

Wydział Matematyki i Nauk Informacyjnych Politechniki Warszawskiej



*“Internetowy serwis do zamawiania artykułów spożywczych  
realizowany na zasadzie modelu P2P”*

Aleksy Miśtał  
Krzysztof Pranczk

v0.6

26.12.2016

Lista zmian w dokumencie:

14.11.2016	Ukończenie aplikacji administratora (skończenie projektu nr 1)	Dodanie szczegółowego opisu zastosowanej architektury	v.0.2
21.11.2016	Wstępna specyfikacja dot. projektu nr 2	Dodanie dostawcy oraz architektury rozwiązania	v.0.3
13.12.2016	Ukończenie projektu nr 2 - kontrola serwisu poprzez Web API	Dodanie dokumentacji powykonawczej dot. projektu nr 2	v.0.4
26.12.2016	Wstępna specyfikacja dot. projektu nr 3	Dodanie Use Case'ów, User Stories dla użytkownika	v.0.5
06.02.2016	Ukończenie projektu nr 3 - strona WWW dla klienta	Dodanie dokumentacji powykonawczej projektu nr 3	v.0.6

## Spis treści

<b>Spis treści</b>	<b>1</b>
<b>Specyfikacja</b>	<b>2</b>
Opis biznesowy	2
Wymagania funkcjonalne	2
Wymagania niefunkcjonalne	8
Architektura rozwiązania	9
Architektura serwisu Web API	10
<b>2. Dokumentacja końcowa (powykonawcza)</b>	<b>11</b>
2.1. Wymagania systemowe	11
2.2. Wykorzystane komponenty	11
2.3. Instrukcja instalacji	12
2.4. Instrukcja uruchomienia	12
2.5. Instrukcja użycia	13
2.6. Model danych	19
2.7. Raport z przeprowadzonych testów	20
2.8. Raport odstępstw od specyfikacji wymagań	21

# 1. Specyfikacja

## 1.1. Opis biznesowy

Niżej opisany system docelowo będzie służył do udostępniania usług osób prywatnych chętnych do zrobienia zakupów w okolicznych punktach. Serwis jest ułatwieniem dla osób chcących zrobić zakupy bez wychodzenia z domu, taniej i szybciej niż w firmach, które się tym specjalizują. Stworzone oprogramowanie, które zostało niżej opisane nie jest udostępniane z użyciem otwartych licencji.

## 1.2. Wymagania funkcjonalne

**Use Cases (wersja desktopowa i wersja webowa)**



rys. 1

## User Stories

ID	Aktor	Nazwa	Opis	Odpowiedź systemu
Admin	Administrator	Przeglądaj użytkowników	Przeglądanie listy zarejestrowanych lub czekających na zatwierdzenie rejestracji użytkowników	Zwrócenie listy wszystkich użytkowników z bazy danych
		Sortuj użytkowników	Sortowanie użytkowników wg. podstawowych atrybutów	Zwrócenie użytkowników z bazy danych spełniających zadane kryteria
		Edytuj dane użytkownika	Edycja danych danego użytkownika głównie w celu zatwierdzenia rejestracji	Aktualizacja wprowadzonych zmian w bazie danych
		Usuń użytkownika	Usunięcie użytkownika na własne życzenie (administratora)	Usunięcie z bazy danych wskazanego użytkownika
Klient		Wyświetl usługi dla danego regionu	Wyświetla usługi dla danego regionu wyszukiwania	Zwraca widok HTML lity usług dla danego regionu
		Pogrupuj usługi wg wysokości ceny lub zakresu cenowego	Wyświetla usługi dla danego regionu wyszukiwania dla zadanych kryteriów	Zwraca widok HTML lity usług dla danego regionu z zadanymi warunkami

	Użytkownik	Zamów usługę sporządzając notatkę dla usługodawcy	Klient klikając na listę usług ma możliwość zamówienia danej usługi z informacją o szczegółach produktów, których potrzebuje	Wyświetla kolejne widoki HTML, które prowadzą użytkownika podczas proces zamawiania, na koniec, wprowadza dane do bazy danych.
		Oceń usługodawcę	Klient ma możliwość oceny usługodawcy, gdy ten zrealizuje swoją usługę poprzez zakładkę usług do oceny	Wyświetlenie widoku HTML służącego do oceniania oraz wprowadzanie zmian w bazie danych
		Wyświetl moje aktualne usługi	Klient wyświetla informacje o aktualnie realizowanych dla niego usługach (potwierdzonych lub nie przez usługodawcę)	Wyświetlenie widoku HTML dla aktualnych usług
		Wyświetl usługi do oceny	Klient wyświetla informacje o usługach zakończonych przez usługodawcę, które są gotowe do oceny	Wyświetlenie widoku HTML usług przeznaczonych do oceny
		Wyświetl moje poprzednie usługi	Klient wyświetla historię swoich zamówień, czyli takich, które zostały przez niego już ocenione	Wyświetlenie widoku HTML usług już zrealizowanych i ocenionych

