

Projekt z przedmiotu:

***Programowanie w języku JAVA***

Temat Projektu:

***System wspomagający******sprzedaż biletów kolejowych***

*Raport II*

*Wykonali:*

*Pancerz Marcin*

*Polak Jacek*

*rok akademicki 2016/2017*

*Spis treści*

[1.Wstęp 3](#_Toc469420261)

[2. Postępy Prac 3](#_Toc469420262)

[2.1 Klient/Serwer 3](#_Toc469420263)

[2.1.1 Aplikacja Serwera 3](#_Toc469420264)

[2.1.2 Aplikacja Kliencka 4](#_Toc469420265)

[2.1.3 Panel Logowania 4](#_Toc469420266)

[2.1.4 Panel Rejestracji 4](#_Toc469420267)

[2.1.5 Panel Użytkownika 4](#_Toc469420268)

[2.1.7 Moduł Daty i Czasu 5](#_Toc469420269)

[2.1.8 Baza danych 5](#_Toc469420270)

[3.Planowane funkcjonalności 5](#_Toc469420271)

[3.1 Wyszukiwarka połączeń 5](#_Toc469420272)

[3.2 Moduł rezerwacji i zakupu biletów 5](#_Toc469420273)

[3.3 Rozkład Dworca 5](#_Toc469420274)

[3.4 Panel Informacyjny 5](#_Toc469420275)

[3.5 Panel Pomocy 5](#_Toc469420276)

[3.6 Panel administracyjny 5](#_Toc469420277)

[3.7 Panel rezerwacji biletów 5](#_Toc469420278)

[3.8 Panel historii zakupionych biletów 5](#_Toc469420279)

[4. Technologia JavaFx i wynikające z niej kwestie 6](#_Toc469420280)

[5. Diagram Klas 6](#_Toc469420281)

[6. Wygląd Aplikacji 7](#_Toc469420282)

[6.1 Menu główne 7](#_Toc469420283)

[6.2 Panel Logowania 8](#_Toc469420284)

[6.3 Panel Rejestracji 9](#_Toc469420285)

[6.4 Panel Użytkownika 10](#_Toc469420286)

[6.5 Panel Profil Użytkownika 11](#_Toc469420287)

# 1.Wstęp

Niniejszy dokument prezentuje postępy prac w tworzeniu aplikacji wspomagającej sprzedaż biletów kolejowych. Zostały w nim opisane funkcjonalności, które na dzień tworzenia dokumentu udało się zaimplementować. Załączony został także diagram klas, wstępnie opisujący strukturę aplikacji. Zwrócona została także uwaga na funkcjonalności, które są w planach do wykonania. W dokumencie zawarte są również screen-shoty prezentujące wygląd aplikacji.

# 2. Postępy Prac

Zgodnie z założeniami projektu, aplikacja powstaje w języku JAVA. Została ona oparta o architekturę klient/serwer. Implementacja aplikacji wsparta została przez technologię JavaFX co zdecydowanie usprawnia pracę, jak również polepsza jej efekt końcowy. Aplikacja współpracuje z bazą danych za pośrednictwem serwera.

***Wykonano***

## 2.1 Klient/Serwer

Aplikacja pisana jest i rozwijana jako dwa człony.

### 2.1.1 Aplikacja Serwera

Aplikacja serwera jest pomostem łączącym aplikację kliencką, która korzysta z funkcjonalności serwera, z bazą danych. Użytkownik zgłasza zapytanie do serwera, a ten w odpowiedni sposób na nie odpowiada. Procesy serwera obsługiwane są wielowątkowo, dzięki czemu możliwym jest aby więcej niż jeden użytkownik korzystał z usług serwera. Na ten moment w aplikacji serwera stworzono kilka klas z nią związanych. Wśród nich można wyodrębnić dwie kategorie. Pierwszą są klasy odpowiedzialne za utworzenie i uruchomienie serwera. Drugą kategorią są klasy odpowiedzialne za współpracę z bazą danych.

### 

### 2.1.2 Aplikacja Kliencka

Aplikacja kliencka jest platformą, z którą bezpośredni kontakt ma klient. Za jej pomocą dokonuje on wszystkich interesujących go procesów, tak więc ważnym jest żeby miała ona przyjemny i wygodny w użyciu interfejs. Dzięki zastosowaniu technologii JavaFx i możliwości stylizacji wyglądu za pomocą plików CSS, kreowanie wyglądu aplikacji staje się bardzo łatwe i przyjemne. Oddzielenie wyglądu aplikacji od jej ścisłej logiki staje się bardzo wydajną i pomocną praktyką.

