**Obraz zawierający tekst

Opis wygenerowany automatycznieObraz zawierający tekst, clipart, wizytówka

Opis wygenerowany automatycznie**

Grzegorz Fijałkowski

nr albumu: 44317

kierunek studiów: Informatyka

specjalność: Inżynieria komputerowa

forma studiów*: stacjonarne*

PROJEKT I IMPLEMENTACJA SERWISU OGŁOSZENIOWEGO

DESIGN AND IMPLEMENTATION OF AN ANNOUNCEMENTS WEBSITE

praca dyplomowa inżynierska

napisana pod kierunkiem:

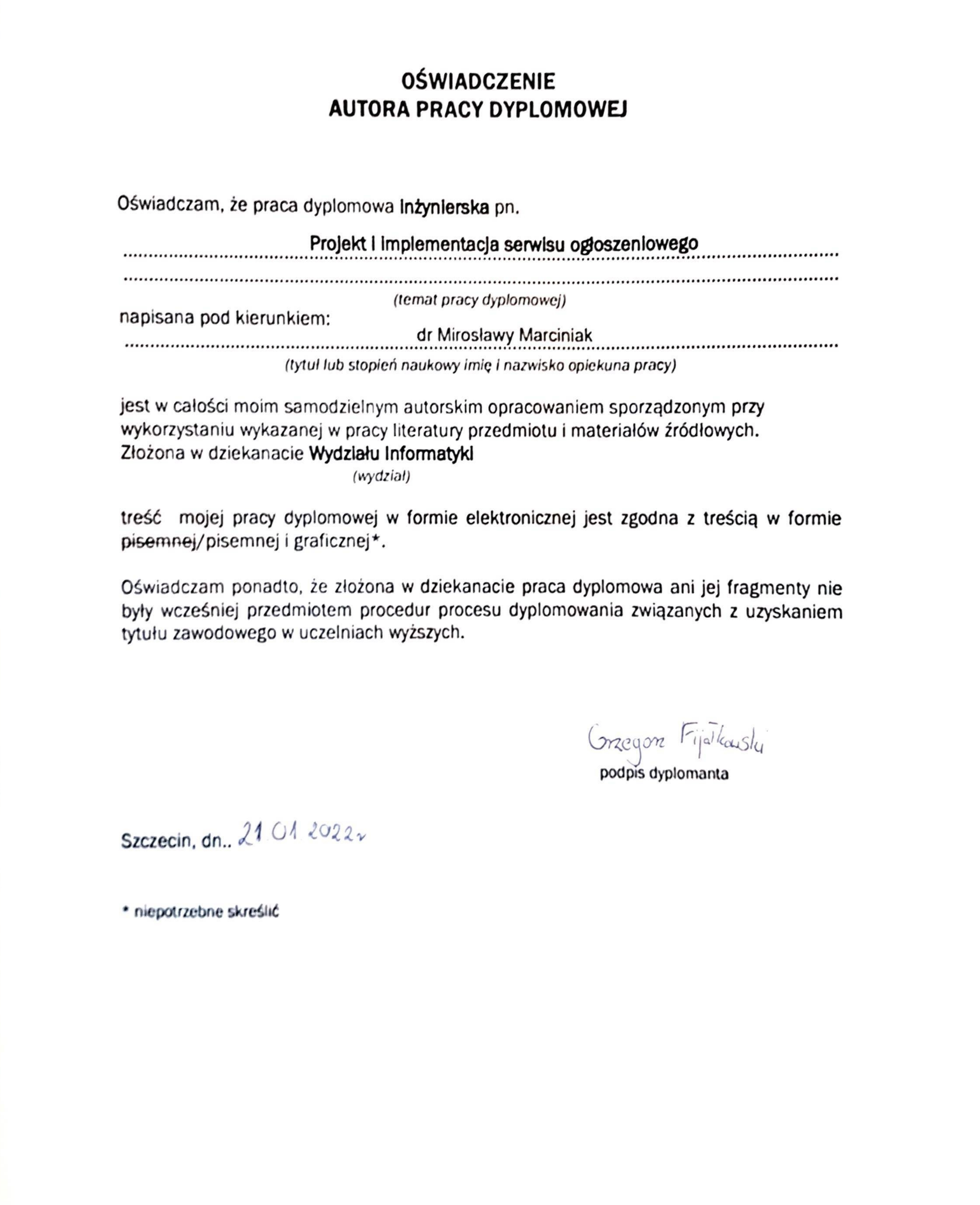
dr Mirosławy Marciniak

Katedra Inżynierii Systemów Informacyjnych

Data wydania tematu pracy: 08.06.2021

Data złożenia pracy:

Szczecin, 2022



Streszczenie

Głównym celem pracy było zbudowanie prototypu internetowego serwisu ogłoszeniowego i opisanie tego procesu. Wykonana aplikacja oferuje możliwość dodawania i przeglądania ogłoszeń kupna/sprzedaży. Na podstawie analizy serwisów internetowych o podobnej tematyce zaprojektowano funkcjonalność systemu, którą przedstawiono na diagramach UML oraz opracowano model danych. Do implementacji systemu wykorzystano technologie webowe oraz stworzony przez Google interfejs programistyczny aplikacji. Na koniec opisano wygląd i sposób działania systemu oraz zamieszczono wyniki jego testowania pod kątem wydajności i bezpieczeństwa danych.

**Słowa kluczowe:** aplikacja internetowa, serwis ogłoszeniowy, Django, Python.

Abstract

The main objective of the thesis was to build a prototype of an online announcements service and describe this process. The application offers a possibility of adding and viewing purchase/sell announcements. Based on analysis of pages with similar themes, the functionality of the system was designed and presented in UML diagrams and a data model was developed. Web technologies and the application programming interface created by Google were used to implement the system. Finally, the appearance and operation of the system is described, and the results of testing it for performance and data security are included.

**Keywords:** web application, announcements service, Django, Python.

**Spis treści**

[Wstęp 5](#_Toc93578361)

[1. Przegląd wybranych serwisów ogłoszeniowych 6](#_Toc93578362)

[1.1. Olx 6](#_Toc93578363)

[1.2. eBay 7](#_Toc93578364)

[1.3. Sprzedajemy 8](#_Toc93578365)

[1.4. Gumtree 8](#_Toc93578366)

[2. Opis projektu systemu 10](#_Toc93578367)

[2.1. Wymagania funkcjonalne 10](#_Toc93578368)

[2.2. Wymagania niefunkcjonalne 11](#_Toc93578369)

[2.3. Diagramy przypadków użycia 12](#_Toc93578370)

[2.4. Diagramy UML 14](#_Toc93578371)

[3. Implementacja serwisu ogłoszeniowego 20](#_Toc93578372)

[3.1. Wybór technologii 20](#_Toc93578373)

[3.2. Architektura systemu 21](#_Toc93578374)

[3.3. Model bazy danych 23](#_Toc93578375)

[3.4. Implementacja wybranych funkcjonalności aplikacji 26](#_Toc93578376)

[3.5. Integracja logowania z kontem Google 28](#_Toc93578377)

[4. Wygląd i działanie prototypu systemu 29](#_Toc93578378)

[5. Wyniki testowania prototypu serwisu 39](#_Toc93578379)

[5.1. Testy wydajnościowe 39](#_Toc93578380)

[5.2. Testy bezpieczeństwa 41](#_Toc93578381)

[6. Podsumowanie 43](#_Toc93578382)

[Bibliografia 44](#_Toc93578383)

[Spis tabel 45](#_Toc93578384)

[Spis rysunków 46](#_Toc93578385)

# Wstęp

W okresie pandemii znaczenie Internetu znacznie nabrało na sile. Wiele szkół i firm przechodziło na tryb zdalny, który oferował masę innowacyjnych rozwiązań. Jakakolwiek interakcja ze społeczeństwem bez połączenia do Internetu była praktycznie niemożliwa. Dlatego też coraz więcej osób zaczęło korzystać z aplikacji webowych, które w głównej mierze pracują na serwerze i korzystają z sieci komputerowej do komunikacji z użytkownikiem. Zasada działania jest podobna do zwykłych aplikacji desktopowych jednak w tym przypadku aplikację webową uruchamia się w przeglądarce bez konieczności pobierania czegokolwiek. To może sprawiać, że cieszą się one coraz większą popularnością. Są one zwyczajnie mniej wymagające, gdyż jedyna rzecz, jaka w tym przypadku jest potrzebna to przeglądarka i połączenie internetowe.

Na przełomie ostatnich lat nastąpiła ewolucja w tej branży. Jeszcze dekadę temu aplikacje realizowały podstawowe funkcjonalności związane z przetwarzaniem danych, a teraz dzięki nim możliwe jest prowadzenie konferencji czy płacenie rachunków za mieszkanie. Wyszukanie ogłoszenia o sprzedaży konkretnego towaru w prasie codziennej lub przeszukiwanie ofert na licznych platformach sklepów internetowych zajmuje za dużo czasu, stąd pomysł na serwis internetowy, zawierający aktualne ogłoszenia o sprzedaży/kupnie lub zamianie, dzięki któremu odwiedzający mogą znaleźć interesującą ich rzecz przy porannej kawie, nie wychodząc z domu.

Tworząc serwis ogłoszeniowy, warto również wspomnieć o ekonomii współdzielenia, która w znacznym stopniu rozwinęła się pod wpływem nowych technologii informatycznych. Według definicji ekonomia współdzielenia polega na efektywniejszym wykorzystaniu zasobu przez współpracę zarówno kupującego jak i sprzedającego [3]. Obecnie na rynku istnieje wiele aplikacji opierających się na takiej strategii, m.in. są to eBay, Uber, Kickstarter i BlaBlaCar. Serwis tworzony na potrzeby tej pracy również opiera się na tej strategii, umożliwiając wymianę niewykorzystanych zasobów. Jedną z zalet takiego rozwiązania jest korzystna cena zasobu w porównaniu do jego oryginalnej ceny. Zyskuje na tym zarówno sprzedawca (towar nie zalega na magazynie), jak i klient.

Celem mojej pracy dyplomowej było zaprojektowanie i wykonanie aplikacji webowej, ułatwiającej wymianę lub sprzedaż niewykorzystanych towarów/zasobów. Ponadto aplikacja ma za zadanie zwiększyć zasięg sprzedaży, docierając do większego grona potencjalnych sprzedawców/klientów, którzy udostępniają swoje zasoby w zależności od woli sprzedającego (odpłatnie lub nieodpłatnie).

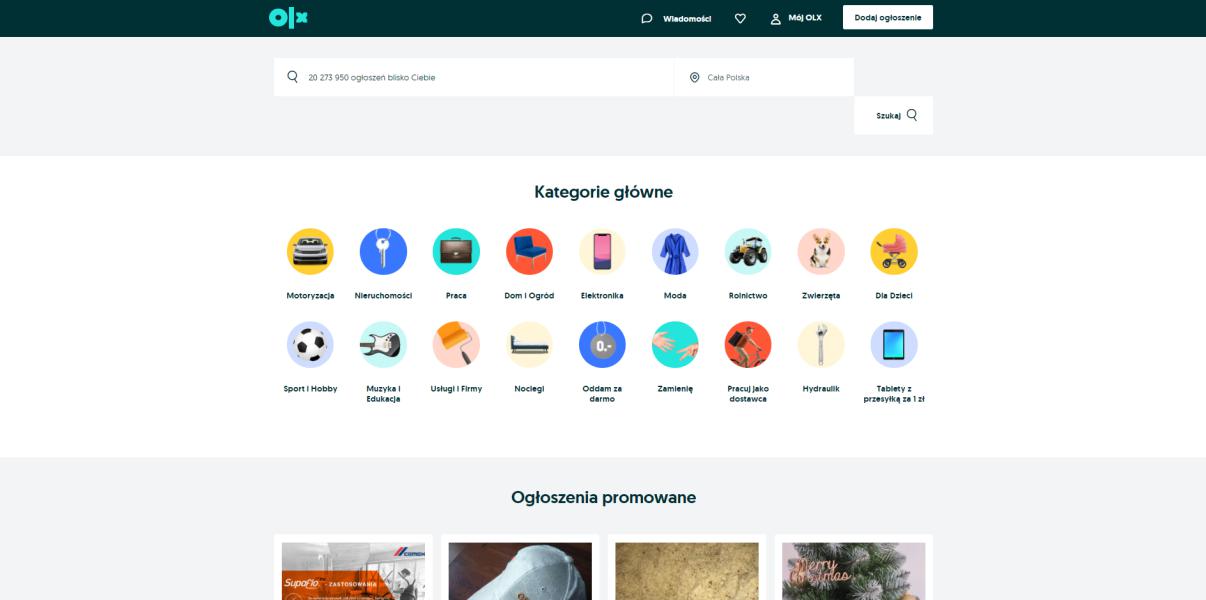
W pracy inżynierskiej został przedstawiony proces projektowania i implementacji aplikacji webowej, realizującej zasady ekonomii współdzielenia. Zakres funkcjonalny projektowanego systemu poprzedzono analizą podobnych rozwiązań popularnych w Internecie. W projekcie systemu szczegółowo opisano przypadki użycia systemu, diagramy UML oraz model danych. Kolejnym etapem był dobór technologii i narzędzi informatycznych stosownych do wybranej architektury systemu. Po przygotowaniu środowiska programistycznego (Django) za pomocą wybranych technologii utworzono internetowy serwis ogłoszeniowy w postaci prototypu systemu, realizującego podstawowe funkcjonalności uwzględnione w projekcie. Ostatnim etapem pracy było testowanie wykonanego prototypu systemu w zakresie wydajności (na podstawie opracowanych scenariuszy) oraz bezpieczeństwa, czyli podatności systemu na zagrożenia.

# Przegląd wybranych serwisów ogłoszeniowych

W dobie Internetu powstaje coraz więcej platform internetowych umożliwiających dzielenie się zasobami na odległość. Ludzie przekonali się do handlu przez Internet, dlatego coraz częściej można spotkać osoby prywatne sprzedające niewykorzystane towary. Wraz z większą świadomością ludzi strony zaczęły się przekształcać w serwisy, które pośredniczą w tym handlu. Na rynku ukształtowało się już wiele gigantów w tej dziedzinie i dlatego właśnie warto zwrócić na nie uwagę przed rozpoczęciem pracy nad tworzeniem serwisu. Strony te zawierają wiele innowacyjnych funkcjonalności i rozwiązań, z których warto brać przykład. Rozpatrzone zostaną zarówno serwisy polskie jak i zagraniczne, ze względu na rozwiązania, które są skierowane na rynek lokalny jak i światowy. Przeglądając Internet w poszukiwaniu serwisów ogłoszeniowych można natrafić na masę stron udostępniających usługi w tym zakresie jednak żeby zredukować ich liczbę zostało wybranych tylko kilka największych, które zostaną opisane poniżej. Podczas przeglądania stron szczególną uwagę należy zwrócić na wykonanie oraz funkcjonalności, które można później zaimplementować w projekcie.

