

# Wypożyczalnia samochodów

## Charakterystyka stanu rzeczywistego.

Projekt ma na celu usprawnienie procesu wynajmu samochodów osobowych. Klient wypożyczalni ma możliwość wypożyczenia na określony okres jednego auta w drodze wyboru z firmowej bazy pojazdów. Docelowo wypożyczalnia (franczyzobiorca) ma mieć swoje placówki we kilku głównych polskich miastach. Każda placówka dysponuje parkingiem z pewną flotą pojazdów. Klient po wypożyczeniu auta z danej placówki ma możliwości jego odstawienia do dowolnej innej. W przypadku odstawienia pojazdu do oddziału, z którego zostało wypożyczone, klient korzysta ze standardowej taryfy. Odstawienie auta do innej placówki wiąże się z wniesieniem dodatkowej opłaty. Dzięki temu zapewnia się w miarę równomierne rozmieszczenie liczby samochodów pomiędzy oddziałami.

Każde auto związane jest ze swoją własną taryfą dzienną oraz posiada pakiet ubezpieczeń wliczonych w stawkę za wynajem.

Klient wypożyczalni nie jest rozliczany z liczby przejechanych kilometrów. System nie uwzględnia mandatów drogowych.

## Wymagania projektowe

Wymagania projektowe zostały zebrane na podstawie dialogu z klientem.

K - Klient

T - Team

T: Ok, otwierasz **wypożyczalnię samochodów** i potrzebujesz **aplikację** - tyle wiemy. W związku z tym mamy mnóstwo pytań. Pierwsze podstawowe pytanie to: Jakiego rodzaju ma być ta aplikacja? Czy ma być to aplikacja webowa czy desktopowa i kto będzie z niej korzystał czy tylko **pracownicy firmy** czy również **wasi przyszli klienci**?

K: A więc chciałbym aby to była przede wszystkim **aplikacja**, która będzie głównym narzędziem pracy moich pracowników. W obecnych założeniach szybka, sprawna i łatwa wymiana informacji pomiędzy poszczególnymi oddziałami firmy to priorytet. Stronę internetową, na której nasi klienci będą mogli zarezerwować pojazd online również będzie istotnym elementem naszej wypożyczalni ale nie jest elementem niezbędnym. Dlatego chciałbym abyśmy najpierw skupili się na aplikacji do zarządzania flotą. Kiedy będzie ona już gotowa będę mógł powoli rozwijać interes, szkolić pracowników itd., wy w tym czasie będziecie pracować nad stroną internetową z rezerwacjami. Kiedy rezerwacje online będą gotowe wtedy rozpoczniemy kampanię reklamową i ruszymy pełną parą. Dlatego istnienie tej drugiej aplikacji również musicie wziąć pod uwagę przy planowaniu całego systemu aby łatwo i sprawnie później wszystko wdrożyć. Czy na tym etapie przychodzi wam jakiś pomysł jakie podejście do sprawy i technologie przy tworzeniu najlepiej wybrać?

T: Powiedziałeś "jedna i druga aplikacja", według nas na pewno lepiej będzie stworzyć jedną aplikację, w której pracownicy będą mieli po prostu większe **uprawnienia** i dostęp do modułu zarządzania flotą. Będzie na pewno szybciej, łatwiej i taniej zarówno stworzyć jak i później utrzymywać taką aplikację. Na pewno doradzamy stworzenie **aplikacji webowej**, dzięki temu dostęp do niej zarówno dla pracowników jak i waszych klientów będzie możliwy na każdym systemie operacyjnym oraz na każdym urządzeniu mobilnym, bez potrzeby tworzenia osobnych aplikacji mobilnych. Dodatkowo pracownicy będą mogli pracować zdalnie z własnego domu, bez konieczności używania VPNa.

