Aplikacja powinna być obsługiwana przez obecne wersje przeglądarek
2.

2.4. Założenia projektowe

Całość aplikacji zaprojektowana zostanie ze wsparciem platformy Docker. Każda odrębna część systemu zamknięta będzie we własnym wirtualnym kontenerze:

1. strona internetowa typu Single-Page Application:
   Architektura SPA pozwala tworzyć strony, które w swoim działaniu bardziej przypominają tradycyjne aplikacje komputerowe. Podczas interakcji użytkownika ze stroną fragmenty widoku są dynamicznie odświeżane zamiast przeładowywania całej strony. Dodatkowo strona powinna przechowywać w pamięci zapytania do serwera aby minimalizować czas oczekiwania na dane.
2. serwer uwierzytelniający użytkowników:
   Strona internetowa działa po stronie klienta, przez co możliwa jest znaczna ingerencja w dane. Aby minimalizować zagrożenia, zaimplementowane zostanie uwierzytelnianie

użytkowników wykorzystujące bezpieczny protokół. Wykorzystany standard powinien zapewnić zarówno uwierzytelnianie jak i autoryzację.

3. serwer dostępowy do danych aplikacji:
   Grafowy dostęp do bazy danych zaimplementowany zostanie za pomocą technologii GraphQL. Serwer umożliwiać będzie pobieranie danych stosując zapytania w języku GraphQL. Dzięki temu relacje między danymi przedstawione są w postaci grafu, co pozwala formować skompilowane zapytania bez potrzeby tworzenia dedykowanych kontrolerów i modeli DTO po stronie API.
4. baza danych:
   Baza danych powinna umożliwiać przechowywanie relacji między obiektami w taki sposób, aby możliwe było reprezentowanie danych w postaci grafu.
5. serwer dostarczający odwrócone proxy:
   Z racji tego, że użytkownicy będą przekierowywani między główną stroną, a stroną do uwierzytelniania, adresy serwerów zarejestrowane będą w odwróconym proxy. Będzie to również główny punkt dostępu do aplikacji, który będzie kierował ruchem. Dodatkowo, będzie obsługiwał szyfrowanie przesyłanych danych protokołem HTTPS.

Rozdział 3

Projekt aplikacji

3.1. Strona internetowa

Do stworzenia strony zdecydowano się użyć frameworku Angular 8. Serwowana będzie ze środowiska uruchomieniowego Node.js. Projekt zbudowany jest z osobnych modułów dla każdej funkcjonalności. Widoki składane są z komponentów — są to elementy zawierające szablon HTML, style CSS oraz logikę i dane w klasie napisanej w języku typescript. Dodatkowo komponenty mogą korzystać z serwisów. Serwisy to specjalne klasy, które są wstrzykiwane jako zależności (ang *Dependency injection*). Zawierają się w nich dane oraz logika dzielone między wieloma komponentami. Nawigacja pomiędzy poszczególnymi widokami odbywa się przy pomocy dedykowanego routera, który przechwytuje nawigację przeglądarki. Dzięki temu nawet nawigacja do tyłu nie powoduje przeładowania całej strony tylko poszczególnych zmienionych elementów.

Do budowy widoków użyta zostanie biblioteka Angular Material, która zawiera często wykorzystywane elementy zbudowane zgodnie z oficjalną specyfikacją *Material design*.

Taka modularna architektura pozwala na wielokrotne używanie tych samych komponentów oraz na zachowanie czytelnej struktury kodu.

3.2. Serwis API

Projekt podzielony zostanie na trzy warstwy:

Api warstwa obsługująca zapytania GraphQL

Core warstwa domenowa definiująca modele encji oraz interfejsy, z których korzysta warstwa Api

Infrastructure warstwa implementująca interfejsy oraz obsługująca dostęp do bazy danych

Podział ten znany jest jako „czysta architektura” (ang. *Clean architecture*). Taki układ umożliwia podział projektu na warstwy, które mają zdefiniowane odpowiedzialności. Dzięki zachowaniu zasady odwrócenia zależności możliwe jest testowanie każdej funkcjonalności z osobna.

