

# Bazy danych I

## Dokumentacja projektu

### Temat projektu:

## **System obliczania wynagrodzenia za pracę**

Kateryna Andrusiak

Wydział Fizyki i Informatyki Stosowanej

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

25 stycznia 2021

## **I. Projekt koncepcji, założenia**

### **1. Zdefiniowanie tematu projektu**

Tematem mojego projektu jest realizacja **systemu obliczania wynagrodzenia za pracę** licząc stawkę godzinową, wszystkie premię, bonusy oraz kary.

### **2. Analiza wymagań użytkownika**

Użytkownik ze strony administratora będzie miał możliwość dodawać nowych pracowników do bazy oraz wprowadzać informacje o nich, wprowadzać pracownikom pewne premię, bonusy, oraz kary. Też będzie miał możliwość dodawać nowe stanowiska, nowe rodzaje premii, bonusów oraz kar. Dostępna też będzie lista pracowników, gdzie można edytować informację o pracowniku oraz usuwanie wybranego pracownika.

Użytkownik ze strony pracownika będzie miał dostęp do swojej historii wypłat. Też będzie możliwość poglądu danych osobowych oraz ich edytowania. Dostępna też będzie możliwość zmiany hasła.

### **3. Zaprojektowanie funkcji:**

- Logowanie.

#### **Administrator:**

- Wyświetlanie listy pracowników:
  - Edytowanie informacji o pracowniku;
  - Usuwanie pracowników.
- Obliczenie wynagrodzenia w wybranym miesiącu licząc stawkę godzinową, ilość godzin w pracy w tym miesiącu, wszystkie premię, bonusy oraz kary:
  - Wprowadzenie ilości godzin w pracy dla wybranego pracownika w wybranym miesiącu;
  - Dodawanie premii dla wybranego pracownika w wybranym miesiącu;
  - Dodawanie bonów dla wybranego pracownika w wybranym miesiącu;
  - Dodawanie kar dla wybranego pracownika w wybranym miesiącu.

- Dodawanie nowych pracowników do bazy oraz informacje o nich.
- Dodawanie nowych stanowisk do bazy.
- Dodawanie nowych premii do bazy.
- Dodawanie nowych bonów do bazy.
- Dodawanie nowych kar do bazy.
- Wylogowanie.

#### Pracownik:

- Pogląd danych osobowych oraz możliwość ich edytowania.
- Pogląd historii wypłat za wybrany okres czasu
- Możliwość zmiany hasła.
- Wylogowanie.

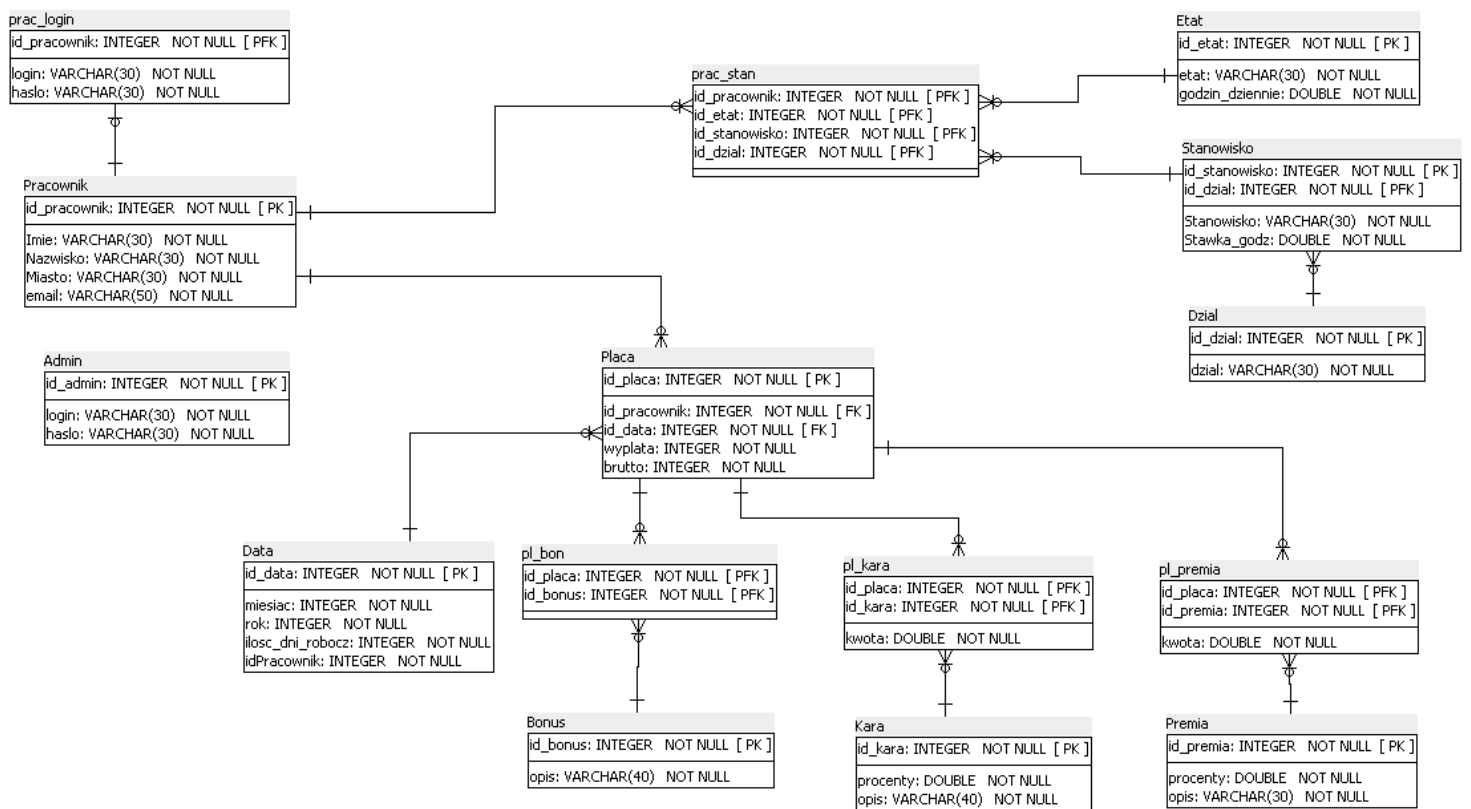
## II. Projekt diagramów (konceptualny)

