



**WYŻSZA SZKOŁA  
INFORMATYKI i ZARZĄDZANIA**  
z siedzibą w Rzeszowie

## **KOLEGIUM INFORMATYKI STOSOWANEJ**

**Kierunek: INFORMATYKA**

**Specjalność: Programowanie**

Przemysław Bęben  
w67219

### ***System do obsługi wypożyczalni samochodów i motocykli***

Prowadzący: mgr inż. Ewa Żesławska

**Praca projektowa programowanie obiektowe C#**

**Rzeszów 2023**



# Spis treści

<b>Wstęp</b>	<b>4</b>
<b>1 Opis założeń projektu</b>	<b>5</b>
1.1 Cele projektu . . . . .	5
1.2 Wymagania funkcjonalne i нефункционалне . . . . .	5
<b>2 Opis struktury projektu</b>	<b>7</b>
2.1 Diagram klas . . . . .	7
2.2 Opis klas . . . . .	8
2.3 Baza danych . . . . .	9
2.4 Wymagania minimalne . . . . .	11
<b>3 Harmonogram realizacji projektu</b>	<b>12</b>
3.1 Diagram Ganta . . . . .	12
3.2 System kontroli wersji . . . . .	12
<b>4 Prezentacja warstwy użytkowej projektu</b>	<b>13</b>
4.1 Strona logowania . . . . .	13
4.2 Panel Użytkownika . . . . .	13
4.2.1 Samochody . . . . .	14
4.2.2 Motocykle . . . . .	14
4.2.3 Moje wypożyczenia . . . . .	15
4.3 Panel Administratora . . . . .	15
4.3.1 Użytkownicy . . . . .	16
4.3.2 Pojazdy . . . . .	16
4.3.3 Wypożyczenia Aktywne . . . . .	18
4.3.4 Wypożyczenia Zakończone . . . . .	18
<b>5 Podsumowanie</b>	<b>20</b>
5.1 Zrealizowane prace . . . . .	20
5.2 Planowane dalsze prace . . . . .	20
<b>Bibliografia</b>	<b>21</b>
<b>Spis rysunków</b>	<b>22</b>

# Wstęp

W dzisiejszym świecie praktycznie każdy pełnoletni człowiek posiada uprawnienia do kierowania pojazdem: samochodem czy też motocyklem. Korzystanie z środków lokomocji ułatwia nam codzienne funkcjonowanie, pozwala nam oszczędzać cenny czas, a także potrafi sprawiać przyjemność. W związku z wszechobecną mobilnością na rynku istnieje rosnące zapotrzebowanie na wypożyczalnie pojazdów. Praktycznie każdy kojarzy gigantyczne firmy jak PANEK CarSharing, natomiast faktem jest, że powstaje coraz więcej małych firm specjalizujących się w wypożyczaniu samochodów, bądź motocykli. Coraz większą liczbę takowych przedsiębiorstw można zaobserwować głównie w mediach społecznościowych gdzie najczęściej są reklamowane. Firmy takie w zdecydowanej większości nie dysponują odpowiednimi środkami finansowymi na zlecenie firmie z branży IT na wykonanie dedykowanego systemu obsługi wypożyczalni, ani tym bardziej nie mają środków na posiadanie własnego działu IT. Z tego powodu obsługa takiego przedsiębiorstwa staje się mozolna, bardzo czasochłonna, zabiera mnóstwo energii i jest realnym problemem. I tu pojawia się moje rozwiązanie - aplikacja do obsługi wypożyczalni pojazdów, prosta, intuicyjna i rozwiązująca większość wyżej wymienionych problemów.

# Rozdział 1

## Opis założeń projektu

### 1.1 Cele projektu

Celem projektu jest utworzenie łatwej w obsłudze aplikacji obsługującej wypożyczalnię samochodów i motocykli. Aplikacja ma być ułatwieniem zarówno dla właścicieli wypożyczalni, jak i dla klientów. Właścicielom ma umożliwiać szybkie i efektywne zarządzanie klientami, oraz flotą pojazdów, natomiast użytkownikom wygodne i intuicyjne korzystanie z usług wypożyczalni. Problemem który będzie rozwiązywał mój projekt jest brak na rynku efektywnego narzędzia do zarządzania wypożyczalnią pojazdów na zakup którego mogłyby zdecydować się małe przedsiębiorstwa. Brak takiego systemu na rynku zdecydowanie zmniejsza efektywność pracy małych firm, a także naraża je na różnorakie błędy spowodowane uciążliwością i czasochłonnością manualnego monitorowania funkcjonowania przedsiębiorstwa. Do rozwiązania problemu, czyli stworzenia aplikacji niezbędne jest przeanalizowanie procesu prowadzenia przedsiębiorstwa z branży wypożyczalni, ażeby sprecyzować funkcjonalności które powinna posiadać aplikacja. Projekt zakłada kroki takie jak analiza wymagań, projektowanie interfejsu klienta i administratora, implementacja funkcjonalności aplikacji, oraz testowanie. Wynikiem prac będzie aplikacja umożliwiająca zarządzanie wypożyczalnią samochodów i motocykli, oferująca intuicyjny interfejs dla zarówno klientów, jak i pracowników wypożyczalni.

