

System Zarządzania Treścią
Jakub Krawczyk 122003

RENT-CAR

1 Opis systemu CMS

CMS został stworzony na potrzeby firmy zajmującej się wypożyczaniem sportowych samochodów. Strona prezentuje flotę firmy oraz umożliwia kontakt z administratorami witryny w celu np. wynajęcia pojazdu. Strona wraz z jej treścią generowana jest na podstawie danych zamieszczonych przez administratora strony. Klient może jedynie przeglądać zawartą treść. Administrator natomiast steruje tym jaką ofertę w danym momencie pokazywać, ile pojazdów ma pojawić się w rzędzie czy oznaczyć pojazd jako wynajęty. Administrator dodaje pojazd tj. podaje jego szczegółowe informacje oraz wskazuje zdjęcie samochodu, które zostanie wrzucone na serwer. Po tej operacji dodaje nową ofertę. Strona nie oferuje możliwości płatności.

2 Opis części systemu

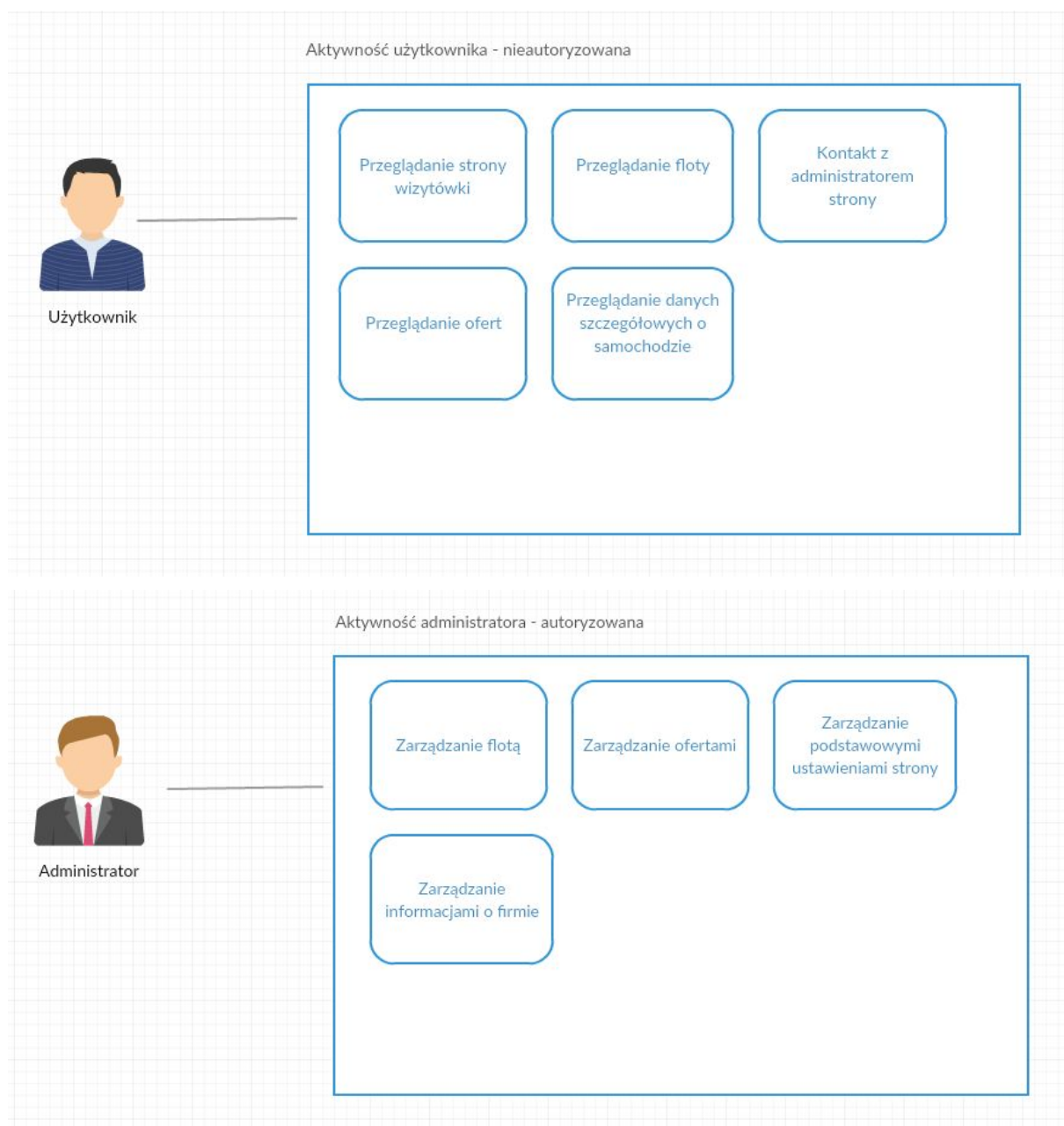
Witryna składa się z czterech podstron. Trzy z nich widoczne w głównej podstronie aplikacji: Main, Rent, Contact oraz Details.