### 4. Zdefiniowanie encji (obiektów) oraz ich atrybutów

Encja	Atrybuty	Opis
Administrator	id_admin, login, hasło	Encja służy do rozpoznawania administratora podczas logowania.
Pracownik	id_pracownik, imie, nazwisko, miasto, email	Encja reprezentuje pracownika firmy oraz zawiera dane osobowe (imię, nazwisko, miasto zamieszkania oraz e-mail)
Prac_login	id_pracownik, login, hasło	Encja służy do rozpoznawania pracownika podczas logowania.
Dział	id_dział, dział	Encja reprezentuje działy w firmie, atrybut dział – nazwa działu.
Stanowisko	id_stanowisko, id_dział, stanowisko, stawka_godz	Encja reprezentuje stanowiska zależące od działu oraz zawiera stawkę godzinową dla każdego stanowiska, atrybut stanowisko – nazwa stanowiska.
Etat	id_etat, etat, godzin_dziennie	Encja reprezentuje możliwe etaty oraz ilość niezbędnych godzin w pracy za dzień, atrybut etat – opis etatu.
Prac_stan	id_pracownik, id_etat, id_stanowisko, id_dział	Encja asocjacyjna służy do powiązania pracownika z etatem, stanowiskiem oraz działem
Data	id_data, miesiąc, rok, ilość_dni_robocz, idPracownik	Encja reprezentuje datę (miesiąc, rok) naliczenia wypłaty, ilość roboczych dni w tym miesiącu dla konkretnego pracownika.
Płaca	id_placa, id_pracownik, id_data, wypłata, brutto.	Encja reprezentuje płacę miesięczną pracownika, atrybut brutto reprezentuje wynagrodzenie za godziny w pracy, wypłata – wynagrodzenie razem z premiami oraz karami.
Premia	id_premia, procenty, opis	Encja reprezentuje premie możliwe do otrzymania oraz ich procentową wielkość od wynagrodzenie za godziny w pracy, atrybut opis – opis premii(nazwa).
Pl_premia	id_placa, id_premia, kwota	Encja asocjacyjna służy do powiązania płacy oraz premii, przechowuje też kwotę naliczonej premii.

Bonus	id_bonus, opis	Encja reprezentuje bonusy możliwe do otrzymania, atrybut opis – opis bonusu(nazwa).
Pl_bon	id_placa, id_bonus	Encja asocjacyjna służy do powiązania płacy ora bonusu.
Kara	id_kara, procenty, opis	Encja reprezentuje kary możliwe do otrzymania oraz ich procentową wielkość od wynagrodzenie za godziny w pracy, atrybut opis – opis kary(nazwa).
Pl_kara	id_placa, id_kara, kwota	Encja asocjacyjna służy do powiązania płacy oraz kary, przechowuje też kwotę naliczonej kary.