Na platformie .Net Core dostępne są dwie aktywnie rozwijane biblioteki implementujące GraphQL: *GraphQL .NET* oraz *Hot Chocolate*. Mimo znacznie mniejszej popularności wybrana została biblioteka *Hot Chocolate*, ponieważ prezentuje wiele możliwości automatyzacji generowania schematu GraphQL oraz analizowania zapytań.

Z racji tego, że GraphQL jest protokołem służącym do przesyłania danych, jest on zaimplementowany w najwyższej warstwie serwisu – Api. W związku z tym nie zawęża sposobu implementacji zapisu danych w warstwie infrastruktury, a więc nic nie stoi na przeszkodzie, aby użyć relacyjnej bazy SQL. Takie rozwiązanie umożliwia wykorzystanie zalet relacyjnych

baz - integralność i niezależność danych, jednocześnie oferując grafowy odczyt danych poprzez GraphQL.

3.3. Baza danych

Baza danych stworzona zostanie w systemie zarządzania relacyjną bazą danych PostgreSQL. Jest to open source'owe oprogramowanie, obecnie jeden z najlepiej rozwiniętych RDBMS-ów.

3.4. Uwierzytelnianie

Jako protokół uwierzytelniania wybrano OpenID Connect. Standard ten rozszerza OAuth2 (służący do autoryzacji) o warstwę identyfikacji użytkowników. Jest obecnie jednym z najbezpieczniejszych standardów uwierzytelniania. Zaimplementowany zostanie przy pomocy biblioteki *IdentityServer4* na platformie *ASP.Net Core 3.0*. Użytkownik po zalogowaniu otrzyma token JWT, który będzie zawierał cyfrową sygnaturę. Dzięki temu jakakolwiek ingerencja w jego strukturę sprawi, iż nie jego walidacja zakończy się niepowodzeniem.

Rozdział 4

Implementacja

4.1. Strona internetowa

4.1.1. Zastosowane technologie

Zgodnie z projektem, strona stworzona została w Angularze. Oprócz podstawowych bibliotek Wykorzystane zostały następujące biblioteki:

flex-layout — biblioteka stworzona przez zespół tworzący Angulara, umożliwiająca stworzenie responsywnego interfejsu. Dostarcza API, które pozwala na definiowanie struktury elementów HTML zależnej od rozmiaru ekranu. Dzięki temu interfejs automatycznie dostosowuje się np. do ekranu telefonu komórkowego.

apollo — wiodący klient GraphQL. Oprócz implementacji protokołu GraphQL, zapewnia również zarządzanie pamięcią podręczną (ang. *cache*) oraz stanem aplikacji (ang. *state management*). Dzięki temu wszystkie wszystkie odpowiedzi z serwisu Api są zapamiętywane, co pozwala na stworzenie aplikacji działającej szybko nawet przy słabym połączeniu z internetem.

oidc-client — biblioteka implementująca protokół OpenID Connect oraz OAuth2. Zapewnia klasy obsługujące proces logowania oraz zarządzania tokenami.

rxjs

Rozdział 5

Uwagi techniczne

5.1. Rysunki

W niniejszym szablonie numeracja rysunków odbywa się automatycznie według następujących reguł: rysunki powinny mieć numerację ciągłą w obrębie danego rozdziału, sam zaś numer powinien składać się z dwóch liczb rozdzielonych kropką. Pierwsza liczbą ma być numer rozdziału, drugą – kolejny numer rysunku w rozdziale. Przykładowo: pierwszy rysunek w rozdziale 1 powinien mieć numer 1.1, drugi – numer 1.2 itd., pierwszy rysunek w rozdziale 2 powinien mieć numer 2.1, drugi – numer 1.2 itd.

