

# Aplikacja Mobilna „Zawodowcy.pl” – projekt zespołowy 2021

---

# Spis treści

---

Opis i motywacje dla powstania projektu .....	3
Członkowie zespołu i zakres obowiązków .....	3
Lista planowanych funkcjonalności: .....	4
Opis technologii:.....	5
Spis Milestone'ów .....	6
Schemat Bazy Danych.....	7
Najczęściej używane zapytania .....	8
Diagramy przypadków użycia .....	9
Diagramy sekwencji.....	12
Diagram klas .....	16
Diagram stanu .....	17
Diagram komponentów.....	18
Design graficzny.....	19

# Opis i motywacje dla powstania projektu

---

Tematyką naszego projektu zespołowego będzie aplikacja mobilna realizująca koncepcję platformy pozwalającej na zamawianie fachowców z szerokiej gamy profesji, za pomocą kilku kliknięć. Celujemy w prostą i przejrzystą oprawę. Aplikacja będzie posiadać podstawowe funkcjonalności tj. system logowania, systemy ocen i opinii, odpowiedni system zapisu na dostępny termin itd.

Realizując naszą aplikację, chcemy odpowiedzieć na zapotrzebowanie na rynku na właśnie takie usługi. Spektakularny sukces takich platform jak Pyszne.pl czy też Glovo, pokazuje że ludzie chętnie korzystają z tego typu rozwiązań. Wiedząc, że takie systemy zarządzania działają i cieszą się sporą popularnością, chcielibyśmy przenieść te rozwiązania na nowy grunt. Mamy świadomość istnienia konkurencyjnych portali tj. fixly.pl czy usterka.pl, ale mamy głębokie przekonanie, że nasze rozwiązanie będzie zarówno satysfakcjonujące dla ludzi młodych, jak i dla mniej obeznanych z nowoczesnymi technologiami, ludzi w podeszłym wieku. Nasza aplikacja, dzięki swojej przejrzystości i intuicyjności, będzie kolejną cegiełką dołożoną do walki przeciw wykluczeniu cyfrowemu.

Projekt zostanie zrealizowany w ciągu 9 miesięcy, gdzie pierwsze 4 miesiące zostaną przeznaczone na odpowiednie rozplanowanie działań. Następnie zespół przejdzie do implementacji, co pozwoli na wydanie aplikacji w czerwcu 2022 roku.

## Członkowie zespołu i zakres obowiązków

---

Kamil Cessak – inżynier Frontend

Szymon Jędrych – inżynier Backend

Paweł Lysko – tester

Kacper Siembida – lider zespołu

# Lista planowanych funkcjonalności:

---

- Rejestracja
  - Standardowe założenie konta.
    - Hasło powinno mieć przynajmniej 8 znaków, jedną dużą i małą literę, znak specjalny i cyfrę.
    - Opcjonalne podanie adresu.
    - Opcjonalna możliwość ustawienia konta jako „Zawodowca”. Wymagane będzie wtedy podanie niezbędnych informacji między innymi: Dane kontaktowe, Nazwa firmy, krótki opis działalności.
- Logowanie
- Resetowanie hasła.
- Wystawianie ofert jako „Zawodowiec”.
  - Podanie Tytułu i opisu oferty.
  - Ustawienie obszaru świadczenia usługi( Miasto, dystans od danego miasta)
  - Wybranie kategorii świadczonej usługi
  - Podanie oczekiwanej ceny za usługę(Możliwe ustawienie widełek)
  - Opcjonalne dodanie zdjęć
  - Opcjonalne ustawienie dostępnych terminów
- Przeglądanie wystawionych ofert.
  - Możliwość filtrowania po mieście i po rodzaju usługi.
  - Możliwość sortowania według ceny rosnąco i malejąco.
- Strona oferty będzie posiadać wszystko podane przez „Zawodowca” podczas wystawiania jej.
- Opinie
  - Możliwość ocenienie(1-5 gwiazdek) i napisania opinii o „Zawodowcu”/usłudze.
  - Możliwość przeglądania opinii.
- Opcjonalny Terminarz fachowca
  - Będzie posiadał zapisane terminy zamówionych usług i wolne terminy do zapisania się.
- Możliwe zamówienie usługi z poziomu aplikacji.
  - Jeżeli „Zawodowiec” posiada terminarz, użytkownik wybiera termin i czeka na akceptację „Zawodowca”
  - Jeżeli nie ma terminarza, użytkownik request o podanie dostępnych terminów przez „Zawodowca”. Po otrzymaniu ich wybiera pasujący.
- Użytkownik posiada listę zamówionych i zakończonych usług
- „Zawodowiec” posiada listę zamówionych i zrealizowanych przez siebie usług.

Opcjonalne funkcjonalności:

- Czat.
- Ustawianie obszaru za pomocą zaznaczania na mapach google.

# Opis technologii:

---

## Frontend:

**React Native** – otwarto-źródłowy zestaw narzędzi dla programistów przeznaczony do tworzenia natywnych, wieloplatformowych aplikacji mobilnych, komputerowych, internetowych stworzony przez firmę Facebook, Inc.

Wybraliśmy go z uwagi na możliwość łatwego stworzenia aplikacji mobilnej zarówno na platformę iOS oraz Android. Technologia React wykazuje się również elastycznością, gdyż łatwo łączy się z wszelkimi technologiami backendowymi, w tym z .NET, który zostanie wykorzystany w naszej aplikacji.

Do warstwy wizualnej użyjemy frameworka **TailwindCSS**, który pozwala na proste kreowanie stylów CSS, bez konieczności pisania ich ręcznie, tylko poprzez wykorzystanie predefiniowanych przez Tailwind klas komponentów.

## Backend:

**ASP.NET Core (.NET 6)** – framework sieciowy od Microsoftu, najnowsza wersja która została zaprezentowana na .NET Conf (dorocznej konferencji Microsoftu). Wybraliśmy tę technologię ze względu na jej popularność, łatwość użycia i dobre, długoterminowe wsparcie.

**SignalR** – framework pozwalający na komunikację real-time między serwerem, a klientem. Aplikacji będzie używać wielu użytkowników w jednym momencie, może się zdarzyć, że kilku z nich będzie chciało w tym samym momencie zamówić usługę na ten sam termin – musimy w jakiś sposób powiadomić resztę z nich, że termin jest już zajęty i aktualizować na bieżąco grafik usługodawcy. W tym celu z pomocą przychodzi nam przykładowo komunikacja real-time po websocketach.

## Baza danych:

**PostgreSQL** – jedna z najpopularniejszych open source relacyjnych baz danych. Przez ponad 25 lat na rynku zdążyła zgromadzić spore grono zwolenników. Zostanie użyta ze względu na to, że nie ustępuje zbyt wiele najlepszym płatnym silnikom bazodanowym i jest całkowicie darmowa.