## 5. Diagram ERD



### III. Projekt logiczny

#### 6. Projektowanie tabel, kluczy, indeksów (plik: *SQL/DDL.sql*)

Tabela	Kolumna	Typ danych	
Administrator	id_admin	INTEGER NOT NULL	[PK]
	login	VARCHAR (30) NOT NULL	
	hasło	VARCHAR (30) NOT NULL	
Pracownik	id_pracownik	INTEGER NOT NULL	[PK] auto inkrementacja
	imie	VARCHAR (30) NOT NULL	
	nazwisko	VARCHAR (30) NOT NULL	
	miasto	VARCHAR (30) NOT NULL	
	email	VARCHAR (50) NOT NULL	
Prac_login	id_pracownik	INTEGER NOT NULL	[PFK]
	login	VARCHAR (30) NOT NULL	
	hasło	VARCHAR (30) NOT NULL	
Dział	id_dział	INTEGER NOT NULL	[PK]
	dział	VARCHAR (30) NOT NULL	
Stanowisko	id_stanowisko	INTEGER NOT NULL	[PK] auto inkrementacja
	id_dział	INTEGER NOT NULL	[PFK]
	stanowisko	VARCHAR (30) NOT NULL	
	stawka_godz		
Etat	id_etat	INTEGER NOT NULL	[PK] auto inkrementacja
	etat	VARCHAR (30) NOT NULL	
	godzin_dziennie	DOUBLE NOT NULL	
Prac_stan	id_pracownik	INTEGER NOT NULL	[PFK]
	id_etat	INTEGER NOT NULL	[PFK]
	id_stanowisko	INTEGER NOT NULL	[PFK]
	id_dział	INTEGER NOT NULL	[PFK]
Data	id_data	INTEGER NOT NULL	[PK] auto inkrementacja
	miesiąc	INTEGER NOT NULL	
	rok	INTEGER NOT NULL	
	ilość dni robocz	INTEGER NOT NULL	
	idPracownik	INTEGER NOT NULL	
Płaca	id_placa	INTEGER NOT NULL	[PK] auto inkrementacja
	id_pracownik	INTEGER NOT NULL	[FK]
	id_data	INTEGER NOT NULL	[FK]
	wypłata	DOUBLE NOT NULL	
	brutto	DOUBLE NOT NULL	
Premia	id_premia,	INTEGER NOT NULL	[PK] auto inkrementacja
	procenty	DOUBLE NOT NULL	
	opis	VARCHAR (30) NOT NULL	
Pl_premia	id_placa	INTEGER NOT NULL	[PFK]
	id_premia	INTEGER NOT NULL	[PFK]
	kwota	DOUBLE NOT NULL	
Bonus	id_bonus	INTEGER NOT NULL	[PK] auto inkrementacja
	opis	VARCHAR (40) NOT NULL	
Pl_bon	id_placa	INTEGER NOT NULL	[PFK]

	id_bonus	INTEGER NOT NULL	[PFK]
Kara	id_kara	INTEGER NOT NULL	[PK] auto inkrementacja
	procenty	DOUBLE NOT NULL	
	opis	VARCHAR (40) NOT NULL	
Pl_kara	id_placa,	INTEGER NOT NULL	[PFK]
	id_kara	INTEGER NOT NULL	[PFK]
	kwota	DOUBLE NOT NULL	

**Widoki (plik: SQL/Widoki.sql):**

<b>Widok lista_pracownikow</b>		
<b>Kolumna</b>	<b>Typ danych</b>	<b>Tabela źródłowa</b>
id_pracownik	INTEGER	Pracownik
imie	VARCHAR	Pracownik
nazwisko	VARCHAR	Pracownik
miasto	VARCHAR	Pracownik
email	VARCHAR	Pracownik
stanowisko	VARCHAR	Stanowisko
opis (dział)	VARCHAR	Dział
etat	VARCHAR	Etat

<b>Widok historia_wyplat</b>		
<b>Kolumna</b>	<b>Typ danych</b>	<b>Tabela źródłowa</b>
id_pracownik	INTEGER	Placa
id_placa	INTEGER	Placa
miesiac	INTEGER	Data
rok	INTEGER	Data
ilosc_dni_robocz	INTEGER	Data
brutto	DOUBLE	Placa
wyplata	DOUBLE	Placa

<b>Widok naliczone_premie</b>		
<b>Kolumna</b>	<b>Typ danych</b>	<b>Tabela źródłowa</b>
id_placa	INTEGER	Placa
kwota	DOUBLE	Pl_premia
opis	VARCHAR	Premia
procenty*100	DOUBLE	Premia

<b>Widok naliczone_kary</b>		
<b>Kolumna</b>	<b>Typ danych</b>	<b>Tabela źródłowa</b>
id_placa	INTEGER	Placa
kwota	DOUBLE	Pl_kara
opis	VARCHAR	Kara
procenty*100	DOUBLE	Kara

<b>Widok udzielone_bony</b>		
<b>Kolumna</b>	<b>Typ danych</b>	<b>Tabela źródłowa</b>
id_placa	INTEGER	Placa
opis	VARCHAR	Bonus

## 7. Analiza zależności funkcyjnych i normalizacja tabel

Tabele są znormalizowane do 3 postaci normalnej 3NF.

## 8. Zaprojektowanie operacji na danych

**Funkcje** *plik: SQL/Funkcje.sql*):

**insert\_data**(mies **int**, r **int**, ilosc\_dni **int**, idPrac **int**) – dodaje dane do tabeli Data. Argumenty: mies – miesiąc, r – rok, ilosc\_dni – ilość dni roboczych, idPrac – id pracownika. Zwraca dodane id\_data.

**stawka\_dzien**(idPrac **int**) – oblicza stawkę dzienną pracownika (czyli stawka godzinowa razy ilość godzin w pracy za dzień). Argument: idPrac – id pracownika. Zwraca policzoną stawkę dzienną.

**insert\_placa**(mies **int**, r **int**, ilosc\_dni **int**, id\_prac **int**) – dodaje dane do tabeli Placa. Argumenty: mies – miesiąc, r – rok, ilosc\_dni – ilość dni roboczych, id\_prac – id pracownika. Zwraca id dodanej płacy.

**wypłata\_mies**(id\_plac **int**, id\_prac **int**) - oblicza stawkę miesięczną pracownika (czyli stawka dzienna razy ilość dni w pracy w tym miesiącu). Argumenty: id\_plac – id\_placa, id\_prac – id pracownika. Zwraca policzoną stawkę miesięczną.

