

POLITECHNIKA WROCŁAWSKA

WYDZIAŁ ELEKTRONIKI

KIERUNEK: Informatyka (INF)
SPECJALNOŚĆ: Inżynieria systemów informatycznych (INS)

PRACA DYPLOMOWA INŻYNIERSKA

Aplikacja webowa wspomagająca zarządzanie
flotą samochodów

A web application supporting cars fleet
management

AUTOR:
Jan Pajdak

PROWADZĄCY PRACĘ:
dr inż. Jarosław Mierzwa, K-9

OPIEKUN:
dr hab. inż. Olgierd Unold Prof. nadzw. PWr,
K-9

OCENA PRACY:

Spis treści

1	Wstęp	1
2	Specyfikacja problemu	2
2.1	Przegląd istniejących rozwiązań	2
2.2	Założenia projektowe	2
2.3	Wymagania funkcjonalne	2
2.4	Wymagania niefunkcjonalne	2
3	Projekt systemu	3
3.1	Technologie	3
3.2	Narzędzia	3
	Bibliografia	4

Rozdział 1

Wstęp

Celem niniejszej pracy dyplomowej jest opracowanie projektu, implementacja oraz wdrożenie systemu umożliwiającego zarządzanie flotą samochodów. Pierwszym etapem projektu jest zebranie wymagań funkcjonalnych i нефункциональных, określenie zakresu pracy oraz przegląd istniejących rozwiązań. Drugi etap projektu to wybór technologii i projekt architektury. Ostatnim celem jest implementacja systemu.

Temat projektu został wybrany ze względu na chęć wykorzystania wiedzy z dziedziny motoryzacji w celu stworzenia aplikacji ułatwiającej zarządzanie pojazdami. Z uwagi na rosnącą popularność rozwiązań związanych z wypożyczaniem samochodów celem projektu jest system, który można opisać jako wewnątrzfirmową wypożyczalnię umożliwiającą jak największe wykorzystanie dostępnej floty pojazdów przez pracowników, którzy nie mają potrzeby posiadania firmowego samochodu na wyłączność.

W rozdziale pierwszym zawarto wstęp oraz krótki opis celu projektu. Rozdział drugi zawiera opis biznesowy problemu, porównanie z istniejącymi rozwiązaniami oraz wymagania. W kolejnym, trzecim rozdziale znajduje się techniczny opis projektu — wykorzystane technologie i narzędzia oraz architektura systemu.

Rozdział 2

Specyfikacja problemu

2.1 Przegląd istniejących rozwiązań

2.2 Założenia projektowe

2.3 Wymagania funkcjonalne

2.4 Wymagania нефunkcjonalne

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Rozdział 3

Projekt systemu

3.1 Technologie

Interfejs użytkownika wykorzystuje platformę *Angular 7*. Podstawowymi elementami w *Angular* są komponenty, każdy z nich złożony z: pliku klasy *TypeScript* zawierającej logikę, wzorca *htm* opisującego wygląd widoku oraz opcjonalnego stylu *css*; w przypadku jego braku styl brany jest z komponentu-rodzica. Warto zwrócić uwagę na język programowania wykorzystywany przez platformę *Angular* — *TypeScript*, będący rozszerzeniem języka *JavaScript*. *TypeScript* dodaje silniejsze typowanie i kładzie większy nacisk na programowanie obiektowe, jednocześnie pozostając w pełni kompatybilnym z *JavaScript*, do którego jest kompilowany i następnie uruchamiany jest w przeglądarce. Proces kompilacji pozwala na usunięcie wielu błędów, które w przypadku *JavaScript* zostałyby zauważone dopiero po uruchomieniu aplikacji.

Interfejs programistyczny oparty został na technologii *ASP.NET Core 2.1* — jest to nowoczesna platforma oferująca działanie na wielu systemach operacyjnych oraz większą wydajność względem starszych rozwiązań firmy Microsoft. Wykorzystany język programowania to obiektowy, kompilowany i statycznie typowany *C# 7.3*. Bardzo ważnym elementem tej części projektu jest *EF (Entity Framework) Core 2.1*, framework ORM (Object-Relational Mapping) pozwalający na konwersję między tabelami bazy danych a klasami *C#*. Jedną z najważniejszych funkcjonalności *EF Core* jest wykorzystana w niniejszym projekcie możliwość utworzenia bazy danych przy użyciu konwencji *Code First*; baza danych jest automatycznie generowana na podstawie klas *C#* znajdujących się w projekcie.

3.2 Narzędzia

W trakcie realizacji projektu wykorzystane zostały narzędzia najczęściej używane przy wybranych technologiach.

Do zarządzania kodem został wykorzystany system kontroli wersji *Git*. Lokalna kopia projektu była synchronizowana ze zdalnym, prywatnym repozytorium znajdującym się na serwisie GitHub. Wykorzystane rozwiązanie pozwala na łatwy dostęp do wcześniejszych wersji projektu oraz zmniejsza ryzyko utraty kodu, gdyż nie jest on przechowywany tylko w jednym miejscu.

Ze względu na wykorzystane technologie, kod był rozwijany z pomocą narzędzi Microsoft, oferujących najlepsze wsparcie dla *TypeScript* oraz *C#*. Aplikacja klienta była rozwijana przy użyciu *Visual Studio Code 1.28*, nowoczesnego edytora który sprawdza

się znakomicie przy tworzeniu interfejsów użytkownika ze względu na zintegrowaną konsolę pozwalającą na łatwe zarządzanie paczkami oraz wielką łatwość dostosowywania do potrzeb użytkownika. W trakcie pracy wykorzystano wiele rozszerzeń, najważniejsze z nich to *TSLint*, linter wykrywający błędy w kodzie *TypeScript* oraz *GitLens* — rozszerzenie wspomagające zarządzanie repozytorium *Git*. Do rozwoju serwisów wykorzystano *Visual Studio 2017* pozwalające na łatwe debugowanie kodu oraz analizę aspektów takich jak wykorzystanie zasobów przez program. *Visual Studio* zostało wzbogacone o narzędzie *JetBrains ReSharper* automatycznie formatujące pliki projektu według zadanego wzorca, zapewniając spójność i przejrzystość kodu.

Interfejs programistyczny testowany był przy pomocy *Postman 6.5.2*, aplikacji pozwalającej na wysyłanie oraz zarządzanie zapytaniem HTTP.

Bibliografia