Rysunki powinny być wyśrodkowane na stronie wraz z podpisem umieszczonym na dole. Podpisy nie powinny kończyć się kropką. Czcionka podpisu powinna być mniejsza od czcionki tekstu wiodącego o 1 lub 2 pkt (w szablonie jest to czcionka rozmiaru `small`). Ponadto należy zachowywać odpowiedni odstęp między rysunkiem, podpisem rysunku a tekstem rozdziału. W przypadku korzystania z szablon odstępy te regulowane są automatycznie. Podpis i grafika muszą stanowić jeden obiekt. Chodzi o to, że w edytorach tekstu typu Office podpis nie scala się z grafiką i czasem trafia na następną stronę, osieracając grafikę. Korzystającym z niniejszego szablonu i otoczenia `\figure` takie osierocenie nigdy się nie zdarzy.

Do każdego rysunku musi istnieć odwołanie w tekście (inaczej mówiąc: niedopuszczalne jest wstawienie do pracy rysunku bez opisu). Odwołania do rysunków powinny mieć postać: „Na rysunku 3.3 przedstawiono...” lub „... co ujęto na odpowiednim schemacie (rys. 1.7)”. Jeśli odwołanie stanowi część zdania, to wtedy wyraz „rysunek” powinien pojawić się w całości. Jeśli zaś odwołanie jest ujęte w nawias (jak w przykładzie), wtedy należy zastosować skrót „rys.”. Jeśli do stworzenia obrazka wykorzystano jakieś źródła, to powinny one być zacytowane w podpisie tegoż rysunku.

Należy pamiętać o tym, że „rysunki” to twory nieżywotne. W związku z tym nie mogą ”pokazywać”. Dlatego „rysunek 1.1 pokazuje ...” jest stylistycznie niepoprawne. Zamiast tego zwrotu trzeba użyć „ na rysunku 1.1 pokazano ...”.

Rysunki można wstawiać do pracy używając polecenia `\includegraphics`. Zalecane jest, aby pliki z grafikami były umieszczane w katalogach odpowiadających numerom rozdziałów czy literom dodatków: `rys01`, `rysA` itd. Sposób wstawiania rysunków do pracy zademonstrowano na przykładze rysunków 5.1 i 5.2.

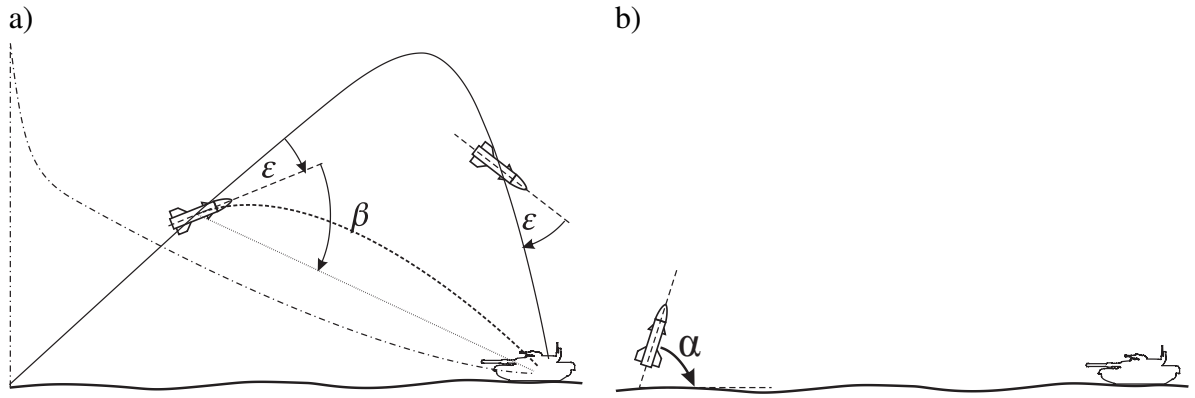
Listing 5.1: Kod źródłowy przykładów wstawiania rysunków do pracy

```
\begin{figure}[ht]
 \centering
  \includegraphics[width=0.3\linewidth]{rys05/kanji-giri}
 \caption{Dwa znaki kanji - giri}
 \label{fig:kanji-giri}
\end{figure}
```

```
\begin{figure}[htb]
 \centering
  \begin{tabular}{@{}ll@{}}
  a) & b) \\
  \includegraphics[width=0.475\textwidth]{rys05/alfa1} &
  \includegraphics[width=0.475\textwidth]{rys05/beta1}
  \end{tabular}
 \caption{Wyznaczanie trajektorii lotu rakiety:
 a) trzy podejścia, b) podejście praktyczne}
 \label{fig:alfabeta}
\end{figure}
```