### 1.2 Wymagania funkcjonalne i нефункционалне

#### Wymagania funkcjonalne

- Proces logowania użytkownika, rozróżniający typ użytkownika(administrator bądź klient)
- Obsługa różnych rodzajów użytkowników
- Zarządzanie pojazdami, dodawanie i usuwanie pojazdów
- Określanie dostępności pojazdów
- Obsługa wypożyczeń
- Naliczanie opłat za wypożyczenie
- Tworzenie pokwitowań za wypożyczenia
- Śledzenie aktywnych i zakończonych wypożyczeń

#### Wymagania нефункционалне

- Bezpieczeństwo - dane użytkowników i pojazdów powinny być zabezpieczone
- Użyteczność - stworzenie prostego i intuicyjnego interfejsu użytkownika

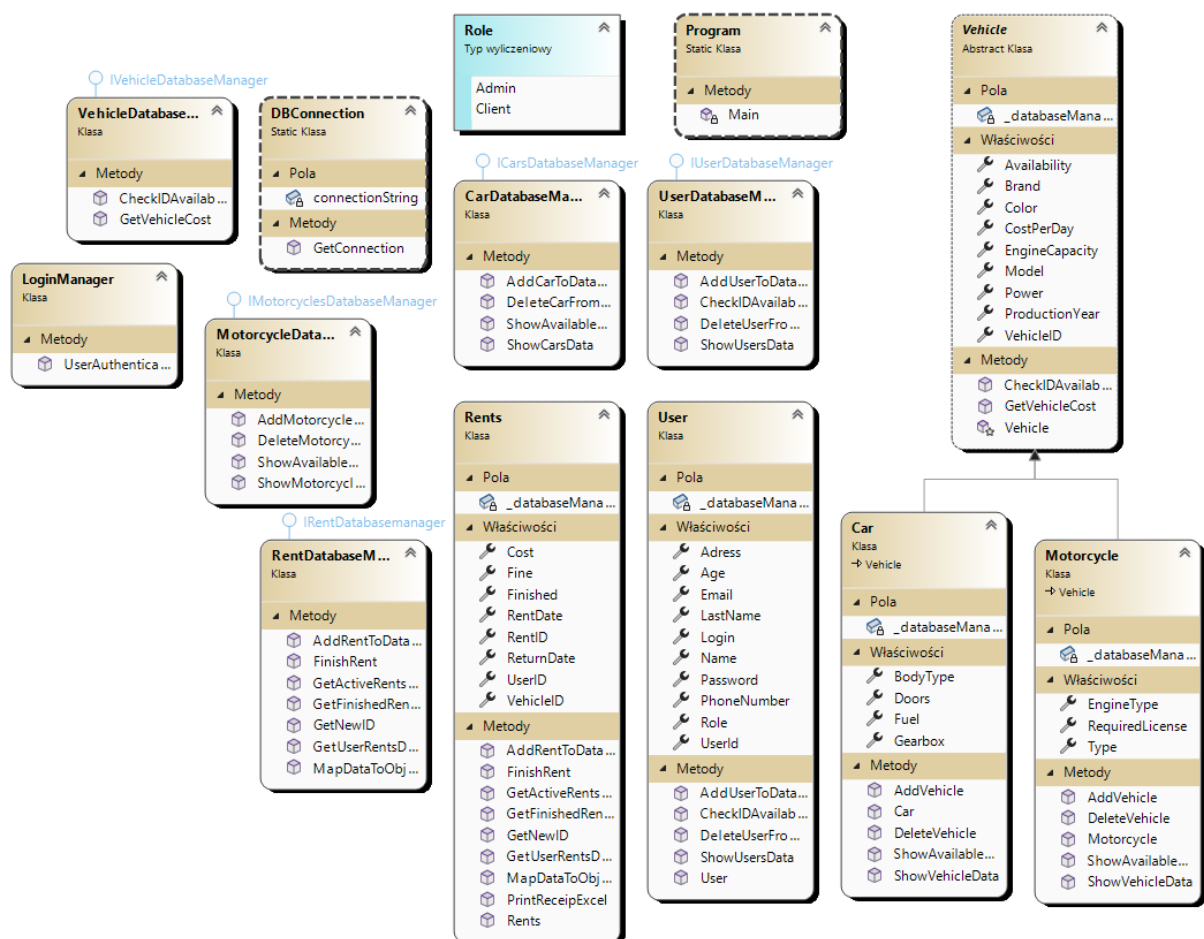
- Wydajność - zapewnienie szybkiego wykonywania operacji
- Dostępność - aplikacja powinna być dostępna przez większość czasu
- Koszty - aplikacja powinna być możliwie tania w wdrożeniu i utrzymaniu