- Main: strona główna aplikacji. Wizytówka wypożyczalni samochodów. Statyczna podstrona.
- Rent: podstrona, która pozwala przejrzeć dostępną flotę oraz ofertę wypożyczalni
- Contact: podstrona kontaktowe. Pozwala nawiązać dialog mailowy z właścicielem wypożyczalni. Pozwala zidentyfikować miejsce firmy na mapie.
- Details: podstrona wyświetlająca szczegółowe informacje o pojeździe.

3 Opis ról i uprawnień

Aplikacja nie posiada rozbudowanego systemu ról i uprawnień. Aplikacja rozróżnia jedynie gościa oraz administratora aplikacji. Brak możliwości rejestracji.

4 Diagram funkcjonalności



5 Schemat struktury bazy danych

Struktura bazy danych nie jest skomplikowana. Posiada standardowe tabele predefiniowane przez framework. W swoim projekcie utworzyłem dodatkowe cztery tabele. Reprezentują one takie obiekty aplikacji jak samochód, oferta, wiadomość czy kontakt oraz relacje między nimi.

▼	rent_car		CREATE TABLE "rent_car" ("brand" varchar(20) NOT NULL, "name" varchar(20) NOT NULL, "productionYear" integer NOT NULL, "course" integer NOT NULL, "engineCapacity" decimal NOT NULL, "power" integer NOT NULL, "color" varchar(10) NOT NULL, "carUUID" char(32) NOT NULL, "photo" varchar(100) NOT NULL, "description" text NOT NULL)
	brand	varchar (20)	`brand` varchar (20) NOT NULL
	name	varchar (20)	`name` varchar (20) NOT NULL
	productionYear	integer	`productionYear` integer NOT NULL
	course	integer	`course` integer NOT NULL
	engineCapacity	decimal	`engineCapacity` decimal NOT NULL
	power	integer	`power` integer NOT NULL
	color	varchar (10)	`color` varchar (10) NOT NULL
	carUUID	char (32)	`carUUID` char (32) NOT NULL
	photo	varchar (100)	`photo` varchar (100) NOT NULL
	description	text	`description` text NOT NULL
▼	rent_contact		CREATE TABLE "rent_contact" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "email" varchar(254) NOT NULL, "name" varchar(255) NOT NULL, "surname" varchar(255) NOT NULL, "phoneNumber" varchar(128) NOT NULL, "city" varchar(50) NOT NULL, "address" varchar(100) NOT NULL, "latitude" decimal, "longitude" decimal)
	id	integer	`id` integer NOT NULL PRIMARY KEY AUTOINCREMENT
	email	varchar (254)	`email` varchar (254) NOT NULL
	name	varchar (255)	`name` varchar (255) NOT NULL
	surname	varchar (255)	`surname` varchar (255) NOT NULL
	phoneNumber	varchar (128)	`phoneNumber` varchar (128) NOT NULL
	city	varchar (50)	`city` varchar (50) NOT NULL
	address	varchar (100)	`address` varchar (100) NOT NULL
	latitude	decimal	`latitude` decimal
	longitude	decimal	`longitude` decimal
▼	rent_message		CREATE TABLE "rent_message" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(255) NOT NULL, "email" varchar(254) NOT NULL, "message" varchar(255) NOT NULL, "isReaded" bool NOT NULL)
	id	integer	`id` integer NOT NULL PRIMARY KEY AUTOINCREMENT
	name	varchar (255)	`name` varchar (255) NOT NULL
	email	varchar (254)	`email` varchar (254) NOT NULL
	message	varchar (255)	`message` varchar (255) NOT NULL
	isReaded	bool	`isReaded` bool NOT NULL
▼	rent_offer		CREATE TABLE "rent_offer" ("available" bool NOT NULL, "price" decimal NOT NULL, "car_id" char(32) NOT NULL, "offerUUID" char(32) NOT NULL)
	available	bool	`available` bool NOT NULL
	price	decimal	`price` decimal NOT NULL
	car_id	char (32)	`car_id` char (32) NOT NULL
	offerUUID	char (32)	`offerUUID` char (32) NOT NULL

6 Opis użytych technologii i rozwiązań

Aplikacja została stworzona z wykorzystaniem znanego frameworka CMSowego - Django. Narzędzie dostarcza wiele gotowych funkcjonalności tj. autoryzacja użytkownika, ORM czy panel zarządzania aplikacją (panel administracyjny). Konfiguracja oraz instalacja frameworka zajmuje dosłownie kilka minut. Wykorzystując dodatkowe paczki możemy otrzymać mocno spersonalizować naszą aplikację czy panel administracyjny. Pobrane paczki to np. 'django-colorful', 'django-tinymce' dodające kontrolkę wybory koloru z panelu administracyjnego czy dodanie RichTextEditora w panelu. Routing aplikacji mamy "out of the box" wykorzystując jedynie plik konfiguracyjny.

