System wspomagania pracy firmy sprzątającej

**Etap 2 Projekt Systemu Informatycznego**

**Mateusz Kita Z710**

# Przedstawienie koncepcji systemu

System wspomagający pracę firmy sprzątającej ma za zadanie ułatwić kontakt klientów z firmą sprzątającą, zarządzanie zespołami sprzątającymi, wycenianie usług oraz zamawianie ekipy sprzątającej.

Klienci podczas przeglądania oferty będą mogli przesłać zapytanie do firmy sprzątającej a pracownicy będą mogli na nie odpowiedzieć. Klienci będą mieli do wyboru kilka usług sprzątania z wyszególnionymi cenami za 1m2 usługi sprzątającej. Po podaniu powierzchni lokalu do sprzątania klient zostanie poinformowany o całkowitym koszcie usługi.

Proces zamówienia będzie przebiegał sprawnie od wyboru konkretnej usługi, przez podanie danych do faktury oraz potwierdzenie zamówienia przez opłacenie go. Ewentualne reklamacje będą obsługiwane przez system wiadomości do firmy sprzątającej.

Dane do faktury będą przechowywane w tabelach bazy danych w celu przyspieszenia procesu składania kolejnych zamówień.

# Funkcje systemu oraz diagram hierarchii funkcji

Projektowany system informatyczny będzie posiadał następujące funkcje:

1. Przeglądanie oferty firmy sprzątającej:

* [Gość] Rejestracja
* [Gość] Logowanie
* [Gość/Klient] Wysłanie zapytania
* [Klient] Wybór usługi
* [Pracownik/Administrator] Zarządzanie usługami
* [Klient] Podanie danych do faktury
* [Klient] Potwierdzenie zamówienia usługi
* [Pracownik/Administrator] Zmiana statusu usługi
* [Pracownik/Administrator] Przypisanie zespołu do usługi
* [Pracownik/Administrator] Edytuj usługę
* [Pracownik/Administrator] Usuń usługę

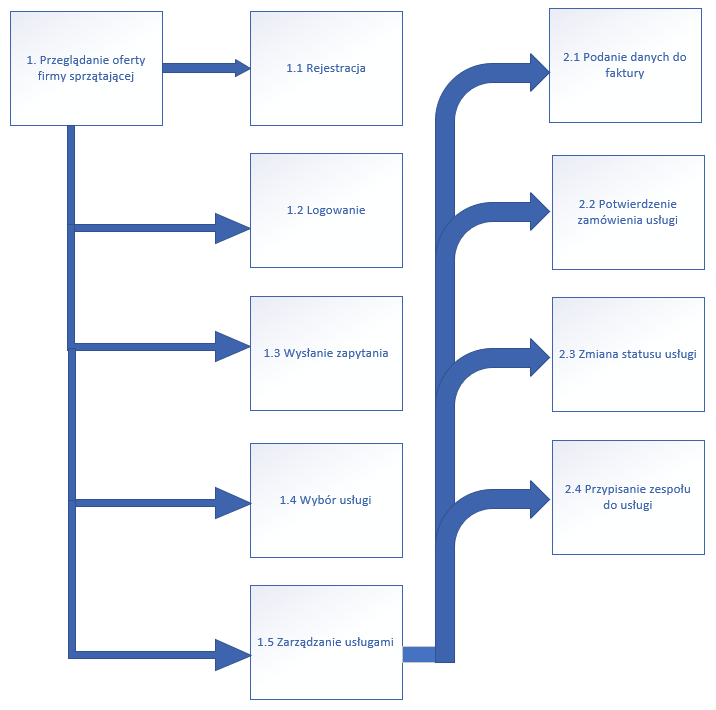
1. Przeglądaj listę klientów

* [Pracownik/Administrator] Wyświetl dane kontaktowe
* [Pracownik/Administrator] Wyślij wiadomość
* [Administrator] Edytuj dane klienta
* [Administrator] Usuń klienta

1. Przeglądaj listę użytkowników

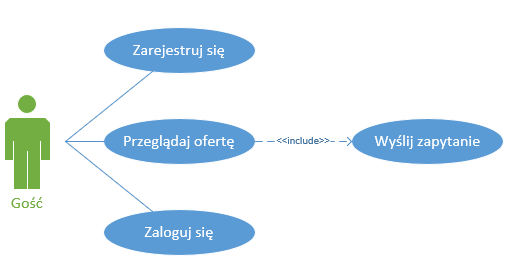
* [Administrator] Edytuj dane użytkownika
* [Administrator] Usuń użytkownika
* [Administrator] Zmień rolę użytkownika

