



**WYŻSZA SZKOŁA
INFORMATYKI i ZARZĄDZANIA**
z siedzibą w Rzeszowie

KOLEGIUM INFORMATYKI STOSOWANEJ

Kierunek: INFORMATYKA

Specjalność: PROGRAMOWANIE

Piotr Dawidowski w68299

Wojciech Jurasz w67239

Kacper Żaba w67277

Pogodowy asystent ubioru

Prowadzący: Dr inż. Mirosław Hajder

PROJEKT ZESPOŁOWY

Rzeszów 2025

Spis treści

1	Opis realizowanego zadania	4
1.1	Cel projektu	4
1.2	Motywacja	4
1.3	Uzasadnienie aktualności, oryginalności i użyteczności:	4
1.4	Dlaczego nasze rozwiązanie jest lepsze	4
2	Instrukcja uruchomienia aplikacji	5
3	Opis architektury i budowy programu	6
3.1	Ogólna koncepcja	6
3.2	Struktura folderów i komponentów	6
3.3	Dane	6
3.4	Architektura systemu	7
4	Szczegółowy opis funkcjonalności i działania programu z perspektywy użytkownika	9
4.1	Interfejs użytkownika	9
4.1.1	Ekran rejestracji	9
4.1.2	Ekran logowania	10
4.1.3	Ekran zarządzania profilem	11
4.1.4	Ekran aplikacji	12
5	Przykłady użycia i demonstracja działania aplikacji	14
5.1	Widok po uruchomieniu aplikacji	14
5.2	Widok po zalogowaniu	15
5.3	Zakładka „Asystent”	16
6	Opis organizacji pracy zespołu	18
6.1	Backend	18
6.2	Frontend	18
6.3	Dokumentacja	19
7	Podsumowanie i wnioski	20
7.1	Podsumowanie osiągniętych rezultatów	20
7.2	Najważniejsze wnioski	20
7.3	Potencjalne ograniczenia	20
7.4	Cechy szczególne i innowacyjne aspekty produktu	21
7.5	Sugestie dotyczące dalszego rozwoju	21
	Bibliografia	22

Rozdział 1

Opis realizowanego zadania

1.1 Cel projektu

Celem projektu jest stworzenie aplikacji pogodowego asystenta ubioru, który na podstawie prognozy pogody oraz preferencji użytkownika, dostarcza rekomendacje dotyczące odpowiedniego ubioru. Aplikacja zbiera dane pogodowe z różnych źródeł i dostarcza użytkownikowi spersonalizowane wskazówki dotyczące codziennego ubioru, uwzględniając zarówno temperaturę, jak i inne czynniki, takie jak wilgotność czy warunki atmosferyczne.

1.2 Motywacja

Współczesny styl życia sprawia, że użytkownicy często nie mają czasu, by sprawdzić prognozy pogodowe i zaplanować swój ubiór odpowiednio do warunków atmosferycznych. Często wynika to z braku czasu na planowanie, niepewności w prognozach pogody lub braku dostosowanych narzędzi, które mogłyby pomóc w codziennym wyborze ubioru. Aplikacja rozwiązuje ten problem, automatycznie dostarczając spersonalizowane rekomendacje.

1.3 Uzasadnienie aktualności, oryginalności i użyteczności:

Dzięki wykorzystaniu technologii takich jak API pogodowe, aplikacja jest w stanie dostarczyć najbardziej aktualne informacje o pogodzie, które są natychmiastowo przekształcane na rekomendacje odzieżowe. Unikalność rozwiązania polega na połączeniu prognoz pogody z analizą preferencji użytkownika, co sprawia, że aplikacja jest bardziej spersonalizowana w porównaniu do standardowych aplikacji pogodowych. Dodatkowo, jej użyteczność polega na dostarczeniu rekomendacji w codziennych sytuacjach, kiedy użytkownicy najczęściej podejmują decyzje dotyczące ubioru. Aplikacja ma potencjał do szerokiego zastosowania, szczególnie w kontekście zwiększonego zainteresowania zdrowiem, komfortem i efektywnym zarządzaniem czasem w codziennym życiu.

1.4 Dlaczego nasze rozwiązanie jest lepsze

Aplikacja oferuje bardziej spersonalizowaną funkcjonalność, w której użytkownik nie tylko otrzymuje prognozę pogody, ale także rekomendacje odzieżowe dostosowane do jego preferencji i aktualnych warunków atmosferycznych. Zastosowanie funkcji dostosowywania do indywidualnych potrzeb oraz dynamiczne uczenie się na podstawie historii preferencji użytkownika wyróżnia aplikację na tle istniejących rozwiązań. Dzięki wykorzystaniu nowoczesnych technologii API, aplikacja jest również szybka i niezawodna w dostarczaniu danych w czasie rzeczywistym.

Rozdział 2

Instrukcja uruchomienia aplikacji

Aby uruchomić aplikację Blazor lokalnie, należy wykonać następujące kroki:

1. **Pobierz projekt** z repozytorium Git lub sklonuj go poleceniem:

```
git clone https://github.com/wjurasz/Pogodowy-asystent-ubioru
```

2. **Otwórz projekt** w środowisku programistycznym, np. Visual Studio lub Visual Studio Code.

3. **Upewnij się, że zainstalowane są wymagane pakiety NuGet.** Można to zrobić w terminalu:

```
dotnet restore
```

4. **Uzupełnij plik `appsettings.json`** odpowiednimi kluczami API, np.:

```
"WeatherApi": {  
  "BaseUrl": "https://api.weatherapi.com/v1",  
  "ApiKey": "4661cd2064114358ad7223220250102"  
},  
"GroqApi": {  
  "ApiKey": "gsk_b97wus7N6eV3qknmkHsbWGdyb3FYgoVE  
tstvRojLzFl1OYMMpiAsq"  
}
```

5. **Wykonaj migracje bazy danych Entity Framework (jeśli dotyczy).**

(a) Otwórz terminal w katalogu głównym projektu.