**update\_pracownik**( id\_prac **int**, nimie **varchar**, nnazwisko **varchar**, nmiasto **varchar**, nemail **varchar**, ndzial **varchar**, nstanowisko **varchar**, netat **varchar**) – edytuje dane o pracowniku. Argumenty: id\_prac – id pracownika, nimie – nowe imię, nnazwisko – nowe nazwisko, nmiasto – nowe miasto , nemail – nowy e-mail, ndzial – nowy dział, nstanowisko – nowe stanowisko, netat- nowy etat)

**delete\_pracownik**(id\_prac **int**) – usuwa dane pracownika ze wszystkich tabel. Argumenty: id\_prac – id pracownika. Zwraca Boolean w zależności czy został pracownik usunięty.

**zmien\_haslo**(idPrac **int**, oldpass **varchar**, newpass **varchar**) – zmienia hasło pracownika. Argumenty: idPrac – id pracownika, oldpass – stare hasło, newpass – nowe hasło.

**Wyzwalacze** (*plik: SQL/Wyzwalacze.sql*):

**pracownik\_test\_insert** – wyzwalacz, który uruchamia funkcję **prac\_insert()** przed dodawaniem nowego rekordu do tabeli Pracownik, jeśli takiego pracownika nie istnieje (identyfikacja po e-mailu) w tabeli to dodaje nowy rekord, w innym przypadku wyrzuca wyjątek.

**stanowisko\_test\_insert** – wyzwalacz, który uruchamia funkcję **stan\_insert()** przed dodawaniem nowego rekordu do tabeli Stanowisko, jeśli takiego stanowiska nie istnieje (identyfikacja po nazwie) w tabeli to dodaje nowy rekord, w innym przypadku wyrzuca wyjątek.

**placa\_test\_insert** – wyzwalacz, który uruchamia funkcję **placa\_insert()** przed dodawaniem nowego rekordu do tabeli Placa, jeśli płacy dla tego placownika w tym miesiącu nie istnieje w tabeli to dodaje nowy rekord, w innym przypadku wyrzuca wyjątek.

**premia\_test\_insert** – wyzwalacz, który uruchamia funkcję **premia\_insert()** przed dodawaniem nowego rekordu do tabeli Premia, jeśli takiej premii nie istnieje (identyfikacja po nazwie) w tabeli to dodaje nowy rekord, w innym przypadku wyrzuca wyjątek.

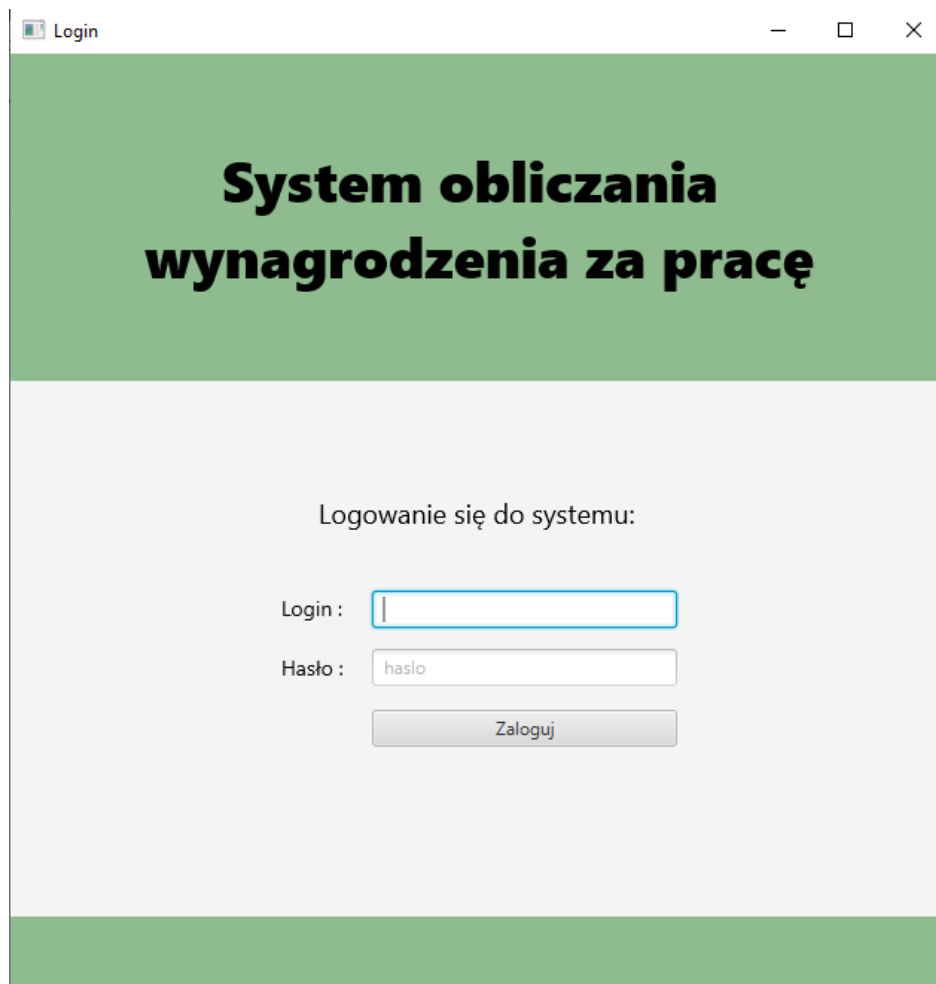
**kara\_test\_insert** – wyzwalacz, który uruchamia funkcję **kara\_insert()** przed dodawaniem nowego rekordu do tabeli Kara, jeśli takiej kary nie istnieje (identyfikacja po nazwie) w tabeli to dodaje nowy rekord, w innym przypadku wyrzuca wyjątek.

