

# **Dokumentacja techniczna do projektu**



## ***“DoctorToothie”***

System do rejestracji oraz zarządzania wizytami w gabinecie/tach stomatologicznym/ch

### **Krótki opis działania Projektu**

Nasza aplikacja umożliwia wygodne oraz skuteczne planowanie oraz zarządzanie zabiegami/wizytami w gabinecie stomatologicznym. System zapewnia funkcjonalność przeglądania grafiku dostępnych lekarzy specjalistów oraz listy wykonywanych przez nich rodzajów usług stomatologicznych. Za pomocą systemu można zarejestrować wizytę oraz śledzić jej przebieg a także sprawdzać historię przebytych zabiegów.

### **Autorzy aplikacji**

- Jakub Janiszewski;
- Patryk Jankowski;

### **Specyfikacja używanej przez nas technologii**

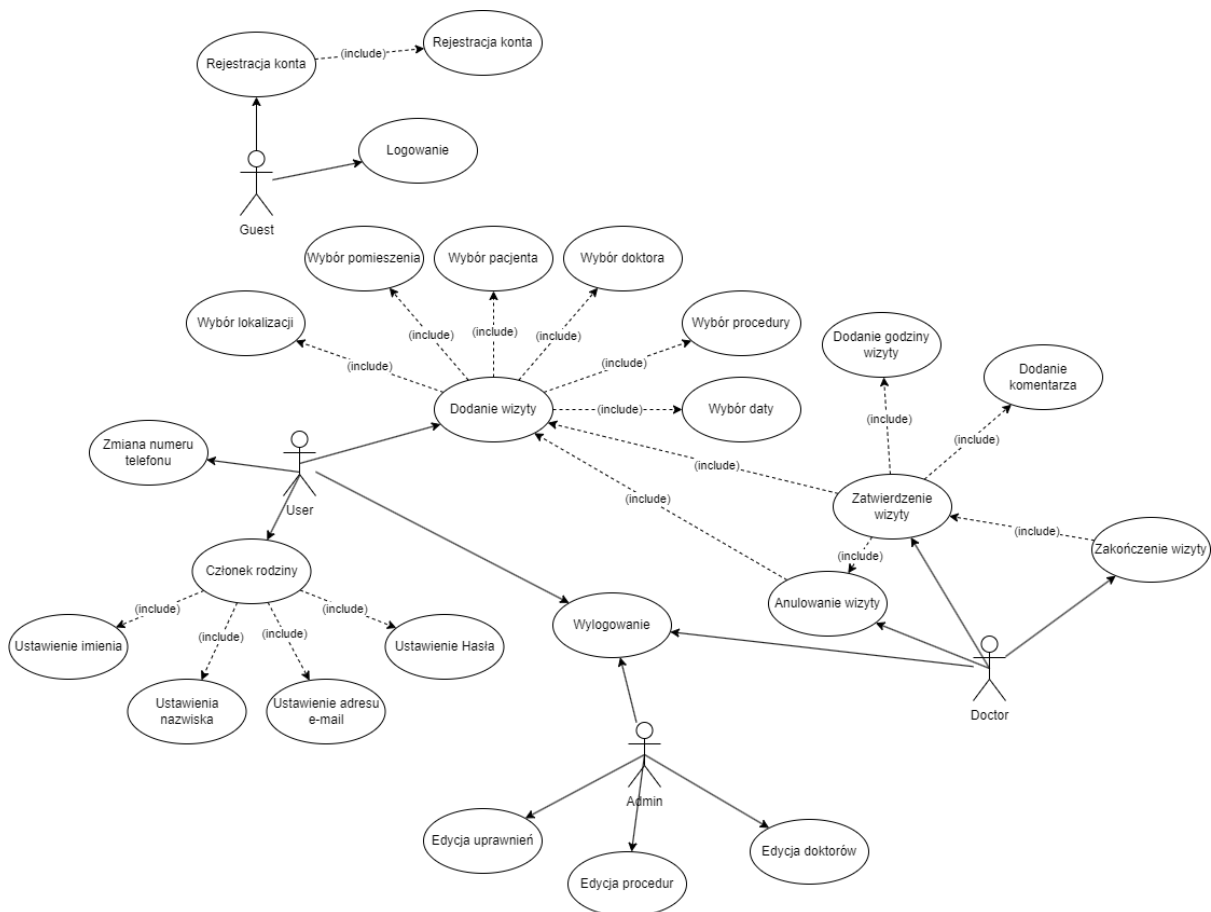
- Język programowania: C#;
- Platforma: .NET 8;
- Baza danych: SQL;
- Dodatkowe biblioteki oraz frameworki:
  - ASP.NET Core MVC dla logiki serwerowej i interfejsów użytkownika;
  - Entity Framework Core dla ORM;

