|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | Dokumentacja techniczna projektu | | HelpMeNow | | Przedmiot: Programowanie zaawansowane  Prowadzący: dr Paweł Płaczek |   Strona główna | WSB w Poznaniu |

Spis treści

[1. Tematyka projektu 2](#_Toc106203852)

[2. Diagram przypadków użycia 2](#_Toc106203853)

[3. Diagram klas 3](#_Toc106203854)

[4. Opis techniczny projektu 3](#_Toc106203855)

[Aplikacja serwerowa 4](#_Toc106203856)

[Aplikacja kliencka 5](#_Toc106203857)

[Modele 6](#_Toc106203858)

[5. Potencjalne możliwe problemy i zagrożenia (do części technicznej) 7](#_Toc106203859)

[6. Scenariusze testów 7](#_Toc106203860)

[7. Spis rysunków i tabel 8](#_Toc106203861)

[8. Lista zmian w dokumencie 8](#_Toc106203862)

## 1. Tematyka projektu

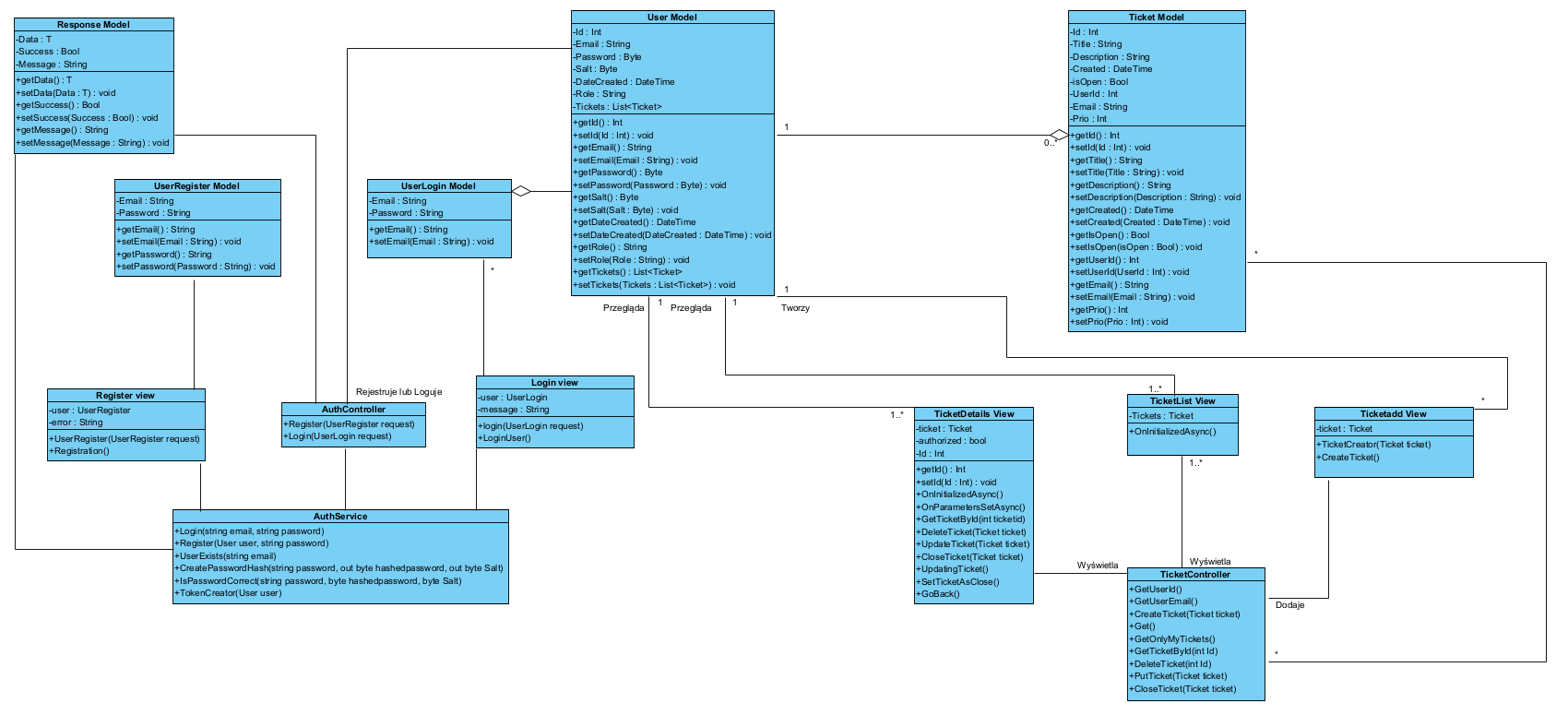
Tematem projektu jest system, który pozwala na zgłaszanie problemów w firmie (System ticketowy)   
System będzie pozwalał na:   
- Tworzenie zgłoszeń i przeglądanie zgłoszeń przez użytkownika

- Tworzenie, zamykanie, przeglądanie oraz edytowanie zgłoszeń przez admina

## 2. Diagram przypadków użycia

Rysunek 1 Diagram przypadków użycia

## 3. Diagram klas



Rysunek 2 Diagram klas

## 4. Opis techniczny projektu

Użyte technologie:

-C#   
-Framework Blazor

-HTML  
-CSS

-Azure SQL Database.  
  