K: Wspaniale. **Jedna zunifikowana aplikacja działająca na wszystkim co ma przeglądarkę.** Brzmi świetnie. Ok przejdźmy do ogólnego zarysu funkcjonalności. Zaczniemy od naszego klienta. Z jego strony będzie to wyglądało tak:

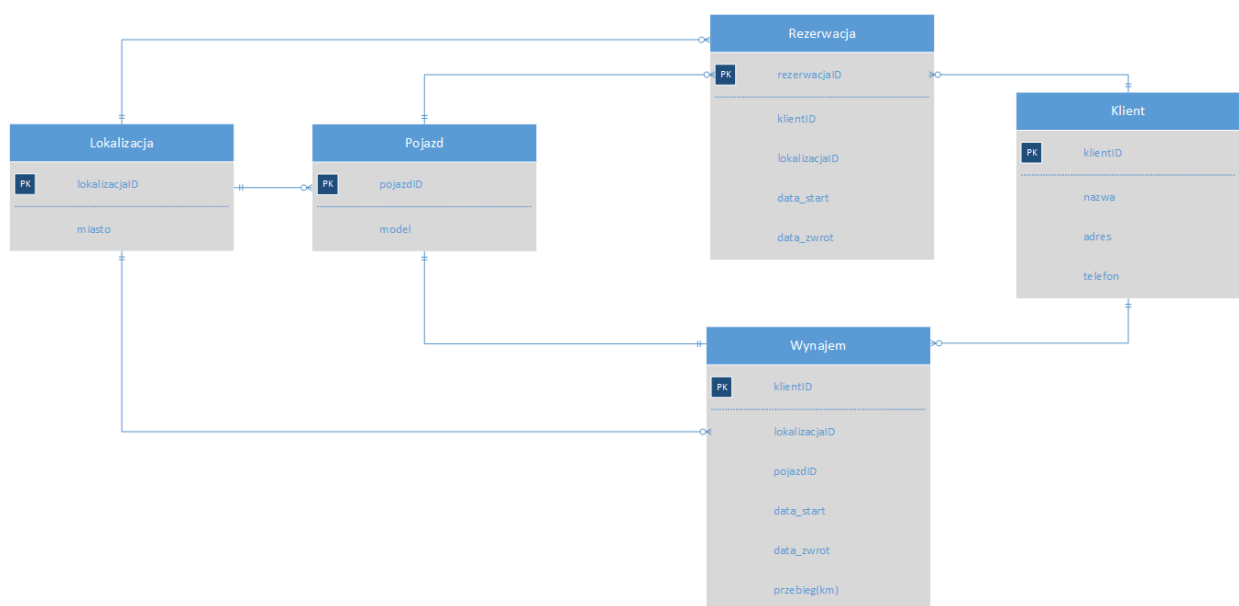
Klient może wejść na naszą stronę, wybrać **miasto** z którego chce wypożyczyć pojazd i zobaczyć **listę wszystkich dostępnych samochodów na daną chwilę wraz z ceną za wypożyczenie**. Z tego miejsca już może dokonać **rezerwacji** i zapłacić **kartą kredytową** jeśli posiada **konto w naszym serwisie** albo założyć nowe. Gdy dojedzie do **wypożyczalni** jedyne co musi wtedy zrobić to okazać prawo jazdy. Nasz pracownik sprawdzi jego dane, wyszuka **rezerwację** w naszym systemie i wyda kluczyki do pojazdu. W systemie jednocześnie zaznaczy, że dany pojazd nie znajduje się już na stanie. Jeśli klient nie chce korzystać z naszej strony to nie musi, może od razu podjechać do wypożyczalni, obejrzeć pojazdy i wybrać taki, który go interesuje. Jeśli chodzi o **zwrot pojazdu**, to klient będzie mógł go zwrócić do tej samej wypożyczalni lub do innego naszego oddziału, w innym mieście za **dodatkową opłatą**.

Z punktu naszego pracownika na początku w aplikacji będzie on musiał widzieć wszystkie **sprawne pojazdy** dostępne w wypożyczalni wraz ze wszystkimi ich szczegółami tj **przebieg**, **stan techniczny**, **stan wizualny**, **cena za wypożyczenie** i to jest moje wymaganie na pierwsze

