SYSTEM ZARZĄDZANIA OŚRODKIEM SZKOLENIA KIEROWCÓW (OSK) DOKUMENTACJA

Spis treści

1. Przeznaczenie projektu

Celem projektu wykonywanego na oba przedmioty („Persystencje" oraz „Fraeworki Biznesowe") z klasy przedmiotów o nazwie „Systemy Klasy Enterprise" było przygotowanie Systemu Zarządzania Ośrodkiem Szkolenia Kierowców (OSK).

System ten ma za zadanie wspomóc zarządzanie takim ośrodkiem przez właścicieli oraz instruktorów, a także pomóc kursantom w wielu aspektach związanych z nauką jazdy, np. zapisy na kurs, itp. Szczegółowe przypadki użycia zostaną ukazane na diagramie UML. System Zarządzania Ośrodkiem Szkolenia Kierowców automatyzuje działanie takie Ośrodka, ogranicza potrzebę udawania się do Ośrodka w sprawach formalnych, które można załatwić „na odległość". Skraca też czas załatwienia wszelkich formalności.

System został napisany w języku Java przy użyciu framework'a Spring. Projekt korzysta także z Maven i został umieszczony na repozytorium GIT. Przy pisaniu tego Systemu korzystano ze środowiska programistycznego IntelliJ IDEA.

W Systemie tym zostało zaimplementowane REST-owe API, a za połączenie z bazą danych PostgreSQL oraz za zarządzanie danymi w niej umieszczonymi odpowiada JPA (Java Persistence API).

2. Baza danych – PostgreSQL

Baza danych na potrzeby korzysta z Systemu Zarządzania Bazą Danych PostgreSQL. Za pomocą mechanizmów JPA i klas encyjnych tworzonych jest osiem tabel, które przedstawiają się następująco:

- Course – znajdują się w niej ogólne dane dotyczące prowadzonych kursów w Ośrodku (kategoria + oznaczenie);
- Course_data – szczegółowe dane o przeprowadzonych kursach, znajdują się tu dane Kursantów, którzy zapisali się na dane kursy;
- Driving_lesson – szczegółowe dane o ustalonych i odbytych jazdach w ramach danego kursu;
- Instructor – dane Instruktorów;
- Internal_exam – szczegółowe dane o ustalonych i odbytych egzaminach wewnętrznych w ramach danego kursu;
- Lecture – szczegółowe dane o ustalonych i odbytych wykładach w ramach danego kursu;

- Participant – dane Kursantów;
- Vehicle – dane pojazdów;

3. Analiza funkcjonalności

Po przeprowadzeniu dokładnej analizy funkcjonalności i wymagań dla Systemu Zarządzania Ośrodkiem Szkolenia Kierowców (OSK) wyszczególniono poszczególnych aktorów i czynności, które powinni mieć możliwość wykonywania na poziomie aplikacji.

ADMINISTRATOR:

- Jako Administrator chcę mieć możliwość dodania/edycji/usunięcia kont Instruktorów.
- Jako Administrator chcę mieć możliwość dodania/edycji/usunięcia z bazy danych pojazdów, którymi prowadzone są jazdy.
- Jako Administrator chcę mieć możliwość dodania/edycji/usunięcia z bazy danych kursów, które są prowadzone w Ośrodku.

INSTRUKTOR:

- Jako Instruktor chcę mieć możliwość sprawdzenia swoich danych.
- Jako Instruktor chcę mieć możliwość ustalenia/zmiany/odwołania wykładu dla danego Kursanta.
- Jako Instruktor chcę mieć możliwość ustalenia/zmiany/odwołania jazdy dla danego Kursanta.
- Jako Instruktor chcę mieć możliwość sprawdzenia stanu płatności za poszczególne kursy od poszczególnych Kursantów.
- Jako Instruktor chcę mieć możliwość przejrzenia listy Kursantów, którzy są zapisani na poszczególne kursy.
- Jako Instruktor chcę mieć możliwość przejrzenia listy ustalonych i prowadzonych przeze mnie wykładów.
- Jako Instruktor chcę mieć możliwość przejrzenia listy ustalonych i prowadzonych przeze mnie jazd.
- Jako Instruktor chcę mieć możliwość wezwania danego Kursanta na egzamin wewnętrzny dotyczący danego kursu.
- Jako Instruktor chcę mieć również możliwość zmiany terminu tego egzaminu lub całkowitego jego odwołania.
- Jako Instruktor chcę mieć możliwość zakończenia kursu danemu Kursantowi po odbyciu wszystkich wykładów, jazd i zdaniu egzaminu wewnętrznego.