Usługodawca	Użytkownik	Dodaj usługę	Dodaje usługę widoczną dla klientów	Dodaje usługę do systemu, zwraca informację o sukcesie albo porażce
		Usuń usługę	Usuwa usługę	Usuwa usługę z systemu, zwraca informację o sukcesie albo porażce
		Wyświetl usługi	Wyświetla usługi danego usługodawcy	Zwraca listę usług wraz z informacjami o nich
		Edytuj usługę	Edytuje informację o usłudze	Aktualizuje informacje o podanej usłudze, zwraca informację o sukcesie albo porażce
		Potwierdź zamówienie	Potwierdza, że wykona konkretną usługę zamówioną przez klienta	Aktualizuje zamówienie, zwraca informację o sukcesie albo porażce potwierdzenia
		Potwierdź realizację usługi	Potwierdza, że zrealizował usługę	Aktualizuje zamówienie, zwraca informację o sukcesie albo porażce potwierdzenia

tab. 1

## User Stories

### 1. Publiczna witryna WWW

- 1.1. Jako anonimowy lub (zalogowany użytkownik) przeszukuję katalog usługodawców w odpowiednim rejonie.

Dowolny użytkownik wchodzi na stronę WWW, wybiera interesujący go rejon zakupów. Wyświetla się katalog usługodawców, którzy mogą zrealizować zakupy w danym regionie. Po wybraniu usługodawcy przechodzi do informacji na jego temat, takich jak imię, nazwisko, dane kontaktowe, ocena oraz dodatkowe informacje o tym jak duże zakupy może zrobić i w jakim czasie.

- 1.2. Jako zalogowany użytkownik wybiera usługodawcę.

Zalogowany użytkownik po wybraniu usługodawcy przechodzi do panelu, w którym sporządza notatkę dla usługodawcy, jakie zakupy i w jakich sklepach chce zrobić. Następnie akceptuje podane informacje i czeka na potwierdzenie telefoniczne od usługodawcy.

- 1.3. Jako zalogowany użytkownik przegląda aktywne usługi.

Zalogowany użytkownik wchodzi do panelu, w którym wyświetlone są jego aktywne usługi. Przy usłudze wyświetlany jest usługodawca.

- 1.4. Jako zalogowany użytkownik przegląda swoje poprzednie usługi.

Zalogowany użytkownik wchodzi do panelu, w którym wyświetlone są jego poprzednio wybrane usługi. Przy usłudze wyświetlany jest usługodawca, jego ocena i notatka widziana publicznie.

- 1.5. Jako zalogowany użytkownik ocenia usługodawcę.

## **2. Interfejs usługodawcy**

Usługodawcą jest użytkownik, który zostanie dodany przez administratora jako usługodawca. Taki użytkownik posiada dostęp (np. bezpośrednio z terminala lub ze stworzonej przez siebie aplikacji) do serwisu poprzez przeznaczone do tych celów Web API.

- 2.1. Usługodawca pobiera dane z serwera odnośnie jego usług

Pobrane dane usługodawca może edytować w dowolnym momencie. Może również dodawać oraz usuwać wybrane przez siebie usługi.



2.2. Usługodawca sprawdza odpowiednim zapytaniem, czy ktoś wybrał jego usługi

Usługodawca może sprawdzić czy ktoś chce skorzystać z jego usług i czeka na potwierdzenie. W takim wypadku podejmuje się zadania albo z niego rezygnuje poprzez Web API.

2.2. Usługodawca może ocenić klientów.

Usługodawca może wyświetlić nieocenione ale zrealizowane usługi. W takim przypadku udostępniona jest dla niego możliwość wystawienia oceny oraz napisanie notki odnośnie klienta.

### 3. Interfejs administratora

3.1. Administrator ma dostęp do listy wszystkich użytkowników i nimi zarządza.

Administratorowi w aplikacji desktopowej wyświetla się lista użytkowników w bazie danych. Wybiera użytkownika lub usługodawcę i może go usunąć lub edytować jego dane (m.in. potwierdzić konto). Dla administratora dostępna jest również opcja wyszukiwania konkretnego użytkownika. Razem z usługodawcą usunięte zostają jego wszystkie oferowane usługi z publicznego katalogu, oprócz aktualnych, których się podjął. Usługodawca nie jest już widziany w publicznym katalogu.

Lista ról użytkowników występujących w historiach (*tab. 1*):

Anonimowy klient	Użytkownik niezalogowany na stronie
Zalogowany klient	Użytkownik zalogowany na stronie
Usługodawca	Użytkownik oferujący swoje usługi
Administrator	Użytkownik odpowiedzialny za zarządzanie systemem.

*tab. 2*

### 1.3. Wymagania niefunkcjonalne

W *tab. 2* przedstawione zostały wymagania niefunkcjonalne.