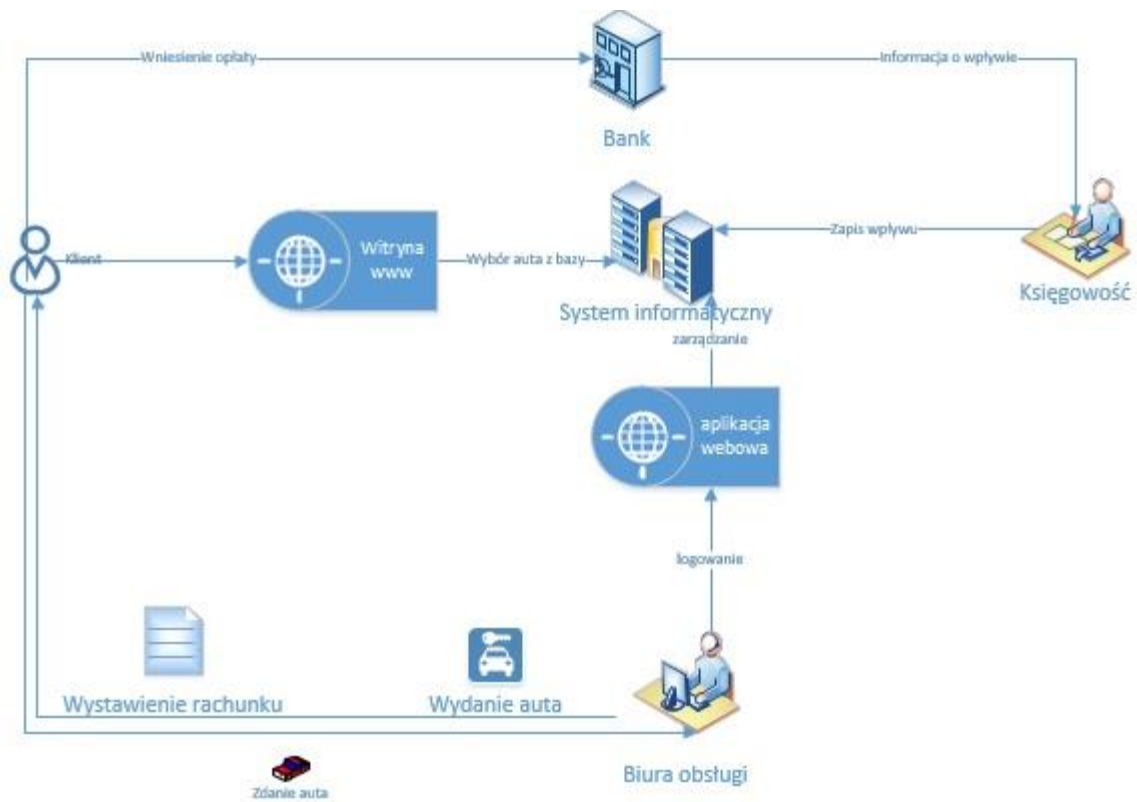
sprinty. Jeśli chodzi o następne, to priorytetowe dla mnie będą następujące funkcjonalności bez których nie będziemy w stanie rozpocząć działalności:

- wydawanie pojazdu dla klienta z bieżącego oddziału i w związku z tym **automatyczna aktualizacja dostępnych pojazdów w systemie**
- przyjmowanie pojazdu od klienta z dowolnego oddziału a co za tym idzie **automatyczne zaznaczenie tego pojazdu jako "do przeglądu"**. Inny pracownik będzie musiał sprawdzić jego **stan techniczny i wizualny** i w razie potrzeby wysłać pojazd do **naprawy lub wyczyszczenia** czy na myjnię. Dopiero po doprowadzeniu pojazdu do dobrego stanu lub stwierdzeniu, że zwrócony przez klienta pojazd nadaje się do ponownego wypożyczenia, zaznaczenie w systemie tego pojazdu jako **"gotowy do wypożyczenia"**
- dział myjni i dział techniczny możliwość wprowadzania informacji o podejmowanych przez nich czynnościach wraz ze wszystkimi kosztami wymiany części i napraw
- wydawanie **dyspozycji transportu** jednego lub wielu pojazdów jednocześnie do innego oddziału, jeśli w bieżącym oddziale jest nadwyżka pojazdów
- sprawdzanie stanu floty innych oddziałów (jeśli np. w jednym oddziale zacznie brakować pojazdów to pracownicy będą musieli zadzwonić do oddziału, który ma nadwyżkę i poprosić o transfer)
- podstawowa (jak na początek) **księgowość**, tj integracja z terminalem płatniczym, drukowanie paragonu, faktury VAT dla pracowników z działu obsługi klienta
- generowanie raportów** z przychodami i kosztami firmy dla księgowości. W przyszłości chciałbym tą gałąź dość mocno rozwinąć ale na początek niech to będzie proste zestawienie.