## Komunikacja z bazą danych:

**Dapper** lub **Entity Framework Core** – w zależności od potrzeb. Pierwszy z nich, Dapper, to Micro ORM który pozwala na elastyczną komunikację z bazą danych przez pisanie własnych procedur. Drugi za to zezwala na komunikację z poziomu kodu, bez potrzeby pisania query.

## Testy:

**XUnit** – domyślny framework do testowania używany w .NET.

**Moq** – najpopularniejsza biblioteka do mockowania w .NET.

Podsumowując, taki dobór narzędzi w oczywisty sposób ułatwi nam pracę, a przynajmniej w znaczący sposób odsunie widmo zastoju i nieoczekiwanych problemów; w miarę możliwości staramy się temu przeciwdziałać już na etapie planowania. Wybraliśmy technologie, które wykazują się elastycznością i łatwością użycia. Oczywiście, wpływ na selekcję miała również ich popularność; dzięki temu 'troubleshooting' będzie mógł być przeprowadzany w mniej chaotyczny sposób. Dodatkowo warto zauważyć, że są to technologie przyszłościowe, cały czas rozwijane i to z nimi najprawdopodobniej chcielibyśmy związać naszą „zawodową” przyszłość.

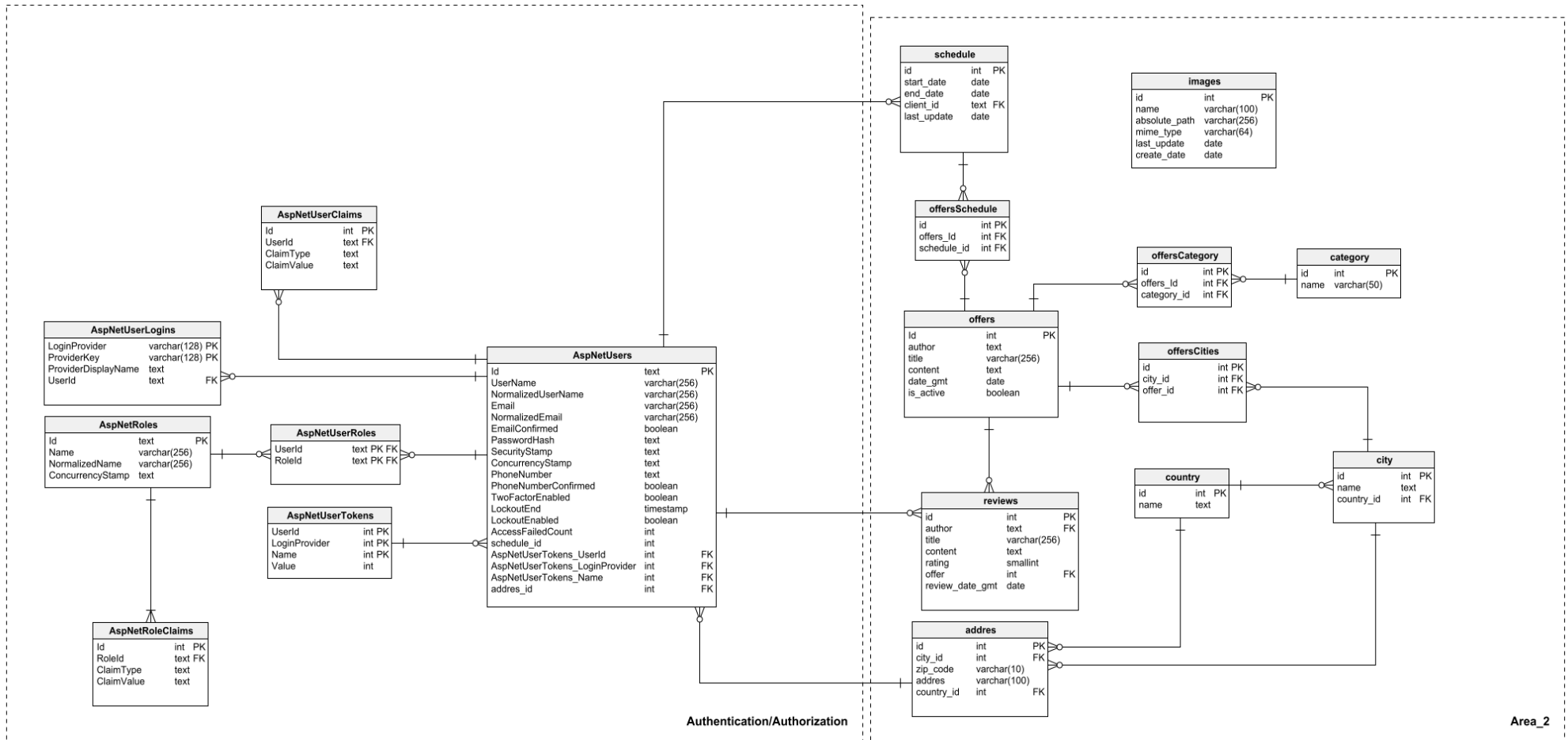
## Spis Milestone'ów

---

- M1 - Utworzeniu wstępnej dokumentacji projektu
- M1 - Zatwierdzenie technologii i ukończenie nietechnicznej dokumentacji projektu
- M2 - Zaprojektowanie bazy danych
- M2 - Utworzenie diagramów
- M2 - Design wyglądu aplikacji i projekt testów
- M3 - Zebranie celów do osiągnięcia (Milestones, epiki, stories)
- M3 - Uporządkowanie i rozpisanie celów na następny semestr

# Schemat Bazy Danych

This is an example database model for PROJECT ZESPOŁOWY 2K21/2K22



# Najczęściej używane zapytania

---

Biorąc pod uwagę przeznaczenie i koncept naszej aplikacji, jak i analizując schemat bazy danych można określić jakie zapytania będą najczęściej realizowane. Z racji faktu, że aplikacja będzie pozwalała na zatrudnianie fachowców z konkretnych profesji, najczęściej zapytania będą dotyczyły kategorii – specjalizacji świadczonych usług i wyszukiwania konkretnych zawodowców.

## ❖ Wyszukiwanie ofert z kategorii 'Remonty'

- `SELECT offers.author AS Autor, offers.title AS Tytuł, category.name AS Kategoria FROM offersCategory, offers, category WHERE offers.id = offersCategory.offers_id AND offersCategory.category_id = category.id AND category.name = 'Remonty'`

## ❖ Wyszukiwanie ofert świadczonych przez użytkownika 'Jan Kowalski'

- `SELECT offers.author AS Autor, offers.title AS Tytuł, category.name AS Kategoria FROM offersCategory, offers, category WHERE offers.id = offersCategory.offers_id AND offersCategory.category_id = category.id AND offers.author = 'Jan Kowalski'`

Nasza aplikacja ma zrzeszać zawodowców z całego kraju, więc często używane będzie wyszukiwanie ofert w danym mieście:

## ❖ Wyszukiwanie ofert z miasta 'Lublin'

- `SELECT offers.author AS Autor, offers.title AS Tytuł, city.name AS Miasto FROM offers, offersCities, city WHERE offers.id = offersCities.offer_id AND offersCities.city_id = city.id AND city.name = 'Lublin'`

Oczywistym jest, że każdemu zależy na jakości wykonanej usługi, więc można założyć, że wyszukiwanie usług po ich ocenie będzie równie częste.