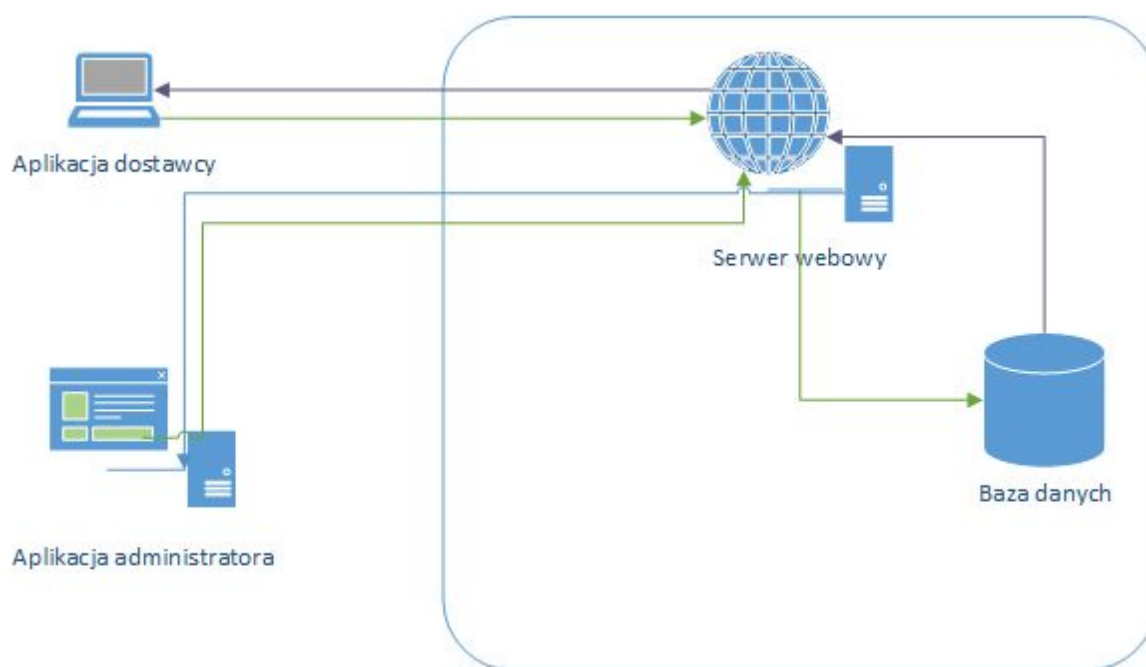
Obszar wymagań	Nr wymagania	Opis
Bezpieczeństwo	1	Z Web API mogą korzystać tylko administrator oraz dostawcy zarejestrowani w serwisie.

Niezawodność	2	Aplikacja typu HA (High Availability) - dostępna 24x7, przerwami serwisowymi jedynie od 3:00 - 5:00, nie dłuższymi niż 2h, nie częściej niż raz w tygodniu.
Wydajność	3	Aplikacja ma odpowiadać na żądania nie dłużej niż 3 s. w przypadku mniej niż 100 równoczesnych połączeń.
Utrzymanie	4	Kopia zapasowa - przechowywana na serwerze, wykonywana każdego dnia o godzinie 4:00.

tab. 3

## 1.4. Architektura rozwiązania

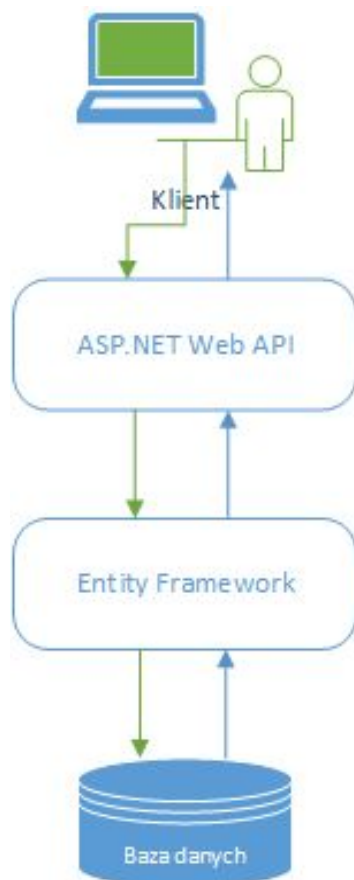
System, który rozważamy przedstawiony jest na rys. 2.



rys.2

## 1.5. Architektura serwisu Web API

Wybrane technologie dla aplikacji administratora i ich wykorzystanie zostały ukazane na kolejnej ryc. (rys. 3)



rys. 3

## 2. Dokumentacja końcowa (powykonawcza)

### 2.1. Wymagania systemowe

Dostarczana aplikacja klienta administratora wymaga:

- ☐ Systemu Windows 7 (lub wyższy)

Dostarczana aplikacja serwera wymaga:

- ☐ Systemu Windows Server 2012R
- ☐ Bazy Sql Server 2012
- ☐ Microsoft IIS 7

### 2.2. Wykorzystane komponenty

Projekt Visual Studio dostarcza wymagane biblioteki za pośrednictwem pakietu NuGet. Wymagane biblioteki do pracy aplikacji dołączane są podczas procesu kompilacji do katalogu pliku wykonywalnego aplikacji.

Aplikacja serwerowa wykorzystuje technologię ASP.NET; potrzebne biblioteki są instalowane podczas zakładania nowego projektu aplikacji w Visual Studio, dlatego też zostały tutaj pominięte.