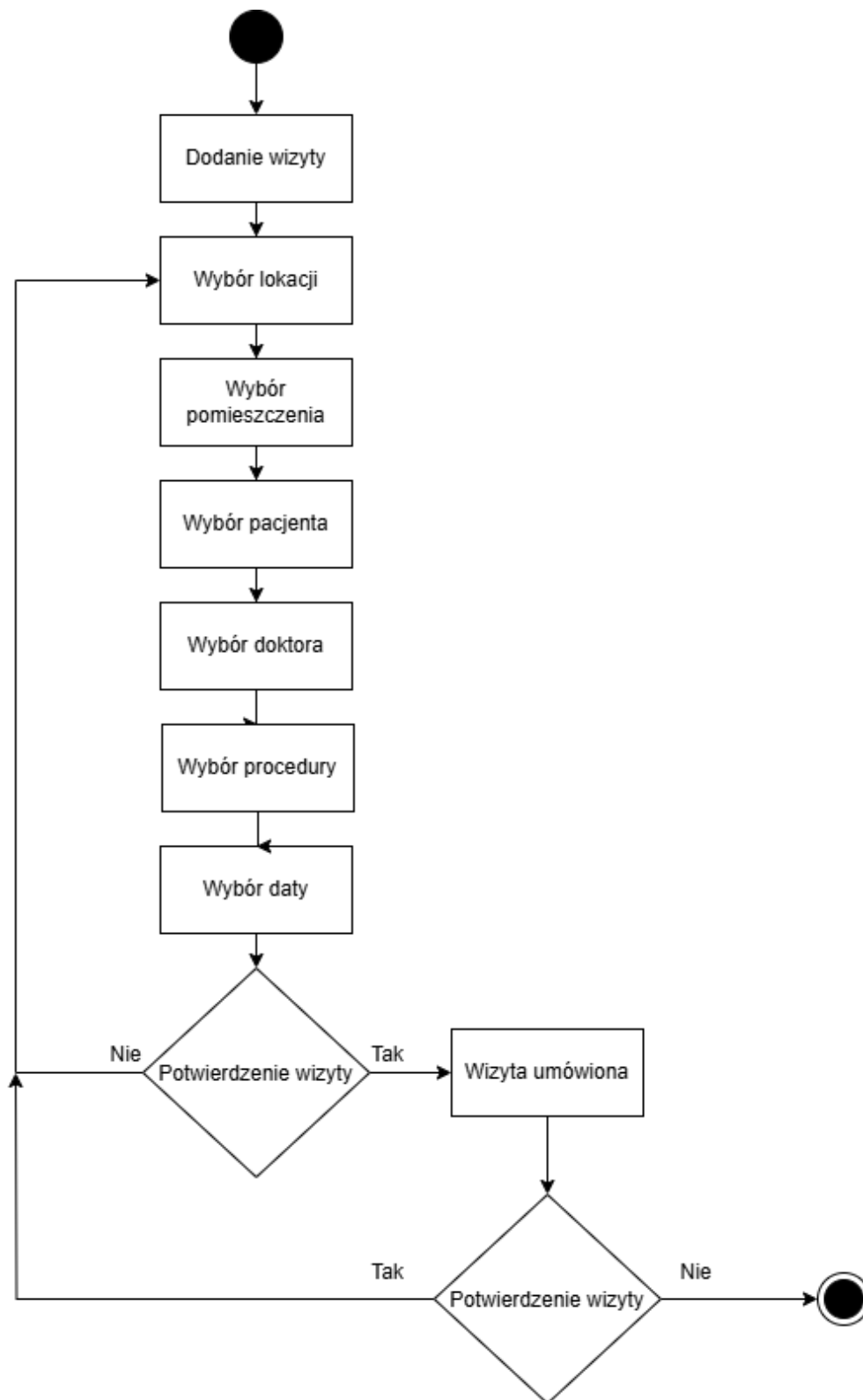
### **Krótką charakterystyka najciekawszych funkcjonalności**