# Rozdział 2

## Opis struktury projektu

Projekt utworzyłem w języku programowania C# wraz z technologią WindowsForms w środowisku Visual Studio. Projekt korzysta z bazy danych Microsoft SQL Server, której struktura została stworzona w programie Microsoft SQL Server Management Studio.

### 2.1 Diagram klas



Rysunek 2.1: Diagram klas

## 2.2 Opis klas

- **Vehicle** - klasa zawierająca właściwości wspólne dla motocykli i samochodów, oraz metody:
  - Metody klasy Vehicle używają metod klasy VehicleDatabaseManager, która jest odpowiedzialna za operacje dotyczące pojazdów na bazie danych
- **Car** - klasa dziedzicząca po klasie Vehicle, implementuje właściwości samochodu
  - Metody klasy Car używają metod klasy CarDatabaseManager, która jest odpowiedzialna za operacje dotyczące samochodów na bazie danych
- **Motorcycle** - klasa dziedzicząca po klasie Vehicle, implementuje właściwości motocykla
  - Metody klasy Motorcycle używają metod klasy MotorcycleDatabaseManager, która jest odpowiedzialna za operacje dotyczące motocykli na bazie danych
- **User** - klasa implementująca użytkowników
  - Metody klasy User używają metod klasy UserDatabaseManager, która jest odpowiedzialna za operacje dotyczące motocykli na bazie danych
- **Rents** - klasa implementująca wypożyczenia
  - **PrintReceiptExcel** - Metoda drukująca paragon za wypożyczenie jako plik .xlsx
  - Reszta metod klasy Rents używa metod klasy RentDatabaseManager, która jest odpowiedzialna za operacje dotyczące wypożyczeń na bazie danych
- **VehicleDatabaseManager** - klasa odpowiedzialna za operacje dotyczące klasy Vehicle na bazie danych
  - **CheckIdAvailability** - metoda sprawdzająca czy podane ID pojazdu nie znajduje się już w bazie danych
  - **GetVehicleCost** - metoda cenę za dzień wypożyczenia danego pojazdu
- **CarDatabaseManager** - klasa odpowiedzialna za operacje dotyczące samochodów na bazie danych
  - **AddCarToDatabase** - metoda dodająca obiekt typu Car do bazy danych
  - **DeleteCarFromDatabase** - metoda usuwająca samochód z bazy danych
  - **ShowCarsData** - metoda zwracająca dane wszystkich samochodów z bazy danych
  - **ShowAvailableCarsData** - metoda zwracająca dane niewypożyczonych samochodów
- **MotorcycleDatabaseManager** - klasa odpowiedzialna za operacje dotyczące motocykli na bazie danych
  - **AddMotorcycleToDatabase** - metoda dodająca obiekt typu Motorcycle do bazy danych
  - **DeleteMotorcycleFromDatabase** - metoda usuwająca motocykl z bazy danych
  - **ShowMotorcyclesData** - metoda zwracająca dane wszystkich motocykli z bazy danych
  - **ShowAvailableMotorcyclesData** - metoda zwracająca dane niewypożyczonych motocykli
- **UserDatabaseManager** - klasa odpowiedzialna za operacje dotyczące użytkowników na bazie danych
  - **AddUserToDatabase** - metoda dodająca obiekt typu User do bazy danych