Wykorzystane pakiety zostały wylistowane w tabeli nr 4.

nr	Komponent	Opis	Licencja
1	Entity Framework 6.1.3	Biblioteka ORM wspomagająca pracę programistów przez stworzenie warstwy dostępu do bazy danych	Apache 2.0
2	MahApps.Metro 1.3.0	Pozwala na wygodniejsze tworzenie bardziej eleganckiego GUI.	MIT
3	Moq 4.5.28	Wykorzystywany podczas testów jednostkowych do zaślepiania obiektów.	BSD

4	Ninject 3.2.2	Pozwala na wygodne wstrzykiwanie zależności (DC)	Apache 2.0
5	NUnit 3.5.0	Wygodne testy jednostkowe.	MIT
6	NUnit3TestAdapter 3.6.0	Przydatny dodatek do Visual Studio, integruje testy napisane w NUnit 3.	NUnit license
7	NUnitTestAdapter 2.0.0	Przydatny dodatek do Visual Studio, integruje testy napisane w NUnit.	NUnit license
8	Microsoft.AspNet.WebApi.Client 5.2.3	Umożliwia komunikację z serwerem poprzez protokół HTTP. Wsparcie dla JSONa oraz XMLa.	Microsoft license
9	Microsoft.AspNet.Mvc 5	Umożliwia stworzenie aplikacji WWW.	Microsoft License

## 2.3. Instrukcja instalacji

Aplikacja klienta administratora dostarczana jest jako plik wykonywalny, w tym przypadku nie ma procesu instalacji.

Przykładowa aplikacja usługodawcy korzystająca z Web API nie wymaga procesu instalacji, przygotowany został plik .exe.

W przypadku serwera:

1. Dostarczone pliki aplikacji serwera powinny zostać dodane jako aplikacja/witryna WWW w panelu IIS. Szczegółowe informacje o procesie instalacji można znaleźć pod adresem: <https://docs.microsoft.com/en-us/aspnet/core/publishing/iis>.
2. Następnie należy wygenerować bazę danych w panelu *SQL Server Management Studio* korzystając z dostarczonego pliku dump.sql.

## 2.4. Instrukcja uruchomienia

Aplikacja klienta administratora dostarczana jest w postaci katalogu z zawartym plikiem wykonywalnym. Wystarczy uruchomić **client-desktop.exe**. Do jej prawidłowego działania niezbędny jest uprzednio skonfigurowany serwer, którego instrukcja instalacji została przedstawiona w poprzednim paragrafie.

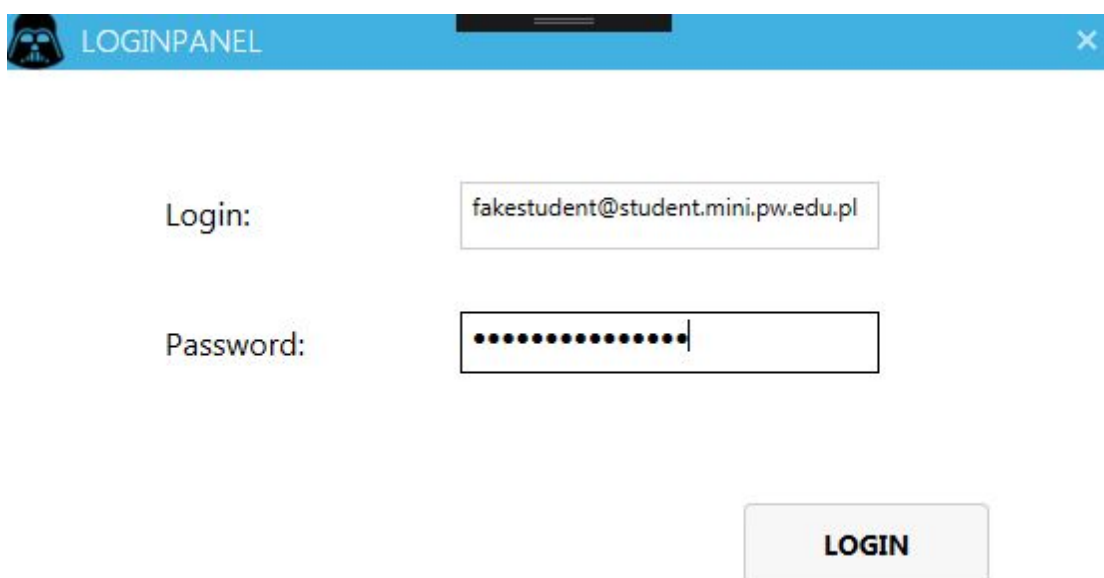
Serwer po instalacji uruchamiany jest automatycznie. Na nim znajdują się witryna WWW.