## Hierarchia funkcji systemu:

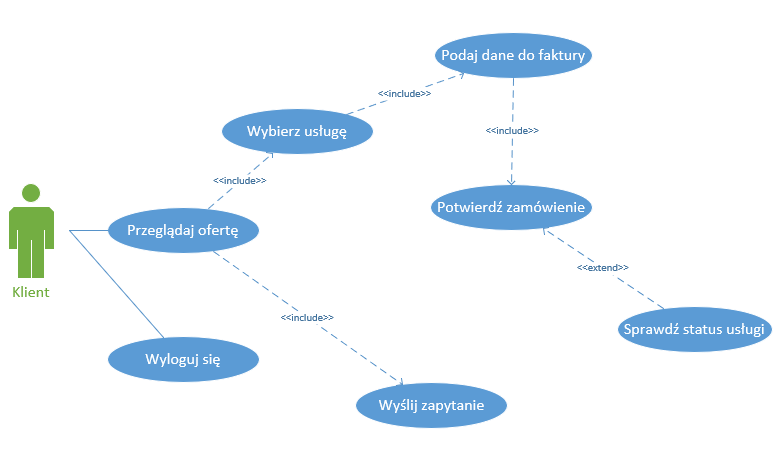


# Określenie aktorów         a. charakterystyka

# Aktorami w systemie będą użytkownicy zarejestrowani oraz niezarejestrowani. Użytkownicy zarejestrowani mogą należeć do jednej z trzech ról systemowych: Klient, Pracownik oraz Administrator. W zależności od roli użytkownikom przydzielone zostaną odpowiednie dostępy do wyświetlania danych bądź ich edycji. Niezarejestrowani użytkownicy będą mogli jedynie przeglądać ofertę firmy, bez możliwości zamówienia usługi. Użytkownicy zarejestrowani jako klienci będą mogli dodatkowo składać zamówienia oraz edytować je. Pracownicy będą mieli przydzielane zlecenia w zależności od swojej dostępności oraz lokalizacji. Administratorzy z kolei będą zarządzać całym systemem.



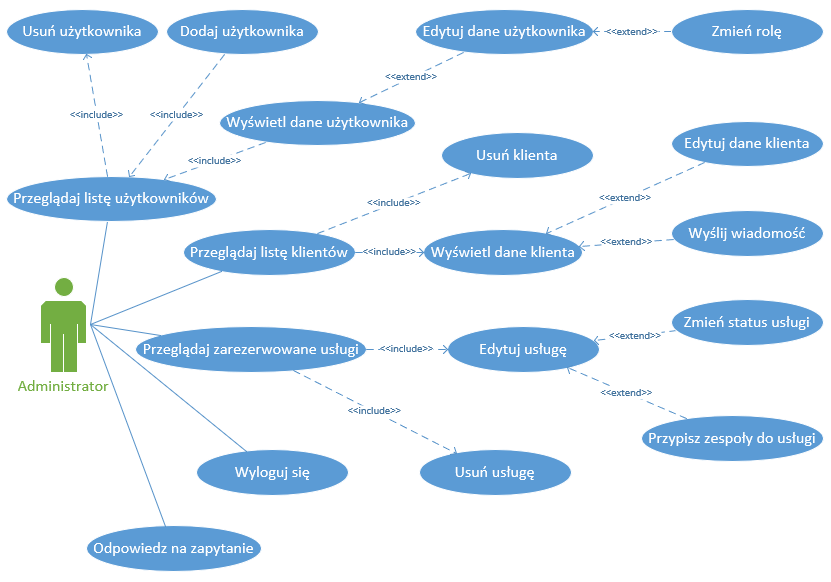
# Gość – użytkownik niezarejestrowany w systemie, mający możliwość logowania oraz rejestracji. W przypadku nie zakładania konta Gość może przeglądać ofertę firmy sprzątającej oraz przesłać zapytanie mailowe do tej firmy.



# Klient - użytkownik zarejestrowany w systemie posiadający możliwość wyboru usługi wraz z jej zarezerwowaniem i potwierdzeniem. Aktywne usługi mogą zostać zmodyfikowane dlatego Klient może sprawdzać ich status oraz może wysyłać zapytania do firmy sprzątającej.