義理

Rys. 5.1: Dwa znaki kanji – giri



Rys. 5.2: Wyznaczanie trajektorii lotu rakiety: a) trzy podejścia, b) podejście praktyczne

Grafiki wektorowe powinny być dostarczone w plikach o formacie pdf. Rozmiar strony w pliku pdf powinien być troszeczkę większy niż zamieszczona na nim grafika (proszę spojrzeć na przykłady grafik wykorzystanych w niniejszym szablonie). Chodzi o to, aby na rysunku nie pojawiała się niepotrzebna biała przestrzeń. Grafiki rastrowe (głównie zrzuty z ekranu bądź zdjęcia) powinny być dostarczane w plikach o formacie png z kompresją bezstratną. Zastosowanie kompresji stratnej, jak jpg, wprowadza niepotrzebne artefakty. Podobnie jak w przypadku grafik wektorowych, grafiki rastrowe nie powinny mieć białych marginesów.

Na rysunkach nie powinno stosować się 100% czarnego wypełnienia, bo robią się plamy przebijające się przez kartkę. Zamiast tego wypełnienie powinno być ok. 90% czerni.

Czcionka na rysunkach nie może być większa od czcionki wiodącej tekstu (jedyny wyjątek to np. jakieś nagłówki). Należy stosować czcionkę kroju Arial, Helvetica bądź tego samego kroju co czcionka dokumentu (`texgyre-termes`).

Jeśli na jednym rysunku pojawić się ma kilka grafik, to zamiast stosować `subfigure` lub inne otoczenia należy wstawić grafiki w tabelę, opisać ją indeksami a) i b), a potem odnieść się do tego w podpisie (rys. 5.2). Czasem pomaga w pozycjonowaniu rysunków użycie komendy: `\vtop{\vskip3ex\hbox{\includegraphics[width=0.475\textwidth]{nazwa}}}`

Na rysunkach nie wolno nadużywać kolorów oraz ozdobników (wiele narzędzi do tworzenia diagramów dostarcza grafikę z cieniowaniem, gradacją kolorów itp. co niekoniecznie przekłada się na czytelność rysunku).

Podczas rozbienia zrzutów z ekranu należy zadbać o to, by taki zrzut był czytelny po wydrukowaniu. Czyli aby pojawiające się literki były wystarczająco duże, a przestrzenie bez treści – relatywnie małe. Przystępując do robienia zrzutu trzeba odpowiednio wyskalować elementy na ekranie. Na przykład robiąc zrzut z przeglądarki FF najpierw należy wcisnąć CTR–0 (domyślne skalowanie), potem CTR–- (zmniejszenie skali o stopień). Potem dobrze jest zawęzić okno przeglądarki tak, by interesująca treść wypełniła je w całości. Jeśli na obserwowanej stronie jest zbyt dużo pustych obszarów, to należy je jakoś zawęzić (sterując wielkością okna przeglądarki lub aktywnymi elementami interfejsu użytkownika). Zrzut bowiem wcale nie musi być odzwierciedleniem 1:1 domyślnego układu obserwowanych elementów. Ważne jest, by na zrzucie z ekranu pokazać interesujący, opisywany fragment i żeby ten fragment był czytelny.

Czasem problemem jest tworzenie zrzutów z ekranu, gdy występują na nim dane wrażliwe. Istnieją dwa sposoby na radzenie sobie z tym problemem. Pierwszy polega na zastąpieniu w systemie danych danych rzeczywistych danymi testowymi – wygenerowanymi tylko do celów prezentacji. Zrzut robi się wtedy na bazie danych testowych. Drugi polega na wykonaniu zrzutu z ekranu, na którym pokazano dane rzeczywiste, i następnie zamianie tych danych już w pliku graficznym za pomocą odpowiedniego edytora (np. `gimp`). Czyli oryginalny zrzut z ekranu należy otworzyć w edytorze, a potem nadpisać oryginalny tekst własnym tekstem. Konieczne jest wtedy dobranie odpowiednich czcionek aby nie było widać wprowadzonych zmian.