**bonus\_test\_insert** – wyzwalacz, który uruchamia funkcję **bonus\_insert()** przed dodawaniem nowego rekordu do tabeli Bonus, jeśli takiego bonusu nie istnieje (identyfikacja po nazwie) w tabeli to dodaje nowy rekord, w innym przypadku wyrzuca wyjątek.

## IV. Projekt funkcjonalny

Projekt jest realizowany w Java za pomocą technologii **JavaFX**, połączenia z bazą danych jest realizowane za pomocą technologii **JDBC**.

### 9. Aplikacja:



The screenshot shows a JavaFX window titled "Login". The window has a green header bar with the text "System obliczania wynagrodzenia za pracę" in bold black font. Below the header, the text "Logowanie się do systemu:" is centered. There are two input fields: "Login :" and "Hasło :". The "Hasło :" field contains the text "hasło". Below the input fields is a button labeled "Zaloguj".

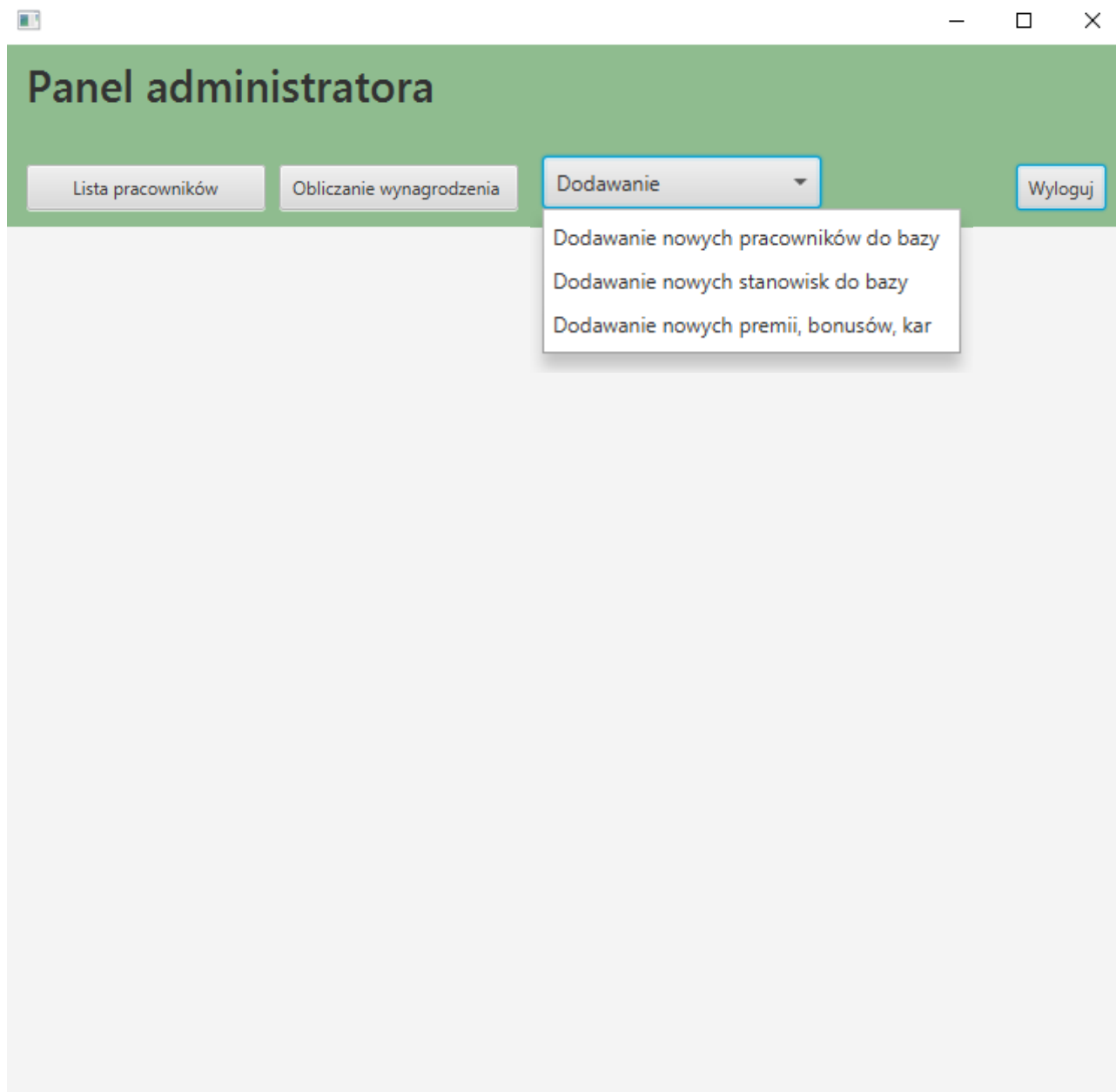
Rys.1. Okno logowania

Dla zalogowania się jako **administrator** należy użyć:

Login: **admin**

Hasło: **admin**

Po zalogowaniu otworzy się nowe okno: strona główna administratora.