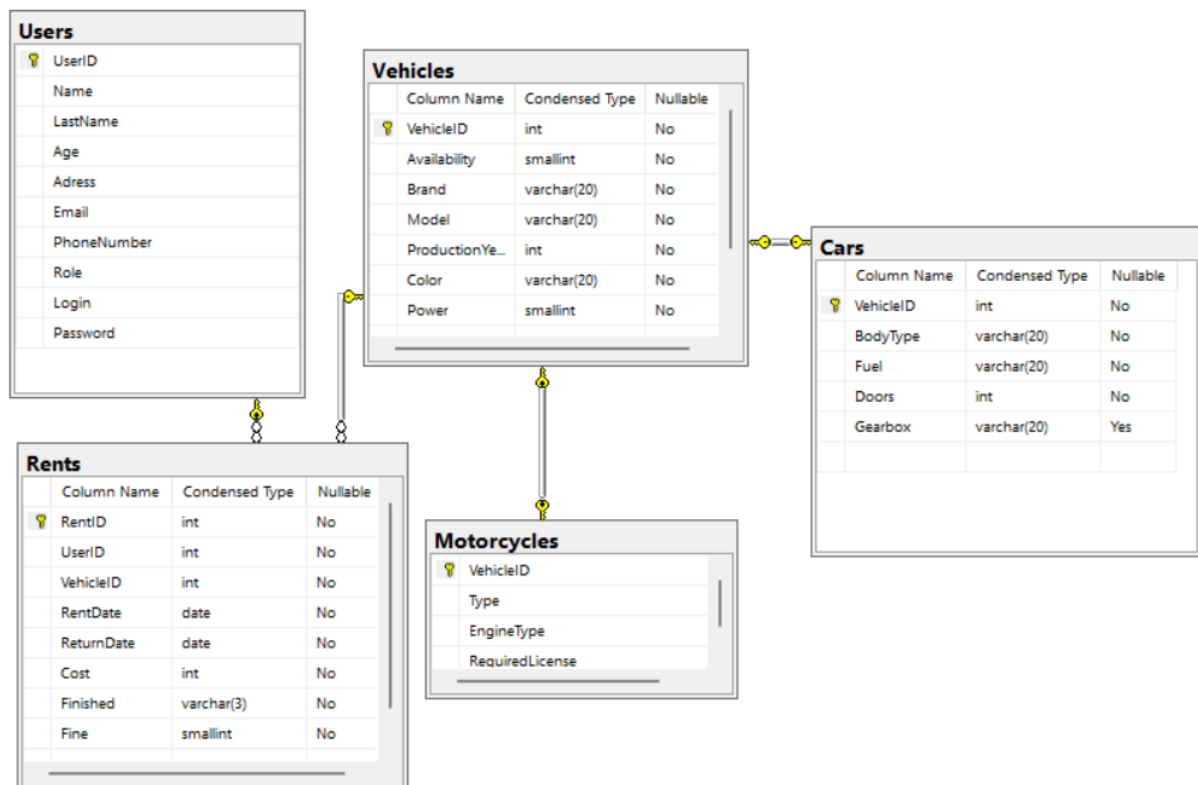
- **DeleteUserFromDatabase** - metoda usuwająca użytkownika z bazy danych
- **ShowUsersData** - metoda zwracająca dane użytkowników
- **CheckIDAvailability** - metoda sprawdzająca czy podane ID użytkownika nie istnieje już w bazie danych
- **RentDatabaseManager** - klasa odpowiedzialna na ioperacje dotyczące wypożyczeń na bazie danych
  - **AddRentToDatabase** - metoda dodająca obiekt typu Rents do bazy danych
  - **MapDataToObject** - metoda zwracająca dane wypożyczenia z bazy danych jako obiekt typu Rents
  - **FinishRent** - metoda kończąca aktywne wypożyczenie
  - **GetNewID** - metoda generująca nowe, wolne ID dla wypożyczenia
  - **GetActiveRentsData** - metoda zwracające dane aktywnych wypożyczeń
  - **GetFinishedRentsData** - metoda zwracająca dane zakończonych wypożyczeń
  - **GetUserRentsData** - metoda zwracająca historie wypożyczeń użytkownika o konkretnym ID
- **LoginManager** - klasa implementująca logowanie do aplikacji
  - **UserAuthentication** - metoda logowania
- **DBConnection** - klasa odpowiedzialna za połączenie z bazą danych
  - **GetConnection** - metoda zwracająca połączenie z bazą danych
- **Program** - główna klasa aplikacji
- **Role** - typ wyliczeniowy przechowujący możliwe role użytkownika

## 2.3 Baza danych

Baza Danych składa się z pięciu następujących tabel:

- **Users** - tabela przechowująca dane użytkowników, posiada kolumny:
  - **UserID** - kolumna przechowująca ID użytkownika
  - **Name** - kolumna przechowująca imie użytkownika
  - **LastName** - kolumna przechowująca nazwisko użytkownika
  - **Age** - kolumna przechowująca wiek użytkownika
  - **Adress** - kolumna przechowująca adres użytkownika
  - **Email** - kolumna przechowująca adres e-mail użytkownika
  - **PhoneNumber** - kolumna przechowująca numer telefonu użytkownika
  - **Role** - kolumna przechowująca role użytkownika
  - **Login** - kolumna przechowująca login użytkownika
  - **Password** - kolumna przechowująca hasło użytkownika
- **Rents** - tabela przechowująca dane wypożyczeń
  - **RentID** - kolumna przechowująca ID wypożyczenia