> Uwaga: takie manipulowanie zrzutami jest usprawiedliwione jedynie w przypadku konieczności ochrony danych wrażliwych czy też lepszego pokazania wybranych elementów. Nie może to prowadzić generowania fałszywych rezultatów!!!

5.2. Wstawianie kodu źródłowego

Kod źródłowy można wstawiać jako blok tekstu pisany czcionką maszynową. Używa się do tego otoczenie `\lstlisting`. W atrybutach otoczenia można zdefiniować tekst podpisu wstawianego wraz z numerem nad blokiem, etykietę do tworzenia odwołań, sposób formatowania i inne ustawienia. Zaleca się stosowanie w tym otoczeniu następujących parametrów:

```
\begin{lstlisting}[label=list:req1,caption=Initial HTTP Request,
                   basicstyle=\footnotesize\ttfamily]
```

Szczególnie przydatne podczas wstawiania większej ilości kodu źródłowego jest zastosowanie parametru `basicstyle=\footnotesize\ttfamily`. Dzięki niemu zmniejsza się czcionka, a przez to na stronie można zmieścić dłuższe linijki kodu. Użycie tak zdefiniowanego parametru nie jest jednak sztywnym zaleceniem. Wielkość czcionki można dobierać do potrzeb.

Listing 5.2: Initial HTTP Request

```
GET /script/Articles/Latest.aspx HTTP/1.1
Host: www.codeproject.com
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml
User-Agent: Mozilla/5.0 ...
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US...
Accept-Charset: windows-1251,utf-8...
```

Można też sformatować kod bez stosowania numerowanego podpisu (wtedy nie zamieszcza się `caption` na liście atrybutów).

```
GET /script/Articles/Latest.aspx HTTP/1.1
Host: www.codeproject.com
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml
User-Agent: Mozilla/5.0 ...
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US...
Accept-Charset: windows-1251,utf-8...
```

Istnieje możliwość wstawiania kodu źródłowego w bieżącej linijce tekstu. Można to zrobić na kilka sposobów:

- korzystając z polecenia `\texttt` ustawiającego czcionkę maszynową, jak w przykładzie `tutaj` (efekt zastosowania komendy `\texttt{tutaj}`). Problemem jednak mogą okazać się znaki podkreślenia i inne znaki kontrolne.
- korzystają z otoczenia `\verb` zapewniającego wypisanie kodu czcionką maszynową jak w przykładzie `tutaj` (efekt zastosowania komendy `\verb|tutaj|`). Problemem jest to, że polecenie `\verb` nie potrafi łamać dłuższego tekstu.
- korzystając z polecenia `\lstin` umożliwiającego wypisanie kodu czcionką ustawianą w opcjach jak w przykładzie `tutaj` (efekt komendy `\lstset{basicstyle=\ttfamily}\lstinline{tutaj}`) lub `tutaj` (efekt komendy `\lstinline[basicstyle=\ttfamily]=tutaj=`).

5.3. Wykaz literatury oraz cytowania

Cytowania powinny być zamieszczane w tekście z użyciem komendy `\cite{}`. Jej argumentem powinien być klucz cytowanej pozycji (lub lista kluczy rozdzielonych przecinkiem bez spacji, jeśli takich pozycji w danym miejscu cytuje się więcej) jaki jest używany w bazie danych bibliograficznych (plik `dokumentacja.bib`). Po kompilacji `bibtex` i `pdflatex` w tekście pojawia się właściwy odsyłacz do pozycji w wykazie literatury (ujęty w kwadratowe nawiasy – zgodnie z tym, co definiuje styl `plabbrv.bst`), zaś w samym wykazie (rozdział Literatura) – zacytowana pozycja. Przykładem cytowania jest: „dobrze to opisano w pracach [2, 1]" (gdzie zastosowano komendę `\cite{JS07,SQL2}`).