Rys.2. Strona główna administratora

Z tego okna można przejść do:

- Listy pracowników
- Obliczania wynagrodzenia
- Dodawania
  - Dodawanie nowych pracowników do bazy
  - Dodawanie nowych stanowisk do bazy
  - Dodawanie nowych premii, bonusów, kar

Oraz można

- Wylogować się





**Panel administratora**

Lista pracowników   Obliczanie wynagrodzenia   Dodawanie   Wyloguj

### Obliczanie wynagrodzenia

Dział:

Pracownik:

Rok:  Miesiąc:

Ilość dni roboczych w tym miesiącu:

Premie:	Bonusy:	Kary:
<input type="checkbox"/> Regulaminowa	<input type="checkbox"/> Karnet na siłownię	<input type="checkbox"/> Opuszczenie pracy bez usprawiedliwienia
<input type="checkbox"/> Uznaniowa	<input type="checkbox"/> Ubezpieczenie na życie	<input type="checkbox"/> Spożywanie alkoholu w czasie pracy
<input type="checkbox"/>	<input type="checkbox"/> Służbowa komórka	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

\*dokonać wielokrotnego wyboru można przytrzymując klawisz SHIFT lub CTRL podczas wybierania dodatkowych elementów po pierwszym wybranym elemencie.

Do wypłaty:

**Rys. 5. Obliczanie wynagrodzenia**

Ta strona służy do obliczania wynagrodzenia dla wybranego pracownika.

Kroki postępowania:

1. Należy wybrać dział, gdzie pracuje pracownik, dla którego należy policzyć wynagrodzenia
2. Dalej należy wybrać pracownika.
3. Następnie potrzebne jest wpisanie roku, domyślnie jest wpisywany obecny rok.
4. Należy wybrać miesiąc, w którym obliczane jest wynagrodzenie.
5. Dalej trzeba wpisać ilość dni liczba musi być z zakresu [0-31].
6. Dalej można wybrać premię, bonusy, kary \*.

\*Te pola nie są obowiązkowe do wybrania, aby wybrać jeden element z listy należy kliknąć na niego, aby dokonać wielokrotnego wyboru trzeba przytrzymać SHIFT lub CTRL i kliknąć.

7. Po wprowadzeniu wszystkich danych należy kliknąć na „Policz”

Zatem ma pojawić się suma do wypłaty oraz dodatkowe informacje.

Dla wyczyszczenia wszystkich pól należy kliknąć „Wyczyść pola”.

The screenshot shows a web application window titled "Panel administratora". At the top, there is a green header bar with the title. Below the header, there is a navigation bar with four buttons: "Lista pracowników", "Obliczanie wynagrodzenia", "Dodawanie" (highlighted with a blue border), and "Wyloguj". The main content area is titled "Dodawanie pracownika do bazy:". It contains a form with the following fields: "Imię:" (text input with placeholder "Imię"), "Nazwisko:" (text input with placeholder "Nazwisko"), "Miasto:" (text input with placeholder "Miasto"), "e-mail:" (text input with placeholder "e-mail"), "Dział:" (dropdown menu with "Wybierz dział"), "Stanowisko:" (dropdown menu with "Wybierz stanowisko"), and "Etat:" (dropdown menu with "Wybierz etat"). Below the form fields is a "Zapisz" button. At the bottom of the form area, there are labels for "Login:" and "Hasło:". At the very bottom left, there is a "Wyczyść pola" button.

Rys. 6. Dodawanie nowych pracowników do bazy danych

Ta strona służy do dodania nowego pracownika do bazy danych.

Aby dodać pracownika należy wypełnić wszystkie pola:

- Imię, Nazwisko i Miasto muszą zawierać tylko litery.
- E-mail musi mieć @ (np. aa@mail.com).
- Po wybraniu działu zostanie otworzona możliwość wybrania stanowiska z tego działu.

Aby dodać pracownika do bazy trzeba kliknąć „Zapisz”, zatem automatycznie zostanie wygenerowane login i hasło pracownika z jakim on będzie mógł zalogować się do serwisu.

Dla wyczyszczenia wszystkich pól należy kliknąć „Wyczyść pola”.

The screenshot shows a web application window titled 'Panel administratora'. The window has a green header bar with the title and a navigation bar below it containing four buttons: 'Lista pracowników', 'Obliczanie wynagrodzenia', 'Dodawanie' (highlighted with a blue border), and 'Wyloguj'. The main content area is titled 'Dodawanie stanowiska do bazy' and contains a form with three input fields: 'Dział:' with a dropdown menu showing 'Wybierz dział', 'Stanowisko:' with a text input containing 'nowe stanowisko', and 'Stawka godzinowa:' with a text input containing 'stawka'. Below these fields is a 'Zapisz' button.

Rys. 7. Dodawanie nowych stanowisk do bazy danych

Ta strona służy do dodania nowego stanowiska do bazy danych.