# 

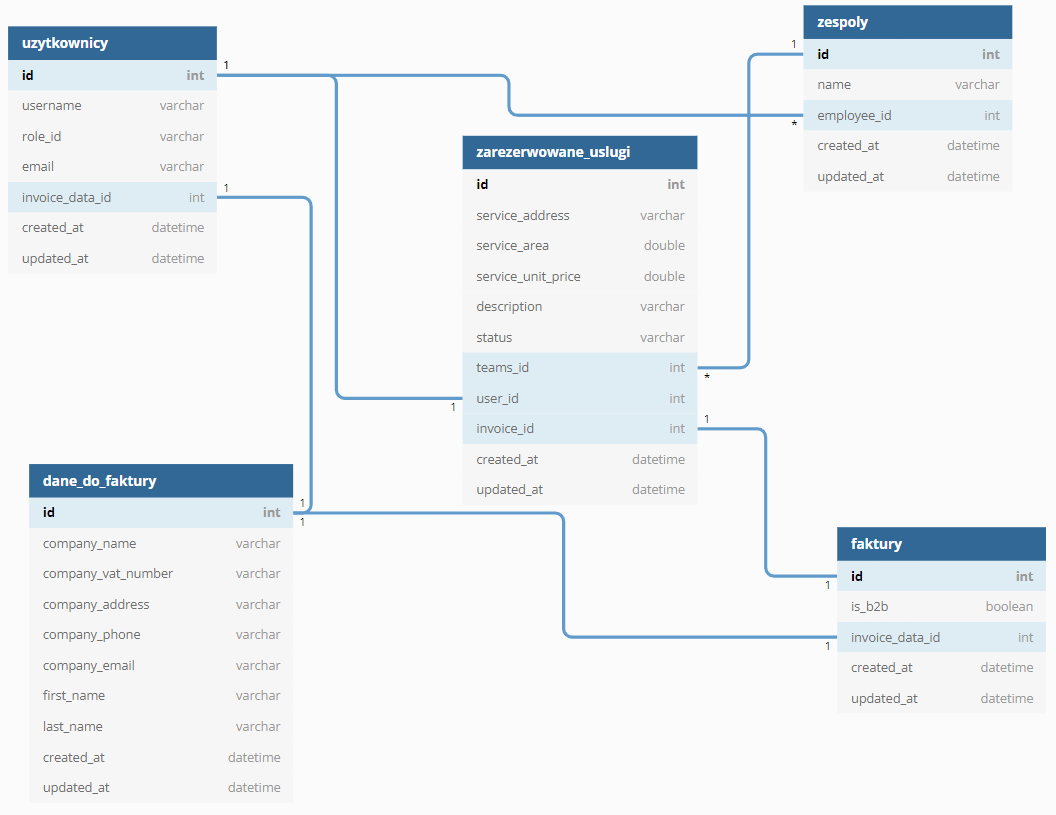
# Pracownik – użytkownik zarejestrowany w systemie oraz zatrudniony w firmie sprzątającej. Do jego uprawnień należy odpowiadanie na zapytania, wyświetlanie danych kontaktowych klientów, wysyłanie im wiadomości, edytowanie zarezerwowanych usług oraz ich edycja i usuwanie.



# Administrator – użytkownik zarejestrowany w systemie oraz zatrudniony w firmie sprzątającej posiadająca uprawnienia do administrowania systemem. Dzięki tym uprawnieniom oprócz możliwości przeglądania listy użytkowników Administrator będzie mógł dodawać, usuwać użytkowników oraz zmieniać ich dane. Ten użytkownik będzie mógł także usuwać klientów oraz edytować ich dane.