- Zarządzanie wizytami całej rodziny za pośrednictwem jednego konta;
- Przeglądanie odbytych wizyt, gdzie lekarz może wprowadzić komentarze, zalecenia;

## Opis sytemu użytkowników

- **Role:** User(pacjent), Doctor(lekarz), Admin(superadmin)
- **Uprawnienia:**
  - Admin(superadmin): [login: Admin hasło:zaq1@WSX]
    - Dodawanie oraz edycja typów zabiegów
    - Dodawanie oraz edycja uprawnień
    - Dodawanie oraz edycja
  - Doktor(superadmin):
    - Wgląd na historyczne wizyty
    - Podejrzenie aktualnie zaplanowanych wizyt
    - Deklaracja czasu wizyty
  - User:
    - Zarejestrowanie członka rodziny
    - Umówienie wizyty
    - Wgląd na historyczne wizyty





## **Instrukcja uruchomienia aplikacji**

1. Uruchom aplikację przez Visual Studio.

2. Kliknij 'Start' lub naciśnij F5 w Visual Studio.

[Visual Studio skompiluje projekt i uruchomi lokalny serwer, na którym będzie można przeglądać aplikację.]

W przypadku, gdy aplikacja się nie uruchomi należy usunąć jeden plik  
(appsettings.Developments.json)

## Modele

- AppUser - rozszerza IdentityUser;
  - FirstName [string]
  - LastName [string]
  - EmployeeLocationId [int?]
  - ParentId [string?]
- Location
  - Id [int]
  - Name [string]
  - Address [string]
  - Image [string?]
- ProcedureType - Model zawierający informacje na temat procedur;
  - Id [int]
  - Title [string]
  - Description [string]
  - BasePrice [int]
- Reservation- zawiera informacje na temat rezerwacji;
  - Id [int]
  - Status [int]
  - CreateById [string]
  - PatientId [int]
  - LocationId [int?]
  - RoomId [int?]
  - DoctorId [string?] ; klucz obcy do AppUser,

- ProcedureTypeId [int?]
  - ProcedureNotes [string?]
- Room- Model zawierający informacje na temat odpowiedniego pomieszczenia;
  - Id [int]
  - Number [string]
  - ParentId [int] ; klucz obcy do Location,

# Kontrolery i Metody

- Appointment -zarządzanie wizytami
  - Index [GET]
  - Review [GET]
  - Accept [POST]
  - Cancel [GET]
  - Done [GET]
- Children
  - Index [GET]
  - New [GET, POST]
  - Edit [GET, POST]
- Doctors
  - Index [GET]
  - New [GET, POST]
  - Edit [GET, POST]
  - Remove [GET]
  - FixRoles [GET]
- Home
  - Index [GET]
  - Error [GET]
- Procedure
  - Index [GET]
  - New [GET, POST]
  - Edit [GET, POST]
- Reserve
  - Index [GET]
  - Cancel [GET]
  - Location [GET, POST]
  - Room [GET, POST]
  - Patient [GET, POST]
  - Doctor [GET, POST]
  - Procedure [GET, POST]
  - Date [GET, POST]

- Review [GET]
  - Reserve [GET]
- Roles
  - Index [GET]
  - EditUser [GET, POST]