Rozszerzenia:  
-Blazored.LocalStorage  
-Microsoft.AspNetCore.Authentication.JwtBearer  
-Microsoft.AspNetCore.Components.Authorization  
-Microsoft.AspNetCore.Components.WebAssembly   
-Microsoft.EntityFrameworkCore  
-Microsoft.EntityFrameworkCore.SqlServer  
-Microsoft.EntityFrameworkCore.Tools   
-Microsoft.VisualStudio.Web.CodeGeneration.Design  
-Swashbuckle.AspNetCore

Aplikacja napisana w architekturze Klient-Serwer.   
Zgłoszenia będą przetrzymywane w bazie danych. Każde zgłoszenie będzie miało osobne ID, dzięki którym będziemy mogli odnieść się do pojedynczego zgłoszenia.   
Do każdego zgłoszenia przypisany jest użytkownik, który zgłosił problem.  
Administrator aplikacji może zamykać i usuwać zgłoszenia, może również dodawać opis do zgłoszenia.  
   
Aplikacja korzysta z modeli (np. model użytkownika lub zgłoszenia).  
Z kontrolerów (kontroler użytkownika (Auth) i zgłoszenia). W kontrolerach znajdują się funkcje takie jak:  
Dodawanie, usuwanie, edycja, wyświetlanie zgłoszenia. Funkcje te wykonują się w aplikacji serwerowej, a wyniki ich działań pojawiają się w aplikacji klienckiej. (Zastosowano protokół http).

### Aplikacja serwerowa

Funkcje z TicketController:   
  
CreateTicket -> Tworzy zgłoszenie.  
   
Get -> Pobieranie listy zgłoszeń (dla administratora).

**GetOnlyMyTicket** -> Pobieranie listy zgłoszeń zalogowanego użytkownika (dla zwykłego użytkownika).

**DeleteTicket** -> Usuwanie zgłoszenia.

**PutTicket** -> Edycja zgłoszeń (W tym wypadku edycja opisu od administratora).

**CloseTicket** -> Ustawianie statusu zgłoszenia na zamknięty.  
  
Do rejestracji i logowania użytkownika użyto specjalnego rozszerzenia pozwalającego na wykonywanie tych akcji.  
Wszystkie działania związane z logowaniem oraz z rejestracją znajdują się w specjalnym serwisie (AuthService) znajdującym się w aplikacji klienckiej.

**Funkcje z AuthService:**  
**Login** –> Logowanie użytkownika.

**Register** -> Rejestrowanie użytkownika.

**UserExist** -> Sprawdzanie czy użytkownik o podanej nazwie znajduje się już w bazie.

**CreatePasswordHash** -> Tworzenie funkcji skrótu oraz soli do podanego hasła.

**IsPasswordCorrect** -> Sprawdzanie czy podane hasło jest takie samo jak hasło zapisane w bazie.

**TokenCreator** -> Tworzenie tokenu do identyfikacji zalogowanego użytkownika  
  
Funkcje z AuthService są używane w kontrolerze AuthController, gdzie są one skondensowane oraz są wysyłane poprzez protokół http do aplikacji klienckiej.  
  
**Funkcje z AuthController:**  
**Login** –> Logowanie użytkownika.  
  
**Register** -> Rejestrowanie użytkownika

### Aplikacja kliencka

Dzięki Blazor framework wszystkie funkcje w aplikacji klienckiej znajdują się w tych samych plikach co HTML.  
  
Aplikacja kliencka wysyła zapytania do aplikacji serwerowej i otrzymuje od niej wyniki tych zapytań.   
  
**Funkcje z strony Register  
  
UserRegister** -> Zapytanie do aplikacji serwerowej o dane nowego użytkownika (Endpoint: AuthController)

**Registration** -> Funkcja, która tworzy nam użytkownika po zakończeniu formularza.

**Funkcje z strony Loginpage:**

**login**-> Zapytanie do aplikacji serwerowej o dane istniejącego już użytkownika (Endpoint: AuthController).

**LoginUser** -> Funkcja, która loguje użytkownika po zakończeniu formularza

**Funkcje z strony TicketList:**Podczas otwierania podstrony ticketlist wyzwalana jest funkcja (OnInitializedAsync)  
która wysyła zapytanie do aplikacji klienckiej o listę zgłoszeń znajdujących się w bazie danych.   
Jeśli użytkownik jest administratorem to zwróci wszystkie zgłoszenia, a jeśli jest normalnym użytkownikiem to zwróci tylko zgłoszenia stworzone przez zalogowanego użytkownika.

