

Wrocław, 07.02.2022 r.

Weronika Najda, 254757  
Arkadiusz Bryłkowski, 253802  
Bazy danych  
P10-57a  
Środa, godz. 07.30

## **Projekt 2: Serwis**

### **1. Cel projektu**

Celem projektu było napisanie aplikacji służącej do obsługi serwisu urządzeń elektronicznych. Aplikacja ma być wykonana w JavaFX za pomocą sterownika JDBC MySQL.

### **2. Opis programu**

Wykonany program ma za zadanie obsługiwać serwisu urządzeń elektronicznych z poziomu serwisu i klienta. Funkcje programu podzielone są dla dwóch rodzajów użytkowników - pracownik serwisu oraz klient. Odpowiednio dla nich został utworzony ekran główny gdzie istnieje możliwość logowania dla nich oraz opcja utworzenia nowego klienta.

Sekcja przeznaczona dla serwisu pozwala na podgląd i obsługę zgłoszeń reklamacji oraz wyświetlenia statystyki wykonanych prac serwisowych w zadanym okresie. Przegląd reklamacji można filtrować ze względu na zakres daty lub stan reklamacji. Po przejrzaniu dowolnego zgłoszenia pracownik może zaktualizować jego stan na:

- Odrzucony
- Ukończony
- W trakcie naprawy

W części przeznaczona dla klienta istnieje możliwość zgłoszenia oraz podgląd statusu wszystkich dotychczasowych reklamacji. Dodatkowo, aplikacja pozwala na zarejestrowanie się w serwisie. Po przejściu do strefy rejestracji i uzupełnieniu wszystkich wymaganych pól, program automatycznie generuje i wyświetla id, które zostaje przypisane do nowego klienta.

### 3. Struktura oprogramowania JAVA

Program jest obsługiwany przez klasy sterujące interfejsem graficznym użytkownika:

- **DbFX** - klasa okna głównego programu podzielona na 3 części:
  - logowanie pracownika serwisu - sprawdza, czy podane dane znajdują się w bazie pracowników serwisu. Przenosi do okna sterowanego przez *DbFXServer*
  - logowanie klienta - sprawdza, czy podane dane znajdują się w bazie klientów. Przenosi do okna sterowanego przez *DbFXClient*
  - utworzenie nowego klienta - przenosi do okna sterowanego przez *DbFXNewClient*
- **DbFXServer** - klasa okna przeznaczonego dla pracowników serwisu zawierająca funkcje:
  - wyświetlenie reklamacji - możliwość filtrowania wyświetlanych danych ze względu na datę reklamacji oraz status
  - aktualizowanie statusu reklamacji - pozwala na aktualizowanie statusu reklamacji dla podanego id reklamacji
  - wyświetlanie statystyk - przenosi do okna sterowanego przez *DbFXStats*
- **DbFXStats** - klasa okna przeznaczonego do wyświetlania statystyk reklamacji i klientów. Wyświetla ona informacje o:
  - ilości klientów
  - ilości różnych statusów reklamacji
  - wykres kołowy wszystkich reklamacji
- **DbFXClient** - klasa okna przeznaczonego dla klientów serwisu zawierająca funkcje:
  - zgłoszenie nowej reklamacji - pozwala na rejestrację nowej reklamacji w systemie
  - podgląd reklamacji - pozwala na uzyskanie aktualnych informacji o wszystkich dotychczasowych reklamacjach klienta
- **DbFXNewClient** - klasa okna przeznaczonego dla nowych klientów zawierająca funkcje:
  - rejestracja w systemie - możliwość zarejestrowania się, a następnie zalogowania jako klient do aplikacji

W klasach sterujących użyto również klasy pomocnicze:

- **DBUtil** jest odpowiedzialna za połączenie, rozłączenie z bazą danych oraz wykonywanie zapytań i zwracanie ich wyników za pomocą klasy *ResultSet*.
  - *dbExecuteQuery* - służy do wykonywania zapytania SQL i zwracania wyniku za pomocą klasy *CachedRowSet*. W projekcie dodano klasę *CachedRowSetWrapper* oraz zależność w Maven, aby uzyskać dostęp do funkcjonalności tej klasy.
  - *dbExecuteUpdate* - służy do wykonywania zapytań modyfikujących bazę (np. wstawienie kolejnego rekordu).