- **UserID** - kolumna przechowująca ID klienta który wypożycza pojazd
- **VehicleID** - kolumna przechowująca ID pojazdu który jest wypożyczany
- **RentDate** - kolumna przechowująca datę wypożyczenia
- **ReturnDate** - kolumna przechowująca deklarowaną datę zwrotu
- **Cost** - kolumna przechowująca cenę za dzień wypożyczenia
- **Finished** - kolumna przechowująca status wypożyczenia
- **Fine** - kolumna przechowująca ewentualną karę za opóźnienie zwrotu
- **Vehicles** - tabela przechowująca dane pojazdów bez podziału na motocykle i samochody
  - **VehicleID** - kolumna przechowująca ID pojazdu
  - **Availability** - kolumna przechowująca status dostępności pojazdu
  - **Brand** - kolumna przechowująca markę pojazdu
  - **Model** - kolumna przechowująca model pojazdu
  - **ProductionYear** - kolumna przechowująca rok produkcji pojazdu
  - **Color** - kolumna przechowująca kolor pojazdu
  - **Power** - kolumna przechowująca moc pojazdu
- **Cars** - tabela przechowująca dane specyficzne dla samochodu
  - **VehicleID** - kolumna przechowująca ID pojazdu
  - **BodyType** - kolumna przechowująca rodzaj nadwozia samochodu
  - **Fuel** - kolumna przechowująca paliwo odpowiednie dla danego samochodu
  - **Doors** - kolumna przechowująca liczbę drzwi w samochodzie
  - **Gearbox** - kolumna przechowująca rodzaj skrzyni biegów samochodu
- **Motorcycles** - tabela przechowująca dane specyficzne dla motocykli
  - **VehicleID** - kolumna przechowująca ID pojazdu
  - **Type** - kolumna przechowująca typ motocykla
  - **EngineType** - kolumna przechowująca rodzaj silnika motocykla
  - **RequiredLicense** - kolumna przechowująca wymagany rodzaj prawa jazdy dla konkretnego motocykla



Rysunek 2.2: Diagram bazy danych

## 2.4 Wymagania minimalne

Żeby uruchomić aplikację na komputerze jednostka ta musi spełniać następujące wymagania minimalne:

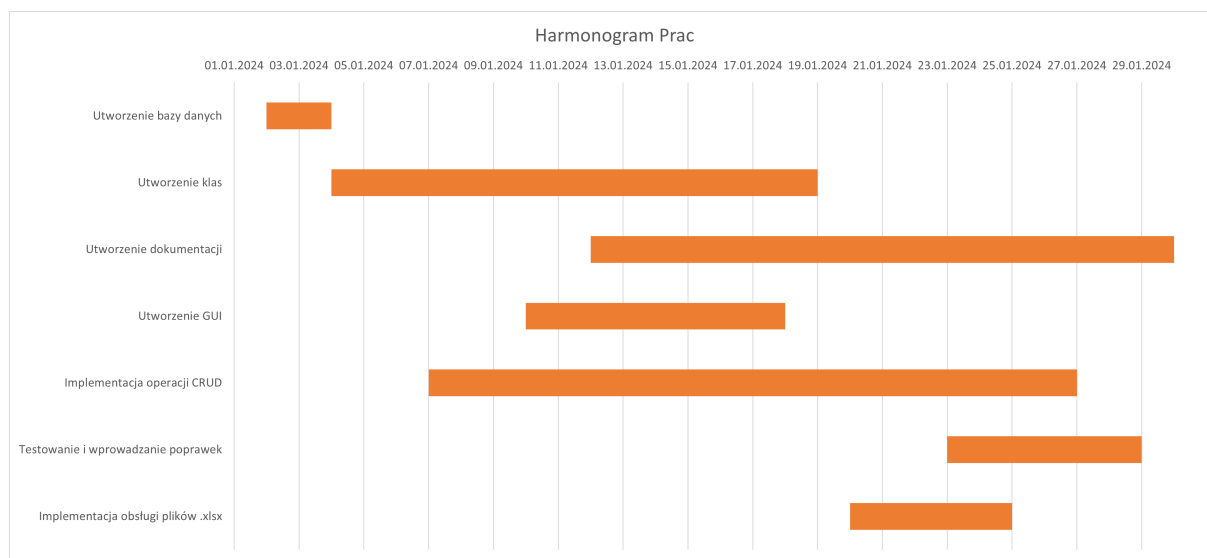
- **Procesor** - dwurdzeniowy procesor o częstotliwości 1 GHz lub równoważny
- **Pamięć RAM** - co najmniej 1 GB RAM
- **Dysk Twardy** - minimum 100 MB
- **Karta Graficzna** - zintegrowana karta graficzna
- **Adress** - kolumna przechowująca adres użytkownika
- **System Operacyjny** - Windows 7,8,10,11

# Rozdział 3

## Harmonogram realizacji projektu

### 3.1 Diagram Ganta

Diagram Ganta dla mojego projektu pokazuje, że najcięższymi wyzwaniami podczas realizacji projektu była implementacja operacji CRUD i utworzenie klas. Czasochłonne okazało się również tworzenie dokumentacji.