**W projekcie został użyty udostępniony serwer wirtualny dostępny z poziomu sieci MiNI lub poprzez SSH.**

## 2.5. Instrukcja użycia

### Aplikacja administratora (Zintegrowany z Web API)

Użytkownik będący administratorem loguje się do aplikacji używając swojego adresu e-mail oraz hasła. Jeśli dane są prawidłowe ma dostęp do panelu administratora, który pozwala na przeprowadzenie operacji CRUD na użytkownikach systemu poprzez Web API. Dodawanie, edytowanie i usuwanie jest bardzo intuicyjne, a przebieg ich jak i logowania pokazują następujące rysunki:



The image shows a web interface for a login panel. At the top, there is a blue header bar with a black mask icon on the left, the text "LOGINPANEL" in white, and a close button (an 'x' in a circle) on the right. Below the header, the login form is displayed. It consists of two rows: the first row has the label "Login:" followed by a text input field containing the email address "fakestudent@student.mini.pw.edu.pl"; the second row has the label "Password:" followed by a password input field filled with black dots. Below these fields, there is a light gray button with the text "LOGIN" in bold black capital letters.

rys. 4

ADMINPANEL				
TYPE	USERNAME	EMAIL	PHONENUMBER	ACTIVATED
Admin	fakestudent@student.mini.pv	fakestudent@student.mini.pv	123123123	
Provider	testprovider@mail.pl	testprovider@mail.pl	127127127	
Customer	testcustomer@mail.pl	testcustomer@mail.pl	123123123	

rys. 5

ADMINPANEL				
TYPE	USERNAME	EMAIL	PHONENUMBER	ACTIVATED
Admin	fakestudent@student.mini.pv	fakestudent@student.mini.pv	123123123	
Provider	testprovider@mail.pl	testprovider@mail.pl	127127127	
Customer	testcustomer@mail.pl	testcustomer@mail.pl	123123123	

Edit User

UserName:

Email:

PhoneNumber:

Save

rys. 6

The screenshot shows a web application interface titled 'ADMINPANEL'. At the top, there are buttons for 'ADD', 'EDIT', 'DELETE', and 'REFRESH'. Below these is a table with the following columns: 'TYPE', 'USERNAME', 'EMAIL', 'PHONENUMBER', and 'ACTIVATED'. The table contains three rows of data:

TYPE	USERNAME	EMAIL	PHONENUMBER	ACTIVATED
Admin	fakestudent@student.mini.pv	fakestudent@student.mini.pv	123123123	
Provider	testprovider@mail.pl	testprovider@mail.pl	127127127	
Customer				

Overlaid on the table is a modal dialog box titled 'ADDUSERWINDOW'. It contains four input fields with labels and 'X' icons for clearing the text:

- UserName:** Contains the text 'Aleksy'.
- Password:** Contains the text '12321mAdw#'.
- Email:** Contains the text 'aleksy@mistal.net'.
- PhoneNumber:** Contains the text '781743005'.

At the bottom of the dialog box, there are four buttons: 'Add Customer', 'Add Provider', 'Add Admin', and 'Cancel'.

rys. 7

### Aplikacja usługodawcy (Web API)

Usługodawca może skorzystać z udostępnionego API i korzystając ze swojego maila i hasła może pobierać informacje i zarządzać swoimi usługami. Przykład prostej aplikacji wykorzystującej API jest zaprezentowany przez aplikację **api-client-provider.exe**.

To jak korzystać z API jest opisane w postaci klas **ApiClient.cs** i **ServiceApi.cs** w kodzie źródłowym projektu **api-client-provider**.

### Witryna WWW

Dla klienta została stworzona strona internetowa, dzięki której może on sprawnie przeglądać oraz zamawiać interesujące go usługi (w drugim przypadku uprzednio logując się na swoje konto użytkownika).

Strona ta działa na serwerze, także wystarczy aby klient przy użyciu przeglądarki internetowej dostał się na ustalony adres URL.

Na kolejnych rysunkach zostaną przedstawione zrzuty ekranu, które przybliżą działanie serwisu. Obsługa jest bardzo intuicyjna, dlatego opis do nich został w tym przypadku pominięty.



## WYGODNE ZAKUPY? WYSTARCZY, ŻE WYBIERZESZ REGION I USŁUGODAWCĘ

NIE MUSISZ WYCHODZIĆ Z DOMU.  
WYBIERZ USŁUGODAWCĘ, KTÓRY PRZYNIESIE ZAKUPY DO CIEBIE.

DLACZEGO WARTO SKORZYSTAĆ

rys. 8

Region	Cena	Sortuj
<input type="text" value="Region np. Warszawa"/>	<input type="text" value="Od"/> <input type="text" value="Do"/>	<input type="text" value="Najtańsze"/> <input type="button" value="Szukaj"/>

### Usługi w Twojej okolicy

Title of the service 1220153972	30,00 zł
Test Short Description of service	Warsaw
★ 4/ 5 (1 ocen)	
Title of the service 109735412	30,00 zł
Test Short Description of service	Warsaw
★ 4/ 5 (1 ocen)	
Title of the service 447377991	30,00 zł
Test Short Description of service	Warsaw
★ 4/ 5 (1 ocen)	
Title of the service 2130765435	30,00 zł
Test Short Description of service	Warsaw
★ 4/ 5 (1 ocen)	

rys. 9

[← Powrót do wyników wyszukiwania](#)

Title of the service 1220153972

Short Description of service

Usługodawca: **Test Provider**

Średnia ocen: 4 / 5 ★

Cena za usługę: 30 zł

Lokalizacja: **Warsaw**Zamówienie: [Zamów zakupy](#)

Aby zamówić, musisz się zalogować

Copyright: 2017. Using theme by [ThemeFisher](#) Developed by Aleksy Miśtał and Krzysztof Franczak

rys. 10

## Panel rejestracji.

Stwórz nowe konto.