KURSANT:

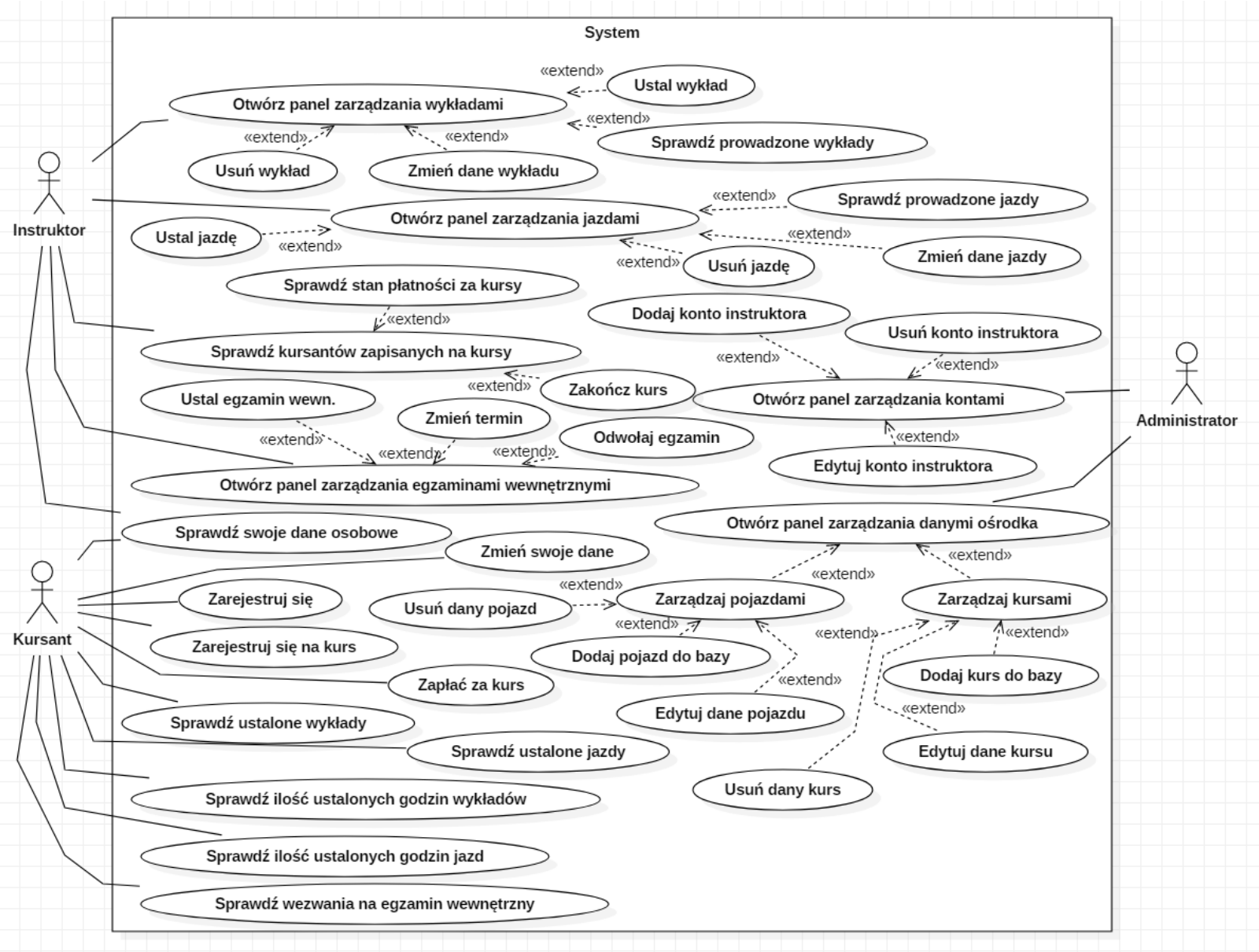
- Jako Kursant chcę mieć możliwość założenia konta w Systemie.
- Jako Kursant chcę mieć możliwość sprawdzenia swoich danych.
- Jako Kursant chcę mieć możliwość zmiany swoich danych.
- Jako Kursant chcę mieć możliwość rejestracji na wybrany kurs.
- Jako Kursant chcę mieć możliwość zapłacenia za kurs, na który jestem zapisany.
- Jako Kursant chcę mieć możliwość podglądu listy ustalonych dla mnie wykładów przez Instruktora.
- Jako Kursant chcę mieć możliwość podglądu listy ustalonych dla mnie jazd przez Instruktora.
- Jako Kursant chcę mieć możliwość sprawdzenia ilości ustalonych/wyjeżdżonych godzin na jazdach.
- Jako Kursant chcę mieć możliwość sprawdzenia ilości ustalonych/odbytych godzin na wykładach.
- Jako Kursant chcę mieć możliwość sprawdzenia, czy jestem wzywany na egzamin wewnętrzny z kursów, na które jestem zapisany.

4. Przypadki użycia

Na podstawie przeprowadzonej analizy funkcjonalności i wymagań, stworzonej listy w podrozdziale 3. Zaprojektowany został diagram przypadków użycia w notacji UML. Do stworzenia tego diagramu wykorzystano program StarUML.

Jak widzimy na poniższym diagramie, uwzględniono trzech aktorów i funkcjonalności, które powinny znaleźć się w tym Systemie.