## Olx

Przeglądając polskie serwisy ogłoszeniowe należy wspomnieć o największym z nich, a dokładniej o OLX. Jest to aplikacja, która dzięki swojemu marketingowi stała się rozpoznawalna w całej Polsce. Zasłynęła również z tego, że jest bardzo prosta w obsłudze i dostosowana do przeciętnego użytkownika. Olx należy do grupy Naspers pochodzącej z południowej Afryki i działa w ponad czterdziestu krajach. W Polsce działa już od 2014 roku.



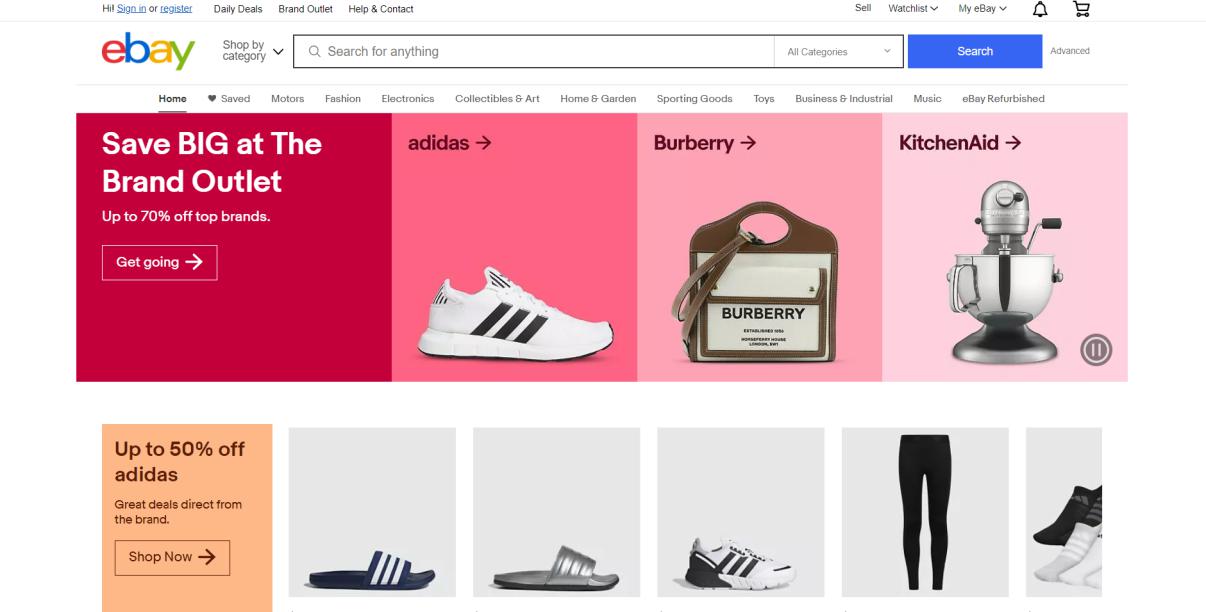
Rys. 1. Strona główna serwisu olx

Na stronie głównej *OLX* znajduje się wyszukiwarka, kategorie oraz ogłoszenia promowane. Są one dobrze rozmieszczone oraz przyjemne dla oka ułożenie strony. Aby dodać ogłoszenie, wystarczy tylko rejestracja, która wymaga jedynie adresu email i hasła. Dla użytkownika, który nie jest obeznany z komputerami jest to niezwykle przejrzyste.

Wielkim plusem strony jest to, że jest ona darmowa. Nie pobiera ona żadnych profitów z dodanych ogłoszeń oraz udanych transakcji. Występują w niej tylko mikropłatności, które pomagają zobaczyć ogłoszenie większej ilości osób. Mimo wszystko klient ma zawsze wybór czy chce skorzystać z tej opcji. Z innych możliwości OLX oferuje również możliwość obserwowania ogłoszenia oraz już w wersji beta możliwość stworzenia swojego profilu kandydata do pracy. Warto również zaznaczyć, że aplikacja wykorzystuje API, dzięki któremu możliwe jest zarejestrowanie się do serwisu przy pomocy konta Google lub Facebook. Według wyników badań GEMIUS/PBI z sierpnia 2021 roku liczba użytkowników tego portalu wynosi prawie 13 milionów, co daje miejsce w pierwszej dziesiątce polskich portali [13].

## eBay

Kolejnym analizowanym serwisem był eBay, który ma zarówno polski, jak i zagraniczny odpowiednik. Nie jest to typowo serwis ogłoszeniowy jak ten wspomniany powyżej, a serwis aukcyjny. Został on założony przez Pierre’a Omidyara w 1995 roku i do dziś działa w 37 krajach. W Polsce zajmuje czołowe miejsce w rankingu serwisów aukcyjnych. Portal eBay jest dobrym przykładem ekonomii współdzielenia, gdyż zajmuje się on sprzedażą używanych towarów.

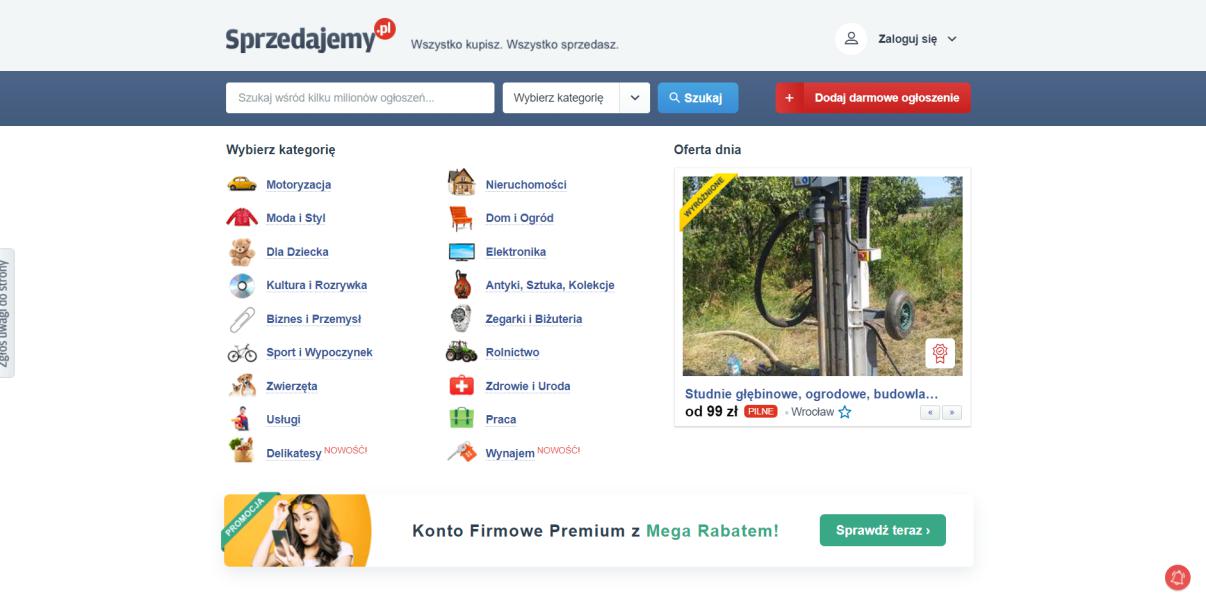


Rys. 2. Strona główna serwisu eBay

Wygląd ogólny strony oraz ułożenie poszczególnych elementów jest bardzo proste, mimo to zostało skonstruowane tak, żeby przykuć uwagę klienta na najpotrzebniejsze rzeczy takie jak popularne wyszukiwania czy okazje dnia. Możliwość wystawienia ogłoszenia występuje wyłącznie przy byciu zalogowanym do serwisu. Sama opcja rejestracji oraz logowania występuje na stronie głównej jednak jest ona dość nieintuicyjna i ciężka do zauważenia. Występują na niej podstawowe rzeczy takie jak wyszukiwarka i kategorie. Liczba transakcji realizowanych na stronie oscyluje w granicy kilku miliardów, co czyni stronę gigantem na rynku światowym.

W kwestii szukania poszczególnego towaru na stronie prezentuje się to dobrze. Po wpisaniu nazwy towaru w wyszukiwarkę albo wyszukaniu jej po kategoriach klient zostaje zalany masą aukcji z poszukiwanym przedmiotem [9].

## Sprzedajemy

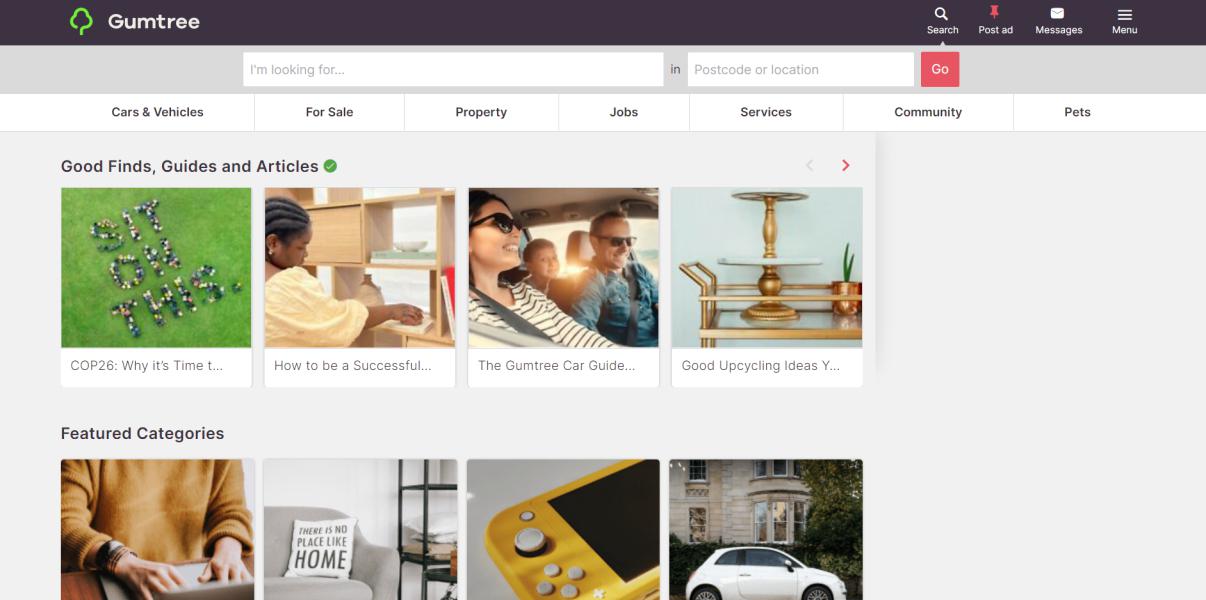
**Sprzedajemy to kolejny większy serwis ogłoszeniowy na polskim rynku. Został on uruchomiony w 2011 roku, czyli trzy lata wcześniej od wspomnianego wcześniej portalu *OLX*. Na początku cieszył się dużym zainteresowaniem na polskim rynku, jednak po powstaniu serwisu *OLX* stracił na popularności, a duża część dotychczasowych użytkowników przeniosła się do konkurencji. Mimo wszystko nadal jest aktywny, a liczba ogłoszeń liczy około kilku milionów. W porównaniu do OLX strona główna serwisu „Sprzedajemy” jest mniej atrakcyjna pod względem wizualnym. Jednakże interfejs jest na tyle intuicyjny, że przeciętny użytkownik

Rys. 3. Strona główna serwisu sprzedajemy

Podczas dodawania ogłoszenia użytkownik nie musi być zalogowany do serwisu. Wystarczy, że podczas tworzenia ogłoszenia poda swoje dane kontaktowe. Istnieje również możliwość zalogowania się za pomocą konta Google lub Facebooka. Podczas umieszczania ogłoszenia klient jest informowany o wszystkich możliwych mikropłatnościach, których jest w tym podejrzeć w swoim profilu w zakładce ogłoszenia. Cały proces obsługi jest bardzo prosty, dlatego też w serwisie zostało wielu aktywnych użytkowników. Niestety ma to też swoje wady, głównie brak innowacyjnych rozwiązań występujących już w serwisie „Olx”. [15]

## Gumtree

**Gumtree** to serwis, który również jak eBay posiada swój polski oraz zagraniczny odpowiednik. Powstał on najwcześniej, a dokładniej w 2000 roku. W Polsce było to cztery lata później. Początkowo obejmował tylko kraje Wielkiej Brytanii (stał się jedną z 30. najpopularniejszych witryn), a później rozpowszechnił się w innych państwach. Polska witryna tego serwisu nie jest już za często używana, co pokazuje liczba aktywnych ogłoszeń w serwisie (ok. 350 tysięcy). Przyczyną spadku popularności może być mało atrakcyjny widok strony głównej serwisu, który wyraźnie odstaje od innych analizowanych w tej pracy serwisów. W założeniu podobny jest do serwisu *sprzedajmy.pl*, ponieważ brakuje mu wielu innowacyjnych technologii, ale za to sam proces dodawania ogłoszenia jest trywialny.

Można wyszukać i dodać ogłoszenia i na tym w zasadzie kończą się wszystkie możliwości tego serwisu. Jeśli chodzi o zagraniczną wersję to spełnia ona swoje zadanie znacznie lepiej. Serwis wykonany bardzo estetycznie, a wszystkie potrzebne opcje takie jak: wyszukiwarka, lista kategorii oraz przykładowe ogłoszenia, znajdują się na stronie głównej. Opcja tworzenia konta jest w pasku nawigacyjnym (widoczna na pierwszy rzut oka). Istnieje możliwość utworzenia konta za pomocą kont Google lub Facebooka. Dodanie ogłoszenia również znajduje się na pasku nawigacyjnym i wymaga zalogowania. [12]

Rys. 4. Strona główna serwisu Gumtree.

**Tabela 1. Ocena wybranych serwisów ogłoszeniowych.**

|  |  |  |
| --- | --- | --- |
| **Nazwa serwisu** | **Zalety** | **Wady** |
|
| **Olx** | Prosty interfejs,  Zamieszczanie ogłoszeń jest darmowe,  Duży wybór kategorii,  Proste dodawanie ogłoszenia. | Brak gwarancji otrzymania zakupionego towaru,  Pojawiające się reklamy. |
|
|
|
| **eBay** | Ochrona kupującego i sprzedającego w razie wystąpienia problemów,  Wygodny w obsłudze,  Szeroki zakres oferty sprzedaży. | Sprzedawanie dóbr jest płatne,  Długi czas dostawy,  Duża konkurencja. |
|
|
|
| **Sprzedajemy** | Wystawianie ogłoszeń jest darmowe,  Przejrzysty interfejs użytkownika. | Brak gwarancji dostarczenia zakupionego towaru,  Słaba strona graficzna interfejsu,  Duża liczba mikropłatności. |
|
|
|
| **Gumtree** | Korzystanie z serwisu jest bezpłatne,  Mała konkurencja. | Ma zasięg lokalny,  Posiada dużą liczbę zabronionych przedmiotów. |
|
|

Źródło: opracowanie własne.