- **Reclamations** zawierająca wszystkie informacje na temat reklamacji. Klasa ta wykorzystuje właściwości JavaFX Beans. Klasa wykorzystywana jest na potrzeby pracownika serwisu.
- **ReclamationsStatus** zawierająca informacje z widoku przeznaczonego tylko dla klienta. Klasa ta wykorzystuje właściwości JavaFX Beans.
- **SceneController** jest wykorzystywany do przekazania wprowadzonego id klienta z ekranu głównego programu do ekranu klienta. Informacja ta jest potrzebna do przefiltrowania na widoku tylko reklamacji dotyczących danego klienta.
- **AdminDAO** - zarządza powiązanymi z reklamacjami operacjami na bazie danych (wyszukiwanie, aktualizowanie). Klasa kontrolera musi być na bieżąco informowana o zmianach na liście reklamacji, dlatego należy wykorzystać klasę kolekcyjną ObservableList dla reklamacji na liście.
  - *getReclamationsList* - metoda konwertuje klasę ResultSet na ObservableList
  - *showAllReclamations* - metoda pozwalająca na wyświetlenie reklamacji. W zależności od liczby parametrów pozwala ona na przefiltrowanie tych danych bądź nie podając żadnego parametru wyświetlenie wszystkich.
  - *updateReclamation* - metoda ta aktualizuje informacje dla podanego id reklamacji
- **ClientDAO** - zarządza operacjami powiązanymi z reklamacjami na bazie danych (dodawanie, wyszukiwanie) oraz rejestracją nowego klienta. Klasa kontrolera musi być na bieżąco informowana o zmianach na liście reklamacji, dlatego należy wykorzystać klasę kolekcyjną ObservableList dla reklamacji na liście.
  - *getReclamationList(ResultSet resultSet)* - metoda konwertuje klasę ResultSet na ObservableList
  - *showReclamations(String id)* - metoda pozwalająca na wyświetlenie wszystkich reklamacji zgłoszonych przez danego klienta
  - *insertReclamation(String serialNo, String device, String type, String price, String date, String id)* - metoda dodająca nową reklamację na podstawie danych podanych przez klienta
  - *getNewID()* - metoda generująca id dla nowego klienta
  - *insertNewClient(String firstName, String lastName, String email, String phone, String password)* - metoda tworząca nowego klienta

## 4. Architektura MYSQL

Baza danych składa się z:

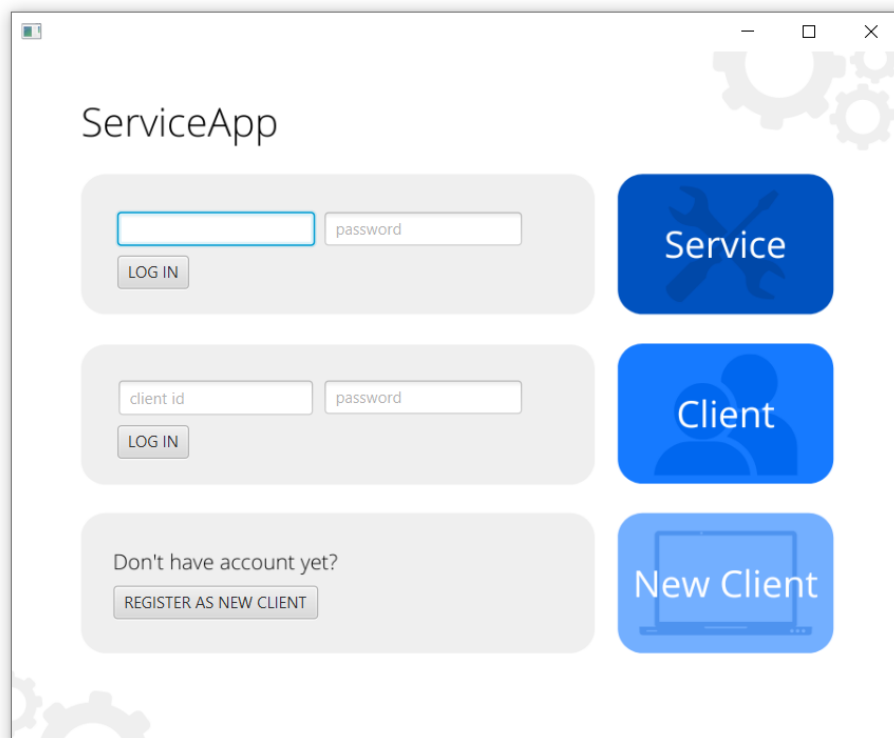
- **tabele:**
  - Clients - klienci serwisu
  - Reclamations - zgłoszone reklamacje
  - Device - zgłoszone urządzenia do reklamacji
  - login\_client - spis loginów i haseł dla klientów
  - login\_service - spis loginów i haseł dla pracowników serwisu
- **widoki:**
  - reclamation\_status\_view - widok zgłoszonych reklamacji
- **wyzwalacze:**
  - before\_reclamations\_insert - ustawienie statusu wprowadzanej reklamacji na 'reported'
- **funkcje:**
  - calculate\_rec\_price - obliczenie kwoty do zapłaty (w przypadku, gdy minął okres gwarancji)
- **procedury:**
  - check\_admin\_user - sprawdzenie, czy wprowadzony login i hasło są poprawne dla pracowników serwisu
  - check\_client\_user - sprawdzenie, czy wprowadzony login i hasło są poprawne dla klientów

