

POLITECHNIKA WROCŁAWSKA  
WYDZIAŁ ELEKTRONIKI

PROJEKT z BAZ DANYCH

**System zarządzania warsztatem samochodowym**

Termin zajęć: Środa, 9:15–11:00

AUTOR/AUTORZY:

Marcin Bober

Indeks: 249426

E-mail: [249426@student.pwr.edu.pl](mailto:249426@student.pwr.edu.pl)

Rafał Rzewucki

Indeks: 248926

E-mail: [248926@student.pwr.edu.pl](mailto:248926@student.pwr.edu.pl)

PROWADZĄCY ZAJĘCIA:

dr inż. Roman Ptak, W4/K9

Wrocław 2020

## Spis treści

1.	Wstęp .....	3
1.1	Cel projektu.....	3
1.2	Zakres projektu.....	3
2.	Analiza wymagań .....	3
2.1	Opis działania i schemat logiczny systemu .....	3
2.2	Wymagania funkcjonalne .....	3
2.3	Wymagania niefunkcjonalne.....	4
2.3.1	Wykorzystywane technologie i narzędzia .....	4
2.3.2	Wymagania dotyczące bazy danych .....	4
2.3.3	Wymagania dotyczące bezpieczeństwa systemu .....	4
2.4	Przyjęte założenia projektowe .....	4
3.	Projekt systemu .....	5
3.1	Projekt bazy danych.....	5
3.1.1	Analiza rzeczywistości i uproszczony model konceptualny.....	5
3.1.2	Model logiczny i normalizacja .....	5
3.1.3	Model fizyczny i ograniczenia integralności danych .....	5
3.1.4	Inne elementy schematu – mechanizmy przetwarzania danych.....	6
3.1.5	Projekt mechanizmów bezpieczeństwa na poziomie bazy danych .....	6
3.2	Projekt aplikacji użytkownika.....	7
3.2.1	Architektura aplikacji i diagramy projektowe .....	7
3.2.2	Interfejs graficzny i struktura menu.....	7
3.2.3	Projekt wybranych funkcji systemu .....	8
3.2.4	Metoda podłączania do bazy danych – integracja z bazą danych .....	8
3.2.5	Projekt zabezpieczeń na poziomie aplikacji .....	8

# 1. Wstęp

## 1.1 Cel projektu

Warsztaty samochodowe często nie mogą sobie poradzić z zapisywaniem i zarządzaniem kolejką napraw pojazdów swoich klientów. Klienci warsztatów samochodowych chcieliby wiedzieć, kiedy ich pojazd będzie mógł zostać naprawiony. Celem projektu jest stworzenie systemu, który umożliwi zarządzanie kolejką napraw przez mechaników warsztatu, co będą mogli na bieżąco obserwować klienci warsztatu.

## 1.2 Zakres projektu

Serwis jest zaprojektowany dla dużego warsztatu samochodowego. Będzie gromadził w bazie danych informacje na temat klientów warsztatu, naprawianych pojazdów, dostępnych części oraz usterek, które zostały naprawione. Co więcej będzie przechowywany rejestr mechaników oraz zapis wszystkich wydarzeń z systemu. Aby umożliwić korzystanie z systemu, zostanie stworzona aplikacja webowa, która w przejrzysty sposób ułatwi dostęp do danych.

# 2. Analiza wymagań

## 2.1 Opis działania i schemat logiczny systemu

Gdy klient wykryje usterkę w swoim pojeździe będzie mógł dodać nową naprawę bezpośrednio do kolejki napraw w warsztacie. Natychmiast po dodaniu nowej naprawy mechanik będzie mógł odpowiednio zmienić status usterki w systemie w zależności od postępu w jej usuwaniu. Klient przez cały czas ma wgląd do tego co aktualnie jest robione w jego sprawie i gdy mechanik zakończy naprawę jego pojazdu, klient będzie wiedział, kiedy może go odebrać.

## 2.2 Wymagania funkcjonalne

Możliwe będzie logowanie jako klient, mechanik lub jako administrator.

Klient ma możliwość:

- dodawania swoich pojazdów do listy pojazdów,
- wprowadzania nowych aut do kolejki napraw,
- przeglądania stanu kolejki napraw,
- przeglądania statusu naprawy swojego pojazdu.

Mechanik ma uprawnienia do:

- organizacja wizyt w warsztacie,
- zarządzanie naprawą.

Administrator ma uprawnienia do:

- przypisywaniem ról do użytkowników,
- zarządzaniem rejestrem użytkowników oraz pojazdów,
- ma również możliwość zarządzać kolejką napraw.