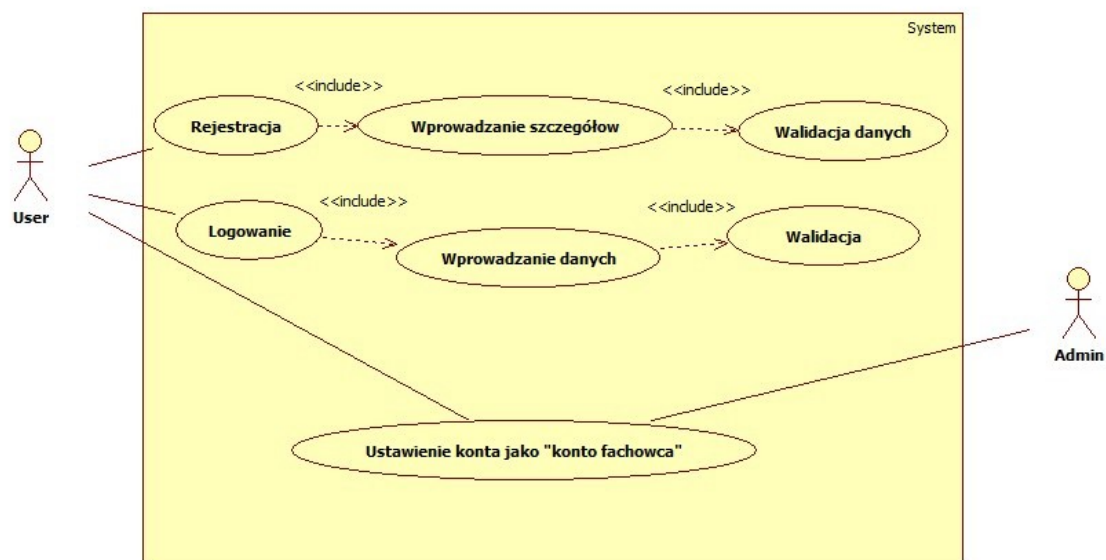
## ❖ Wyszukiwanie ofert o ratingu równym 6

- `SELECT offers.author AS Autor, offers.title AS Tytuł, reviews.name AS Ocena FROM offers, reviews WHERE offers.id = reviews.offer AND reviews.rating = 6`



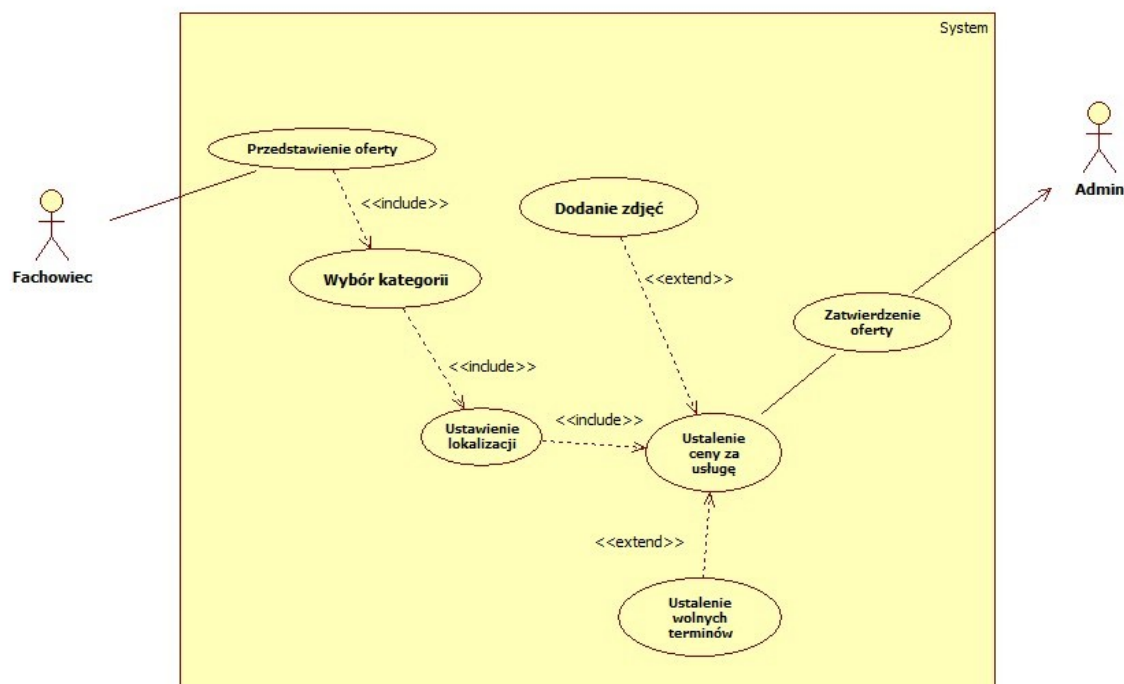
# Diagramy przypadków użycia

## 1. Rejestracja i logowanie



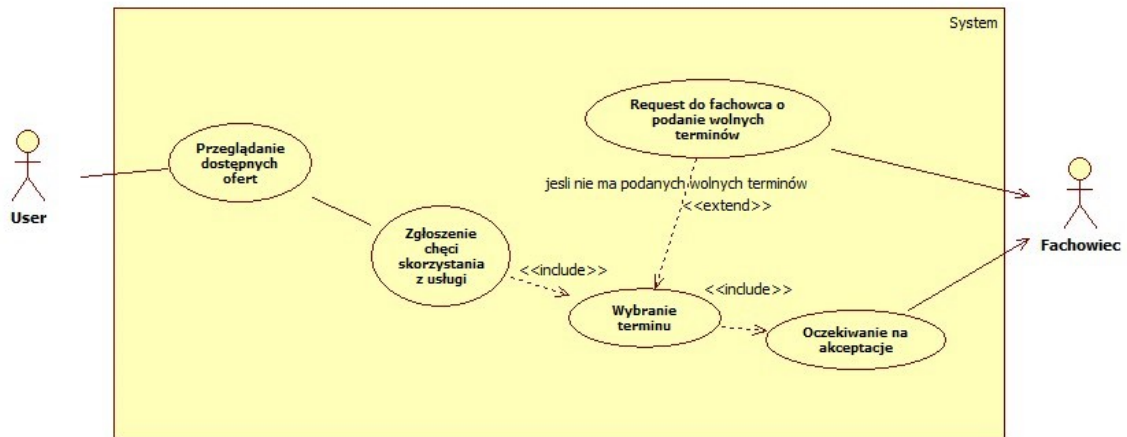
Nazwa	Rejestracja i logowanie
Scenariusz	Użytkownik rejestruje się/ loguje pomyślnie
Aktorzy	User, Admin
Warunki początkowe	Użytkownik musi mieć zainstalowaną aplikację
Niezmienniki	Użytkownik chce się zarejestrować/zalogować
Opis	<ol style="list-style-type: none"> <li>1.Użytkownik uruchamia aplikację</li> <li>2.Wybiera czy chce się zarejestrować czy zalogować</li> <li>3.Podaje dane</li> <li>4.System sprawdza poprawność danych</li> <li>5.Po pomyślnej weryfikacji system zwraca stosowną wiadomość</li> <li>6.Użytkownik ma możliwość ustawienia konta jako „konto fachowca”, jeśli się na to decyduje to musi oczekiwać na zatwierdzenie przez Admina</li> <li>7. Po zatwierdzeniu przez Admina, Użytkownik ma konto typu „konto fachowca”</li> </ol>
Warunki końcowe	Użytkownik jest zarejestrowany/zalogowany i jeśli wyraża taką chęć ma „konto fachowca”
Źródło możliwych błędów	3. W przypadku podania błędnych danych, użytkownik zostanie poproszony o ich ponowne wprowadzenie.