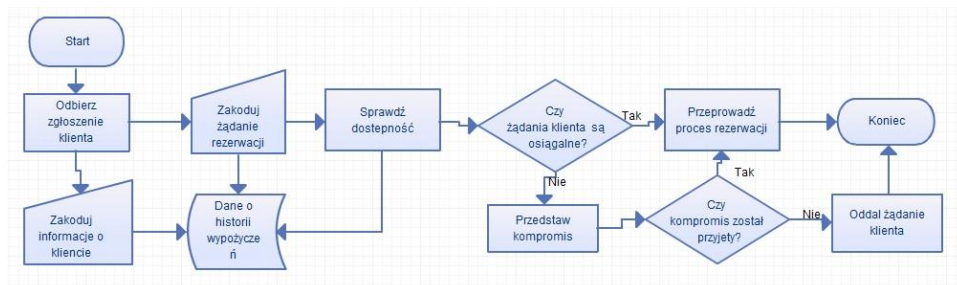
## Diagram ERD



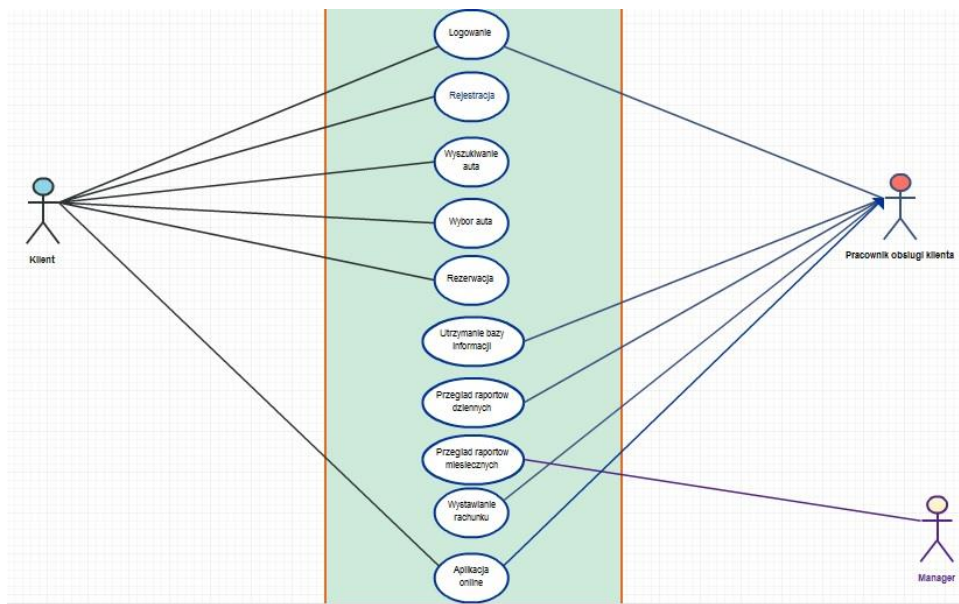
## Diagram WFD



## Diagram BPM - odbiór zgłoszenia klienta (osobiste albo telefoniczne)



## Diagram USECASE



## Wzorzec projektowy

Jako że ma to być aplikacja nowoczesna powinna ona być zgodna z wzorcem projektowym **MVC** -Model, Widok (warstwa prezentacji) , Kontroler (obsługuje zadania usera). Aplikacje Django mają narzucony z góry ten wzorzec. Plik odpowiedzialny za bycie kontrolerem nazywa się **Views**, pliki widoków umieszczone są w folderze **templates** (szablony) natomiast plik odpowiedzialny za model to **models.py** zawiera zmienną model (auta) , zmienną content przechowującą zawartość - opis auta i zmienną published data dodania.

Projekt aplikacji prezentuje prawidłowe działanie wzorca model-widok-kontroler. Aplikacja dynamicznie pobiera dane z bazy i wprowadza do przeglądarki. Główna strona indeksowa jest szablonem pobierającym rekordy z bazy, jedno auto to jeden rekord.