## 2.3 Wymagania niefunkcjonalne

### 2.3.1 Wykorzystywane technologie i narzędzia

Aplikacja dostępowa zostanie zbudowana w języku wysokiego poziomu jakim jest PHP. Jest to prosty język wysokiego poziomu, który w prosty sposób pozwala zwizualizować dane dla użytkowników systemu oraz umożliwia interakcję z systemem. Dzięki niemu i wbudowanym w niego metodą możliwe jest połączenie z bazą danych oraz wykonywanie do niej zapytań.

### 2.3.2 Wymagania dotyczące bazy danych

Baza danych musi obsługiwać przechowywanie danych w formie tabel. Aplikacja będzie wymagać minimum 3 tabel o różnych rozmiarach, przy czym chcemy zachować możliwość rozbudowania aplikacji w przyszłości.

Szacunkowa liczba napraw w ciągu roku wynosi około 1000, przy czym musimy założyć, że każda naprawa będzie przeprowadzana na innym pojeździe. W najbardziej rozbudowanym przypadku każdy pojazd będzie miał innego właściciela, toteż zakładamy średnią ilość nowych użytkowników w ciągu roku na poziomie równym 1000. W sumie daje to około 3000 nowych rekordów w bazie danych w ciągu jednego roku. Dla każdego silnika baz danych jest to bardzo mała ilość, dlatego nie determinuje to wyboru silnika bazy.

Bardziej miarodajnym czynnikiem będzie ilość odczytów danych z bazy, ponieważ mechanicy w warsztacie prawdopodobnie będą cały czas zalogowani do systemu i co pewien okres czasu dane wyświetlane w serwisie będą musiały zostać zaktualizowane, tak aby zawsze można mieć dostęp do najnowszych danych.

Przy dodawaniu nowych danych do bazy będzie używany mechanizm transakcji, który umożliwi wycofanie wprowadzonych zmian, jeśli wystąpi jakikolwiek błąd przy dodawaniu. Jako silnik bazy danych został wybrany MariaDB z kilku istotnych powodów:

- jest on w pełni darmowy,
- jego uruchomienie nie wymaga praktycznie żadnej konfiguracji,
- nie ma wygórowanych wymagań sprzętowych,
- przechowuje dane również po zaniku zasilania,
- przez swoją prostotę budowy i działania operacje tak zapisu jak i odczytu z bazy będą wykonywane w bardzo krótkim czasie,
- posiada wbudowaną obsługę transakcji.

### 2.3.3 Wymagania dotyczące bezpieczeństwa systemu

Dla klientów zostanie zaprojektowany system logowania z użyciem adresów email i haseł. Hasła będą hashowane, a połączenia z serwerem będą szyfrowane co stanowi podstawę bezpieczeństwa systemu.

Dla mechaników zostanie zbudowany uproszczony system logowania przy pomocy pinu. Znacząco ułatwi to i przyspieszy pracę mechaników przy wprowadzaniu nowych danych.

## 2.4 Przyjęte założenia projektowe

- dostarczenie przystępnej aplikacji dla warsztatów i ich klientów,
- stworzenie stabilnej i odpornej na błędy aplikacji dostępowej do bazy danych,

- utworzenie aplikacji bazodanowej, która będzie gwarantować spójność oraz bezpieczeństwo danych.

### 3. Projekt systemu

#### 3.1 Projekt bazy danych

##### 3.1.1 Analiza rzeczywistości i uproszczony model konceptualny

W rzeczywistości występują klienci warsztatu samochodowego, ich pojazdy oraz usterki tych pojazdów. Wymagany jest model, który pozwoli jak najefektywniej przechowywać dane o każdym przedmiocie. Odzwierciedlenie tej części rzeczywistości w postaci modelu bazy danych zostało przedstawione na modelu logicznym jak i fizycznym bazy danych.

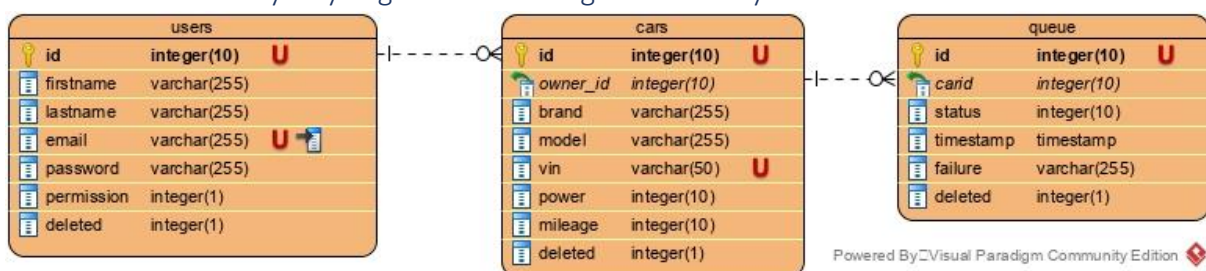
##### 3.1.2 Model logiczny i normalizacja

