

UNIwersytet Zielonogórski

Wydział Informatyki, Elektrotechniki i Automatyki

Platforma .NET – Projekt
Prowadzący: dr inż. Marek Sawerwain

Tytuł raportu/sprawozdania

Wykonał: Damian Radecki, Grupa dziekańska: 33-INF-SSI-SP
Projekt realizowano razem z:
Damian Kurkiewicz
Data oddanie projektu: DD mmmm YYYY

Ocena:

Spis treści

1 Wprowadzenie	2	4.2 Modele bazy danych	4
1.1 Aplikacje czasu rzeczywistego	2	4.3 Modele DTO	4
1.2 Opis działania	2	4.4 Implementacja kontrolerów	4
1.3 Grupa docelowa	2	4.5 Problemy i ich rozwiązania	4
2 Użyte technologie	2	5 Testy	4
2.1 .Net Core	2	5.1 Testy jednostkowe	4
2.2 Entity Framework	4	5.2 Testy integracyjne	4
2.3 SignalR	4	6 Opis wkładu własnego w realizację projektu	4
2.4 MySql	4	6.1 Stworzenie i konfiguracja bazy danych	4
2.5 Angular	4	6.2 Stworzenie systemu logowania i rejestracji	4
3 Projekt	4	6.3 Autoryzacja i autentykacja użytkowników	4
3.1 Struktura projektu	4	6.4 Konfiguracja środowiska i serwera	4
3.2 Use Cases	4	6.5 Strona internetowa opracowana w Angular	4
3.3 Struktura bazy danych	4	6.6 Implementacja zarządzania wiadomościami	4
3.4 Komunikacja z web servicem	4	6.7 Implementacja zarządzania powiadomieniami	4
4 Implementacja	4	6.8 Implementacja zarządzania chatami	4
4.1 Diagram klas	4	6.9 Obsługa SignalR	4
		7 Podsumowanie	4
		7.1 Wnioski	4
		7.2 Do zrealizowania przy dalszym rozwoju	4

Spis listingów

Motto:
Pisanie raportu przywilejem każdego studenta.

1 Wprowadzenie

1.1 Aplikacje czasu rzeczywistego

Aplikacje działające na żywo oferują wiele korzyści, które są sporym ułatwieniem dla użytkowników podczas używania takiej aplikacji. Czynności wykonywane bez odświeżania strony nie tylko skracają czas wykonywania czynności czy obsługiwanie samej witryny to jeszcze znacznie ułatwiają komunikację, unikają blokowania strony i tworzą bardziej intuicyjny interfejs. Takie aplikacje internetowe stają się normą w dzisiejszych czasach. Każde przeładowanie strony jest nie komfortowe i stwarza pewnego rodzaju niebezpieczeństwo wykradnięcia danych. Serwisy internetowe obsługujące komunikację real-time z klientem są lepiej zabezpieczone i działają wydajnościowo lepiej. Powstawało wiele technologii do wsparcia komunikacji na żywo, które działają zarówno po stronie witryny i serwera. Są to między innymi WebSocket, SignalR, RabbitMQ czy Apache Kafka. Wszystkie z nich są dziś globalnie używane do wsparcia przekazu informacji.

1.2 Opis działania

Projekt chatu na żywo jest aplikacją, która wspiera komunikację między użytkownikami, aby ich konwersacje nie działały w stylu w jakim działa klasyczny serwer e-mail. Założenie projektu są takie, aby użytkownicy bez przeładowania strony mogli wymieniać między sobą wiadomości. Dodatkowo wszelkie powiadomienia przychodzą również bez zbędnego odświeżania witryny. Sprawia to, że witryna jest bardziej intuicyjna, łatwiejsza i szybsza w obsłudze. Taka architektura aplikacji jest przyjazna użytkownikowi, od którego będzie wymagana minimalny wysiłek w trakcie używania strony. Celem takiej aplikacji jest też maksymalne bezpieczeństwo wspierane przez beareen token i autoryzację użytkowników z zachowaniem szyfrowania danych poufnych. Architektura zapewnia, że nie będziemy otrzymywać wiadomości od niezaakceptowanych użytkowników lecz daje możliwość uczestnictwa w czatach posiadających osoby nieznajome poprzez mechanizm grup. W grupach każdy użytkownik może zaprosić swoich znajomych co może spowodować komunikację między nieznanymi w danym czacie.

1.3 Grupa docelowa

Grupą docelową są wszyscy użytkownicy którzy cenią sobie bezpieczeństwo i wygodę. Chcą szybko skomunikować się ze swoimi przyjaciółmi bądź grupą docelową bez żadnych opóźnień czy niepotrzebnych przeładowań strony. Są pewni tego, że ich dane są przechowywane w bezpiecznym miejscu i nikt nie wkradnie się na ich konto. Mogą to być zarówno firmy, które chcą komunikować się między sobą i ewentualnie z klientami poprzez utworzenie chatu dla grupy użytkowników jak i dla szkół, uniwersytetów, grup pracowników, przyjaciół czy kolegów.

2 Użyte technologie

2.1 .Net Core

Popularny, nowoczesny i wydajny framework oparty o otwartoźródłową implementację, który został wydany w 2016 roku do ogólnego przeznaczenia. Stanowi zestaw bibliotek pozwalający tworzyć wieloplatformowe aplikacje o wysokim stopniu bezpieczeństwa. Framework ten pozwala na pisanie aplikacji przeznaczonych do obliczeń chmurowych, IoT oraz jak w naszym przypadku do pisania web serwisu.



Rysunek 1: Logo .Net Core

Framework .Net Core został przez nas wybrany, ponieważ jest to nowa oraz dobrze prosperująca technologia wprowadzająca dużą dawkę świeżości podczas tworzenia nowego oprogramowania. Posiada wsparcie dla tworzenia web serwisów opartych o metodykę REST, poprzez dodanie nowych i gotowych do działania bibliotek. Platforma .Net Core jest znacznie wydajniejsza od .Net Framework. Wprowadza znaczące usprawnienia przekładające się na szybkość działania pisanych programów.

2.2 Entity Framework

2.3 SignalR

2.4 MySql

2.5 Angular

3 Projekt

3.1 Struktura projektu

3.2 Use Cases

3.3 Struktura bazy danych

3.4 Komunikacja z web servicem

4 Implementacja

4.1 Diagram klas

4.2 Modele bazy danych

4.3 Modele DTO

4.4 Implementacja kontrolerów

4.5 Problemy i ich rozwiązania

5 Testy

5.1 Testy jednostkowe

5.2 Testy integracyjne

6 Opis wkładu własnego w realizację projektu

6.1 Stworzenie i konfiguracja bazy danych

6.2 Stworzenie systemu logowania i rejestracji

6.3 Autoryzacja i autentykacja użytkowników

6.4 Konfiguracja środowiska i serwera

6.5 Strona internetowa opracowana w Angular

6.6 Implementacja zarządzania wiadomościami

6.7 Implementacja zarządzania powiadomieniami

6.8 Implementacja zarządzania chatami

6.9 Obsługa SignalR

7 Podsumowanie

7.1 Wnioski

7.2 Do zrealizowania przy dalszym rozwoju