OPIS I STRUKTURA PROJEKTU

Magdalena Jaśkiewicz 233968

Marcelina Matuszewska 233986

**Zaawansowane programowanie aplikacji bazodanowych**

Projekt

Wypożyczalnia samochodów

Celem stworzenia aplikacji było zastosowanie zdobytej przez nas wiedzy podczas zajęć laboratoryjnych. Zadaniem projektu było wykonanie aplikacji dla firmy zajmującej się wypożyczaniem samochodów

**Język programowania**: Java

**Serwer baz danych**: Microsoft SQL Server Managment

**Środowisko programistyczne**: IntelliJ IDEA

**Dodatkowe wykorzystane technologie:**

* Postman
* Maven
* Spring Boot
* Spring Security

Struktura projektu aplikacji – backend została przedstawiona poniżej (rys. 1):

|  |
| --- |
|  |
| Rysunek 1. Struktura projektu aplikacji - backend. |

Konfiguracja właściwości aplikacji zostały ustawione w pliku *application.properties*, który jest plikiem tekstowym korzystającym z wbudowanych właściwości (rys. 2):

|  |
| --- |
|  |
| Rysunek 2. Plik application.properties. |

Łączenie z serwerem Microsoft SQL Server Managment było wykonywane w następujący sposób (rys. 3):

|  |
| --- |
|  |
| Rysunek 3. Łączenie z serwerem MS SQL Server Managment. |

Z poziomu Microsoft SQL Server Managment wygenerowano skrypt z przykładową strukturą bazy danych (plik o nazwie*: car\_rental.sql*). Tabele odpowiednio: klienci   
(rys. 4), wypożyczenia (rys. 5), samochody (rys. 6) zostały zaprezentowane poniżej:

|  |
| --- |
|  |
| Rysunek 4. Tabela dbo.customer zawierająca wszystkich klientów. |

|  |
| --- |
|  |
| Rysunek 5. Tabela dbo.rental zawierająca wypożyczenia. |

|  |
| --- |
|  |
| Rysunek 6. Tabela dbo.car zawierająca dostępne w bazie samochody. |

ZAŁOŻENIA PROJEKTOWE – FUNKCJONALNOŚĆ

W projekcie aplikacji założyłyśmy dwie role aplikacyjne z podziałem na poszczególne funkcjonalności, które zostały przedstawione na schemacie (rys. 7) oraz opisane poniżej:

* **ADMIN** - osoba zarządzająca stroną posiadająca najwięcej uprawnień   
  (m.in. usuwanie klientów, samochodów, wypożyczeń z bazy oraz ich dodawanie)
* **USER** - klient, osoba korzystająca z aplikacji, która ma możliwość logowania, przegląd oferty samochodów w wypożyczalni oraz dodawanie swojego wypożyczenia.

|  |
| --- |
|  |
| Rysunek 7. Diagram sekwencji zdarzeń. |

Fragment klasy *SecurityConfig* przedstawiający role aplikacyjne (admin i user) oraz ich podział na dane funkcjonalności został przedstawiony poniżej (rys. 8):

|  |
| --- |
|  |
| Rysunek 8. Fragment kodu przedstawiający role aplikacyjne. |

Kodowanie hasła zostało wykonane w następujący sposób (rys. 9 - 10):

|  |
| --- |
|  |
| Rysunek 9. Kodowanie hasła. |

|  |
| --- |
|  |
| Rysunek 10. Kodowanie hasła – przykładowy klient. |

Rezultat kodowania został przedstawiony poniżej (rys. 11) :

|  |
| --- |
|  |
| Rysunek 11. Rezultat kodowania hasła – przykładowy klient. |

b. (w sprawozdaniu) przedstawić przypadki użycia dla każdej  
grupy (graficznie oraz opisać). W tym celu należy wykorzystać  
diagramy UML

4) przedstawić zaprojektowaną strukturę bazy danych lub diagram encji.  
Na diagramach powinny zostać naniesione ograniczenia w tym relacje // nie wiem czym te dwa różnią się od siebie.

// 5) Dołączyć skrypt z przykładową strukturą bazy danych .  
Z poziomu aplikacji administrator powinien mieć możliwość utworzenia całej bazy z danymi przykładowymi (skasuje starą bazę i utworzy nową z całą strukturą i danymi).  
6) Aplikacja powinna zawierać:  
- moduł logowania  
- moduł dla poszczególnych roli aplikacyjnych  
- możliwość wybierania, sortowania i przeglądania danych wg. funkcjonalności  
- moduł przedstawiający w sposób graficzny pewne elementy czasowe (np. statystyki sprzedaży w danym roku lub miesiącu)   
// - tryb połączenia z bazą danych,

- kodowanie hasła (dowolną metodą), które jest przechowywane w bazie danych.  
// - bazę danych w wybranym środowisku BD (SQL Server, ORACLE lub inny)