Baza danych będzie składać się z 3 tabel i zawierać wymienione dane:

1. Użytkownicy:
  - identyfikator rekordu,
  - imię, nazwisko, email,
  - hasło – w formie zaszyfrowanej,
  - uprawnienia użytkownika,
  - flaga czy rekord został usunięty.
2. Samochody
  - identyfikator rekordu,
  - identyfikator właściciela,
  - marka, model,
  - numer VIN, moc oraz przebieg,
  - flaga czy rekord został usunięty.
3. Kolejka napraw
  - identyfikator rekordu,
  - identyfikator naprawianego samochodu,
  - status naprawy,
  - czas dodania,
  - usterka do naprawy,
  - flaga czy rekord został usunięty.

Tabele zostaną połączone relacjami jeden do wielu, co zagwarantuje spójność danych. Nie będzie na przykład możliwości dodania nowego samochodu, który nie ma właściciela.

##### 3.1.3 Model fizyczny i ograniczenia integralności danych



#### 3.1.4 Inne elementy schematu – mechanizmy przetwarzania danych

W tabeli „users” zostanie utworzony index na kolumnie „email”, który usprawni wyszukiwanie użytkowników poprzez podanie adresu e-mail.

#### 3.1.5 Projekt mechanizmów bezpieczeństwa na poziomie bazy danych

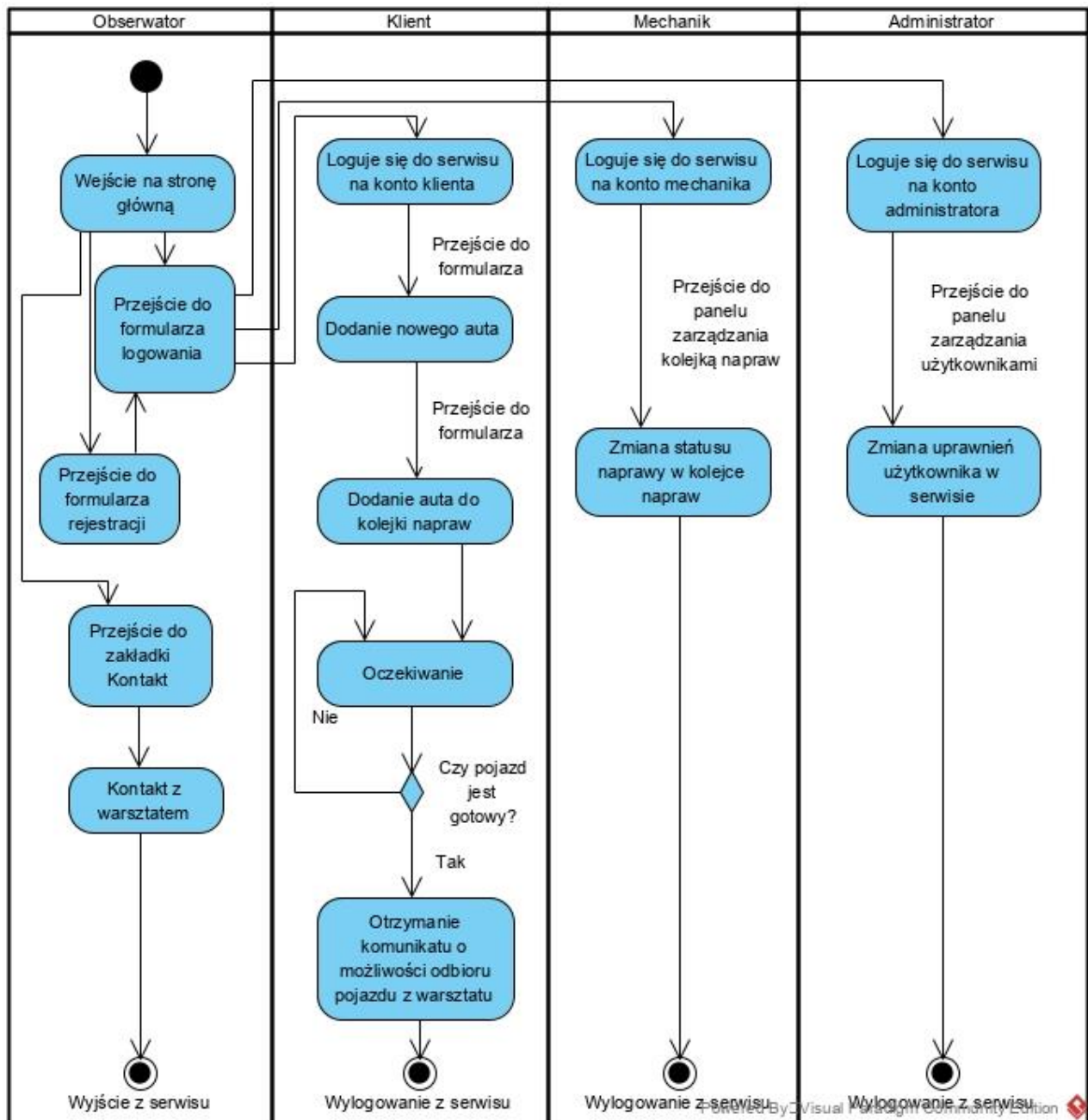
Zabezpieczenie przed nieuprawnionym dostępem będzie się odbywać poprzez zabezpieczenie bazy danych hasłem, które będzie przechowywane w formie zaszyfrowanej. Dostęp do hasła jak i do bazy danych będzie miał administrator, właściciel bazy danych oraz osoby wskazane przez właściciela.