Wprowadzona jest dwupoziomowa hierarchia szablonów: **specyficzne** dla rekordów oraz szablon **główny**, dla całego projektu. Określony rekord jest otoczony zewnętrzną zawartością. Zewnętrzny element z menu i stopką jest elementem nadrzędnym, a treść elementem podrzędnym.

## Zarządzanie konfiguracją

Aplikacja posiada wbudowany panel administracyjny. Osoba posiadająca odpowiednie uprawnienia, w domyśle admin, może zarządzać aplikacją z poziomu przeglądarki. Tworząc bazę danych utworzyliśmy jednocześnie konto superużytkownika (python

to pojawi się pole logowania. Oprócz dodawania i przeglądania użytkowników w panelu została zarejestrowana nasza aplikacja poprzez utworzenie nowego skryptu pythona "admin.py" dzięki temu jako admin mamy dostęp do całej treści bazy.

Już na etapie rozwoju oprogramowania można ułożyć pewne historyjki odzwierciedlające zachowania użytkownika względem aplikacji (testy behawioralne). Przykładowa historyjka do przetestowania może brzmieć następująco: Jako zalogowany użytkownik po kliknięciu "wyloguj" powinienem otrzymać komunikat o prawidłowym wylogowaniu. Pozytywny wynik testów definiuje, że wymagania zostały spełnione.

Demo aplikacji połączone jest z bazą danych SQLITE, zawierającą m.in. tabelę z listą aut.

Lista aut może być uzupełniana/modyfikowana zarówno z poziomu bazy, jak również z poziomu panelu administracyjnego:

Oczywiście klient zarządzający aplikacją będzie mógł w przyszłości dodawać własne tabele, modyfikować je, dodawać kolumny itp. bez konieczności kontaktu z twórcami aplikacji. Jako następną w kolejności można dodać np. tabelę z listą klientów.

Widok dodawania z poziomu bazy danych:

Table Name: articles\_article

Enter Field Values

|                           |                                  |   |   |
|---------------------------|----------------------------------|---|---|
| 1. id ( integer )         | Primary key (will autoincrement) | ➡ | 🔗 |
| 2. title ( varchar(150) ) | Empty string                     | ➡ | 🔗 |
| 3. content ( text )       | Empty string                     | ➡ | 🔗 |
| 4. published ( datetime ) | Empty string                     | ➡ | 🔗 |
| 5. image ( varchar(100) ) | Empty string                     | ➡ | 🔗 |

Lub z poziomu panelu administracyjnego:

Administracja Django

Początek > Articles > Articles > Dodaj article

Witaj, admin. Pokaż stronę / Zmiana hasła / Wyloguj się

**Dodaj article**

Tytuł:

Zawartość:

Data publikacji: Data:  Dzisiaj Czas:  Teraz

Obrazek:  Nie wybrano pliku.

Program posiada opcję rejestracji i logowania



Logowanie | Rejestracja

CMS Start Artykuły Kontakt

Użytkownik:

Hasło:

Zaloguj

TWOJE KONTO

Ustawienia profilu

Powiadomienia

Wyloguj

Widok po zalogowaniu:

Logowanie | Rejestracja

CMS Start Artykuły Kontakt

**Witaj: admin**

Wyloguj się

TWOJE KONTO

Ustawienia profilu

Powiadomienia

Wyloguj

I po wylogowaniu:

Logowanie | Rejestracja

CMS Start Artykuły Kontakt

**Prawidłowe wylogowanie**

TWOJE KONTO

Ustawienia profilu

Powiadomienia

Wyloguj

Oprócz prostego zarządzania treścią z poziomu przeglądarki z dowolnego komputera z przeglądarką admin ma możliwość łatwego rozwijania aplikacji np. poprzez dodawanie wtyczek, choćby przykładowo moduł obsługujący płatności PayPal. W przypadku, gdy z biegiem czasu rozrośnie się lista aut możliwych do wypożyczenia admin może dodać obsługę paginacji. Bez jakichkolwiek zmian w widokach cała lista zostanie automatycznie podzielona wg indywidualnych upodobań.

Kwestią dalszego rozwinięcia projektu jest m.in. powiązanie klienta z wypożyczonym autem, datą jego wypożyczenia i planowaną datą zwrotu oraz tworzenie archiwum wypożyczeń.