Co do zawartości rekordów bibliograficznych - style bibtexowe potrafią „skracać" imiona (czyli wstawiać, jeśli taka wola, inicjały zamiast pełnych imion). Niemniej dobrze jest od razu przyjąć jakąś konwencję. Proponuje się, aby w rekordach od razu wstawiane były inicjały zamiast pełnych imion.

Niekiedy tytuły prac zawierają wyrazy z dużymi i małymi literami. Takie tytuły należy brać w podwójne nawiasy klamrowe, aby `bibtex` nie zamienił ich na postać, w której poza pierwszą literą pozostałe są małe.

Jeśli jakiś cytowany zasób pochodzi z Internetu, to jego rekord w pliku `bib` powinien wyglądać jak niżej.

```
@INPROCEEDINGS{SQL2,
  title={{A MySQL-based data archiver: preliminary results}},
  author={Bickley, M. and Slominski, Ch.},
  booktitle = {{Proceedings of ICALEPCS07}},
        month = oct,
        day = {15--19},
        year={2007},
  note={\url{http://www.osti.gov/scitech/servlets/purl/922267}
        [dostęp dnia 20 czerwca 2015]}
}
```

A to inny przykład rekordu danych bibliograficznych:

```
@TechReport{JS07,
        author = {Jędrzejczyk, J. and Śródka, B.},
        title  ={Segmentacja obrazów metodą drzew decyzyjnych},
        year = {2007},
        institution = {Politechnika Wrocławska, Wydział Elektroniki}
}
```

5.4. Indeks rzeczowy

Generowanie indeksu po trosze wygląda jak generowanie wykazu literatury – wymaga kilku kroków. Podczas pierwszej kompilacji `pdflatex` generowany jest plik z rozszerzeniem `*.idx` (zawierający „surowy indeks"). Następnie, bazując na tym pliku, generowany jest plik z rozszerzeniem `*.ind` zawierający sformatowane dane. Ten krok wymaga uruchomienia odpowiedniego narzędzia oraz zastosowania plik z definicją stylu `Dyplom.ist`. W kroku ostatnim dokonuje się kolejnej kompilacji `pdflatex` (dzięki niej w wynikowym dokumencie pojawi się Indeks rzeczowy). Domyślnie Indeks rzeczowy zostanie sformatowany w układzie dwukolumnowym.

Oczywiście aby to wszystko zadziałało w kodzie szablonu należy umieścić odpowiednie komendy definiujące elementy indeksu rzeczowego (`\index`) oraz wstawiające sformatowany Indeks rzeczowy do dokumentu wynikowego (`\printindex`). Więcej informacji o tworzeniu indeksu rzeczowego można znaleźć na stronie `https://en.wikibooks.org/wiki/LaTeX/Indexing`. Poniżej przedstawiono przykłady komend użytych w szablonie do zdefiniowania elementów indeksu rzeczowego:

- `\index{linia komend}` – pozycji główna.
- `\index{generowanie!-- indeksu}` – podpozycja.

Generowanie pliku `*.ind` można inicjować na kilka sposobów:

- poprzez wydanie odpowiedniego polecenia bezpośrednio w linii komend

  ```
  makeindex Dyplom.idx -t Dyplom.ilg -o Dyplom.ind -s Dyplom.ist
  ```

- poprzez odpalenie odpowiedniego narzędzia środowiska. Na przykład w `TeXnicCenter` definiuje się tzw. `output profiles`:

  ```
  makeindex "%tm.idx" -t "%tm.ilg" -o "%tm.ind" -s "%tm.ist"
  ```

  a samo generowanie pliku `*.ind` zapewni wybranie pozycji menu `Build/Makeindex`.
- korzystając z odpowiednio sparametryzowanych pakietów i komend wewnątrz kompilowanego dokumentu (czyli od razu przy okazji jego kompilacji).

  ```
  \DisemulatePackage{imakeidx}
  \usepackage[noautomatic]{imakeidx}
  % jeśli chcemy, by indeks by generowany automatycznie programem makeindex:
  %\usepackage[makeindex]{imakeidx}
  % a tak ponoć można przekazać opcje do programu generującego indeks:
  %\makeindex[options=-s podrecznik -L polish -M lang/polish/utf8]
  %\makeindex[options=-s podrecznik]
  \makeindex
  ```