Aby dodać stanowisko należy:

- Wybrać dział
- Wpisać nazwę
- Wpisać stawkę godzinową

Zatem kliknąć „Zapisz” aby dodać do bazy danych. Pola zostaną wyczyszczone automatycznie po dodaniu stanowiska do bazy.

**Panel administratora**

Lista pracowników   Obliczanie wynagrodzenia   **Dodawanie**   Wyloguj

### Dodawanie nowych premii, bonusów, kar

**Premia:**

Opis:    Procenty:   

**Kara:**

Opis:    Procenty:   

**Bonus:**

Opis:   

Rys. 8. Dodawanie nowych premii, kar oraz bonusów do bazy danych

Ta strona służy do dodania nowych premii, kar oraz bonusów do bazy danych.

Aby dodać do bazy danych **premię** należy wypełnić pole opis oraz wpisać procenty [0-100%].  
Zatem kliknąć „Dodaj premię”.

Aby dodać do bazy danych **karę** należy wypełnić pole opis oraz wpisać procenty [0-100%]. Zatem kliknąć „Dodaj karę”.

Aby dodać do bazy danych **bonus** należy wypełnić pole opis. Zatem kliknąć „Dodaj bonus”.

Aby wylogować się z konta administratora należy kliknąć „Wyloguj”.

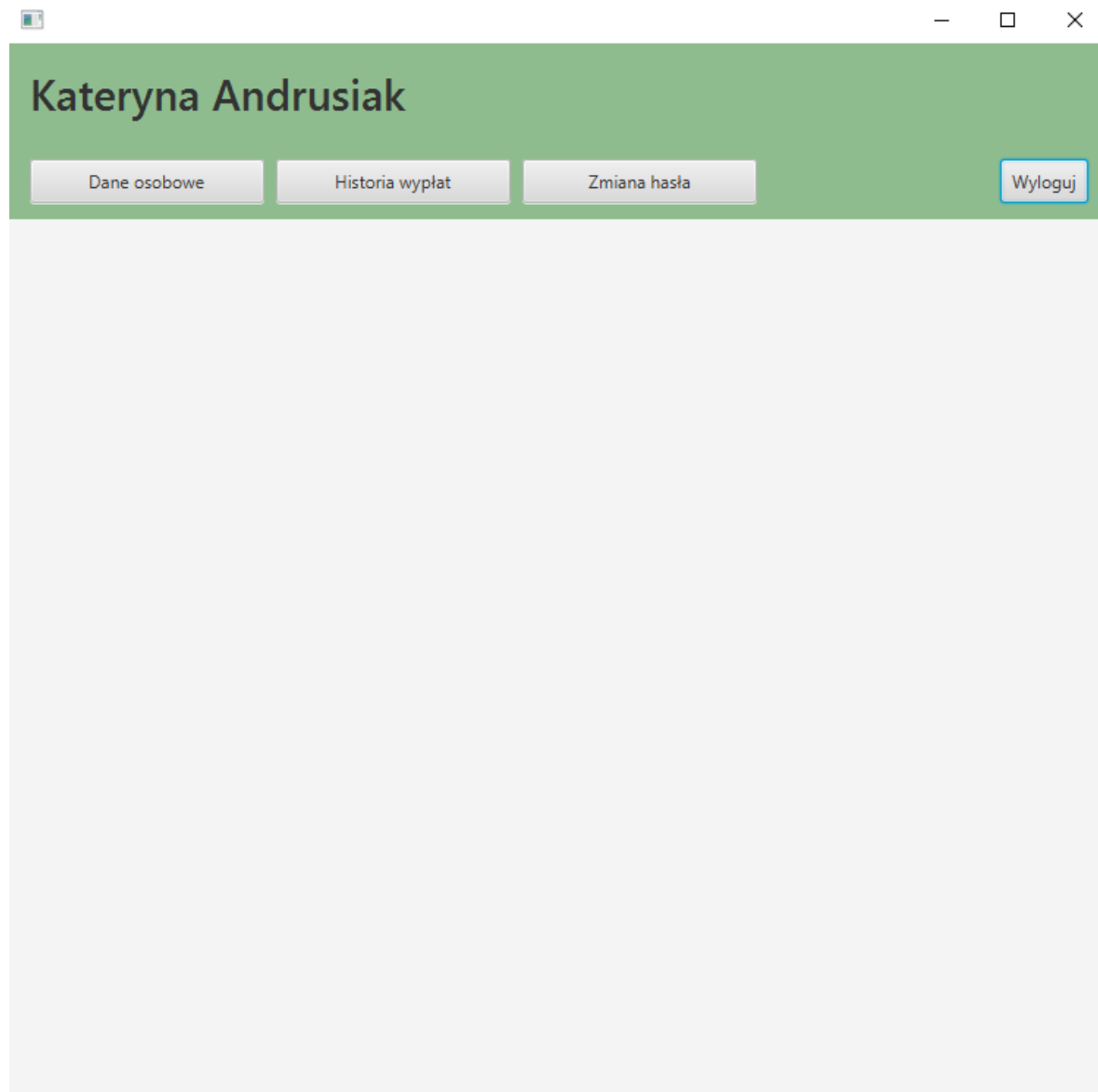
Dla zalogowania się jako **pracownik** należy użyć:

Login: **p1Andrusiak**

Hasło: **1Andrusiak**

**Lub jakichkolwiek dla innego istniejącego w bazie pracownika.**

Po zalogowaniu otworzy się nowe okno: strona główna pracownika.