# Opis projektu systemu

Z biegiem czasu otaczająca nas rzeczywistość coraz bardziej się zmienia. Przechodzimy przełom technologiczny. Wraz z jego rozwojem pojawia się więcej wymagań, które każda aplikacja powinna posiadać. Producenci oprogramowania szukają coraz bardziej innowacyjnych rozwiązań, które mogłyby wpłynąć pozytywnie na odbiór aplikacji przez klienta. Analiza popularnych serwisów ogłoszeniowych pokazała, że niektóre z nich odchodzą od schematu handlowania wyłącznie dobrami fizycznymi i udostępniają także oferty pracy. Może to być odpowiedzią na coraz większe wymagania użytkowników, których można łatwo stracić przez rozwijającą się konkurencję.

Podczas tworzenia serwisu ważne jest, żeby wcześniej sporządzić listę wymagań, które powinny być zawarte w aplikacji. Według Karla Wiegersa i Joy Beatty zrozumienie wymagań zapewnia, że programista będzie pracował nad należytym problemem i wyszuka dla niego najlepszego rozwiązania. Bez tego nie dałoby rady określić, czy projekt spełnia swoje wymagania oraz, czy został w pełni ukończony [5]. Wyróżnia się dwa rodzaje wymagań, czyli wymagania funkcjonalne i niefunkcjonalne. Te pierwsze określają, co system powinien robić, a drugie, jak powinien być zbudowany.

## Wymagania funkcjonalne

**Tworzenie, modyfikowanie i usuwanie ogłoszeń**

Jako serwis ogłoszeniowy jego głównym zadaniem jest właśnie tworzenie ogłoszeń, które potem widnieją na liście. Dodatkowo użytkownik ma możliwość modyfikowania i usuwania swoich ogłoszeń. W zasadzie te trzy funkcje w głównej mierze czynią tę aplikację serwisem ogłoszeniowym. Ogłoszenia powinny widnieć na stronie do nich przeznaczonej oraz prezentować się w sposób przyjemny dla oka, żeby przyciągać uwagę innych użytkowników.

**Wyświetlanie profilu użytkownika**

Drugi dość ważny aspekt każdej aplikacji to profil użytkownika. Daje on możliwość edycji danych osobowych oraz dostęp do wszystkich wystawionych przez użytkownika ogłoszeń w jednym miejscu. Jest to szczególnie przydatne udogodnienie dla użytkowników ze względu na to, że w razie błędnego wpisania danych użytkownik w każdej chwili może naprawić swój błąd. To samo dotyczy monitorowania zamieszczonych przez niego ogłoszeń.

**Przegląd ogłoszeń wybranego użytkownika**

Serwis powinien udostępniać opcję przejrzenia ogłoszeń użytkownika. Po wejściu w ogłoszenie na samym dole obok danych sprzedającego powinien widnieć przycisk, który przenosi użytkownika na podstronę, na której powinny znajdować się wszystkie ogłoszenia konkretnego sprzedającego w jednym miejscu.

**Wybór konkretnej kategorii**

Kolejnym wymaganiem serwisu ogłoszeniowego powinna być lista dostępnych kategorii. Przeglądając wiele serwisów, można stwierdzić, że każdy z nich posiada mniejszą lub większą listę kategorii. Jest to dobre rozwiązanie, ponieważ mogą być użytkownicy, którzy mają sprecyzowanych wymagań.

**Wyszukiwarka**

Wyszukiwarka jest funkcją, którą znajduje się we wszystkich wspomnianych wcześniej serwisach ogłoszeniowych. Jest ona skierowana do użytkowników, którzy wiedzą, czego szukają, ponieważ umożliwia ona wyszukanie interesującego produktu przez wpisanie odpowiedniej frazy w pole wyszukiwarki. Po rozpoczęciu przeszukania wyszukiwarka przenosi użytkownika na podstronę, na której znajdują się wszystkie ogłoszenia z szukaną frazą.

**Wysyłanie i odbieranie wiadomości**

Klient i sprzedawca muszą się jakoś ze sobą kontaktować, a często kontakt przez e-mail albo telefon jest utrudniony przez różne czynniki. Dlatego serwis powinien mieć możliwość wysyłania i odbierania wiadomości. Przycisk przenoszący użytkownika do panelu wysyłania powinien być dostępny po wejściu w ogłoszenie. Interfejs wysyłania wiadomości powinien zawierać takie etykiety jak nazwę odbiorcy, temat wiadomości i jej opis. Możliwość przejrzenia wiadomości powinna występować w profilu za pomocą przycisku, który przeniesie użytkownika do skrzynki odbiorczej. W skrzynce odbiorczej powinna być opcja wykonywania pewnych akcji takich jak: usunięcie, archiwizowanie i zaznaczenie, jako przeczytane lub nieprzeczytane. Każda wiadomość powinna zawierać takie informacje jak nadawca, temat, czas  otrzymania i opis wiadomości.

## Wymagania niefunkcjonalne

**Dostępność**

Wystarczy posiadać komputer i dostęp do Internetu, żeby wejść do serwisu. Działa on na serwerze, dzięki czemu jest dostępny całodobowo. A cała mechanika sprowadza się do wpisania adresu strony w przeglądarkę.

**Szybkość**

System powinien obsługiwać wielu klientów jednocześnie oraz dzięki swojej szybkości przetwarzać masę ogłoszeń w tej samej chwili. Dzięki tej funkcjonalności system nie powinien wariować w przypadku zbyt dużej ilości klientów. Serwis w jednym czasie powinien przetwarzać ogłoszenia użytkowników z całego kraju, dlatego taka funkcjonalność jest wymagana.

**Niezawodność**

Niskie prawdopodobieństwo awarii systemu oraz w przypadku niedostępności aplikacji szybka jej naprawa. Nawet najsłynniejsze aplikacje posiadają od czasu do czasu problemy bądź awarie, które utrudniają korzystanie z nich, dlatego tak ważna jest szybka ich naprawa.

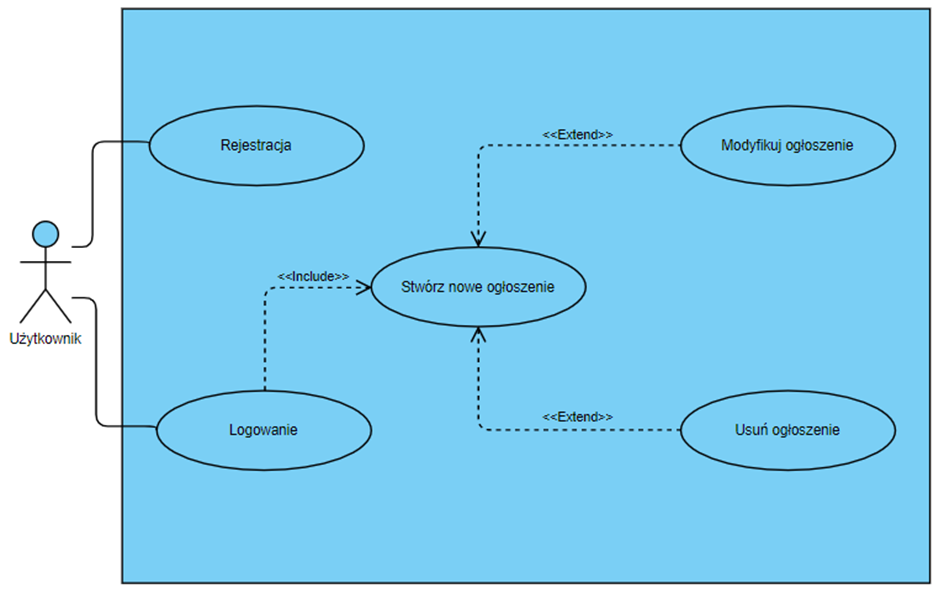
**Użyteczność**

Aplikacja została stworzona z myślą o klientach, dlatego tak ważna jest jej użyteczność. Przy weryfikacji tej funkcjonalności brane pod uwagę są takie rzeczy jak łatwość korzystania z serwisu oraz sama estetyka, która powinna zachęcać klientów do korzystania ze strony. Sam serwis spełnia wszystkie te aspekty. Nawet użytkownik, który nie miał dużej styczności z komputerami, powinien dojść do wszystkiego sam. Proces rejestracji, logowania oraz wstawiania ogłoszeń jest łatwy w obsłudze.

## Diagramy przypadków użycia

Aktorem na diagramach jest użytkownik korzystający z serwisu. Opcja tworzenia ogłoszenia jest dostępna dopiero po zalogowaniu się na stronie, a jeżeli użytkownik nie ma jeszcze konta, musi się wpierw zarejestrować. Po zalogowaniu się na górnym pasku pojawia się dodatkowa opcja “Dodaj ogłoszenie”. Po wejściu na nią przekierowuje użytkownika do podstrony, na której możliwe jest dodanie ogłoszenia, wcześniej wpisując odpowiednie parametry. Po potwierdzeniu wszystkiego ogłoszenie zostanie dodane i można je podejrzeć na stronie z innymi wystawionymi rzeczami. Istnieje wtedy możliwość modyfikacji i usunięcia takiego ogłoszenia, która odbywa się dość w prosty sposób. Wystarczy kliknąć odpowiednie do tego przyciski, które są umieszczone przy ogłoszeniu, jeśli jest się jego twórcą.

Rys. 5. Diagram przypadków użycia - tworzenie ogłoszenia

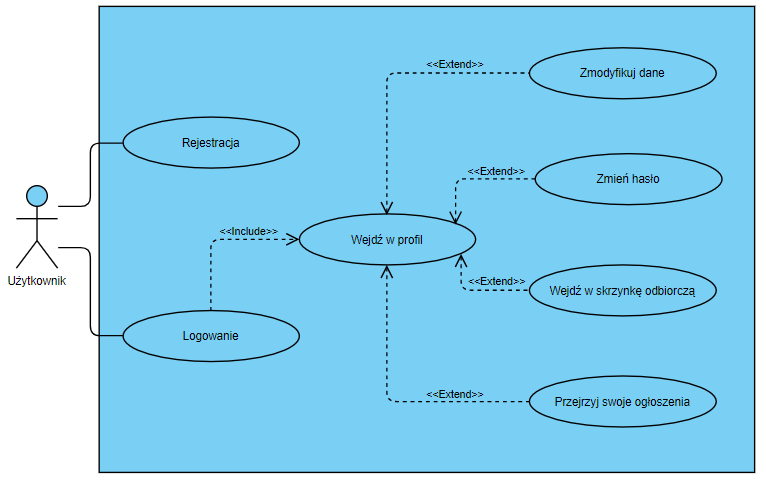


Drugim rozważanym przypadkiem jest wyszukiwanie ogłoszeń. Tym razem aktor, czyli użytkownik, nie musi się logować do systemu, żeby móc skorzystać z tej opcji. Na stronie głównej serwisu znajduje się wyszukiwarka, która umożliwia użytkownikowi wyszukanie ogłoszenia przez podanie przez niego odpowiedniej frazy. Dalej pojawi się lista ogłoszeń, która zostanie skojarzona z daną frazą. Wyszukiwarka wychwytuje słowa klucz i tak użytkownik może znaleźć ogłoszenia, zawierające podaną frazę. Po pojawieniu się listy, użytkownik ma możliwość podglądu wybranego przez siebie ogłoszenia, a jeśli jest jego twórcą, może jeszcze je zmodyfikować lub usunąć.

Obraz zawierający tekst, osoba

Opis wygenerowany automatycznie

Rys. 6. Diagram przypadków użycia - wyszukiwanie ogłoszeń

Następny przypadek odpowiada za podgląd profilu użytkownika. Tutaj tak samo, jak w poprzednich przypadkach, aktorem jest użytkownik, który wykonuje konkretne czynności.

Rys. 7. Diagram przypadków użycia – profil użytkownika.

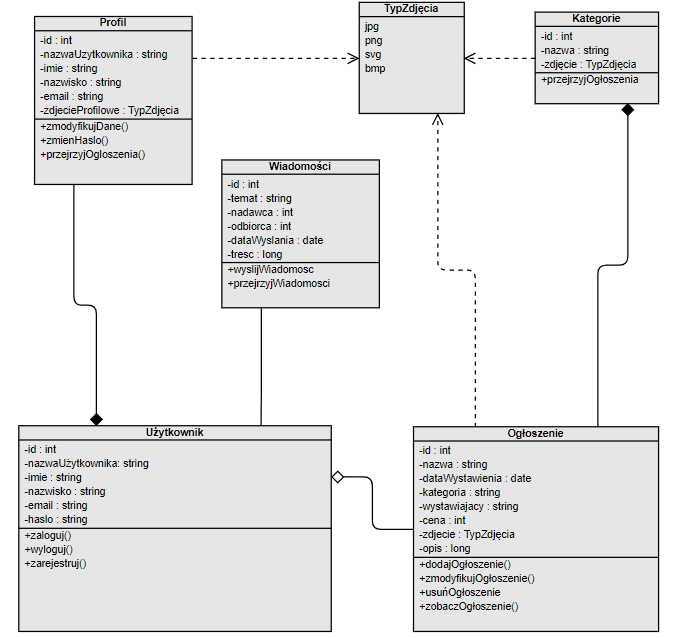
Wejście do profilu użytkownika wymaga wcześniejszego logowania, a więc jeżeli użytkownik wcześniej nie założył konta, to najpierw musi się zarejestrować, a dopiero potem zalogować. Użytkownik zostanie wtedy przeniesiony na stronę główną serwisu i będzie mógł dostrzec pewne zmiany jak na przykład zamiast opcji logowania, będzie nazwa profilu użytkownika w górnym pasku, która po naciśnięciu przenosi użytkownika do podstrony, na której będzie mógł przejrzeć swój profil. Będą tam wyświetlały się podstawowe dane takie jak imię, nazwisko i adres e-mail, które będzie można modyfikować. Pojawi się również możliwość przeglądu wystawionych ogłoszeń, dzięki czemu będzie można zebrać je wszystkie w jednym miejscu.

## Diagramy UML

Diagramy przypadków użycia opisane wcześniej prezentują funkcjonalność systemu z perspektywy użytkownika. W tej części pracy przedstawiono diagramy UML, które wykorzystuje się do graficznego przedstawienia planu systemu. Jest to szczególnie przydatne rozwiązanie, ponieważ pozwala w prosty sposób pokazać działanie systemu. Skrót UML (ang. ***Unified Modelling Language***) oznacza w języku polskim ujednolicony język modelowania**.** Joseph Schmuller stwierdził, że *“Celem diagramów jest pokazanie wielu perspektyw systemu; ten zestaw perspektyw to model. UML-owy model systemu to coś w rodzaju wykonanego w skali modelu budynku wraz z jego artystyczną interpretacją”* [4].