  Niestety, `makeindex` jest narzędziem, które umieszcza część pozycji w grupie `Symbols`, a nie w grupach związanych z literkami alfabetu (w związku z czym indeksowany element zaczynający się od polskiej literki trafia do grupy `Symbols`, jak np. `\index{Światło}`. Jeśli chce się zamieszczać w indeksie symbole matematyczne, to dobrze jest to robić jak w następujacym przykładzie: `\index{$asterisk@$\ast$}` czy też `\index{c@$\mathcal{C}$}`, tj. dostarczając przy okazji klucz do sortowania. Lepiej w tym względzie radzą sobie inne

narzędzia, jak `texindy` lub `xindy` dostępne pod linuxem. Korzystając z nich uzyskuje się grupy polskich literek w indeksie rzeczowym (hasła zaczynające się od polskich literek już nie trafiają do grupy Symbols). Przykład polecenia wydanego z linii komend, w którym wykorzystano `texindy` zamieszczono poniżej (zakładamy kodowanie plików w UTF8, można dla niniejszego szablonu zmienić na cp1250):

```
texindy -L polish -M lang/polish/utf8 Dyplom.idx
```

To polecenie wygeneruje `Dyplom.ind` o zawartości:

```
\begin{theindex}
  \providecommand*\lettergroupDefault[1]{}
  \providecommand*\lettergroup[1]{%
      \par\textbf{#1}\par
      \nopagebreak
  }

  \lettergroup{G}
  \item generowanie
    \subitem -- indeksu, 27
    \subitem -- wykazu literatury, 27

  \indexspace

  \lettergroup{L}
  \item linia komend, 27

  \indexspace

  \lettergroup{Ś}
  \item \'Swiat\IeC {\l }o, 28

\end{theindex}
```

Aby mieć większą kontrolę automatyczne generowanie indeksu zostało w niniejszym szablonie wyłączone (indeks trzeba wygenerować samemu, wydając polecenie `makeindex` lub zalecane `texindy`).

5.5. Inne uwagi

Dobrym sposobem na kontrolę błędów występujących podczas kompilacji jest wstawiania linijki `\end{document}` w wybranym miejscu dokumentu. Jest to szczególnie przydatne w przypadkach, gdy błędy te są trudne do zidentyfikowania (gdy wygenerowane przez kompilator numery linii z błędami nie są tymi, w których błędy występują). Wystarczy wtedy przestawić wspomnianą linijkę do kolejnych miejsc, aż znajduję to miejsce, gdzie występuje problem.

Aby osiągnąć apostrofy maszynowe (czyli takie złożone z samych kresek) należy użyć polecenia `"{}jak tutaj{}"` (podwójny apostrof i podwójny apostrof z na wszelki wypadek umieszczonymi nawiasami klamrowymi, nawiasy są potrzebne z tej racji, iż podwójny apostrof przed niektórymi literkami zamienia je na literki z akcentami). W efekcie otrzymamy "jak tutaj". Jeśli natomiast apostrofy mają być drukarskie (czyli złożone z kropek i kresek), to należy użyć polecenia `,,jak tutaj''` (dwa pojedyncze przecinki i dwa pojedyncze apostrofy). W efekcie otrzymamy „jak tutaj”. Można też użyć znaków apostrofów odpowiednio zakodowanych „jak tutaj”, tylko że czasem trudno pisze się takie apostrofy w środowiskach kompilacji projektów latexowych.