Imię	<input type="text"/>
Nazwisko	<input type="text"/>
Numer telefonu	<input type="text"/>
Email	<input type="text"/>
Hasło	<input type="password"/>
Potwierdź hasło	<input type="password"/>
Typ użytkownika	<div>Klient</div>
<input type="button" value="Zarejestruj się"/>	

Copyright: 2017. Using theme by [ThemeFisher](#) Developed by Aleksy Miśtał and Krzysztof Franczak

rys. 11

## Panel logowania.

Zaloguj się by swobodnie korzystać z serwisu.

Email   
Poprawny adres email jest wymagany.

Hasło   
Hasło jest wymagane.

☐ Zapamiętaj mnie

[Zarejestruj się](#)Copyright: 2017. Using theme by [Themefisher](#) Developed by [Aleksy Mistał](#) and [Krzysztof Franczk](#).

rys. 12

## Moje Usługi

[Aktywne usługi](#) [Usługi do oceny](#) [Poprzednie usługi](#)

Title of the service 1220153972

Test Provider ★ 4/5 (1 ocen)

30,00 zł

Oczekiwanie na potwierdzenie realizacji

Title of the service 1220153972

Test Provider ★ 4/5 (1 ocen)

30,00 zł

Oczekiwanie na akceptację

Title of the service 1220153972

Test Provider ★ 4/5 (1 ocen)

30,00 zł

Oczekiwanie na akceptację

Title of the service 1220153972

Test Provider ★ 4/5 (1 ocen)