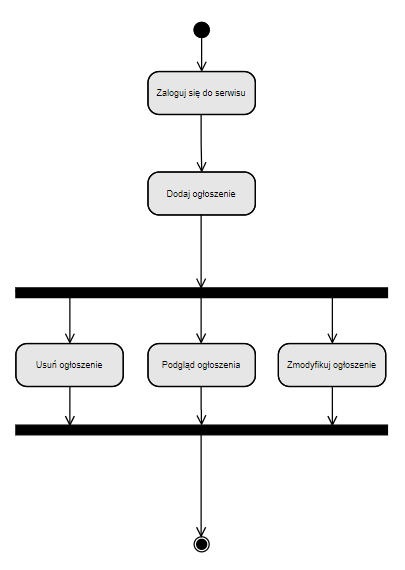
Za pomocą diagramu klas przedstawia się klasy obiektów umieszczonych w systemie oraz zachodzące między nimi związki. Z kolei diagram czynności umożliwia opisanie w krokach pewnej funkcjonalności systemu. Zasadnicza różnica między nim a diagramem przypadków użycia polega na tym, że diagram czynności pokazuje w krokach, jak system dojdzie do zamierzonego rezultatu. Diagram przypadków użycia podchodzi do tego inaczej i pokazuje tylko, jaką funkcjonalność system udostępnia. Dodatkowo warto wspomnieć o diagramie pakietów, który jest również jednym z istotnych diagramów UML, gdyż umożliwia zaprezentowanie pakietów i ukazanie relacji, zachodzących między nimi.

**Diagram klas**

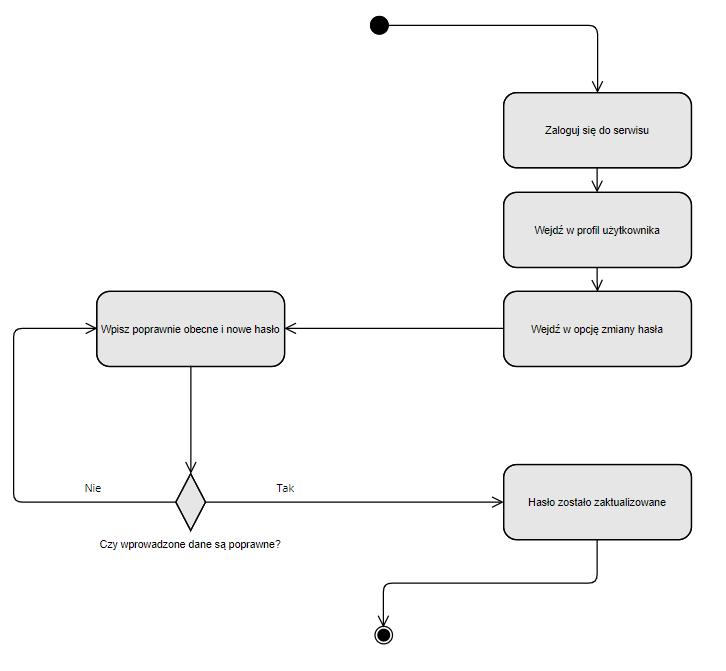
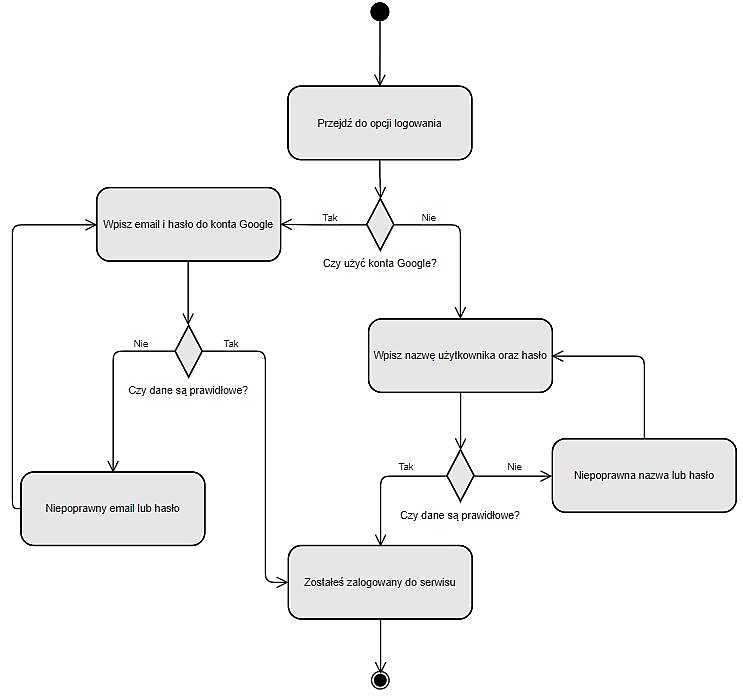


Rys. 8. Diagram klas

**Diagramy czynności**

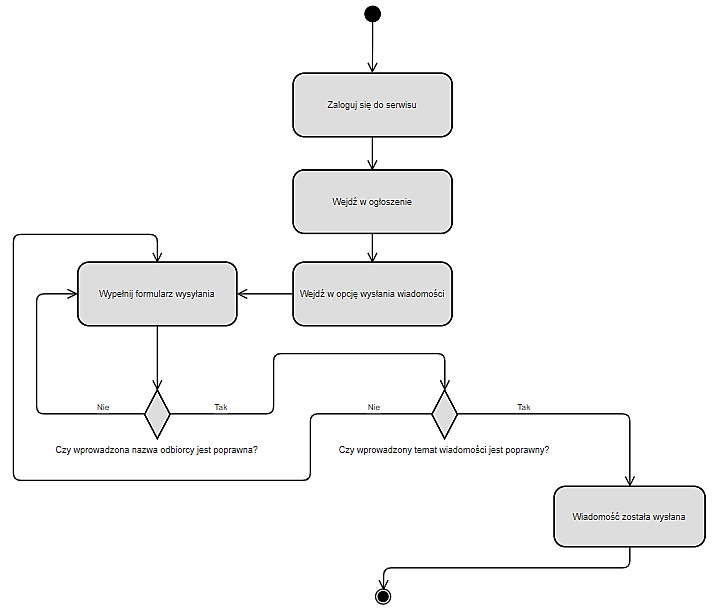


Rys. 9. Diagram czynności - zarządzanie ogłoszeniami



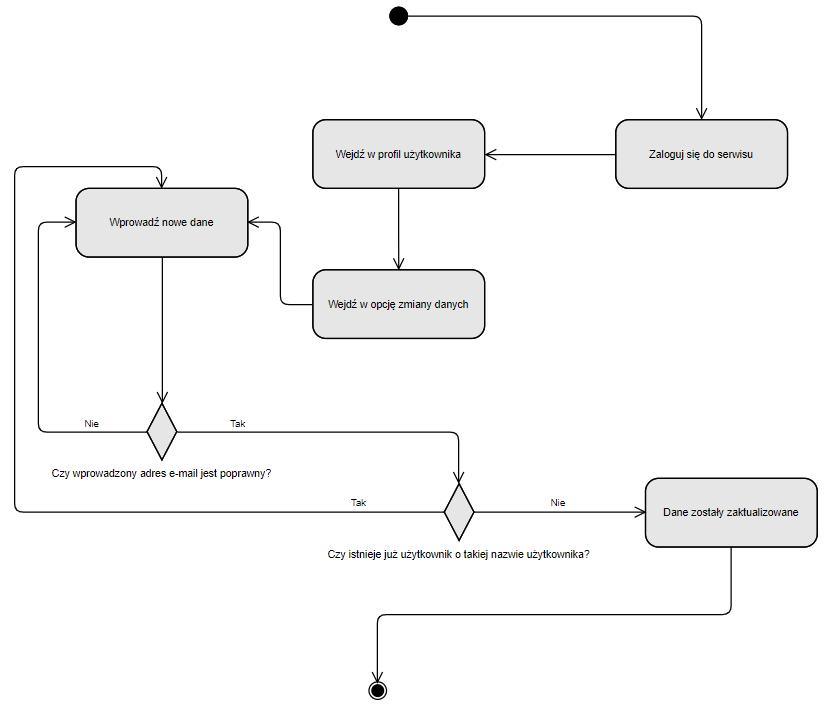
Rys. 11. Diagram czynności - logowanie do serwisu

Rys. 10. Diagram czynności - zmiana hasła



Rys. 12. Diagram czynności - edycja danych.

Rys. 13. Diagram czynności - wysyłanie wiadomości.

Opracowanie diagramów UML przed rozpoczęciem implementacji jest niezwykle istotne, gdy dzięki temu programista może podjąć ważne decyzje dotyczące systemu jeszcze przed implementacją, co pozwoli zaoszczędzić wiele czasu. Niestety jednym z najczęściej popełnianych błędów podczas realizacji projektu jest brak wcześniejszego etapu projektowania. Skutkuje to tym, że późniejsza praca nad implementacją jest mało efektywna, ponieważ wymaga ciągłych zmian.

# Implementacja serwisu ogłoszeniowego

## Wybór technologii

W kolejnym rozdziale pracy przedstawiam technologie i narzędzia programistyczne, które wykorzystałem podczas realizacji projektu.

**Django**

*Django* jest darmowym szkieletem wykorzystywanym do tworzenia aplikacji webowych. Jest on napisany w całości w języku programowania *Python*, przy czym wykorzystuje wiele dodatkowych narzędzi, dzięki którym tworzenie aplikacji jest szybsze i łatwiejsze. Zajmuje się on większością problemów, jakie można napotkać podczas tworzenia stron internetowych pozwalając skupić się wyłącznie na pisaniu aplikacji. Dużą zaletą *Django* jest również to, że posiada rozbudowaną dokumentację i aktywną społeczność. Tworząc nowy projekt w *Django*, nie trzeba tworzyć wszystkiego od nowa, ponieważ udostępnia on wiele funkcjonalności między innymi wbudowaną bazę danych oraz rozbudowany panel administratora [8].

**Python**

*Python* jest jednym z najbardziej popularnych języków programowania wysokiego poziomu. Został opracowany na początku lat 90. XX wieku przez holenderskiego programistę Guido van Rossum. W zamyśle chciał on stworzyć język programowania, który miał być łatwy oraz intuicyjny, ale silny jak konkurencyjne języki. Dzięki swojej prostocie zajmuje czołowe miejsce w rankingu najczęściej używanych języków programowania na świecie. Posiada on rozbudowany pakiet bibliotek standardowych, a dzięki swojej wszechstronności umożliwia programowanie obiektowe, funkcyjne i strukturalne. Dodatkowym plusem jest to, że *Python* jest stale udoskonalany, dzięki czemu co jakiś czas dostaje aktualizacje naprawiające stare błędy albo dodające nowe funkcjonalności [6].

**Bootstrap**

*Bootstrap* — szkielet, którego twórcami są programiści Twittera. Ułatwia on pracę nad stroną wizualną aplikacji webowej dając gotowe rozwiązania CSS i JavaScript. Aby móc z niego korzystać należy podpiąć arkusz stylów do pliku HTML, który pozwala załadować CSS od *Bootstrap* w aplikacji. Zawiera ciekawe komponenty przydatne do tworzenia serwisu ogłoszeniowego takie, jak: komponent Cards, który nadaje się idealnie na makietę dla pojedynczego ogłoszenia. Dodatkowo dzięki niemu możliwe jest rozmieszczenie konkretnych komponentów w rzędzie, co na pewno zostanie wykorzystane przy wyświetlaniu ogłoszeń. Ważnym aspektem szkieletu *Bootstrap* jest również responsywny system siatki, pozwalający dzięki systemowi dwunastu kolumn tworzyć układy o różnych kształtach i rozmiarach. Wykorzystuje szereg kontenerów, wierszy i kolumn do układania zawartości. Umożliwia również dostosowanie układu tabeli do szerokości urządzenia, z którego korzysta użytkownik [11].

**SQLite3**

*SQLite3* — jest to relacyjna baza danych stworzona przez Richarda Hippa. Wybór tej bazy danych podyktowany został faktem, że nie wymaga konfiguracji oraz jest prosta w obsłudze. Poza ty, przy tworzeniu projektu w Django automatycznie wybieraną bazą jest SQLite3. Dodatkowo wielką jej zaletą jest to, że zajmuje niewiele miejsca na dysku, więc jest świetnym rozwiązaniem w przypadku mniejszych projektów (takim jak serwis ogłoszeniowy).

**Google API**

*Google API* — jest to stworzony przez Google interfejs programistyczny aplikacji. Pozwala on na integrację i komunikację z usługami Google. Dzięki temu dowolne aplikacje mogą wykorzystywać funkcjonalności istniejących usług przy korzystaniu z tych interfejsów programistycznych aplikacji [10].

**Apache JMeter**

*Apache JMeter* — jest to oprogramowanie *open source* firmy Apache, umożliwiające testowanie działania funkcjonalności i mierzenie wydajności oprogramowania. *JMeter* jest napisany w całości w języku Java. Będzie on używany w projekcie w ostatniej fazie testowania, gdzie zostaną przeprowadzone testy wydajnościowe sprawdzające, jak serwis sobie radzi przy większym obciążeniu. *Apache JMeter* początkowo służył wyłacznie do testowania aplikacji internetowych, ale później rozszerzono go o dodatkowe funkcje testowe [7].

**OWASP ZAP**

*OWASP ZAP* — kolejne narzędzie *open source* służące do testowania systemu. ZAP umożliwia testowanie aplikacji pod kątem testów penetracyjnych i jest on zarówno elastyczny, jak i rozszerzalny. W fazie testowania zostanie użyty do przeprowadzenia testów bezpieczeństwa aplikacji. Jest pośrednikiem między przeglądarką a aplikacją internetową, co pozwala na przechwytywanie i sprawdzanie wiadomości wysyłanych między nimi [14].

## Architektura systemu

Oczywistym wyborem przy projektowaniu aplikacji webowych jest architektura klient-serwer, która umożliwia komunikację między warstwami funkcjonalnymi. Opisując architekturę aplikacji, należy wspomnieć o wzorcach architektonicznych, które są ich nieodłącznym elementem. Najpopularniejszym wzorcem wykorzystywanym podczas tworzenia aplikacji internetowych jest wzorzec MVC (ang. *Model-View-Controller*).

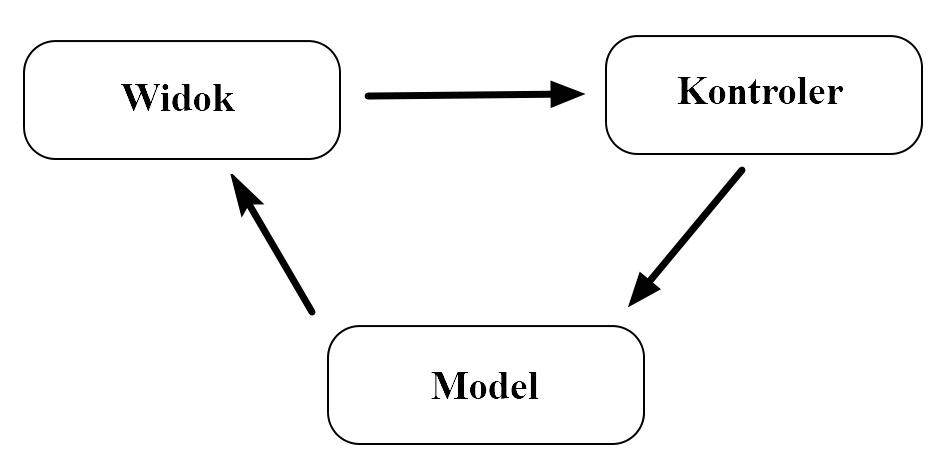
MVC umożliwia podział aplikacji na trzy warstwy:

* Model;
* Widok;
* Kontroler.