# Projekt źródła danych (ERD, opis szczegółowy encji)

## Diagram relacji między tabelami źródła danych



## Szczegółowy opis encji w postaci tabelarycznej

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nazwa tabeli | Opis tabeli | Nazwa pola | Typ pola | Opis pola |
| uzytkownicy | Przechowuje dane dotyczące kont użytkowników systemu | id | int | Klucz główny |
|  |  | username | varchar | Nazwa użytkownika, potrzebna do zalogowania się |
|  |  | role\_id | varchar | Nazwa roli użytkownika |
|  |  | invoice\_data\_id | int | Klucz obcy. Id w tabeli dane\_do\_faktury |
|  |  | created\_at | datetime | Data utworzenia rekordu |
|  |  | updated\_at | datetime | Data ostatniej modyfikacji |
| dane\_do\_faktury | Przechowuje dane klienta potrzebne do wystawienia faktur | id | int | Klucz główny |
|  |  | company\_name | varchar | Nazwa przedsiębiorstwa |
|  |  | company\_vat\_number | varchar | Numer NIP |
|  |  | company\_address | varchar | Adres przedsiębiorstwa |
|  |  | company\_phone | varchar | Numer telefonu |
|  |  | company\_email | varchar | Email |
|  |  | first\_name | varchar | Imię |
|  |  | last\_name | varchar | Nazwisko |
|  |  | created\_at | datetime | Data utworzenia rekordu |
|  |  | updated\_at | datetime | Data ostatniej modyfikacji |
| zarezerwowane\_usługi | Przechowuje informacje o zarezerwowanych usługach | id | int | Klucz główny |
|  |  | service\_address | varchar | Adres miejsca usługi |
|  |  | service\_area | double | Powierzchnia do sprzątania |
|  |  | service\_unit\_price | double | Cena za 1m2 usługi sprzątania |
|  |  | status | varchar | Status zlecenia |
|  |  | description | varchar | Opis zlecenia |
|  |  | teams\_id | int | Klucz obcy. Id w tabeli zespoly |
|  |  | user\_id | int | Klucz obcy. Id w tabeli uzytkownicy |
|  |  | invoice\_id | int | Klucz obcy. Id w tabeli faktury |
|  |  | created\_at | datetime | Data utworzenia rekordu |
|  |  | updated\_at | datetime | Data ostatniej modyfikacji |
| faktury | Przechowuje informacje o wystawionych fakturach | id | int | Klucz główny |
|  |  | is\_b2b | boolean | Informacja czy klient posiada działalność gospodarczą |
|  |  | invoice\_data\_id | int | Klucz obcy. Id w tabeli dane\_do\_faktury |
|  |  | created\_at | datetime | Data utworzenia rekordu |
|  |  | updated\_at | datetime | Data ostatniej modyfikacji |
| zespoły | Przechowuje dane zespołów sprzątających – listę pracowników należących do zespoły oraz przypisane zlecenia. | id | int | Klucz główny |
|  |  | name | varchar | Nazwa zespołu |
|  |  | employee\_id | int | Klucz obcy. Id w tabeli uzytkownicy |
|  |  | created\_at | datetime | Data utworzenia rekordu |
|  |  | updated\_at | datetime | Data ostatniej modyfikacji |

# Architektura systemu

Projekt utworzony będzie za pomocą stosu technologicznego MERN oznaczającego 4 technologie odpowiedzialne za poszczególne warstwy systemu informatycznego:

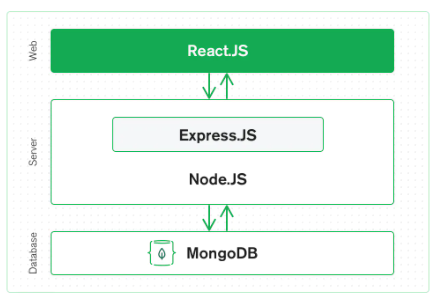
MongoDB – technologia bazy danych

ExpressJS – framework dla technologii NodeJS

ReactJS – biblioteka odpowiedzialna za zarządzanie graficznym interfejsem użytkownika

NodeJS – środowisko uruchomieniowe dla kodu JavaScript

## Schemat architektury projektowanego systemu



źródło: <https://www.mongodb.com/mern-stack>

Część front-endowa projektu będzie obsługiwana przez aplikację napisaną z użyciem biblioteki ReactJS. Jest to bardzo popularna technologia internetowa z dużą społecznością sympatyków skupioną wokół niej. Umożliwia ona zagnieżdżanie znaczników HTML wewnątrz kodu JavaScript, a wykorzystanie wirtualnego DOM sprawia, że interfejsy tworzone za jej pomocą są szybkie i dynamiczne. Do zarządzania stanem aplikacji użyta zostanie biblioteka Redux a do wyeliminowania błędów związanych ze swobodnym typowaniem zmiennych JavaScript użyty zostanie nadzbiór tego języka – TypeScript.

Za część back-endową projektu będzie odpowiadała technologia NodeJS wraz z frameworkiem ExpressJS. Ta para technologii również posiada dużą społeczność tworzącą funkcjonalne biblioteki i rozszerzenia, m.in. adaptery do komunikacji z bazą danych MongoDB.

W celu integracji części front i back-endowej zostanie stworzone API, po którym będzie komunikowała się aplikacja.