Na potrzeby aplikacji stworzono trzech użytkowników posiadających uprawnienia:

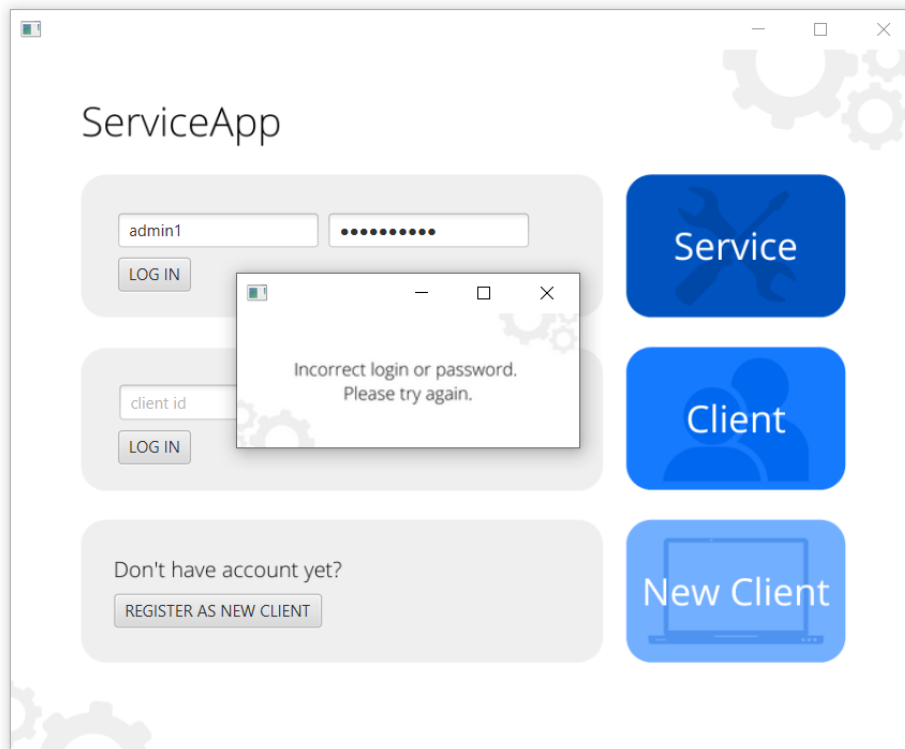
- **access@localhost**
  - EXECUTE na procedurach check\_admin\_user, check\_client\_user
- **admin@localhost**
  - SELECT na Reclamations, Clients
  - UPDATE na Reclamations
- **client@localhost**
  - SELECT na reclamation\_status\_view, Clients
  - INSERT na reclamation\_status\_view, Clients, Device, Reclamations, login\_client
  - EXECUTE funkcji calculate\_rec\_price
  - TRIGGER na tabeli Reclamations