Każda z warstw pełni inną rolę, ale współpracują ze sobą, żeby zapewnić poprawne działanie systemu. Warstwa *model* odpowiada za struktury danych, warstwa *widok* za stronę wizualną aplikacji, a warstwa *kontroler* za obsługę żądań.

Jak zauważyli Lech Madeyski i Michał Stochmiałek *“wartość szkieletu architektonicznego MVC leży w dwóch podstawowych zasadach. Pierwsza to separacja prezentacji i modelu, umożliwiająca zmianę interfejsu użytkownika (np. udostępnienie usług aplikacji poprzez interfejs graficzny i tekstowy). Druga zasada to separacja widoku i kontrolera”*[2].

Zasada działania systemu opartego na modelu MVC polega na tym, że użytkownik wysyła żądanie HTTP, które zostaje dostarczone do *kontrolera* (rys.14). Następnie kontroler przetwarza te dane i wysyła je do modelu, który z kolei zwróci odpowiednie dane do warstwy *widok*. Pierwsza warstwa - klienta, nazywana jest często warstwą prezentacji. Odpowiada za stronę wizualną aplikacji i jest tworzona przy użyciu takich języków jak HTML oraz CSS. Następną jest warstwa aplikacji, która odpowiada za przetwarzanie informacji. Ostatnią jest warstwa danych i do jej zadań należy udostępnianie danych.

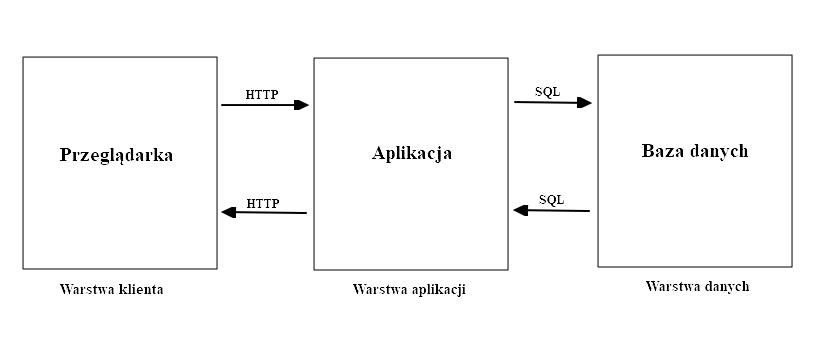


Rys. 14. Schemat MVC

Szkielet *Django* korzysta zewzorca MVC. Jest to szczególnie przydatne, ponieważ podział aplikacji na trzy moduły znacznie przyspiesza pracę programisty. W *Django* wygląd poszczególnych modułów definiuje się w różnych plikach:

* Model w pliku *models.py*,
* Widok w pliku *views.py*,
* Kontroler w pliku *urls.py*.

Schemat przedstawiony na rys. 15 pokazuje działanie aplikacji bazującej na architekturze trójwarstwowej. Warstwa aplikacji będzie działała na platformie Django i będzie napisana w języku programowania Python. Będzie ona pośrednikiem pomiędzy przeglądarką, a bazą danych. W teorii komunikacja między serwerem, a przeglądarką odbywa się przy pomocy protokołu HTTP, który formułuje stronę dla użytkownika przez jego żądania. Następnie serwer wysyła zapytania SQL do bazy danych w celu uzyskania potrzebnych informacji.



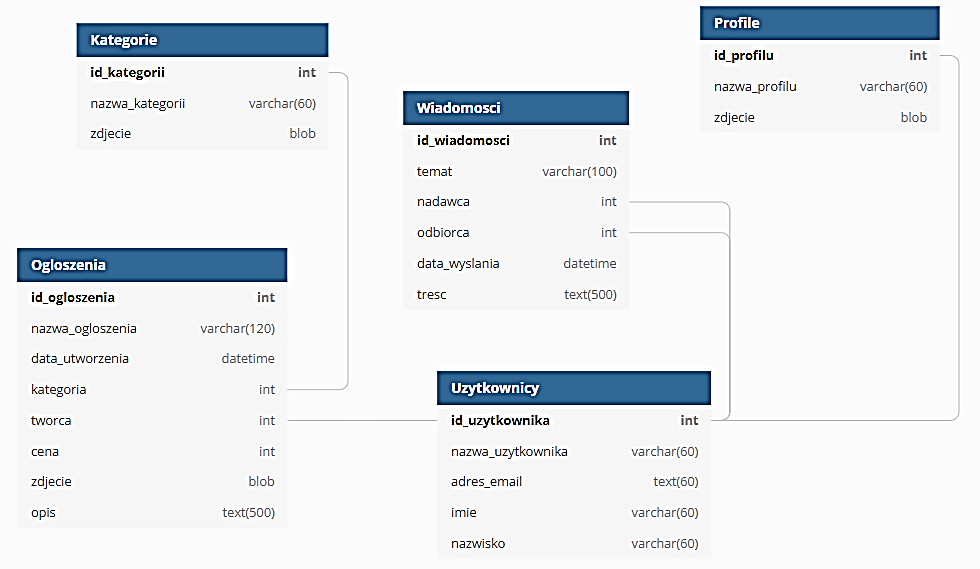
Rys. 15. Schemat architektury trójwarstwowej.

## Model bazy danych

Po wyborze narzędzi pracy następuje etap, w którym należy zastanowić się chwilę nad ogólnym modelem bazy danych. W trakcie tworzenia aplikacji internetowej bardzo ważną rolę pełni baza danych, która przechowuje dowolne informacje zapisywane na stronie. Dzięki niej możliwe jest przykładowo logowanie do serwisu po wcześniejszej rejestracji. W swojej książce Michael J. Hernandez stwierdził, że “*relacyjna baza danych przechowuje dane w relacjach, które są przez użytkowników postrzegane jako tabele. Każda relacja składa się z krotek, zwanych rekordami, oraz atrybutów, zwanych polami*” [1].

W *Django* baza danych powstaje automatycznie przy tworzeniu projektu. W zależności od programisty można ją dowolnie skonfigurować. Na starcie Django oferuje SQLite3, jest to lekka, relacyjna baza danych, która idealnie nadaje się do prostych projektów. Udostępnia on również opcję zmiany bazy danych w ustawieniach projektu, jednak w tym przypadku serwis korzysta z wyżej wymienionej.

Do poprawnego działania baza danych serwisu potrzebuje jedynie kilku tabel. Serwis skupia się głównie na użytkownikach wystawiających ogłoszenia, dlatego baza danych SQLite3 nada się idealnie do przechowywania tego typu informacji. Baza serwisu przechowuje informacje o użytkownikach, ogłoszeniach, kategoriach i profilach (rys.16).



Rys. 16. Schemat bazy danych.

**Tabela 2. Opis szczegółowy pól tabel bazy danych.**

|  |  |
| --- | --- |
| Tabela **Użytkownicy** | **Opis pól** |
| id\_użytkownika | Identyfikator użytkownika |
| nazwa\_użytkownika | Nazwa użytkownika podana przy rejestracji |
| adres\_email | Adres email użytkownika |
| imie | Adres email użytkownika |
| nazwisko | Nazwisko użytkownika |

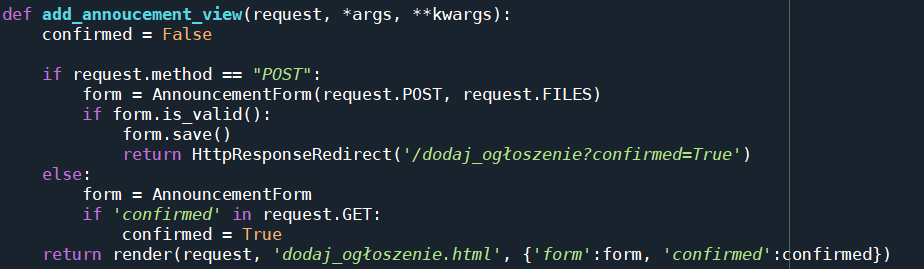
**Tabela 3. Opis szczegółowy pól tabel bazy danych (cd.).**

|  |  |
| --- | --- |
| Tabela **Kategorie** |  |
| id\_kategorii | Identyfikator kategorii |
| nazwa\_kategorii | Nazwa kategorii |
| Tabela **Ogłoszenia** |  |
| id\_ogłoszenia | Identyfikator ogłoszenia |
| nazwa\_ogłoszenia | Nazwa ogłoszenia |
| data\_utworzenia | Data utworzenia ogłoszenia |
| kategoria | Kategoria do wyboru z listy |
| twórca | Użytkownik który stworzył ogłoszenie |
| cena | Cena podana przy tworzeniu ogłoszenia |
| zdjęcie | Zdjęcie wgrane przy tworzeniu ogłoszenia |
| opis | Opis ogłoszenia |
| Tabela **Profile** |  |
| id\_profilu | Identyfikator profilu |
| nazwa\_profilu | Nazwa obecnie zalogowanego użytkownika |
| zdjęcie | Awatar użytkownika |
| Tabela **Wiadomości** |  |
| id\_wiadomosci | Identyfikator wiadomości |
| temat | Temat wysłanej wiadomości |
| nadawca | Nadawca wiadomości |
| odbiorca | Odbiorca wiadomości |
| data\_wyslania | Data wysłania wiadomości |
| tresc | Treść wysłanej wiadomości |

Źródło: opracowanie własne.

## Implementacja wybranych funkcjonalności aplikacji

**Funkcja umożliwiająca dodanie ogłoszenia**

Opisując zaimplementowane funkcjonalności w systemie, należy zacząć od funkcji dodawania ogłoszenia. Jest to jedna z najbardziej podstawowych funkcjonalności, na której opiera się cały serwis. Umożliwia ona dodanie ogłoszenia do serwisu po wcześniejszym wypełnieniu odpowiedniego formularza.

Powyższy rysunek przedstawia funkcję, która odpowiada za dodawanie ogłoszeń. Jej mechanizm polega na sprawdzeniu, czy przesłana metoda żądania jest typu POST. Jeśli formularz jest poprawny, to zapisuje go i przekierowuje użytkownika do podstrony, na której wyświetlony jest komunikat o pomyślnym dodaniu ogłoszenia.

Rys. 17. Implementacja funkcji umożliwiającej dodanie ogłoszenia.

**Funkcja umożliwiająca modyfikacje ogłoszenia**

Obraz zawierający tekst, monitor, ekran, zrzut ekranu

Opis wygenerowany automatycznieKolejną podstawową funkcjonalnością systemu jest możliwość dowolnej modyfikacji dodanego ogłoszenia. Zasada działania jest podobna do poprzednio opisywanej funkcji, z tą różnicą, że zawiera inne elementy. W tym przypadku funkcja pobiera identyfikator ogłoszenia wybranego do modyfikacji. Następnie dane konkretnego ogłoszenia udostępnia za pomocą formularza i jeśli jest wypełniony prawidłowy, to zapisuje go i przekierowuje użytkownika do strony z ogłoszeniami wraz z wyświetlonym komunikatem o pomyślnym zmodyfikowaniu ogłoszenia.

Rys. 18. Implementacja funkcji umożliwiającej modyfikację ogłoszenia.

**Funkcja umożliwiająca wyszukanie ogłoszeń po nazwie**

Obraz zawierający tekst

Opis wygenerowany automatycznieWyszukiwarka znajdująca się na stronie głównej serwisu daje możliwość wyszukania ogłoszeń po dowolnej frazie. Żeby to było możliwe, należało najpierw zaimplementować funkcję, która zwraca ogłoszenia w zależności od wpisanej frazy (rys.19).

W pierwszej kolejności sprawdzane jest, czy metoda żądania jest typu POST. Jeśli tak, to funkcja tworzy zmienną, która zawiera wyszukiwaną frazę. Następnie na podstawie tej zmiennej wybierane są ogłoszenia, które zostają wyświetlone na stronie. W przeciwnym przypadku system przenosi użytkownika do strony, na której znajduje się komunikat informujący o wystąpieniu błędu podczas wyszukiwania frazy.

Rys. 19. Implementacja wyszukiwarki.

**Funkcja umożliwiająca wyświetlanie profilu**

Obraz zawierający tekst

Opis wygenerowany automatycznieKażdy użytkownik w swoim profilu ma dostęp do takich opcji, jak: skrzynka odbiorcza, przegląd zamieszczonych ogłoszeń, zmiana hasła i zmiana podstawowych danych. Dzięki użyciu specjalnego rozwiązania profil użytkownika jest tworzony automatycznie przy rejestracji konta. Dodatkowo profil jest inny dla każdego użytkownika i żeby go wyświetlić, należy być zalogowanym na swoim koncie.

Do zaimplementowania funkcjonalności znajdującej się na rysunku 20 użyto *DetailView*, czyli widoku generycznego, który wyświetla dane pojedynczego egzemplarza modelu. Jest to szczególnie przydatne rozwiązanie podczas implementacji profili, ponieważ pozwala zaoszczędzić wiele czasu. Najpierw w klasie zainicjowany zostaje model i nazwa szablonu, z którego korzysta funkcjonalność. Następnie w funkcji wywoływana jest metoda *get\_context\_data,* pozwalająca uzyskać dodatkowe informacje w szablonie. Dzięki niej możliwe jest dodanie zmiennej o nazwie *page\_user*, która jest różna dla każdego użytkownika.

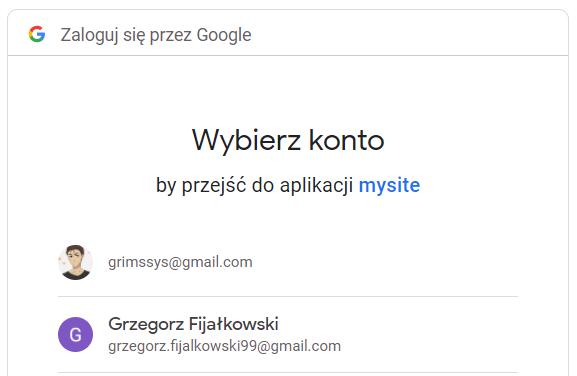
Rys. 20. Implementacja profilu użytkownika.

## Integracja logowania z kontem Google

Jedną z ciekawszych funkcjonalności zaimplementowanych w systemie jest logowanie za pomocą konta Google. Wraz z rozwojem technologii powstaje coraz więcej aplikacji, z czego każda z nich wymaga rejestracji. Korzystanie z Google API pozwala trzymać to wszystko w jednym miejscu i logować się na różne aplikacje za pomocą jednego konta Google. W przypadku implementacji tego rozwiązania w Django należy użyć pakietu *django-allauth*, który zajmuje się uwierzytelnianiem, rejestracją, zarządzaniem oraz uwierzytelnieniem kont zewnętrznych.