Rysunek 3.1: Diagram Ganta dla projektu

### 3.2 System kontroli wersji

Używanym przezemnie systemem kontroli wersji podczas pracy nad projektem był GIT. Repozytorium z plikami źródłowymi projektu dostępne jest w serwisie GitHub pod adresem: <https://github.com/pbeben3/VehicleRentalApp>

## Rozdział 4

# Prezentacja warstwy użytkowej projektu

### 4.1 Strona logowania

Po uruchomieniu aplikacji pokazuje się strona logowania. Strona posiada pola do wpisania loginu i hasła. Pola te posiadają walidacje na wypadek wpisania niepoprawnych danych. Użytkownik po wpisaniu danych i przejściu procedury logowania zostaje przekierowany do panelu administratora, bądź panelu użytkownika w zależności jaka rola jest przypisana do jego konta.



The image shows a login form on an orange background. At the top is a line-art icon of a car. Below it are two input fields: the first is labeled 'Login:' and the second is labeled 'Hasło:'. Under the password field is a button labeled 'Zaloguj się'. At the bottom of the form is a link labeled 'Wyjdź'.

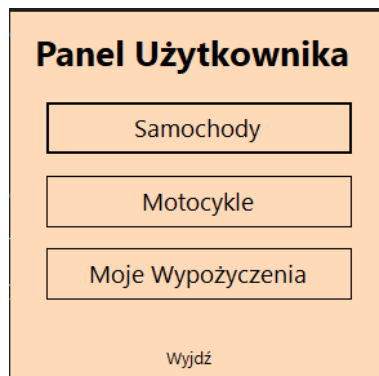
Rysunek 4.1: Panel logowania użytkownika

### 4.2 Panel Użytkownika

Po zalogowaniu się klienta ukazuje mu się Panel Użytkownika w którym ma trzy opcje do wyboru:

- Samochody
- Motocykle
- Moje Wypożyczenia

Oczywiście może też kliknąć w przycisk wyjdź, co zamknie aplikację i wyloguje go z konta.



Rysunek 4.2: Panel Użytkownika

### 4.2.1 Samochody

Po wybraniu opcji Samochody z Panelu Użytkownika klientowi pokazuje się panel zawierający listę dostępnych samochodów, oraz opcje wypożyczenia któregoś poprzez podanie jego numeru ID a następnie kliknięcie przycisku Rezerwuj. Pola do których użytkownik wprowadza dane posiadają walidację. Użytkownik posiada też opcję powrotu do Panelu Użytkownika poprzez kliknięcie Menu w lewym dolnym rogu.

Dostępne Samochody												
	VehicleID	Brand	Model	ProductionYear	Color	Power	EngineCapacity	BodyType	Fuel	Doors	Gearbox	CostPerD
▶	3	Ford	Mustang	2020	black	350	5000	Coupe	Gasoline	3	Automatic	1200
	4	Volkswagen	Golf	2012	White	105	12000	Hatchback	Diesel	5	Manual	450
•												

**Rezerwuj Samochód**  
 ID Samochodu:   
 Data Wypożyczenia:   
 Data Zwrotu:

[Menu](#)

Rysunek 4.3: Panel Samochody

### 4.2.2 Motocykle

W przypadku wybrania z Panelu Użytkownika opcji Motocykle użytkownikowi ukazuje się panel analogiczny do panelu Samochody, ale dotyczący wypożyczeń motocykli.

**Dostępne Motocykle**

	VehicleID	Brand	Model	ProductionYear	Color	Power	EngineCapacity	Type	EngineType	RequiredLicense	CostPerDay
▶	5	Yamaha	R1	2018	Blue	175	600	Sport	4 stroke	a	500
*											

**Rezerwuj Motocykl**  
 ID Motocykla: \_\_\_\_\_  
 Data Wypożyczenia: \_\_\_\_\_  
 Data Zwrotu: \_\_\_\_\_  

Rezerwuj

Menu

Rysunek 4.4: Panel Motocykle

### 4.2.3 Moje wypożyczenia

Ostatnią opcją jaką klient może wybrać z Panelu Administratora jest opcja Moje Wypożyczenia. Po wybraniu jej użytkownikowi wyświetli się panel w którym pokazane są wszystkie wypożyczenia dokonane z jego konta.