## 2. Wystawienie oferty przez fachowca



Nazwa	Wystawienie oferty przez fachowca
Scenariusz	Fachowiec wystawia ofertę
Aktorzy	Fachowiec, Admin
Warunki początkowe	Fachowiec musi mieć zainstalowaną aplikację, być zalogowany i posiadać „konto fachowca”
Niezmienne	Fachowiec chce wystawić ofertę
Opis	1. Fachowiec klika w zakładkę pozwalającą składać oferty. 2. Zostaje poproszony o podanie tytułu i wybranie kategorii. 3. W następnym kroku musi określić swoją lokalizację i zasięg w jakim jest w stanie realizować swoje usługi. 4. Następnie Fachowiec jest proszony o podanie ceny (możliwość ustalenia widełek) swojej usługi. 5. Fachowiec ma również możliwość dodania tematycznie pasujących zdjęć swoich poprzednich prac, jak również ustalenia wolnych terminów. 6. Admin zatwierdza ofertę
Warunki końcowe	Oferta fachowca jest dostępna dla przeglądających
Źródło możliwych błędów	W przypadku podania błędnych danych w punktach 2 – 5, fachowiec zostanie poproszony o ich ponowne wprowadzenie.

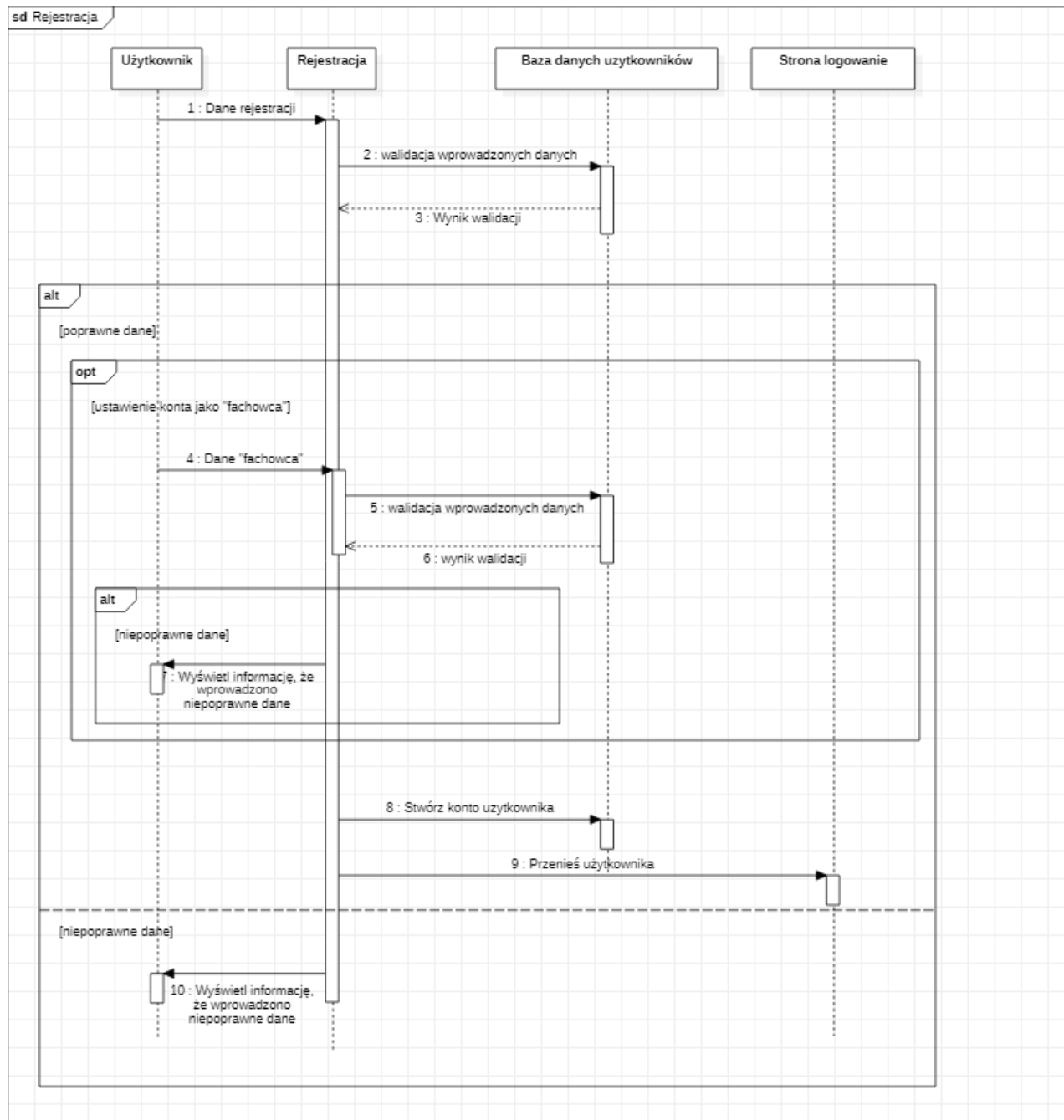
### 3. Użytkownik zamawia usługę



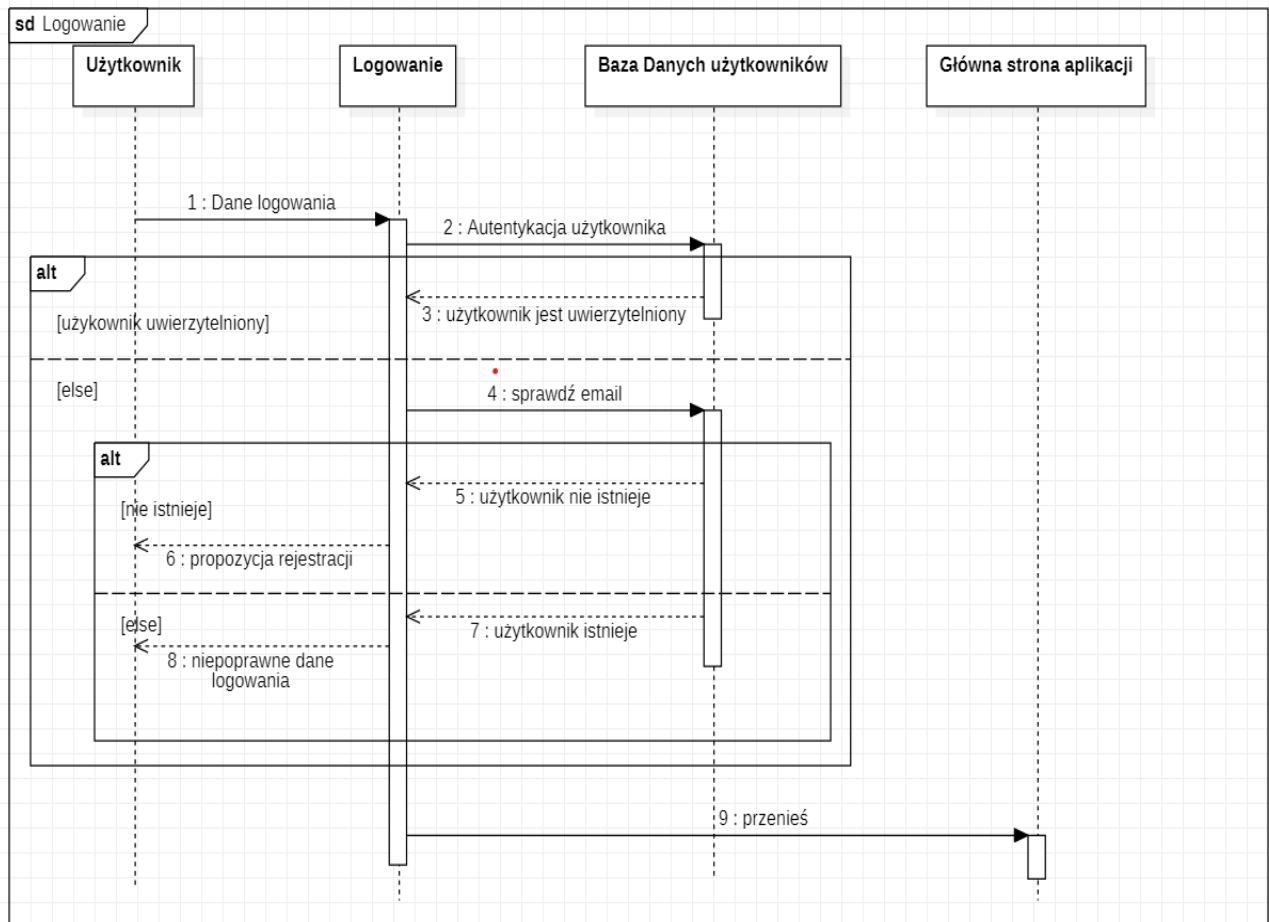
Nazwa	Użytkownik zamawia usługę
Scenariusz	Użytkownik zamawia wyszukaną przez siebie ofertę
Aktorzy	Użytkownik, Fachowiec
Warunki początkowe	Użytkownik musi mieć zainstalowaną aplikację i być zalogowany
Niezmienniki	Użytkownik chce zamówić usługę
Opis	<ol style="list-style-type: none"> <li>1. Użytkownik klika w zakładkę pozwalającą szukać ofert.</li> <li>2. Po znalezieniu odpowiadającej oferty, użytkownik zgłasza chęć skorzystania z usługi.</li> <li>3. Następnie użytkownik wybiera odpowiadający mu termin z wolnych terminów fachowca. <ol style="list-style-type: none"> <li>3a. W przypadku gdy fachowiec nie ma ustawionego terminarza, zostanie poproszony o jak najszybsze podanie wolnych terminów.</li> </ol> </li> <li>4. Po obopólnym ustaleniu odpowiedniej daty fachowiec akceptuje wykonanie usługi</li> </ol>
Warunki końcowe	Użytkownik jest umówiony na konkretną datę z fachowcem.
Źródło możliwych błędów	Fachowiec nie podaje wolnych terminów