Rys.9. Strona główna pracownika

Z tego okna można przejść do:

- Danych osobowych
- Historii wypłat
- Zmiany hasła

Oraz można

- Wylogować się

The screenshot shows a web application window with a green header bar containing the name 'Kateryna Andrusiak'. Below the header is a navigation bar with four buttons: 'Dane osobowe', 'Historia wypłat', 'Zmiana hasła', and 'Wyloguj'. The main content area is titled 'Dane osobowe' and displays the following information:

	Kateryna Andrusiak
Miasto:	Kraków
Email:	katia@email
Dział:	IT
Stanowisko:	Tester Aplikacji
Etat:	Pełen etat

At the bottom of the main content area is a button labeled 'Edytuj'.

Rys.10. **Dane osobowe pracownika**

Na tej stronie są wyświetlone dane zalogowanego pracownika.

Dostępna też możliwość edytowania danych osobowych pracownika, aby z niej skorzystać należy kliknąć „Edytuj”.

The screenshot shows the same web application window as in Rys.10, but with a modal window titled 'Edytowanie' open. The modal window contains the following fields and controls:

Imię:	<input type="text" value="Kateryna"/>	Dział:	<input type="text" value="IT"/>
Nazwisko:	<input type="text" value="Andrusiak"/>	Stanowisko:	<input type="text" value="Tester Aplikacji"/>
Miasto:	<input type="text" value="Kraków"/>	Etat:	<input type="text" value="Pełen etat"/>
e-mail:	<input type="text" value="katia@email"/>		

At the bottom of the modal window is a button labeled 'Zapisz zmiany'.

Rys.11. **Edytowanie danych pracownika**

Po zmianie należy kliknąć „Zapisz zmiany”.



Dane osobowe

### Historia wypłat

Zmiana hasła

Wyloguj

## Historia wypłat

[illegible]

Na tej stronie są wyświetlona historia wypłat pracownika.

Dane są posortowane według daty.

Jest też możliwość sortowania kolumn klikając na ich nazwy.



Kateryna Andrusiak

Dane osobowe Historia wypłat Zmiana hasła Wyloguj

### Zmiana hasła

Stare hasło:

Nowe hasło:

Potwierdź nowe hasło:

Rys.13. Zmiana hasła pracownika

Ta strona służy do zmiany hasła pracownika.

Aby zmienić hasło należy:

- Wpisać stare hasło.
- Wymyślić i wpisać nowe hasło
- Potwierdzić nowe hasło, wprowadzając jeszcze raz.

Po wypełnieniu wszystkich pól należy kliknąć „Zmień”.

**Aby wylogować się ze swojego konta należy kliknąć „Wyloguj”.**

## 10. Makropolecenia

Dla ułatwienie wprowadzenia danych są automatycznie wyciągane dane z bazy do list.

Też przy każdej próbie dodania danych do bazy wyskakują okienka z komunikatem udało się/nie udało się.

## V. Dokumentacja

### 11. Wprowadzanie danych

Wstępnie wprowadzone dane zostały umieszczone w pliku *SQL/DML.sql*.

Ręcznie dane zostały wprowadzone do tabel:

- Admin
- Etat
- Dział

Automatycznie z aplikacji do tabel:

- Pracownik
- Prac\_login
- Stanowisko
- Prac\_stan
- Data
- Placa
- Bonus
- Pl\_bon
- Kara
- Pl\_kara
- Premia
- Pl\_premia

### 12. Dokumentacja użytkownika

Najpierw należy rozarchiwować archiwum z projektem. Dalej znaleźć i uruchomić **Project.jar**, zatem ma otworzyć się okno z rysunku 1, dalej postępowania zgodnie z instrukcjami. Też uruchomić projekt można na serwerze Paskal za pomocą polecenia:

```
java -jar /home/stud2018/8andrusiak/bd1_labs/Project_BD/Project.jar
```

Baza danych stworzona jest w PostgreSQL i umieszczona na hostingu ElephantSQL (dla działania aplikacji potrzebne jest połączenie z Internetem). Wszystkie dane potrzebne dla poprawnego działania aplikacji są wprowadzone do bazy danych.

### 13. Opracowanie dokumentacji technicznej

Dokumentacja została dodana do kodu źródłowego aplikacji.

### 14. Wykaz literatury

- Wykłady: [https://newton.fis.agh.edu.pl/~antek/index.php?sub=db\\_wykl](https://newton.fis.agh.edu.pl/~antek/index.php?sub=db_wykl)
- Ćwiczenia: [https://newton.fis.agh.edu.pl/~antek/index.php?sub=db\\_cw](https://newton.fis.agh.edu.pl/~antek/index.php?sub=db_cw)
- Wikipedia: <https://www.wikipedia.org>
- Stackoverflow: <https://stackoverflow.com>
- Udemy kursy: <https://www.udemy.com>
- <https://www.postgresql.org>
- <https://www.tutorialspoint.com/postgresql/index.htm>
- <https://www.w3schools.com/sql/default.asp>
- <https://www.postgresqltutorial.com>