Ze względu na zastosowanie technologii JavaFx w procesie tworzenia aplikacji można wyróżnić

### 2.1.3 Panel Logowania

Udało się wykonać w pełni działający Panel Logowania użytkownika. Proces logowania został dokładnie zabezpieczony. W takcie logowania aplikacja kliencka wysyła zapytanie do serwera czy dany użytkownik jest zarejestrowany, a następnie odbywa się proces walidacji wprowadzonych przez użytkownika danych. Ważną kwestią okazało się zaimplementowanie sprawdzania poprawności struktury adresu e-mail do tego celu została wykorzystana klasa EmailValidator - Apache Commons. Cały proces rejestracji opatrzony został komunikatami informującymi użytkownika o wynikłym problemie. Gdy proces logowania przebiegnie pomyślnie, użytkownik przenoszony jest do Panelu Użytkownika.

### 2.1.4 Panel Rejestracji

Udało się zrealizować również Panel Rejestracji użytkownika. W panelu tym użytkownik proszony jest o podanie danych do rejestracji. Dane są poddane procesowi walidacji, podobnie jak w procesie logowania, wykorzystana została klasa EmailValidator - Apache Commons . Hasło użytkownika również musi składa się z określonej ilości znaków, jeśli ilość ta okaże się nie poprawna użytkownik otrzyma adekwatny komunikat. Gdy użytkownik pomyślnie przejdzie ten proces, przenoszony jest do Panelu Użytkownika, jeśli nie, proszony jest o poprawienie pól formularza powodujących błąd.

### 2.1.5 Panel Użytkownika

Panel użytkownika jest na etapie ciągłego rozwoju. Na ten moment udało zrealizować się funkcję wylogowywania użytkownika.

Przejście do Profilu użytkownika - funkcjonalność opisana w dalszej części dokumentu.

Reszta funkcjonalności w planach.

#### 2.1.6 Profil użytkownika

Panel ten skupia się na danych personalnych użytkownika. W celach informacyjnych wyświetla je, jak również umożliwia ich edycję. Panel ten współpracuje z bazą danych, w celu zmienienia danych użytkownika. Adres e-mail wyłączony jest z możliwości zmiany, gdyż jest on identyfikatorem użytkownika(Login) , który nie może ulec zmianie.

### 2.1.7 Moduł Daty i Czasu

W aplikacji został zaimplementowany moduł odpowiedzialny za bieżącą datę i czas. Jest on uruchamiany jako osobny wątek, z którego każdy z kontrolerów, za pomocą odpowiednich metod może pobierać potrzebne dane. Następnie dane te wyświetlane są na bieżącym panelu użytkownika. We wszystkich panelach załączony jest widok na bieżącą datę i czas.

### 2.1.8 Baza danych

Baza danych jest systematycznie rozwijana. Jej ogólna struktura została już utworzona.

# 3.Planowane funkcjonalności

## 3.1 Wyszukiwarka połączeń

## 3.2 Moduł rezerwacji i zakupu biletów

## 3.3 Rozkład Dworca

## 3.4 Panel Informacyjny

## 3.5 Panel Pomocy

## 3.6 Panel administracyjny

## 3.7 Panel rezerwacji biletów

## 3.8 Panel historii zakupionych biletów

# 4. Technologia JavaFx i wynikające z niej kwestie

W procesie tworzenia aplikacji zastosowano technologię JavaFx korzystającą z wzorca architektonicznego Model-View-Controller. Wynika z tego, że powstało i powstanie wiele klas odpowiedzialnych za taki podział, są to klasy Modelu, klasy Kontrolerów jak również pliki FXML odpowiedzialne za wygląd aplikacji. Dodatkowym plikiem jest arkusz styli CSS, który ułatwia modelowanie wyglądu.

Z racji wymogu odświeżania widoków, w ramach reakcji na zmiany wynikające z działań modelu, w aplikacji wykorzystano wzorzec projektowy *Obserwator* - ułatwia to znacznie system odświeżania widoków.

# 5. Diagram Klas

Diagram klas prezentuje zbiór klas wewnętrznych aplikacji - poprzez sformułowanie "wewnętrznych" rozumie się klasy związane tylko z ogólnie rozumianym problemem rezerwacji i sprzedaży biletów kolejowych, a więc te związane z aplikacją kliencką, ale w formie logiki a nie zastosowanej technologii JavaFx. Tak więc wszystkie klasy związane ze wzorcem architektonicznym Model-View-Controller nie zostały zamodelowane na diagramie. Niewątpliwie w trakcie dalszego rozwoju aplikacji diagram ulegnie znacznym zmianom.