## 5. Przykładowe działanie programu

Ekran początkowy:



Błędne wprowadzenie loginu lub hasła:



Zalogowanie się jako pracownik serwisu:

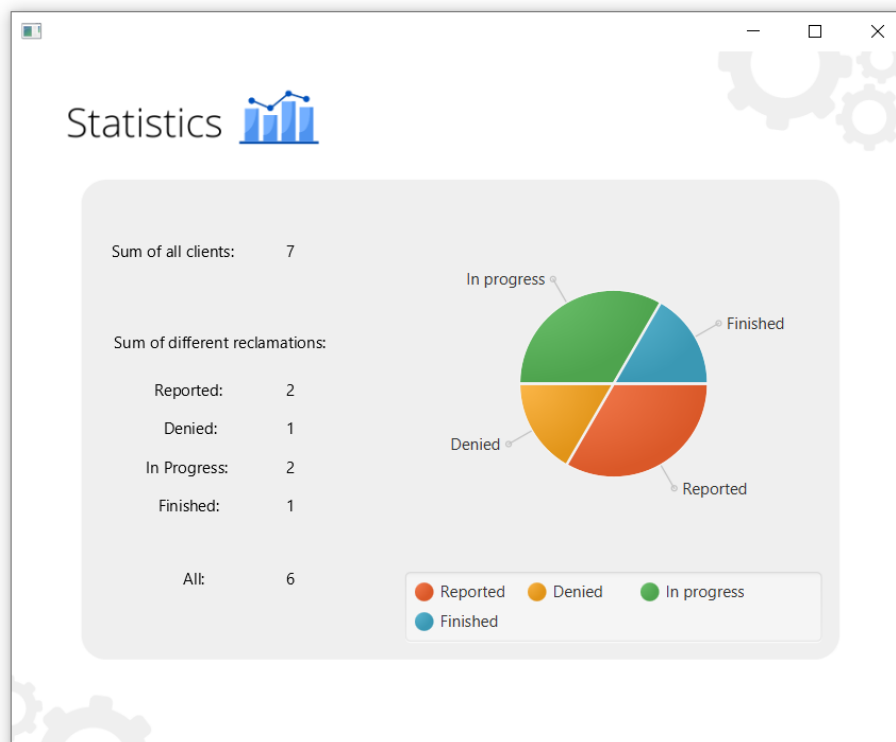
The screenshot shows the 'Service' application interface. At the top left, there is a blue square button with a wrench and screwdriver icon, labeled 'Welcome!' and 'Log out'. To the right, there are two main sections. The first section, 'Get recent statistics', contains a 'Show all' button. The second section, 'Update reclamtion status', contains a text input field labeled 'Enter id', a dropdown menu labeled 'Select statu...', and an 'Update' button. Below these sections, there is a date range selector with 'From' and 'To' fields, a 'Select status' dropdown, and a 'Show data' button. The 'From' field is set to 'Select date' and the 'To' field is set to '05.02.2022'. The 'Select status' dropdown is set to 'All'. Below the date range selector, there is a table with the following columns: ID, Price, Date, Client, Device Id, and Status. The table is currently empty, displaying 'No content in table'.

Wyświetlenie reklamacji w okresie 2008 - 2022 o statusie "In progress" oraz aktualizacja statusu reklamacji o id = 7:

The screenshot shows the 'Service' application interface with the search results for complaints. The 'Update reclamtion status' section now has the 'Enter id' field set to '7' and the 'Select statu...' dropdown set to 'In progress'. The 'Show data' button is highlighted. The table below shows the search results for complaints in the period 2008 - 2022 with the status 'In progress'.

ID	Price	Date	Client	Device Id	Status
1	0.0	2011-04-20	2	CGD826	In progress
6	0.0	2022-02-05	1	ASDE345	In progress
7	0.0	2022-02-05	1	BN789	In progress

Wyświetlanie statystyk serwisu:



Ekran widziany po zalogowaniu się jako klient:

The Client dashboard features a blue sidebar with a "Welcome!" message and a "Log out" button. The main content area includes a form to "Add new reclamtion" and a table for "My reclamations".