Oto sposoby ustawienia odstępów między liniami:

- używając komendy `\linespread{...}` (akceptowalne), przy czym atrybutem tej metody jest współczynnik zależny od wielkości czcionki. Dla czcionki wiodącej 12pt odstęp półtora linii osiągnie się komendą `\linespread{1.241}`. Dla innych czcionek wiodących wartości tego parametru są jak w poniższym zestawieniu.

```
10pt 1.25 dla \onehalfspacing
     1.667 for \doublespacing,
                 ponieważ ,,basic ratio'' = 1.2
                (\normalfont posiada \baselineskip rozmiaru 12pt)
11pt 1.213 dla \onehalfspacing oraz 1.618 dla \doublespacing,
     ponieważ ,,basic ratio'' = 1.236
                (\normalfont posiada \baselineskip rozmiaru 13.6pt)
12pt 1.241 dla \onehalfspacing oraz 1.655 dla \doublespacing,
     ponieważsince ''basic ratio'' is 1.208
                (\normalfont has a \baselineskip of 14.5pt)
```

  Kłopot w tym, że raz ustawiony odstęp będzie obowiązywał do wszystkich czcionek (nie działa tu żadem mechanizm zmiany współczynnika w zależności od wielkości czcionki akapitu).
- używając pakietu `setspace` (niezalecane). Ponieważ klasa `memoir` emuluje pakiet `setspace`, w preambule dokumentu należałoby umieścić:

```
\DisemulatePackage{setspace}
\usepackage{setspace}
```

  a potem można już sterować odstęp komendami:

```
\singlespacing
\onehalfspacing
\doubelspacing
```

  Ten sposób pozwala na korzystanie z mechanizmu automatycznej zmiany odległości linii w zależności od wielkości czcionki danego akapitu.
- korzystając bezpośrednio z komend dostarczonych w klasie `memoir` (zalecane):

```
\SingleSpacing
\OnehalfSpacing
\DoubleSpacing
```

  Ten sposób również pozwala na korzystanie z mechanizmu automatycznej zmiany odległości linii w zależności od wielkości czcionki danego akapitu.

Na koniec jeszcze uwaga o rozmiarze pliku wynikowego. Otóż `pdflatex` generuje pliki `pdf`, które zazwyczaj mogłyby być nieco lepiej skompresowane. Do lepszego skompresowania tych plików można użyć programu `ghostscript`. Wystarczy w tym celu wydać komendę (pod windowsami):

```
gswin64 -sDEVICE=pdfwrite -dCompatibilityLevel=1.4 -dNOPAUSE -dQUIET -dBATCH
-sOutputFile=Dyplom-compressed.pdf Dyplom.pdf
```

Rozdział 6

Podsumowanie

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

6.1. Sekcja poziomu 1

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Nam id nulla a adipiscing tortor, dictum ut, lobortis urna. Donec non dui. Cras tempus orci ipsum, molestie quis, lacinia varius nunc, rhoncus purus, consectetuer congue risus.

6.1.1. Sekcja poziomu 2

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Sekcja poziomu 3

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Paragraf 4 Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

6.2. Sekcja poziomu 1

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Literatura

[1] M. Bickley, C. Slominski. A MySQL-based data archiver: preliminary results. *Proceedings of ICALEPCS07*, Paz. 2007. `http://www.osti.gov/scitech/servlets/purl/922267` [dostęp dnia 20 czerwca 2015].

[2] J. Jędrzejczyk, B. Śródka. Segmentacja obrazów metodą drzew decyzyjnych. Raport instytutowy, Politechnika Wrocławska, Wydział Elektroniki, 2007.

Dodatek A

Tytuł dodatku

Zasady przyznawania stopnia naukowego doktora i doktora habilitowanego w Polsce określa ustawa z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki (Dz.U. nr 65 z 2003 r., poz. 595 (Dz. U. z 2003 r. Nr 65, poz. 595). Poprzednie polskie uregulowania nie wymagały bezwzględnie posiadania przez kandydata tytułu zawodowego magistra lub równorzędnego (choć zasada ta zazwyczaj była przestrzegana) i zdarzały się nadzwyczajne przypadki nadawania stopnia naukowego doktora osobom bez studiów wyższych, np. słynnemu matematykowi lwowskiemu – późniejszemu profesorowi Stefanowi Banachowi.

W innych krajach również zazwyczaj do przyznania stopnia naukowego doktora potrzebny jest dyplom ukończenia uczelni wyższej, ale nie wszędzie.

Dodatek B

Opis załączonej płyty CD/DVD

Tutaj jest miejsce na zamieszczenie opisu zawartości załączonej płyty. Należy wymienić, co zawiera.

Indeks rzeczowy