**Funkcje z strony TicketAdd:  
  
TicketCreator**-> Zapytanie do aplikacji serwerowej o dane nowego zgłoszenia (EndPoint: TicketController)  
  
**CreateTicket** -> Funkcja, która tworzy nam zgłoszenie po zakończeniu formularza.  
  
**Funkcje z strony TicketDetails:**Użytkownik, który naciśnie na tytuł zgłoszenia w liście zgłoszeń zostanie przeniesiony do strony, gdzie może zobaczyć detale tego zgłoszenia.   
  
Podczas otwierania podstrony ticketdetails wyzwalana jest funkcja (OnInitializedAsync)

która sprawdza rolę użytkownika.  
  
**GetTicketByID** -> Pobiera id otwartego zgłoszenia (Endpoint: TicketController)  
  
**OnParametersSetAsync ->** Podczas inicjalizowania strony przypisuje jej Id wybranego zgłoszenia.  
  
**DeleteTicket ->** Zapytanie do aplikacji serwerowej o usunięcie wybranego zgłoszenia (EndPoint: TicketController)  
  
**DeletingTicket ->** Usuwanie zgłoszenia po naciśnięciu odpowiedniego przycisku.  
  
**UpdateTicket ->** Zapytanie do aplikacji serwerowej o zaktualizowanie wybranego zgłoszenia (EndPoint: TicketController)  
  
**UpdatingTicket ->** Aktualizowanie zgłoszenia po naciśnięciu odpowiedniego przycisku.  
  
**CloseTicket ->** Zapytanie do aplikacji serwerowej o ustawienie statusu zgłoszenia na zamknięty (EndPoint: TicketController).  
  
**SetTicketAsClose->** Zamknięcie zgłoszenia po naciśnięciu odpowiedniego przycisku.   
  
**GoBack ->** Powrót do poprzedniej strony (ListaTicketów) po naciśnięciu odpowiedniego przycisku.

### Modele

**UserRegister:**

Email -> Adres Email nowego użytkownika.  
Password -> Hasło nowego użytkownika.  
**UserLogin**Email -> Adres Email użytkownika.

Password -> Hasło użytkownika.  
  
**User:**Id –> Identyfikator użytkownika.

Email –> Adres Email użytkownika.  
Password -> Hasło z funkcją skrótu.

Salt -> Sól do hasła.  
DateCreated -> Data utworzenia użytkownika.  
Role -> Rola użytkownika.

**Ticket:**Id –> Identyfikator zgłoszenia.  
Title -> Tytuł zgłoszenia.  
Description -> Opis zgłoszenia.  
Created -> Data utworzenia zgłoszenia.  
IsOpen -> Status zgłoszenia.  
Solution -> Opis rozwiązania zgłoszenia.  
Prio-> Priorytet zgłoszenia.   
  
**Response:**Data-> Jest to typ generyczny, przyjmuję wartość na której chcemy działać.   
Message -> Przypisujemy wiadomość,  
Success -> Przypisujemy true, albo false w zależności czy funkcja wykonała poprawnie swoje zadanie czy nie.

## 5. Potencjalne możliwe problemy i zagrożenia (do części technicznej)

-Brak dostępu do serwera

-Awaria bazy danych

## 6. Scenariusze testów

Test case 1:

Test tworzenia ticketu

- test połączenia z bazą danych (czy wyszuka nam użytkownika i przypisze go do ticketu, czy Ticket zapisze sie w bazie)

- test generowania numerów ticketów,

- kategoryzacja ticketów – nadanie priorytetu,

Test case 2:

Test dodawania komentarzy do ticketów

- sprawdzenie, czy jesteśmy w stanie dodać komentarz jako admin

Test case 3:

Test zamykania ticketu

- sprawdzenie, czy jako Administrator / pracownik helpdesku jesteśmy w stanie zamknąć ticket

## 7. Spis rysunków i tabel

[Rysunek 1 Diagram przypadków użycia 1](#_Toc106202712)

[Rysunek 2 Diagram klas 2](file:///C:\Users\yooma\Downloads\technical_doc_template-PawLid-2.docx#_Toc106202713)

## 8. Lista zmian w dokumencie

|  |  |  |
| --- | --- | --- |
| Rewizja | Imię i nazwisko | Opis |
| 1 | Paweł Kitta | Stworzenie tematyki projektu. |
| 2 | Paweł Kitta | Stworzenie diagramów. |
| 3 | Lidia Grefkiewicz | Opis techniczny projektu. |
| 4 | Lidia Grefkiewicz | Stworzenie scenariuszy testów. |
| 5 | Lidia Grefkiewicz | Stworzenie spisu rysunków i tabel |