Proces integracji aplikacji z Google API przebiega w następujących krokach:

1. Instalacja pakietu *django-allauth* i dodanie do INSTALLED\_APPS kilku wierszy konfiguracji.
2. Dodanie zainstalowanego pakietu do adresów URL aplikacji.
3. W konsoli interfejsów API Google utworzyć nowy projekt, a następnie zarejestrować aplikację i wypełnić ekran akceptacji OAuth.
4. Stworzenie nowego identyfikatora klienta OAuth i nadanie mu nazwy wraz z uzupełnieniem identyfikatorów url.
5. W panelu administratora należy wejść w obszar strony i dodać nową ustawiając 127.0.0.1:8000 jako nazwę domeny i nazwę wyświetlaną.
6. Ponownie w panelu administratora należy wejść w obszar aplikacje społecznościowe i dodać nową uzupełniając dane kluczami uzyskanymi w punkcie czwartym.
7. Dodać kod w szablonie pozwalający na logowanie za pomocą konta Google.

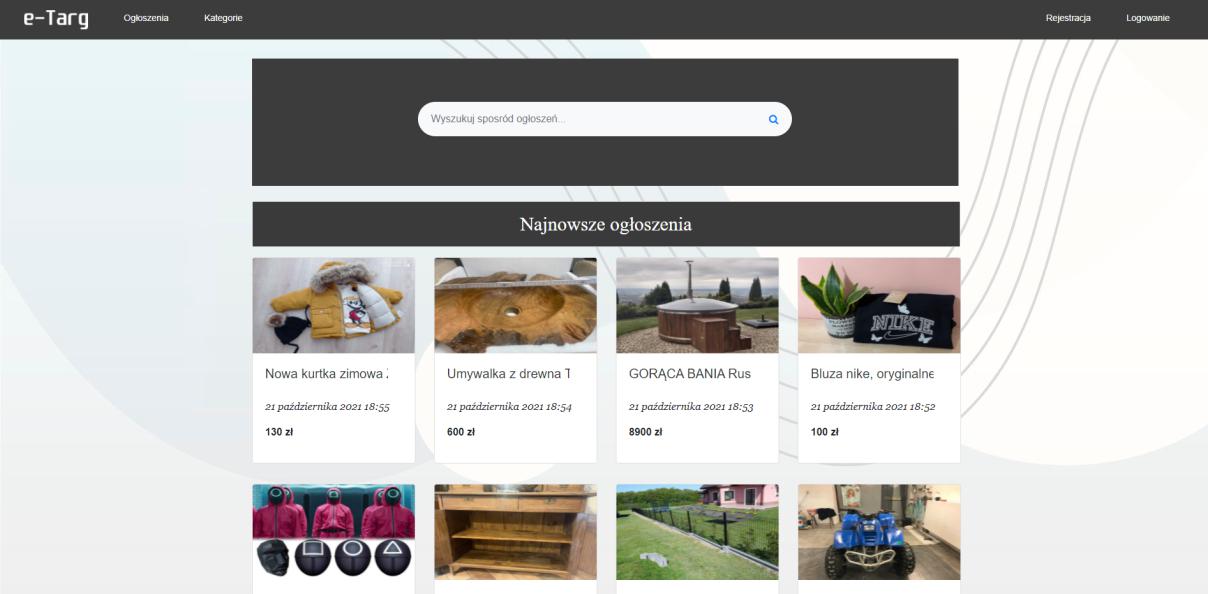


Rys. 21. Interfejs logowania Google.

# Wygląd i działanie prototypu systemu

**Strona główna**

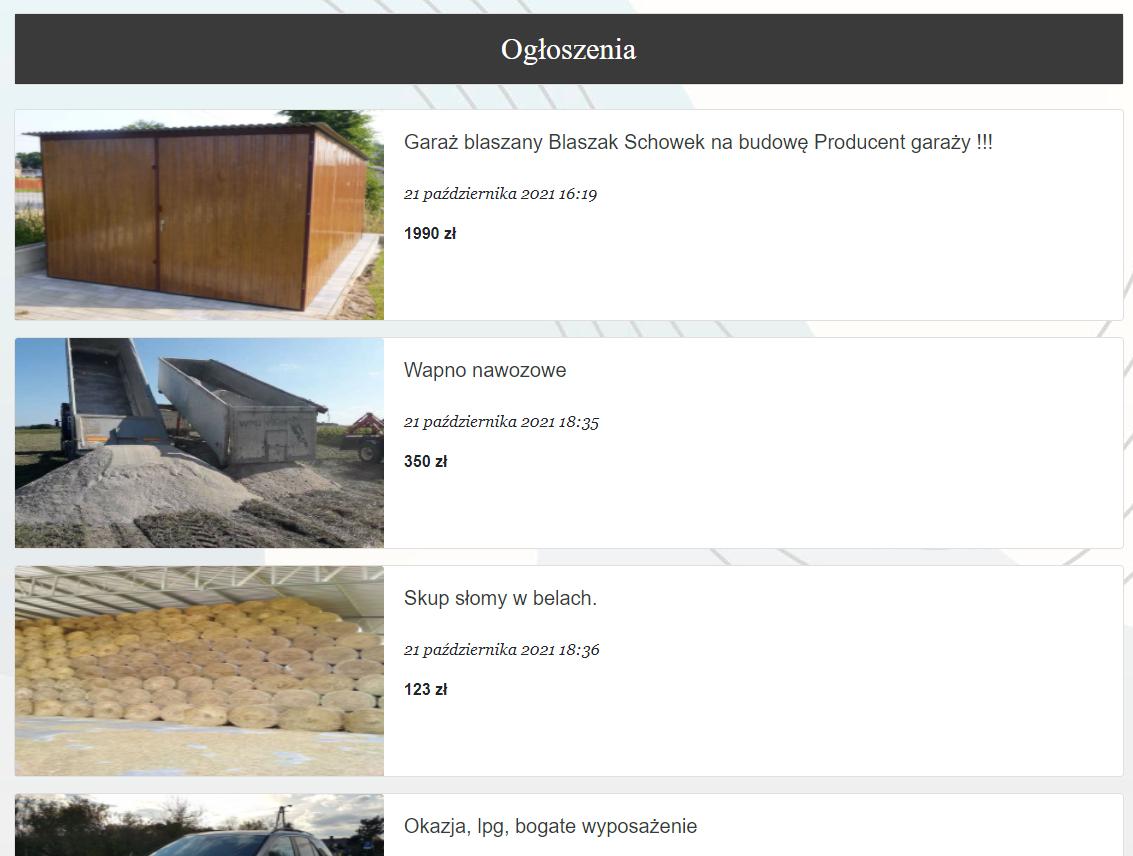
Po zalogowaniu się w systemie na ekranie wyświetlona zostaje strona główna aplikacji (rys.22) . Układ strony powstał na wzór największych serwisów ogłoszeniowych. Na samej górze znajduje się pasek nawigacyjny, który zapewnia użytkownikowi łatwy dostęp do wszystkich dostępnych opcji serwisu. Jest on aktywny cały czas pozwalając na zwinne poruszanie się po stronach. W środkowej sekcji zlokalizowana jest wyszukiwarka umożliwiająca użytkownikowi wyszukanie pożądanego przez niego ogłoszenia w dowolnym momencie. Pod wyszukiwarką widnieje lista dwunastu najnowszych ogłoszeń ułożonych po cztery w rzędzie. Taki zabieg pomaga sprzedającym zwrócić uwagę większej liczby klientów na ich ogłoszenie. Dodatkowo pozostawia to otwartą furtkę do dalszej rozbudowy serwisu, np. o ogłoszenia promowane lub mikropłatności.



Rys. 22. Strona główna aplikacji.

**Lista ogłoszeń**

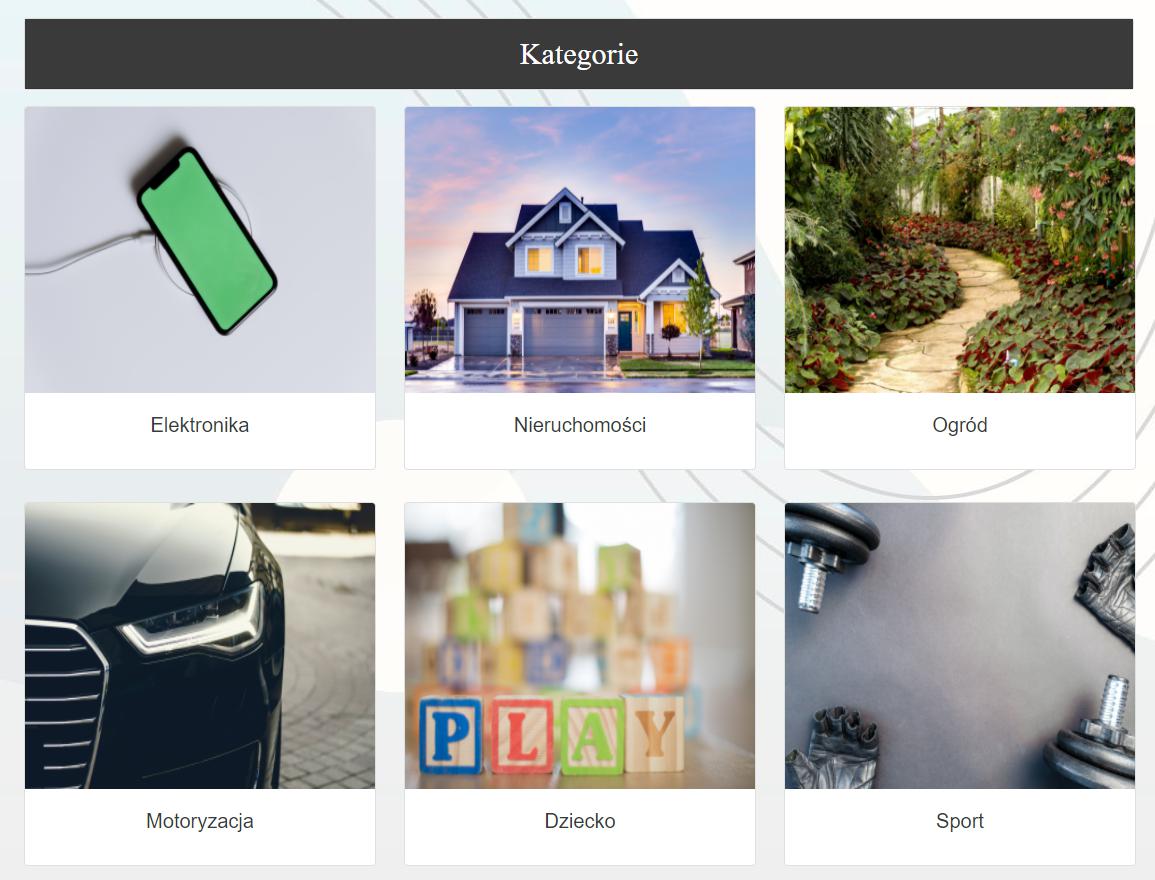
Lista ogłoszeń wyświetla wszystkie ogłoszenia dodane w serwisie i uporządkowuje je jedno pod drugim. Można się do niej dostać za pomocą paska nawigacyjnego, klikając na odpowiednią pozycję. Lista dzieli się na strony, na których może się znajdować maksymalnie siedem ogłoszeń. Na samym dole widnieje lista dostępnych stron do wyboru (rys. 23).



Rys. 23. Lista ogłoszeń.

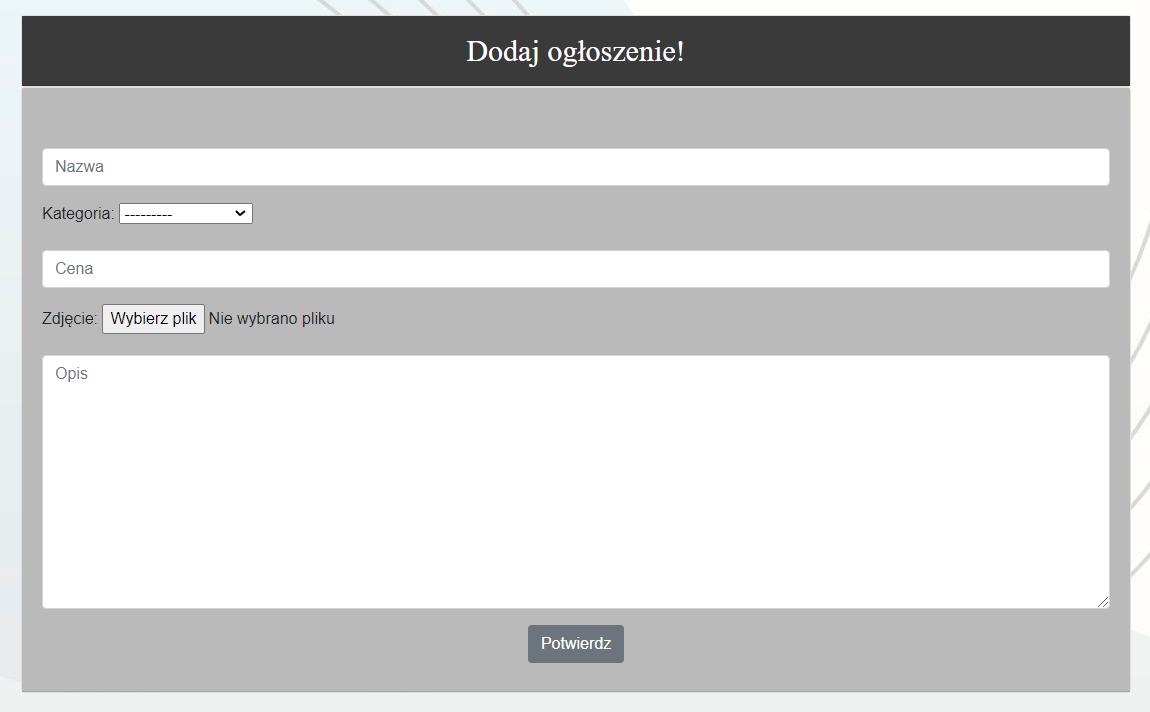
**Lista kategorii**

Lista kategorii pozwala użytkownikowi na zawężenie grupy poszukiwań danego ogłoszenia do ogłoszeń podpiętych pod daną kategorię. Podobnie jak w przypadku listy ogłoszeń można się do niej dostać używając paska nawigacyjnego oraz klikając odpowiednią pozycję. W prototypie systemu uwzględniono sześć różnych kategorii. Jednak w przypadku dalszego rozwijania aplikacji zostaną dodane kolejne.

**Dodawanie ogłoszenia**

Rys. 24. Lista kategorii.

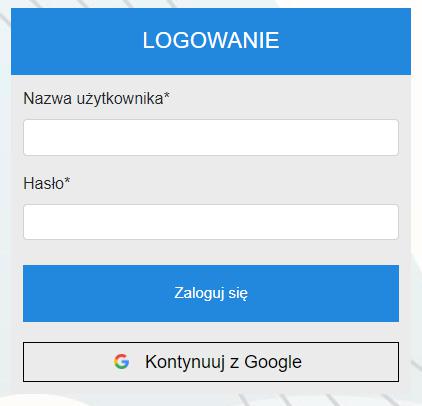
Jest to opcja, która pozwala zarejestrowanym użytkownikowi dodawać ogłoszenia. Dostępna jest jedynie dla zalogowanych użytkowników. Bycie niezalogowanym spowoduje, że strona nie udzieli praw dostępu do niej i wyświetli odpowiedni komunikat o błędzie. Po wypełnieniu odpowiednich pól formularza i potwierdzeniu, ogłoszenie zostaje dodane na stronę (i zapisane w bazie danych). Formularz ogłoszenia zawiera takie pola, jak: nazwa, cena, opis, kategoria i zdjęcie. Przy czym dwa ostatnie nie są polami tekstowymi - w przypadku *kategorii* należy wybrać jedną pozycję z listy wartości, natomiast dla *zdjęcia* należy wybrać jedno ze zdjęć znajdujących się na urządzeniu użytkownika. Całość jest prosta w swojej budowie, dlatego użytkownik, który nie miał nigdy dużej styczności z komputerem, poradzi sobie bez problemu.