**Moje Wypożyczenia**

	RentID	UserID	Name	LastName	VehicleID	RentDate	ReturnDate	Cost	Finished	Fine
▶	1	2	Robert	Lewandowski	3	30.01.2024	05.02.2024	0	No	0
*										

Menu

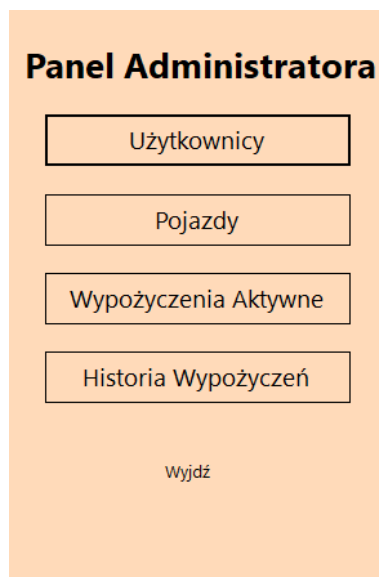
Rysunek 4.5: Panel Moje Wypożyczenia

## 4.3 Panel Administratora

Po zalogowaniu się przez administratora ukazuje się Panel Administratora który posiada cztery opcje do wyboru:

- Użytkownicy
- Pojazdy
- Wypożyczenia Aktywne
- Historia Wypożyczeń

Administrator może też kliknąć przycisk Wyjdź, co spowoduje zamknięcie aplikacji i wylogowanie go z konta.



Rysunek 4.6: Panel Administratora

### 4.3.1 Użytkownicy

Po wybraniu z Panelu Administratora opcji Użytkownicy użytkownikowi wyświetla się panel w którym widzi listę wszystkich użytkowników aplikacji, oraz może nimi zarządzać, dodawać, bądź usuwać. Istnieje opcja kliknięcia przycisku Menu i powrotu do Panelu Administratora.

#### Dodawanie Użytkownika

UserID:

Name:

LName:

Age:

Address:

Email:

PNumber:

Role:

Login:

Password:

#### Usuwanie Użytkownika

UserID:

UserID	Name	LastName	Age	Adress	Email	PhoneNumber	Role	Login	Password
1	Janusz	Kowalski	30	Rzeszów 120a	janusz@gmail.c...	123456789	admin	jkow	jkow
2	Robert	Lewandowski	21	Mielec 32	robert@gmail.c...	987654321	klient	rlew	rlew
3	Patryk	Rogal	40	Tyczyn 2	progal@gmail...	987465309	Client	progal	Progal123
4	Jan	Gruba	36	Kielnarowa 243	mgruba@gmail...	967513409	Client	mgruba	M12345kol
*									

Rysunek 4.7: Panel Użytkownicy

### 4.3.2 Pojazdy

Po wybraniu z Panelu Administratora opcji Pojazdy użytkownikowi ukazuje się panel administrowania pojazdami dostępnymi w wypożyczalni. Na panelu znajdują się lista samochodów, lista motocykli, pole do usuwania motocykla wraz z przyciskiem, pole do usuwania samochodu wraz z przyciskiem, przycisk Dodaj Samochód, który przekierowuje do formularza dodawania samochodu, przycisk Dodaj Motocykl przekierowujący do formularza dodawania motocykla, przycisk Odśwież, który odświeża listy pojazdów i przycisk Menu pozwalający na powrót do Panelu Administratora. Wszystkie pola do których użytkownik wprowadza wartość posiadają walidację.



**Dodaj Samochód**

**Usuń Samochód**

ID

Usuń

Samochody												
VehicleID	Availability	Brand	Model	ProductionYear	Color	Power	EngineCapacity	BodyType	Fuel	Doors	Gearbox	
1	0	Toyota	Rav4	2005	Green	150	2000	Terrain	Gasoline	5	Manual	
3	0	Ford	Mustang	2020	black	350	5000	Coupe	Gasoline	3	Automatic	
4	1	Volkswagen	Golf	2012	White	105	12000	Hatchback	Diesel	5	Manual	
*												

**Dodaj Motocykl**

**Usuń Motocykl**

ID

Usuń

Motocykle												
VehicleID	Availability	Brand	Model	ProductionYear	Color	Power	EngineCapacity	Type	EngineType	RequiredLicense	CostPerDay	
2	0	KTM	EXC300	2020	Orange	80	250	enduro	4 stroke	a2	300	
5	1	Yamaha	R1	2018	Blue	175	600	Sport	4 stroke	a	500	
*												

Menu

Odśwież

Rysunek 4.8: Panel Pojazdy

### Dodawanie Samochodu

VehicleID:

Brand:

Model:

PYear:

Color:

Power:

ECapacity:

CostPerDay:

BType:

Fuel:

Doors:

Gearbox:

Rysunek 4.9: Formularz dodawania samochodu

### Dodawanie Motocykla

VehicleID:

Brand:

Model:

PYear:

Color:

Power:

ECapacity:

CostPerDay:

Type:

EType:

RLicense:

Rysunek 4.10: Formularz dodawania motocykla

### 4.3.3 Wypożyczenia Aktywne

Po wybraniu z Panelu Administratora opcji Wypożyczenia Aktywne użytkownikowi ukazuje się panel wyświetlający listę trwających wypożyczeń. W dolnej części panelu użytkownik może wpisać ID aktywnego wypożyczenia po czym użyć przycisku zakończ. Spowoduje to zmianę statusu wypożyczenia na zakończone i naliczenie należności klientowi. Pole do wpisywania ID wypożyczenia posiada walidację. Ponadto na panelu znajduje się przycisk Menu przenoszący użytkownika spowrotem do Panelu Administratora.

**Aktywne Wypożyczenia**

	RentID	UserID	Name	LastName	VehicleID	RentDate	ReturnDate	Cost	Finished	Fine
▶	1	2	Robert	Lewandowski	3	30.01.2024	05.02.2024	0	No	0
*										

**Zakończ Wypożyczenie**  
ID:

Menu

Rysunek 4.11: Panel Aktywne Wypożyczenia

### 4.3.4 Wypożyczenia Zakończone

Po wybraniu z panelu Administratora opcji Wypożyczenia Zakończone użytkownikowi pokazuje się panel zawierający listę zakończonych wypożyczeń. Na dole panelu znajduje się pole w które użytkownik wpisze ID zakończonego wypożyczenia, a następnie może kliknąć przycisk Drukuj. Spowoduje to utworzenie paragonu danego wypożyczenia w pliku arkusza kalkulacyjnego Excel. Pole w którym użytkownik wprowadza ID posiada walidację. Na panelu dodatkowo znajduje się przycisk kMenu, którego kliknięcie spowoduje przejście spowrotem do Panelu Administratora.

Zakończone Wypożyczenia

	RentID	UserID	Name	LastName	VehicleID	RentDate	ReturnDate	Cost	Finished	Fine
▶	1	2	Robert	Lewandowski	3	30.01.2024	05.02.2024	-4800	Yes	0
*										

Drukuj Paragon

ID:

Drukuj

Menu

Rysunek 4.12: Panel Zakończone Wypożyczenia

# Rozdział 5

## Podsumowanie

### 5.1 Zrealizowane prace

W projekcie udało się zrealizować:

- Interfejs graficzny wykonany w technologii WindowsForms
- Zaprojektowanie oraz stworzenie bazy danych, a następnie nawiązanie z nią połączenia z poziomu aplikacji
- System logowania użytkownika
- Obsługę zarządzania pojazdami w bazie danych
- Obsługę zarządzania użytkownikami w bazie danych
- Obsługę zarządzania wypożyczeniami w bazie danych
- Funkcjonalność wypożyczania pojazdu przez klienta
- Generowanie paragonów, oraz zapisywanie ich do pliku .xlsx
- Walidację danych wprowadzanych przez użytkowników
- Obsługę wyjątków

### 5.2 Planowane dalsze prace

W przyszłości planuje wykonać następujące prace:

- Ulepszyć GUI poprzez dodanie elementów jak pola wyboru, czy rozwijane listy z wyborami, oraz ogólne wizualne poprawienie interfejsu
- Dodać metody automatycznie generującej ID pojazdu
- Zaimplementować metodę importującą dane pojazdu z pliku Excela

# Bibliografia

- [1] <https://learn.microsoft.com/en-us/dotnet/csharp/> z dnia 30.01.2024
- [2] <https://www.w3schools.com/cs/index.php> z dnia 30.01.2024
- [3] Jacek Matulewski, *C#. Lekcje programowania. Praktyczna nauka programowania dla platform .NET i .NET Core*, Helion, Gliwice 2021.

# Spis rysunków

2.1	Diagram klas . . . . .	7
2.2	Diagram bazy danych . . . . .	11
3.1	Diagram Ganta dla projektu . . . . .	12
4.1	Panel logowania użytkownika . . . . .	13
4.2	Panel Użytkownika . . . . .	14
4.3	Panel Samochody . . . . .	14
4.4	Panel Motocykle . . . . .	15
4.5	Panel Moje Wypożyczenia . . . . .	15
4.6	Panel Administratora . . . . .	16
4.7	Panel Użytkownicy . . . . .	16
4.8	Panel Pojazdy . . . . .	17
4.9	Formularz dodawania samochodu . . . . .	17
4.10	Formularz dodawania motocykla . . . . .	17
4.11	Panel Aktywne Wypożyczenia . . . . .	18
4.12	Panel Zakończone Wypożyczenia . . . . .	19