# Diagramy sekwencji

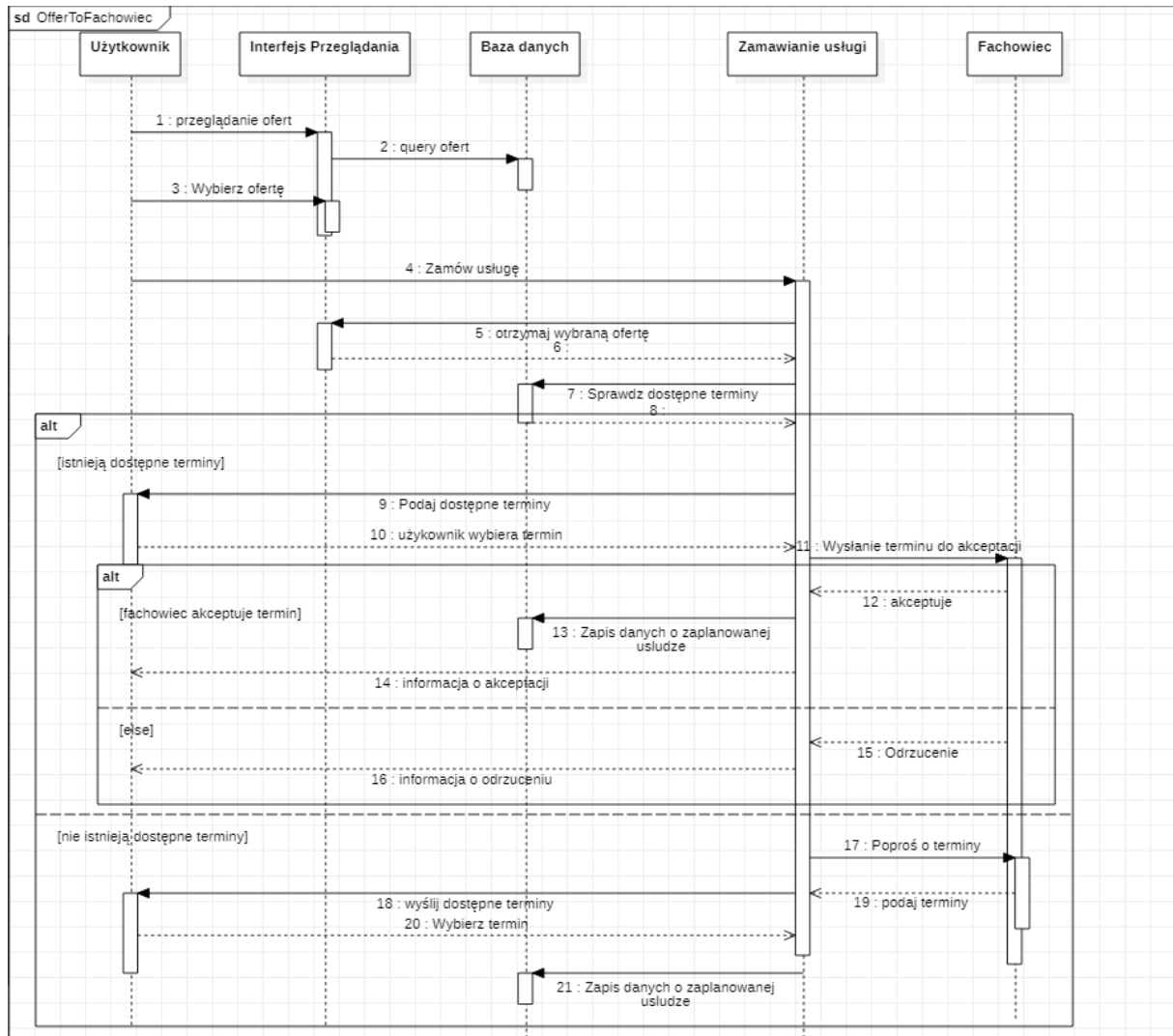
## Rejestracja



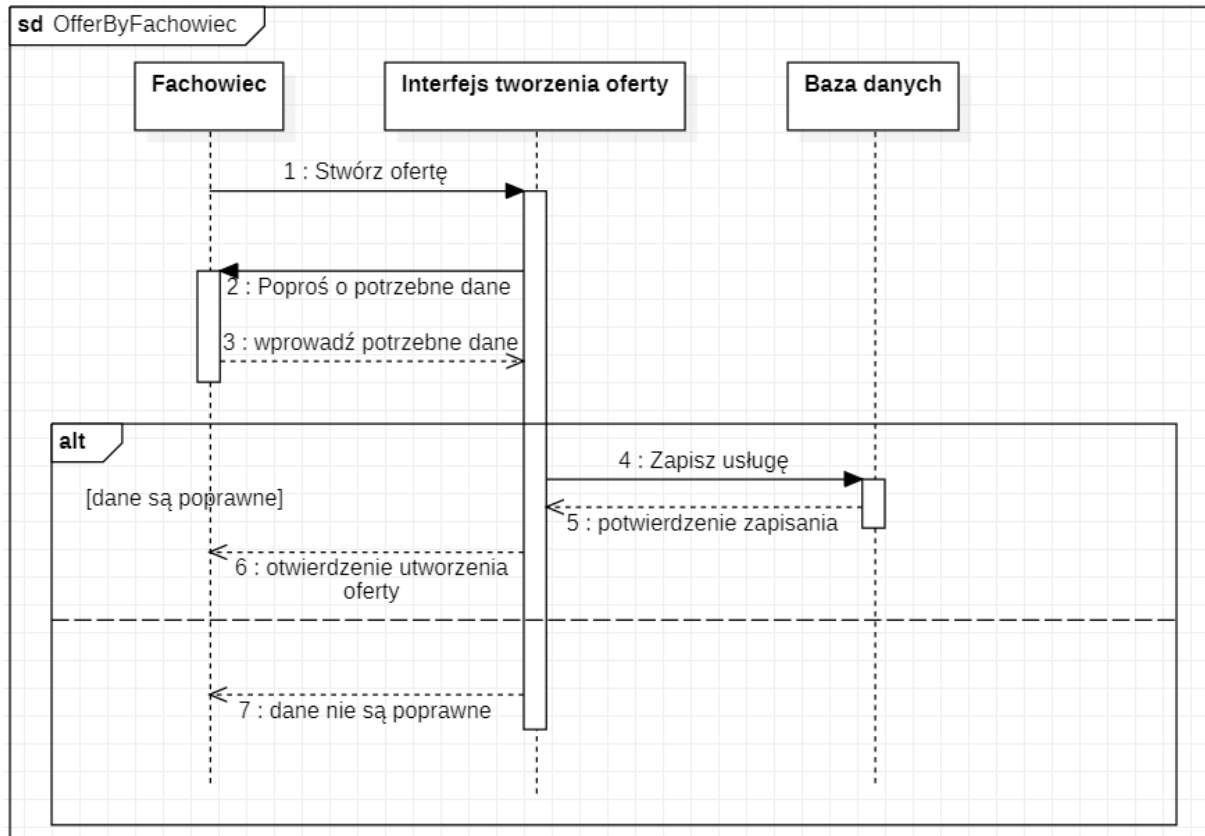