Rys. 25. Formularz dodawania ogłoszenia.

**Panel Logowania**

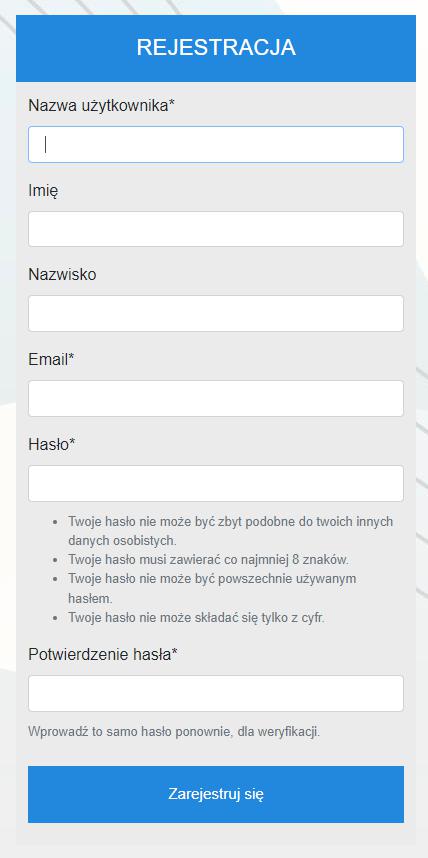
Logowanie jest dostępne dla użytkownika tylko, jeśli wcześniej zarejestrował się, używając odpowiedniego do tego formularza. Istnieje jeszcze dodatkowa opcja, która pozwala na zalogowanie się do serwisu, używając konta Google. Wystarczy kliknąć przycisk “*Kontynuuj z Google”*, który przeniesie użytkownika do autoryzowanej strony logowania Google. Dostęp do panelu logowania można uzyskać, klikając odpowiednią pozycję na pasku nawigacyjnym, która znajduje się po prawej stronie zaraz obok pozycji z rejestracją. W panelu znajdują się dwa pola tekstowe, które należy prawidłowo wypełnić, aby pomyślnie zalogować się do serwisu. Jest to nazwa użytkownika oraz hasło, które było podawane podczas rejestracji. W przypadku, kiedy użytkownik wprowadzi błędne dane, wyświetla się na górze komunikat o błędnych danych. W przypadku powodzenia przenosi użytkownika na stronę główną z komunikatem mówiącym o zalogowaniu się w serwisie



Rys. 26. Panel logowania użytkownika.

**Panel rejestracji**

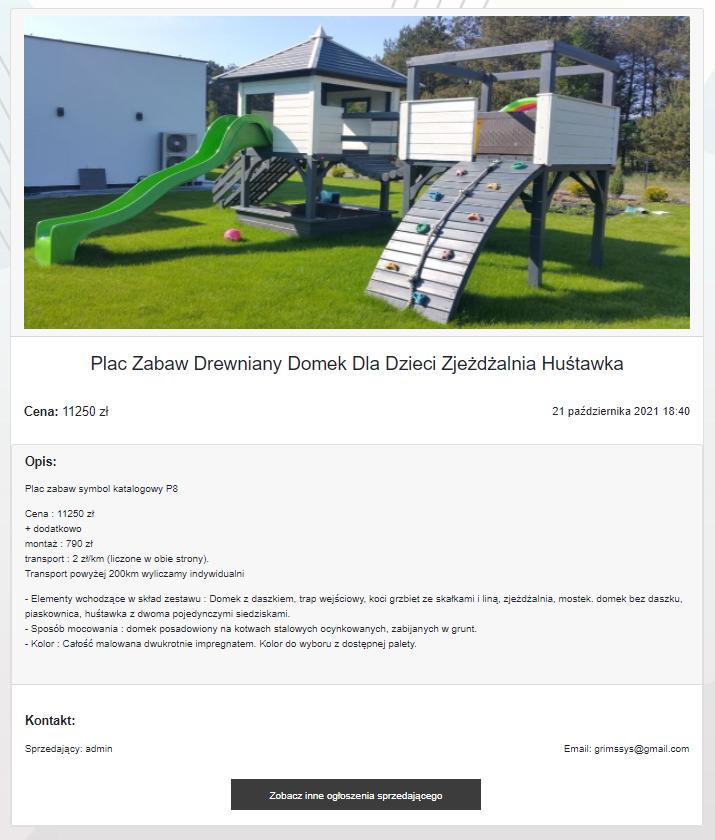
Rejestracja do serwisu jest wymagana, jeżeli użytkownik chce wystawiać ogłoszenia i zarządzać nimi. Panel rejestracji zawiera cztery wymagane pola i są to po kolei nazwa użytkownika, e-mail, hasło oraz potwierdzenie hasła. Dodatkowo panel posiada dwa niewymagane pola, a dokładniej imię i nazwisko. Można je w każdej chwili zmienić w profilu po rejestracji. Podobnie jak w przypadku logowania, kiedy rejestracja zakończy się powodzeniem, przeniesie użytkownika na stronę główną z komunikatem mówiącym o udanej rejestracji w serwisie



Rys. 27. Panel rejestracji użytkownika.

**Podgląd ogłoszenia**

Po wybraniu ogłoszenia z listy użytkownik zostanie przekierowany do strony z jego pełnym opisem. Na samej górze umiejscowione jest zdjęcie ogłoszenia, które po kliknięciu powiększy się do oryginalnych rozmiarów. Pod zdjęciem znajduje się nazwa, cena i data zamieszczenia ogłoszenia. Najwięcej miejsca jest przygotowane na opis ogłoszenia, który umiejscowiony jest w centralnym punkcie. W ostatnim fragmencie ogłoszenia znajdują się dane do kontaktu oraz przycisk pozwalający na podejrzenie innych ogłoszeń sprzedającego



Rys. 28. Podgląd ogłoszenia.

**Profil użytkownika**

Profil użytkownika powstaje automatycznie po rejestracji w serwisie. Aby uzyskać do niego dostęp należy wejść w pozycję z nazwą użytkownika na pasku nawigacyjnym. Udostępnia on takie funkcje jak zmiana podstawowych danych, zmiana hasła i podgląd wystawionych ogłoszeń. Profil użytkownika składa się z awatara, imienia, nazwiska i adresu e-mail. Awatar można dowolnie modyfikować w każdej chwili w opcji zmiany danych.



Rys. 29. Profil użytkownika.

**Wysyłanie wiadomości**

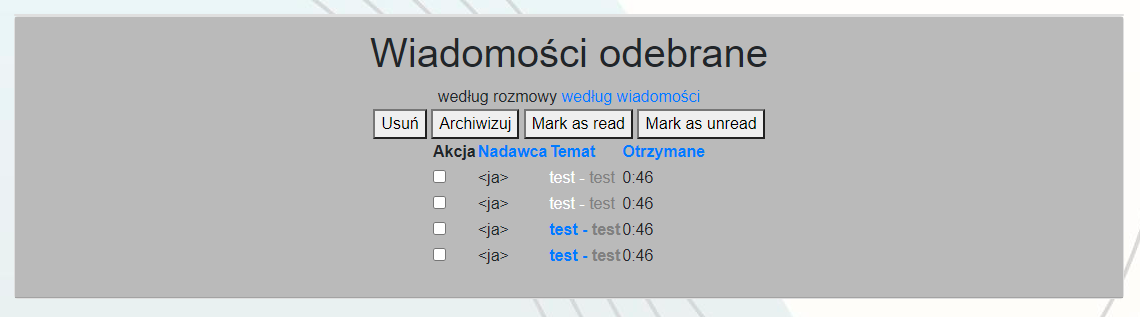
Obraz zawierający tekst

Opis wygenerowany automatycznieFormularz wysyłania wiadomości dostępny jest po wejściu w odpowiednią opcję w podglądzie ogłoszenia. Umożliwia on komunikację z drugim użytkownikiem serwisu, pozwalając na efektywne zarządzanie czasem. W formularzu wysyłania znajdują się trzy  etykiety do wypełniania takie jak odbiorca, temat i treść wiadomości. W przypadku etykiety odbiorca należy wpisać poprawną nazwę użytkownika serwisu, która znajdowała się w podglądzie ogłoszenia. W przeciwnym wypadku formularz nie pozwoli wysłać wiadomości. Podobnie sprawa ma się do tematu wiadomości, gdzie nie można zostawić jej pustej, ponieważ wyświetli się komunikat mówiący o tym, że trzeba wypełnić to pole. Jeśli wszystkie dane zostały wpisane prawidłowo to po chwili przetwarzania, system  wyśle wiadomość do odbiorcy, a użytkownik zostanie przeniesiony do strony ogłoszenia, z którego wysyłał wiadomość. Jednocześnie pojawi się komunikat o udanym procesie wysyłania.

Rys. 30. Formularz wysyłania wiadomości.

**Skrzynka odbiorcza**

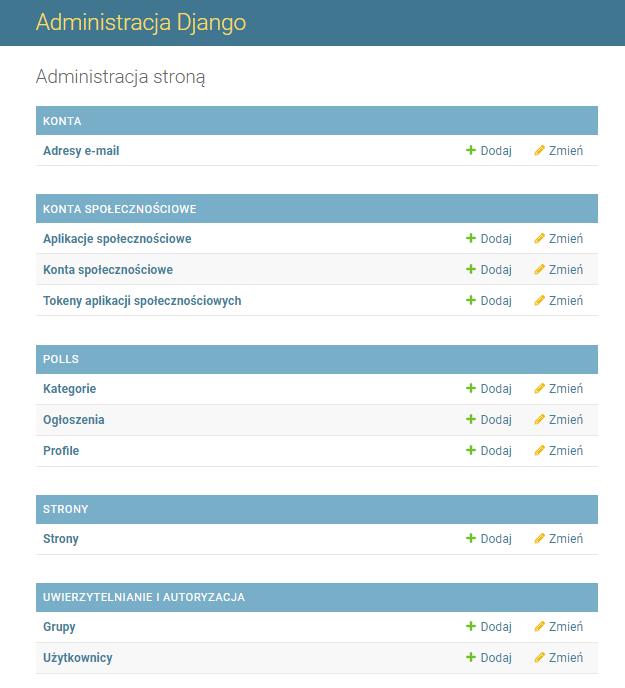
Kolejną funkcjonalnością związaną z wiadomościami jest skrzynka odbiorcza, która zawiera wszystkie otrzymane wiadomości. Dostępna jest ona w profilu użytkownika po wejściu w odpowiednią opcję. Znajdują się w niej informacje takie jak nadawca, temat i czas otrzymania wiadomości. Po wejściu w pojedynczą wiadomość wyświetla się interfejs z treścią oraz pole do odpisania. Dodatkowo istnieje kilka możliwości zarządzania wiadomościami.



Rys. 31. Skrzynka odbiorcza.

**Panel administratora**

Automatyczny panel administratora pozwala zaoszczędzić sporo czasu programiście, który nie musi go projektować od początku. Jest to jedna z głównych zalet Django, która daje mu przewagę nad innymi narzędziami do budowy aplikacji internetowych. Szkielet Django zapewnia automatyczny interfejs administratora, który jest zdatny do użytku zaraz po rozpoczęciu projektu. Umożliwia on odczyt, tworzenie i modyfikowanie danych w modelu. Żeby dostać się do panelu, należy wpisać normalny adres aplikacji w przeglądarce oraz dodać po ukośniku “admin”. W tym momencie przeglądarka przekierowuje użytkownika do panelu administratora, gdzie konieczne jest zalogowanie, podając dane administratora serwisu. Pierwszy raz konto administratora tworzy się w linii poleceń. Na rysunku 32 jest zrzut ekranu aplikacji, przedstawiający panel administratora. Zawiera modele, które można automatycznie modyfikować na stronie. Jest to przydatna funkcja ze względu na to, że pozwala na łatwe zarządzanie danymi znajdującymi się w serwisie.



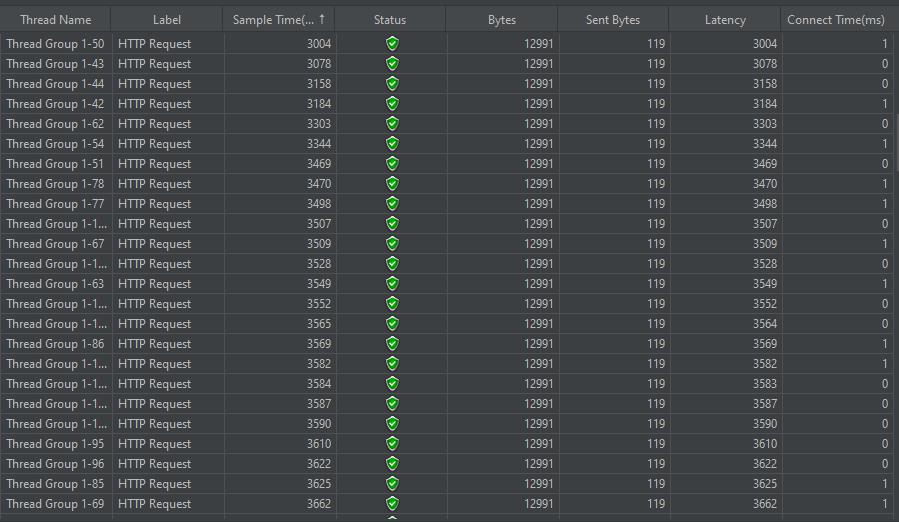
Rys. 32. Panel administratora.

# Wyniki testowania prototypu serwisu

Ostatni etap pracy polegał na testowaniu aplikacji za pomocą różnych narzędzi informatycznych. Testowanie pozwala określić poprawność zaimplementowanych funkcjonalności systemu oraz w zależności od rodzaju testu, stabilność i bezpieczeństwo danych. W pracy przeprowadzono dwa rodzaje testów, wybranych ze względu na ich funkcjonalność. W pierwszej kolejności wykonano testy wydajnościowe, mające na celu sprawdzenie jak zachowuje się aplikacja pod obciążeniem. Następnie za pomocą testu bezpieczeństwa oceniono podatność systemu na zagrożenia z Internetu.

## Testy wydajnościowe

Domyślnym założeniem aplikacji jest ciągły napływ nowych użytkowników korzystających z serwisu. Wraz z większym ruchem na stronie mogą pojawiać się różnego rodzaju problemy wydajnościowe polegające między innymi na zwiększeniu czasu oczekiwania na odpowiedź aplikacji. W najgorszym przypadku może to doprowadzić do zatrzymania działania systemu. Biorąc pod uwagę możliwość wystąpienia tych problemów, należy zrozumieć jak ważne jest wykonywanie testów wydajnościowych pozwalających zasymulować obciążenie aplikacji. Wyniki testowania jest pokazany na rysunku 33.