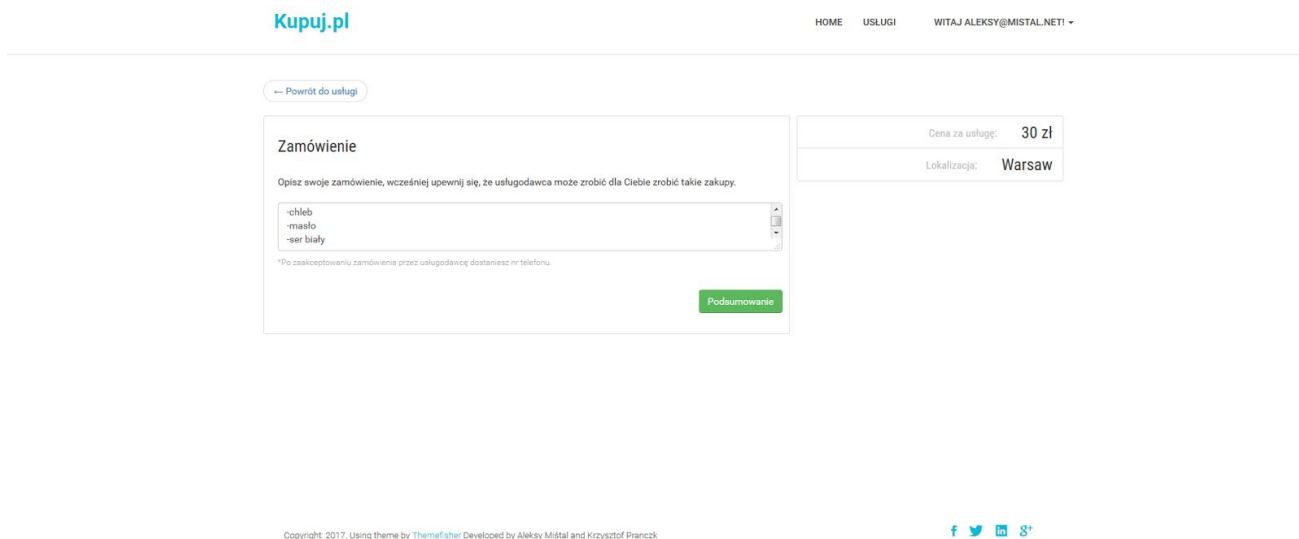
30,00 zł

Oczekiwanie na akceptację

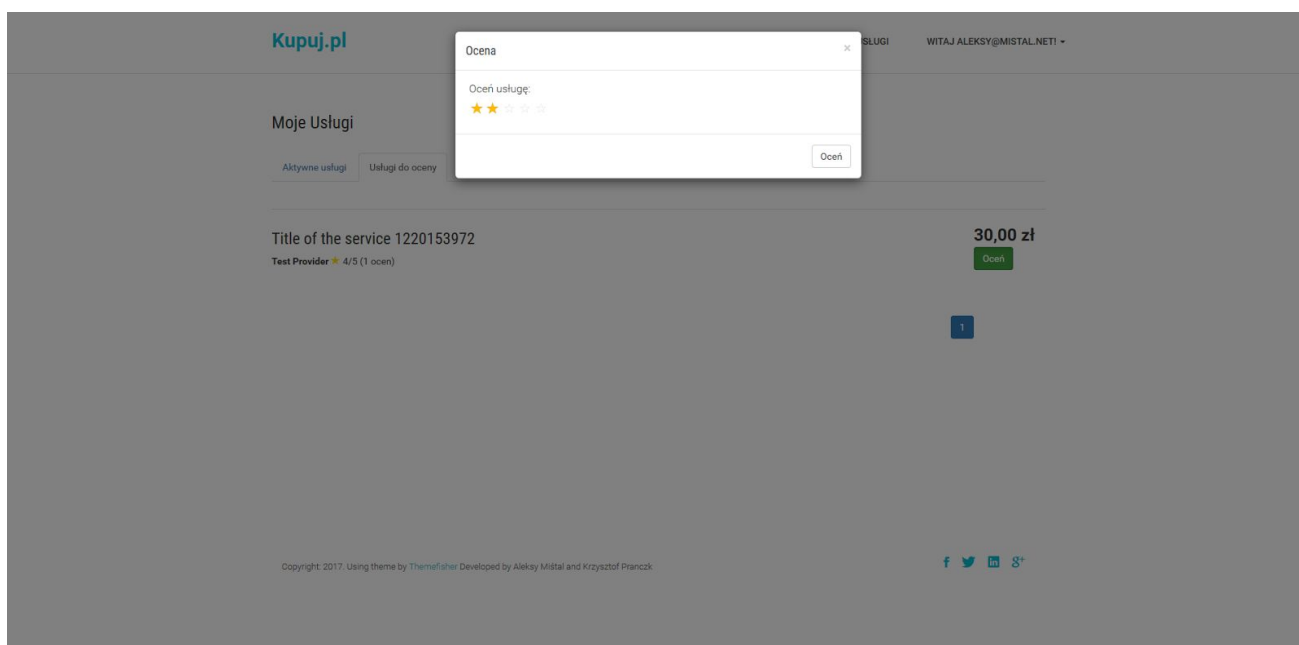
1 2 »

Copyright: 2017. Using theme by [Themefisher](#) Developed by [Aleksy Mistał](#) and [Krzysztof Franczk](#).