Na następnej stronie umieszczony został diagram przypadków użycia tego Systemu za pomocą diagramu UML.


System
Instruktor
Kursant
Administrator
Ustal wykład
Otwórz panel zarządzania wykładami
Sprawdź prowadzone wykłady
Usuń wykład
Zmień dane wykładu
Otwórz panel zarządzania jazdami
Sprawdź prowadzone jazdy
Ustal jazdę
Usuń jazdę
Zmień dane jazdy
Sprawdź stan płatności za kursy
Sprawdź kursantów zapisanych na kursy
Dodaj konto instruktora
Usuń konto instruktora
Zakończ kurs
Otwórz panel zarządzania kontami
Ustal egzamin wewn.
Zmień termin
Odwołaj egzamin
Edytuj konto instruktora
Otwórz panel zarządzania egzaminami wewnętrznymi
Otwórz panel zarządzania danymi ośrodka
Sprawdź swoje dane osobowe
Zmień swoje dane
Zarejestruj się
Usuń dany pojazd
Zarządzaj pojazdami
Zarządzaj kursami
Zarejestruj się na kurs
Dodaj pojazd do bazy
Dodaj kurs do bazy
Zapłać za kurs
Edytuj dane pojazdu
Sprawdź ustalone wykłady
Edytuj dane kursu
Sprawdź ustalone jazdy
Usuń dany kurs
Sprawdź ilość ustalonych godzin wykładów
Sprawdź ilość ustalonych godzin jazd
Sprawdź wezwania na egzamin wewnętrzny
«extend»

5. RestAPI

W celu komunikacji pomiędzy klientem a bazą danych wykorzystano RestAPI. Do każdej funkcjonalności widocznej na powyższym diagramie przypadków użycia stworzono metody GET/PUT/PATCH/POST/DELETE.

Następnie w programie POSTMAN przygotowano kolekcję requestów GET/PUT/PATCH/POST/DELETE. Utworzona kolekcja requestów znajduje się w repozytorium GIT w katalogu „request_collection_postman".

Aby wszystkie utworzone requesty działały zgodnie z ID podanymi w adresach URL każdego requesta typu GET/PATCH/PUT/DELETE należy w odpowiedniej kolejności wywołać umieszczone w tej kolekcji requesty typu POST (podaję kolejność po nazwach nadanych tym requestom):

- Dodanie pierwszego kursanta (ID 1);
- Dodanie drugiego kursanta (ID 2);
- Dodanie pierwszego instruktora (ID 3);
- Dodanie drugiego instruktora (ID 4);
- Dodanie trzeciego instruktora (ID 5);
- Dodanie pierwszego kursu (ID 6);
- Dodanie drugiego kursu (ID 7);
- Dodanie trzeciego kursu (ID 8);
- Dodanie pierwszego pojazdu (ID 9);
- Dodanie drugiego pojazdu (ID 10);
- Dodanie trzeciego pojazdu (ID 11);
- Zapisanie się na kurs A przez kursanta o ID 1 (course_data – ID 12);
- Zapisanie się na kurs B przez kursanta o ID 2 (course_data – ID 13);
- Ustalenie pierwszego wykładu przez instruktora o ID 3 dla kursanta o ID 1 (lecture – ID 14);
- Ustalenie drugiego wykładu przez instruktora o ID 3 dla kursanta o ID 1 (lecture – ID 15);
- Ustalenie wykładu przez instruktora ID 4 dla kursanta ID 2 (lecture – ID 16);
- Ustalenie pierwszej jazdy przez instruktora ID 4 dla kursanta ID 2 na kursie ID 6 pojazdem ID 9 (driving_lesson – ID 17);
- Ustalenie drugiej jazdy przez instruktora ID 4 dla kursanta ID 2 na kursie ID 6 pojazdem ID 9 (driving_lesson – ID 18);
- Ustalenie jazdy przez instruktora ID 3 kursantowi ID 1 na kursie ID 7 pojazdem ID 10 (driving_lesson – ID 19);
- Ustalenie egzaminu wewnętrznego przez instruktora ID 3 dla kursanta ID 1 na kursie ID 7 (internal_exam – ID 20);

- Ustalenie egzaminu wewnętrznego przez instruktora ID 3 dla kursanta ID 2 na kursie ID 6 (internal_exam – ID 21.

Jest też druga opcja na wykonanie tych requestów w odpowiedniej kolejności. Można skorzystać ze skryptu o rozszerzeniu .bat o nazwie „addall.bat", który umieszczony jest również w repozytorium GIT w katalogu o nazwie „request_curl_scripts".

Po wykonaniu requestów typu POST w odpowiedniej kolejności można wykonywać requesty typu GET/PUT/PATCH/DELETE w dowolnej kolejności wedle uznania i potrzeby.

6. Autor projektu

Autorem projektu jest:

Wojciech Niewęgłowski (Student I roku Informatyki II stopnia UMCS).

Link do projektu na Github:

https://github.com/RespectPL/SKE-FBP_OSKProject