Rys. 33. Wyniki testu wydajnościowego prototypu serwisu ogłoszeniowego.

Źródło: opracowanie własne.

Do tworzenia testów wydajnościowych zostało użyte narzędzie *Apache Jmeter*. Jest to najpopularniejsze rozwiązanie na rynku do testowania obciążenia oraz analizy wydajności. Na potrzeby testów zostały przygotowane cztery scenariusze, w których serwer ma obsłużyć określoną liczbę żądań w czasie jednej sekundy. Podczas wyboru ilości żądań wzięte zostały realne możliwości serwera i jest po kolei 50, 100, 200 i 500 żądań na sekundę.

Testowanie za pomocą programu *JMeter* rozpoczyna się od stworzenia planu testów. Następnym krokiem jest stworzenie grupy wątków, gdzie znajdują się takie informacje jak symulowana liczba użytkowników, czas, w jakim mają zostać wysłane żądania i ilość żądań. Gdy parametry w grupie wątków zostaną ustawione, to należy dodać żądanie http, w którym należy uzupełnić dane takie jak protokół, adres strony i numer portu. Ostatnim etapem jest dodanie elementu, który przechowuje wyniki żądań HTTP i przedstawia je w dowolnej formie dla użytkownika. W tabeli 4. zostały przedstawione wyniki testowania, badające czas oczekiwania na odpowiedź serwera i ilość potrzebnego czasu do pełnego załadowania strony. Testy były prowadzone dla żądania GET, które miało otworzyć stronę główną serwisu. Analizując otrzymane wyniki można określić, jaka liczba żądań wykonanych w ciągu sekundy przejdzie prawidłowo, a jaka nie dostanie żadnej odpowiedzi od serwera.

**Tabela 4.** **Porównanie testów.**

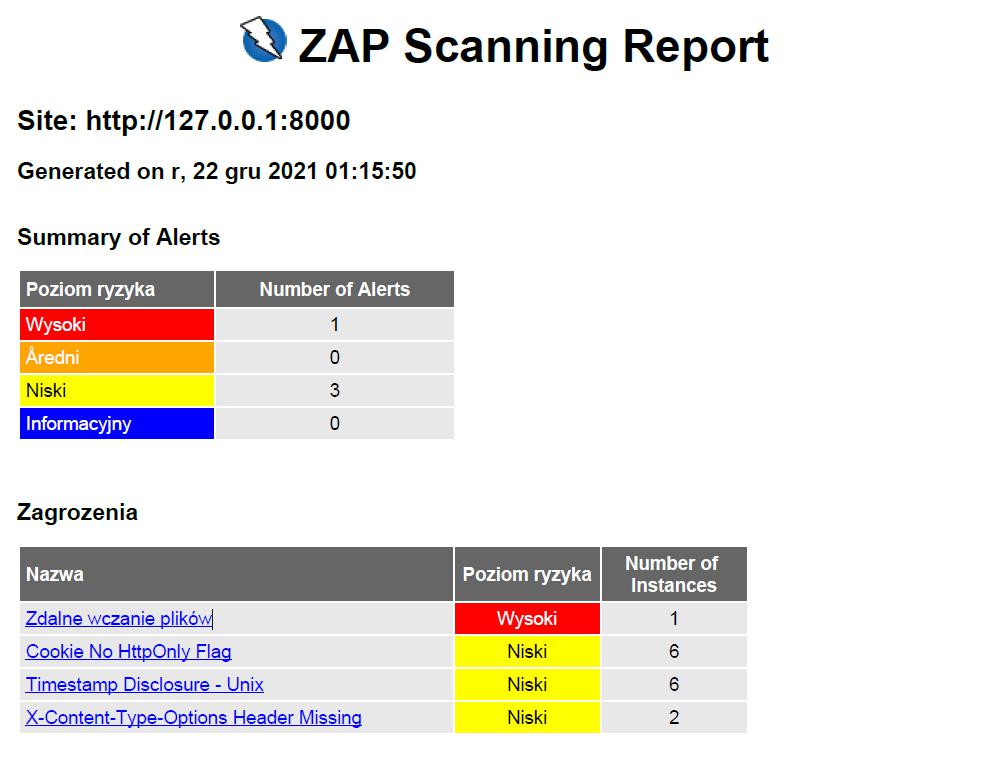
|  |  |  |
| --- | --- | --- |
| **Ilość żądań** | **Powtórzenia** | **Opis wyniku testowania** |
| 50 | 10 | Nie wystąpiły żadne komplikacje, a wszystkie testy zostały pomyślnie ukończone z dobrym czasem oczekiwania. |
| 100 | 10 | Podobnie jak w pierwszym przypadku wszystkie testy zostały zakończone pozytywnie, jednak czas oczekiwania wzrósł. |
| 200 | 10 | Mały odsetek testów zakończył się negatywnie jednak serwer nadal dobrze radzi sobie z taką ilością żądań na sekundę, ponieważ w większości testy zakończyły się pozytywnie. |
| 500 | 10 | Niecała połowa testów zakończyła się pozytywnie, co pokazuje, że serwer nie radzi już sobie z obsługą tylu żądań na sekundę. |

Źródło: opracowanie własne.

Na podstawie testów okazało się, że optymalna liczba żądań, jaką może obsłużyć serwer wynosi około 175. Przy większej ilości żądań podczas testu w JMeter zaczyna pojawiać się błąd, który mówi o tym, że aplikacja może być przeciążona. Należy wziąć pod uwagę fakt, że konfiguracja systemu ma wpływ na to jak działa JMeter i im jest ona lepsza tym system może generować większe obciążenie. Wraz z rozwojem aplikacji liczba żądań jaką może obsłużyć serwer powinna zostać zwiększona.

## Testy bezpieczeństwa

Po zakończeniu testów wydajnościowych przyszła pora na sprawdzenie bezpieczeństwa systemu. Testy bezpieczeństwa są jednymi z ważniejszych, ponieważ badają podatność systemu na zagrożenia, a co za tym idzie, pozwalają wykryć problem zanim wykorzysta go osoba nieautoryzowana. Dzięki nim możliwe jest potwierdzenie, że system jest w pełni bezpieczny albo wręcz przeciwnie, że zawiera luki, które programiści muszą naprawić.

***ZAP*** (Zed Attack Proxy) jest *open sourcowym* narzędziem tworzonym przez organizację OWASP (Open Web Application Security Project) wspomagającym testy penetracyjne, sprawdzające poziom bezpieczeństwa aplikacji webowych. Działa na zasadzie proxy, dzięki czemu pozwala nie tylko na podglądanie żądań wysyłanych do serwera aplikacji i odpowiedzi z serwera otrzymywanych, ale daje również możliwość debugowania, modyfikowania oraz wysyłania własnych żądań. Jest narzędziem dostosowanym do obsługi HTTP, oferuje prosty sposób na rozszyfrowanie HTTPS poprzez dodanie własnego certyfikatu do przeglądarki. Dostarcza automatyczne skanery, jak również narzędzia pomagające manualnie testować aplikację. Wyróżnia się cztery poziomy stopnia ryzyka i są to poziom wysoki, średni, niski i informacyjny.

Rys. 34. Wyniki testu bezpieczeństwa prototypu serwisu ogłoszeniowego.

Źródło: opracowanie własne.

Raport testu bezpieczeństwa wykonany za pomocą narzędzia OWASP ZAP dla wykonanego przeze mnie serwisu internetowego przedstawia rys.34. W raporcie znajdują się informacje na temat wykrytych zagrożeń, stopienia ich ryzyka i ich szczegółowy opis.

W przypadku testowania prototypu serwisu ogłoszeniowego wykryto cztery zagrożenia o stopniu wysokim i niskim. W tym przypadku należy podjąć decyzję, czy takie zagrożenie jest akceptowalne, czy należy je usunąć. Stwierdziłem, że mimo wykryte zagrożenie wysokiego poziomu nie wpływa bezpośrednio na działanie systemu (nie zostawia „otwartych drzwi” osobom nieautoryzowanym), dlatego jest akceptowalne i aplikacja nie wymaga korekty.

# Podsumowanie

Głównym celem pracy było opracowanie projektu serwisu ogłoszeniowego i jego implementacja za pomocą technologii webowych. Do implementacji projektu wykorzystane zostały nowoczesne narzędzia programistyczne stosowane przez wiele dużych firm informatycznych. Z kolei podczas testowania wykonanej aplikacji zostały użyte najbardziej popularne programy, umożliwiające nie tylko sprawdzenie poprawność działania zaimplementowanej funkcjonalności dla różnych obciążeń serwera, jak również jej odporność na zagrożenia płynące z Internetu.

Patrząc na rozwój największych serwisów ogłoszeniowych na świecie stwierdziłem, że utworzony przeze mnie prototyp aplikacji to dopiero zalążek profesjonalnego serwisu ogłoszeniowego i istnieje wiele opcji jego rozwoju. W pierwszej kolejności przy dalszej rozbudowie aplikacji należałoby dodać inne kategorie. Niestety w chwili obecnej serwis nie posiada ich zbyt wiele, a większa liczba kategorii na pewno zachęciłaby potencjalnych klientów do korzystania z serwisu. Dodatkowo warto byłoby jeszcze zaimplementować podkategorie, które umożliwiłby łatwiejsze wyszukiwanie towaru. Spośród innych rozszerzeń wartych rozważenia można zaliczyć: dodawanie ogłoszeń do obserwowanych, rozwinięcie opcji wysyłania wiadomości czy dodanie nowych funkcjonalności w profilu użytkownika.

# Bibliografia

**Literatura**

1. Hernandez M. Projektowanie baz danych dla każdego, Helion, 2014.
2. Madeyski L., Stochmiałek M., Architektura nowoczesnych aplikacji internetowych. W: Inżynieria oprogramowania po roku 2004: nowe wyzwania. (red. Jerzy Górski). WNT, Warszawa, 2004.
3. Paczkowski T., Kamela A., Szyl M., Ekonomia współdzielenia - Sharing economy narodziny nowego systemu ekonomicznego? Przyszłość, szanse i zagrożenia. Współczesne problemy ekonomiczne w badaniach młodych naukowców T. 4, Teoria i praktyka, Wydawnictwo Uniwersytetu w Białymstoku, 2020, s. 60–73.
4. Schmuller J., UML dla każdego. Helion, 2003.
5. Wiegers K., Beatty J., Specyfikacja oprogramowania. Inżynieria wymagań. Wydanie III. Helion, 2014.

**Źródła internetowe**

1. About Python, <https://pythoninstitute.org/what-is-python> , (dostęp dn. 22.12.2021).
2. Apache JMeter, <https://jmeter.apache.org> (dostęp dn. 22.12.2021).
3. Django introduction, <https://developer.mozilla.org/en-US/docs/Learn/Serverside/Django/Introduction> (dostęp dn. 22.12.2021)
4. eBay, http://www.ebay.com, (dostęp dn. 18.11.2021).
5. Google APIs, <https://en.wikipedia.org/wiki/Google_APIs>   
    (dostęp dn. 22.12.2021).
6. Grid system – Bootstrap, <https://getbootstrap.com/docs/4.0/layout/grid>,

(dostęp dn. 12.12.2021).

1. Gumtree, <http://www.gumtree.com> , (dostęp dn. 24.11.2021).
2. Olx, <http://www.OLX.pl> (dostęp dn. 20.11.2021).
3. OWASP ZAP, <https://www.zaproxy.org/getting-started>   
    (dostęp dn. 22.12.2021).
4. Sprzedajemy, <http://sprzedajemy.pl> , (dostęp dn. 04.12.2021)

# Spis tabel

[**Tabela 1. Ocena wybranych serwisów ogłoszeniowych.** 9](#_Toc93763096)

[**Tabela 2. Opis szczegółowy pól tabel bazy danych.** 24](#_Toc93763097)

[**Tabela 3. Opis szczegółowy pól tabel bazy danych (cd.).** 25](#_Toc93763098)

[**Tabela 4.** **Porównanie testów.** 40](#_Toc93763099)

# Spis rysunk**ów**

[Rys. 1. Strona główna serwisu olx 6](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578327)

[Rys. 2. Strona główna serwisu eBay 7](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578328)

[Rys. 3. Strona główna serwisu sprzedajemy 8](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578329)

[Rys. 4. Strona główna serwisu Gumtree. 9](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578330)

[Rys. 5. Diagram przypadków użycia - tworzenie ogłoszenia 12](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578331)

[Rys. 6. Diagram przypadków użycia - wyszukiwanie ogłoszeń 13](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578332)

[Rys. 7. Diagram przypadków użycia – profil użytkownika. 13](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578333)

[Rys. 8. Diagram klas 15](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578334)

[Rys. 9. Diagram czynności - zarządzanie ogłoszeniami 16](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578335)

[Rys. 11. Diagram czynności - logowanie do serwisu 17](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578336)

[Rys. 10. Diagram czynności - zmiana hasła 17](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578337)

[Rys. 12. Diagram czynności - edycja danych. 18](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578338)

[Rys. 13. Diagram czynności - wysyłanie wiadomości. 19](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578339)

[Rys. 14. Schemat MVC 22](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578340)

[Rys. 15. Schemat architektury trójwarstwowej. 23](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578341)

[Rys. 16. Schemat bazy danych. 24](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578342)

[Rys. 17. Implementacja funkcji umożliwiającej dodanie ogłoszenia. 26](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578343)

[Rys. 18. Implementacja funkcji umożliwiającej modyfikację ogłoszenia. 26](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578344)

[Rys. 19. Implementacja wyszukiwarki. 27](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578345)

[Rys. 20. Implementacja profilu użytkownika. 27](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578346)

[Rys. 21. Interfejs logowania Google. 28](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578347)

[Rys. 22. Strona główna aplikacji. 29](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578348)

[Rys. 23. Lista ogłoszeń. 30](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578349)

[Rys. 24. Lista kategorii. 31](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578350)

[Rys. 25. Formularz dodawania ogłoszenia. 32](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578351)

[Rys. 26. Panel logowania użytkownika. 33](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578352)

[Rys. 27. Panel rejestracji użytkownika. 34](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578353)

[Rys. 28. Podgląd ogłoszenia. 35](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578354)

[Rys. 29. Profil użytkownika. 36](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578355)

[Rys. 30. Formularz wysyłania wiadomości. 36](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578356)

[Rys. 31. Skrzynka odbiorcza. 37](#_Toc93578357)

[Rys. 32. Panel administratora. 38](#_Toc93578358)

[Rys. 33. Wyniki testu wydajnościowego prototypu serwisu ogłoszeniowego. 39](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578359)

[Rys. 34. Wyniki testu bezpieczeństwa prototypu serwisu ogłoszeniowego. 41](file:///C:\Users\Mirka\Desktop\Dyplomanci\formatki%20prac\Rok%20akademicki%202021\Fijałkowski%20Grzegorz_ok\Praca_Fijalkowski_19sty.docx#_Toc93578360)