rys. 13



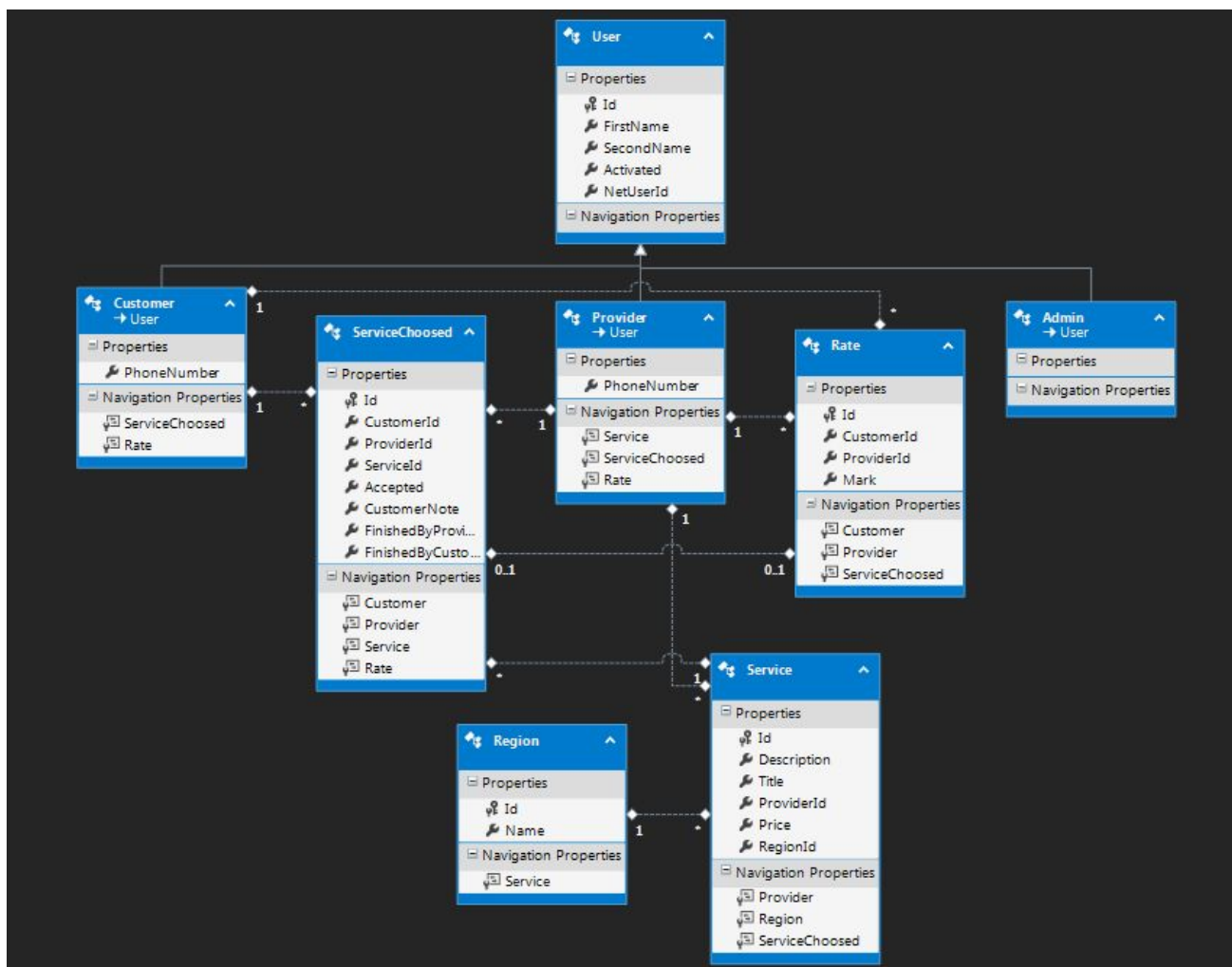
rys. 14



rys. 15

## 2.6. Model danych

Baza danych jest przechowywana na serwerze z wykorzystaniem SQL Server. Skrypt SQL tworzący struktury bazy danych znajduje się w pliku **DatabaseModel.edmx**. Baza danych różni się od bazy danych z wcześniejszych wersji projektu, ponieważ teraz jest zintegrowana z bazą danych, ASP.NET, która to przechowuje zarejestrowanych użytkowników. Widoczne to jest na modelu z rys. 8 - użytkownik posiada jeszcze jedno pole **NetUserId**, które to odpowiada właśnie za wspomnianą już integrację.



rys. 16

## 2.7. Raport z przeprowadzonych testów

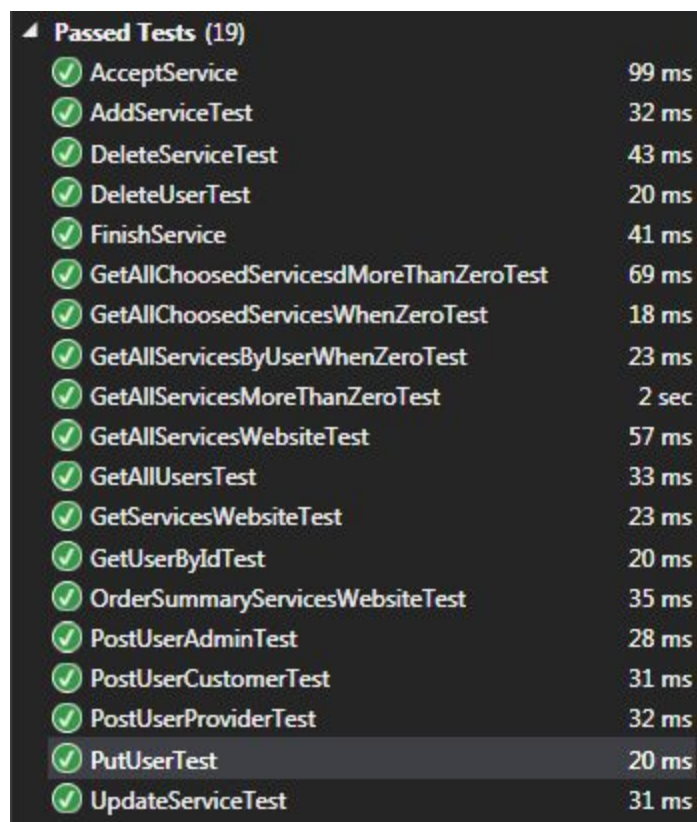
Aplikacja administratora jak i serwerowa została przetestowana przy użyciu testów jednostkowych.

W aplikacji administratora zostały przetestowane widoki. Wynik przeprowadzonych testów jest widoczny na rys. 9.

Run All   Run...   Playlist : All Tests	
▲ Passed Tests (4)	
✓ AddUserTest	194 ms
✓ DeleteUserTest	8 ms
✓ UpdateUserDataTest	5 ms
✓ UsersReadTest	6 ms

rys. 17

Aplikacja serwerowa również została przetestowana. W tym przypadku sprawdzane były metody odpowiadające za prawidłowe funkcjonowanie Web API oraz akcje odpowiednich kontrolerów odpowiadających za zwracanie widoków na stronę WWW (rys.10):



Passed Tests (19)	
✓ AcceptService	99 ms
✓ AddServiceTest	32 ms
✓ DeleteServiceTest	43 ms
✓ DeleteUserTest	20 ms
✓ FinishService	41 ms
✓ GetAllChoosedServicesdMoreThanZeroTest	69 ms
✓ GetAllChoosedServicesWhenZeroTest	18 ms
✓ GetAllServicesByUserWhenZeroTest	23 ms
✓ GetAllServicesMoreThanZeroTest	2 sec
✓ GetAllServicesWebsiteTest	57 ms
✓ GetAllUsersTest	33 ms
✓ GetServicesWebsiteTest	23 ms
✓ GetUserByIdTest	20 ms
✓ OrderSummaryServicesWebsiteTest	35 ms
✓ PostUserAdminTest	28 ms
✓ PostUserCustomerTest	31 ms
✓ PostUserProviderTest	32 ms
✓ PutUserTest	20 ms
✓ UpdateServiceTest	31 ms

rys. 18

## 2.8. Raport odstępstw od specyfikacji wymagań

### 1. Zrezygnowano z oceniania klientów przez usługodawców

Wprowadziłoby to duży stopień skomplikowania jak na potrzeby tego projektu.