## Logowanie



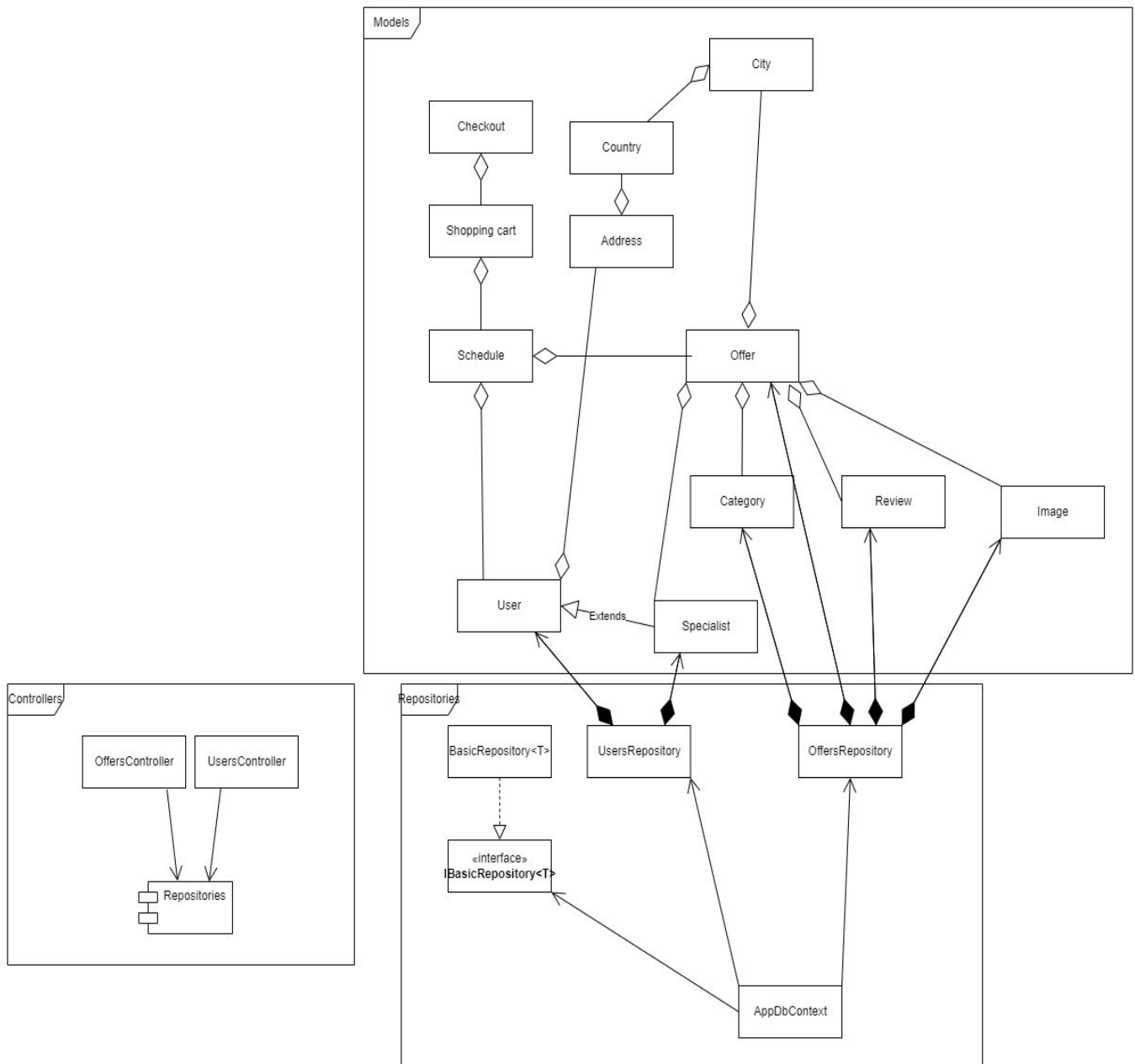
## Korzystanie z oferty oferowanej przez fachowca