Aplikacja ze względu na swoją prostotę nie wykorzystuje frameworków JavaScriptowych. W użyciu jest HTML oraz framework CSS'owy - Bootstrap.

Do analizy ruchu wykorzystano Google Analytics pozwalające na stworzenie niemalże każdego rodzaju raportu. Dzięki niemu potrafimy identyfikować zachowania użytkownika w naszej aplikacji. Wiemy gdzie najchętniej zagląda a jakie obszary aplikacji są dla niego nieosiągalne.

Dodatkowo wykorzystano Google Maps w celu identyfikacji lokalizacji miejsca firmy.

Brand:	<input type="text" value="Porsche"/>
Name:	<input type="text" value="Cayman"/>
ProductionYear:	<input type="text" value="2019"/>
Course:	<input type="text" value="100"/>
EngineCapacity:	<input type="text" value="2,5"/>
Power:	<input type="text" value="365"/>
Color:	<input type="text" value="Black"/>
Description:	<div><div><div>B <i>I</i> <u>U</u> ABC Styles Paragraph</div><div> </div><div>Font Family Font Size </div></div><div><p>A super fast car for demanding people</p></div><div>Path: p Words: 7</div></div>
Photo:	<div>Currently: gallery/porshecayman.jpg <input type="button" value="Clear"/></div> <div>Change: <input type="button" value="Wybierz plik"/> Nie wybrano pliku</div>

Zrzut ekranu pokazujący przykładowe okno edycji samochodu. Mamy tutaj m.in. predefiniowane pole z wyborem koloru, pole tzw. RichTextEditor oraz pole to uploadu nowego zdjęcia samochodu.

7 Instrukcja instalacji

Repozytorium z kodem aplikacji znajduje się [tutaj](#). Aplikacja została napisana w pełni w Django. Django wykorzystuje język programowania Python w najnowszej wersji, dlatego od niego powinniśmy rozpocząć zabawę. Pełny opis instalacji Pythona oraz Django można znaleźć [tutaj](#). Aby uruchomić serwer należy wpisać następującą komendę `'python manage.py runserver'`. Istnieje ryzyko braku wszystkich bibliotek. W tym celu interpreter pythona wypisze nam instrukcje jak zainstalować brakujące biblioteki.

8 Instrukcja użytkownika

Po pomyślnej instalacji oraz uruchomieniu serwera administrator ma już pełny dostęp do naszej aplikacji. Po wejściu na adres serwera z dopiskiem `/admin` i zalogowaniu danymi `'admin' 'admin'` może edytować dostępne treści.

9 Repozytorium

<https://github.com/kuubsOn/RentCarCMS>



Rent your dream car

Only with us for a small amount of money you can rent a car

[Contact with us!](#)

Availability

Cars available 24/7 even on holidays.

Clear rules

You know how much you will pay for the loan before booking.

High vehicle class

High-class cars are fully equipped.

Deadlines tailored to you

You give away when you want and the amount will be automatically recalculated.

Choose your dream car

Only with us for a small amount of money you can rent a car

Audi A7



Production: 2017

Course: 40

Engine capacity: 4.2

Power (HP): 460

Color: WHITE

A super fast car for demanding people

[Details](#)

Audi A8



Production: 2018

Course: 100

Engine capacity: 5.2

Power (HP): 602

Color: BLACK

A super fast car for demanding people

Rent car

[Home](#) [Rents](#) [Contact](#)

Audi A7

Production: 2017

Course: 40

Engine capacity: 4.2

Power (HP): 460

Color: WHITE



A super fast car for demanding people...

© Company 2017

Rent car

[Home](#) [Rents](#) [Contact](#)

Address

Name: Jakub Krawczyk

Surname: Krawczyk

City: Poznań (Poznań-Nowe Miasto)

Address: Centrum

Email: jakubkrawczyk2014@gmail.com

Phone number: +48123456789





Send message

Type your name and surname

Type your email

Type your message

Send