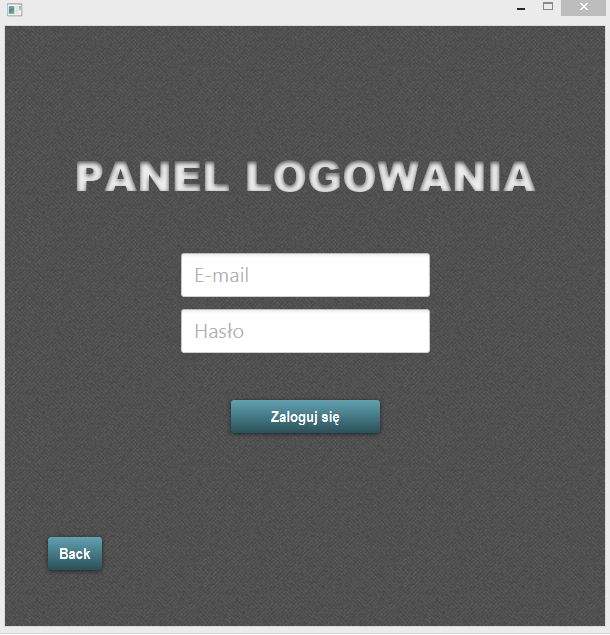
*Z racji słabej czytelności diagramu po załączeniu go do dokumentu, został on dołączony jako załącznik nr 1*

# 6. Wygląd Aplikacji

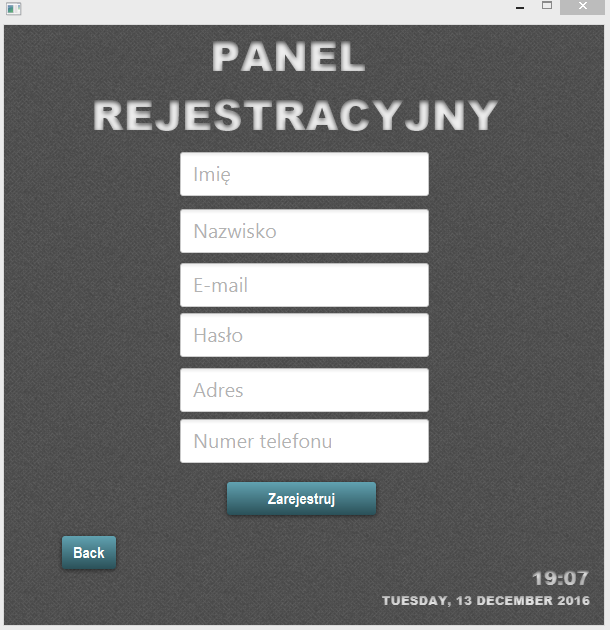
## 6.1 Menu główne



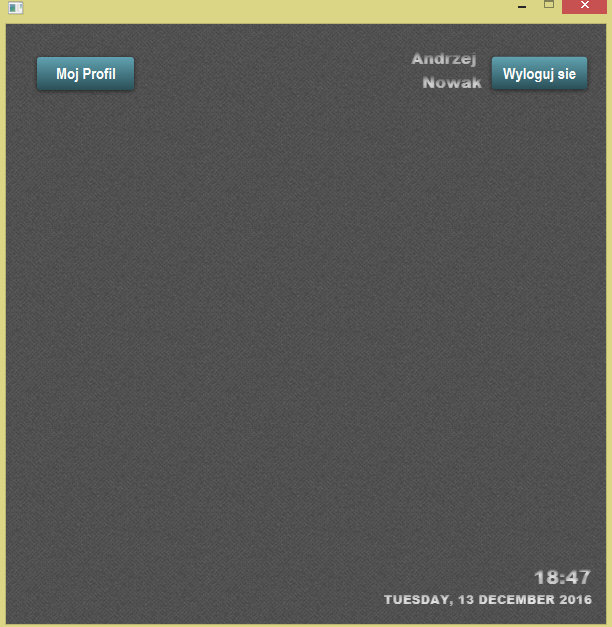
## 6.2 Panel Logowania



## 6.3 Panel Rejestracji



## 6.4 Panel Użytkownika



## 6.5 Panel Profil Użytkownika