### Tworzenie i obsługa oferty przez fachowca

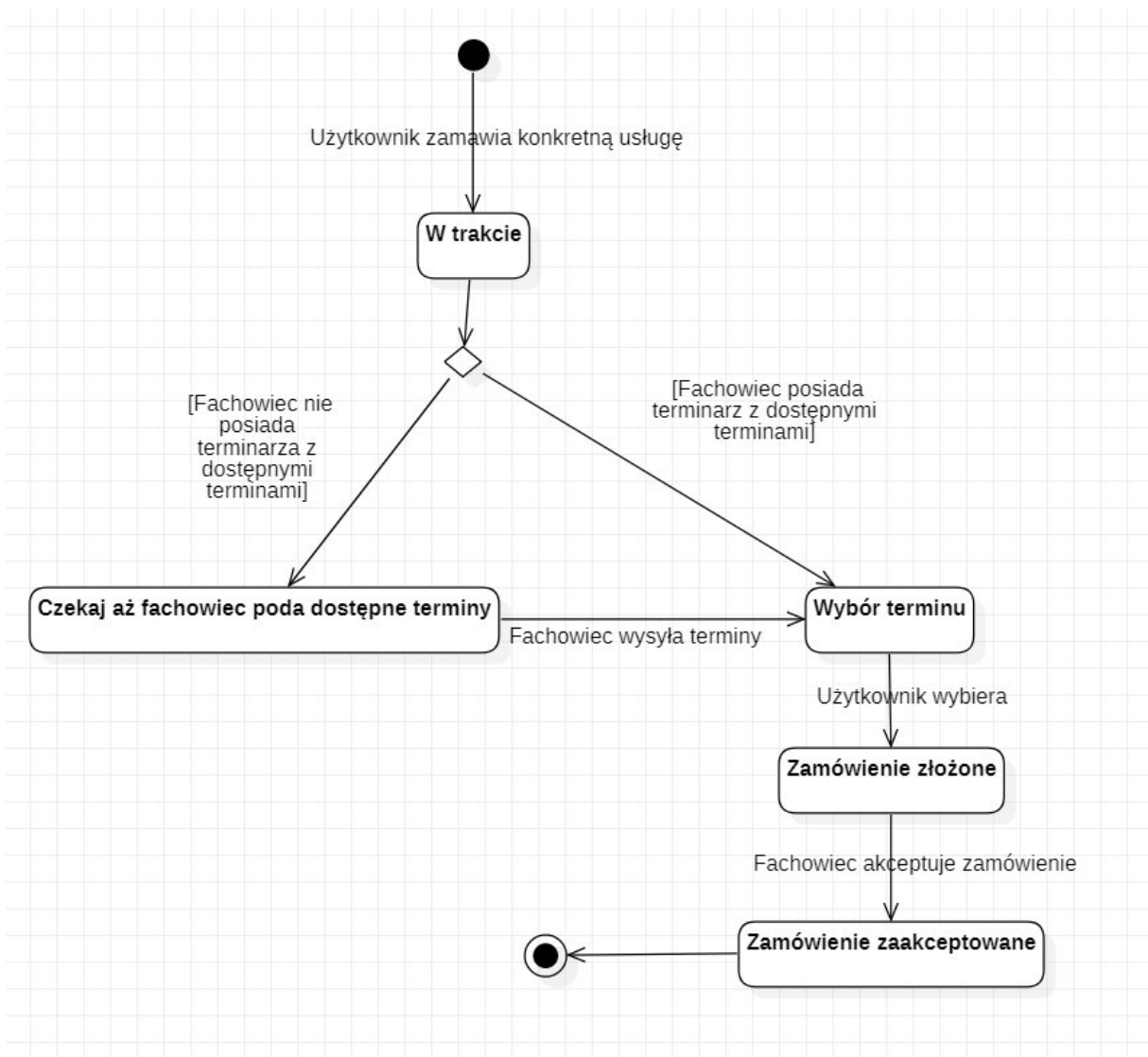


# Diagram klas

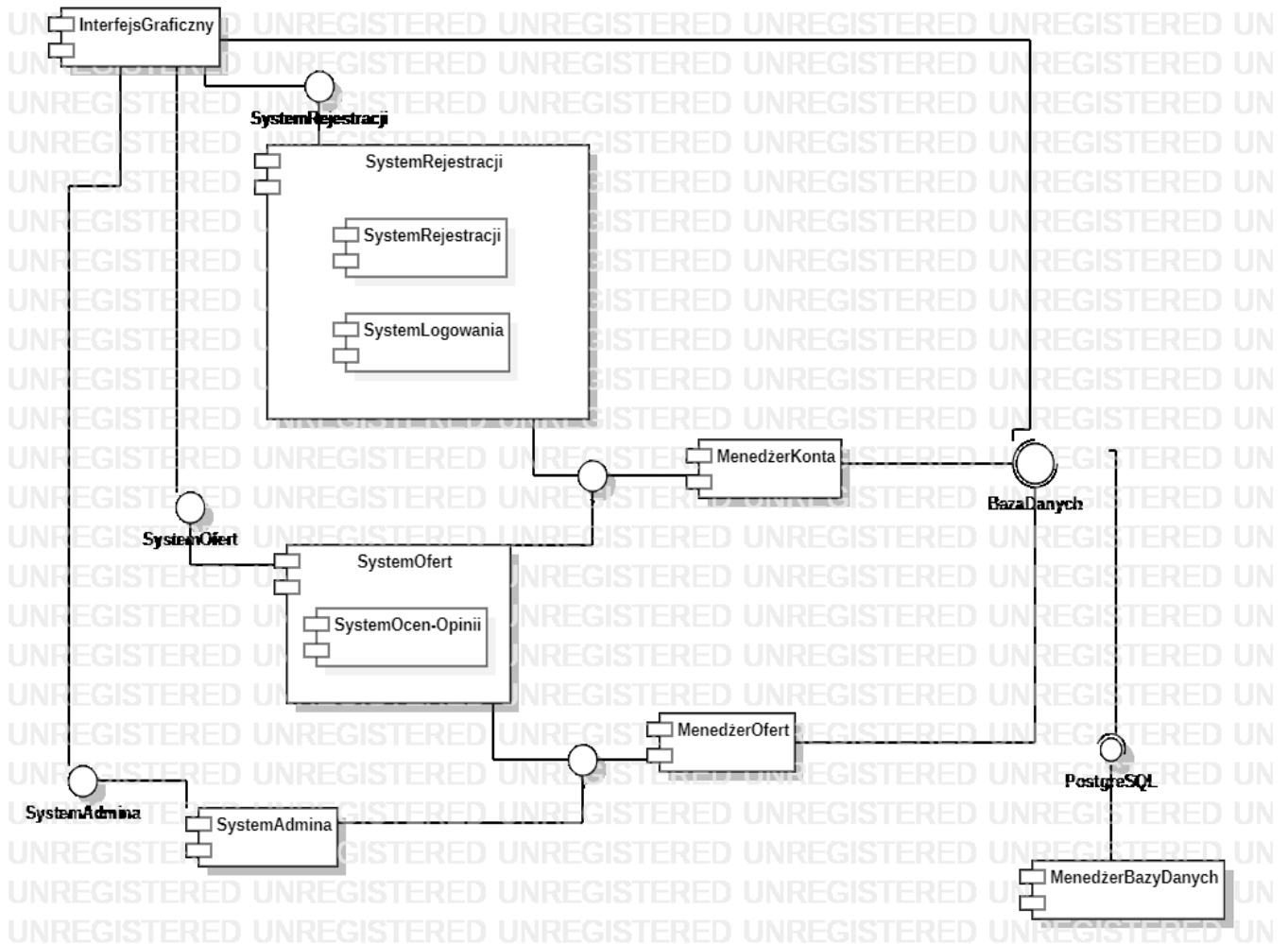




# Diagram stanu



# Diagram komponentów

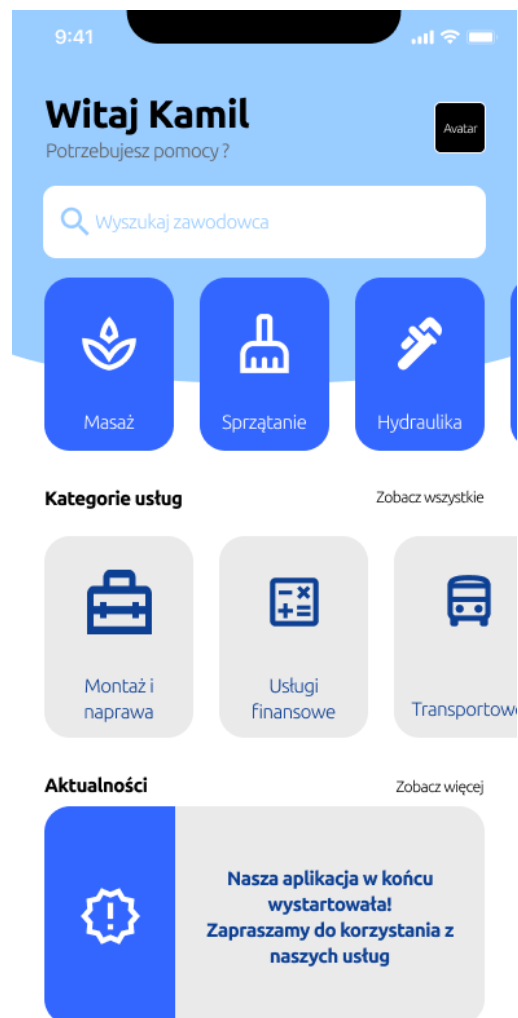


# Design graficzny

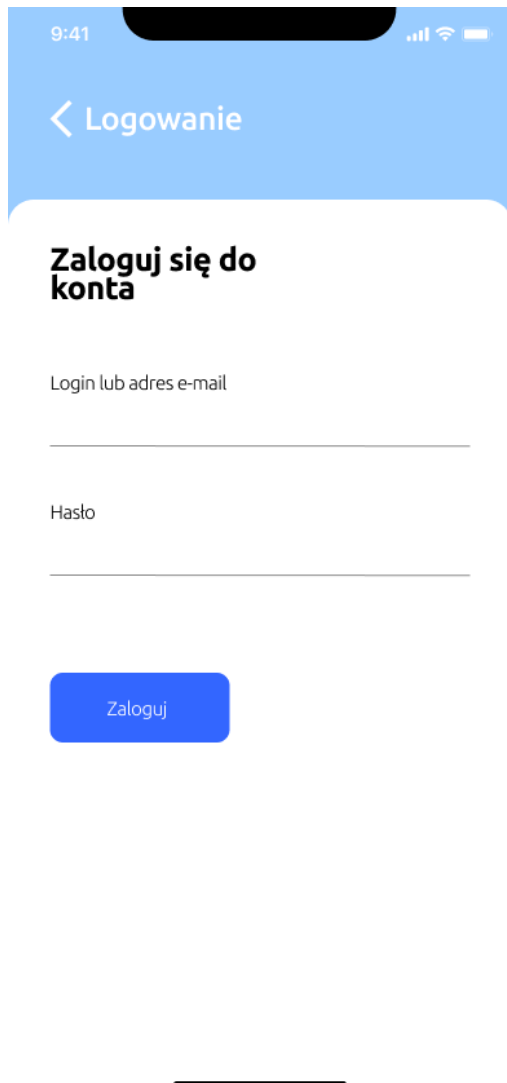
Ekran powitalny



Ekran startowy



## Ekran logowania



9:41

< Logowanie

**Zaloguj się do konta**

Login lub adres e-mail

\_\_\_\_\_

Hasło

\_\_\_\_\_

Zaloguj

\_\_\_\_\_

## Ekran rejestracji



9:41

< Rejestracja

**Utwórz konto**

Adres e-mail

\_\_\_\_\_

Nazwa użytkownika

\_\_\_\_\_

Imię

\_\_\_\_\_

Nazwisko

\_\_\_\_\_

Miejscowość

\_\_\_\_\_

Hasło

\_\_\_\_\_

Zarejestruj

\_\_\_\_\_

## Lista usług