**Add new reclamtion form:**

- Input fields: 2, device name, type, serial no., device price, bought date.
- Buttons: Add

**My reclamations table:**

ID	Device	Serial no.	Price	Status
No content in table				

Dodanie nowej reklamacji i wyświetlenie aktywnych dla danego klienta (sprawdzenie czy gwarancja ulega unieważnieniu):

The 'Client' dashboard features a blue sidebar with a 'Welcome!' message and a 'Log out' button. The main content area includes a form to 'Add new reclamation' and a table of 'My reclamations'.

**Add new reclamation form:**

2	Samsung
Tablet	VTY678
1230	02.02.2018

**My reclamations table:**

ID	Device	Serial no.	Price	Status
2	Samsung gal...	CGD826	0.0	In progress
2	Samsung	VTY678	123.0	reported

Ekran rejestracji nowego użytkownika:

The 'New Client' registration form includes a blue sidebar with a 'Sign up for free!' message and a 'Back' button. The main content area has a 'Please provide all data' form and a 'Log in' section.

**Please provide all data form:**

first name	last name
email	
password	
phone number	Sign in

**Log in section:**

Your Client ID is...



Dodawanie nowego użytkownika (Client ID zostaje wygenerowane automatycznie):

New Client

Please provide all data

Awlyn Hamilton

a.hamilton@gmail.com

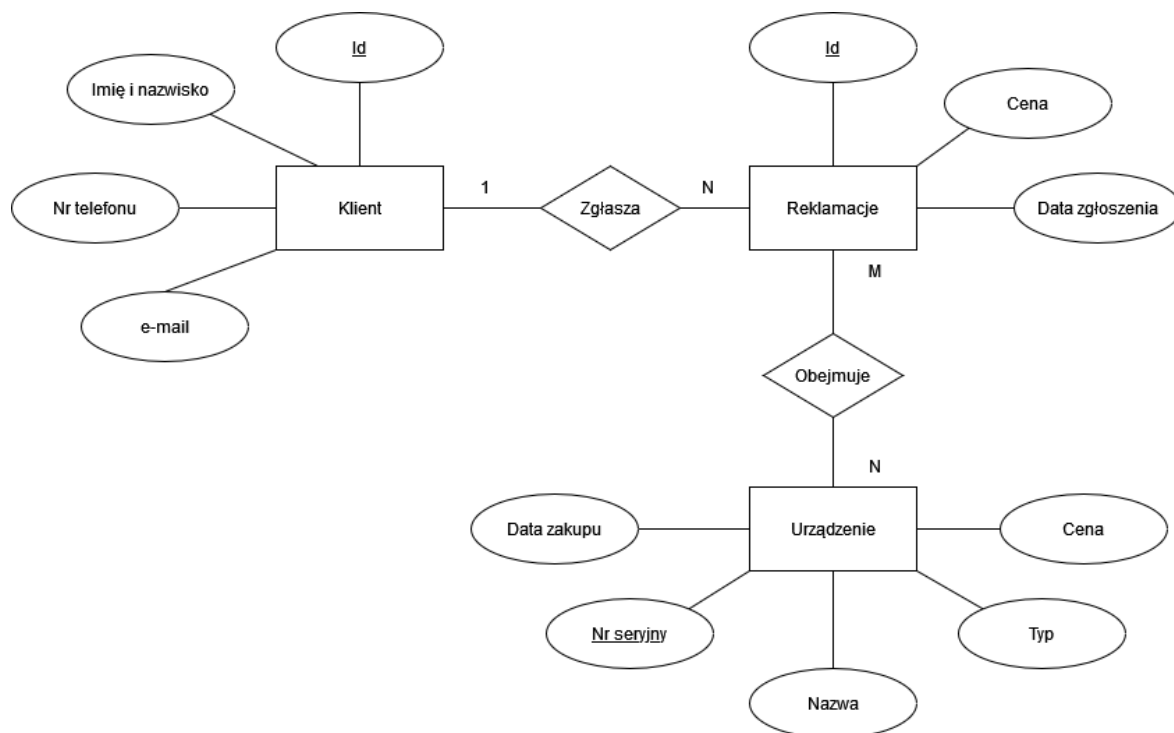
password123

456784320 Sign in

Back

Your Client ID is... 8 Log in

## 6. Diagram ER



## 7. Diagram EER