Każdy mechanizm będzie miał do siebie przypisany pin, który będzie zmieniany co pewien okres czasu, za pomocą którego będzie mógł w szybki sposób zalogować się do serwisu.

Zabezpieczenie przed utratą danych będzie zapewnione poprzez wykonywanie cyklicznych kopii zapasowych całej bazy danych. Po uruchomieniu usługi w warsztacie można również rozszerzyć bezpieczeństwo poprzez przechowywanie dodatkowej bazy danych w chmurze jako kopię tylko do odczytu, która również może umożliwić odzyskać dane w przypadku awarii serwera umieszczonego w warsztacie.

## 3.2 Projekt aplikacji użytkownika

### 3.2.1 Architektura aplikacji i diagramy projektowe



### 3.2.2 Interfejs graficzny i struktura menu

Użytkownik ma możliwość przejścia w następujące miejsca w serwisie:

- Strona główna – wyświetla stronę główną serwisu,
- Kolejka napraw – wyświetla aktualną kolejkę napraw w warsztacie
- Panel Klienta – przechodzi do panelu klienta:
  - Dodaj auto – klient ma możliwość dodać nowe auto do serwisu,
  - Dodaj auto do kolejki – w tym miejscu klient może dodać swój pojazd do kolejki napraw.
- Kontakt – przenosi na stronę, na której znajdują się dane kontaktowe,
- Zaloguj – przenosi do formularza umożliwiającego zalogowanie,

- Wyloguj – wylogowanie użytkownika z serwisu.

### 3.2.3 Projekt wybranych funkcji systemu

Aplikacja ma zapewnić swobodny dostęp do kolejki napraw, dlatego wymagany jest przejrzysty interfejs, który pozwoli użytkownikowi w prosty sposób zalogować się do serwisu. Logowanie odbywać się będzie za pomocą adresu e-mail oraz hasła wprowadzonych w odpowiednim formularzu. Jeśli dane nie będą się zgadzać z tymi zapisanymi w bazie danych, zostanie wyświetlony odpowiedni komunikat. Po pomyślnym logowaniu użytkownik ma możliwość przejścia do Panelu Klienta, w którym będzie wyświetlona kolejka napraw.

### 3.2.4 Metoda podłączania do bazy danych – integracja z bazą danych

Połączenie z bazą danych będzie odbywać się poprzez rozszerzenie języka PHP o nazwie PDO. Zapewnia ono powłokę, którą można użyć, aby komunikować się z bazą danych i wykonywać wszelkie zapytania.

### 3.2.5 Projekt zabezpieczeń na poziomie aplikacji

Każdy użytkownik będzie miał przypisany indywidualny zestaw adresu e-mail i hasła, dzięki którym możliwe będzie zalogowanie do serwisu. Dodatkowo aplikacja będzie monitorowała zmianę adresu IP zalogowanego użytkownika, aby wykryć przechwycenie sesji przez innego użytkownika

Program przed dodaniem nowego użytkownika do bazy, jego email zostanie sprawdzony funkcje „filter\_input” z filtrem „FILTER\_VALIDATE\_EMAIL”, co więcej hasło zostanie zaszyfrowane poprzez użycie wbudowanej w język PHP funkcji password\_hash(). Domyślnie używa ona algorytmu bcrypt, który gwarantuje optymalną złożoność szyfrowania.