(b) Wpisz polecenie:

```
dotnet ef database update
```

(c) Upewnij się, że połączenie do bazy danych w `appsettings.json` jest poprawne.

6. **Uruchom aplikację lokalnie:**

```
dotnet watch
```

7. **Korzystaj z funkcjonalności aplikacji:**

- Podaj imię, preferencje pogodowe, środek transportu i nazwę miasta.
- Zapytaj chat, aby otrzymać rekomendacje.

Rozdział 3

Opis architektury i budowy programu

3.1 Ogólna koncepcja

System „Asystent pogodowy ubioru” to aplikacja wspierająca użytkownika w doborze odzieży na podstawie prognozy pogody oraz osobistych preferencji temperaturowych. Aplikacja pozwala użytkownikowi na rejestrację i logowanie, podanie miasta, w którym przebywa, oraz informacji o preferowanej temperaturze. Na tej podstawie generowana jest rekomendacja ubioru dopasowana do aktualnych warunków pogodowych.

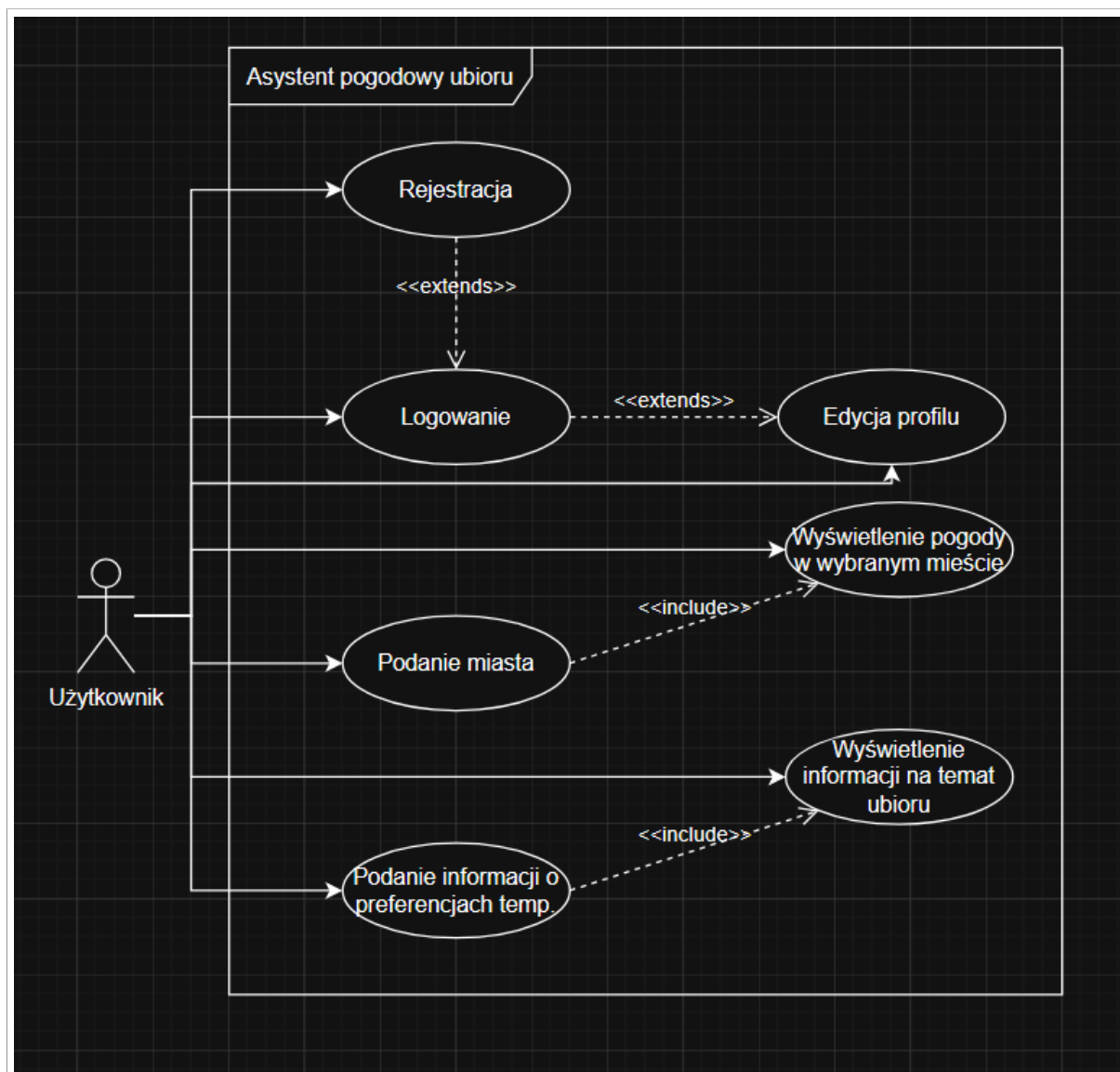
3.2 Struktura folderów i komponentów

- **Components/Account** – komponenty odpowiedzialne za logikę uwierzytelniania użytkownika, integracja z systemem tożsamości ASP.NET.
- **Pages** – główne strony aplikacji, w tym formularz pogodowy.
- **Data/WeatherService.cs** – klasa pobierająca dane pogodowe z zewnętrznego API.
- **GroqApiClient.cs** – klient odpowiedzialny za wysyłanie zapytań do modelu językowego (LLM) i otrzymywanie rekomendacji.
- **ChatMessage.cs** – klasa reprezentująca wiadomości kontekstowe do i od LLM.

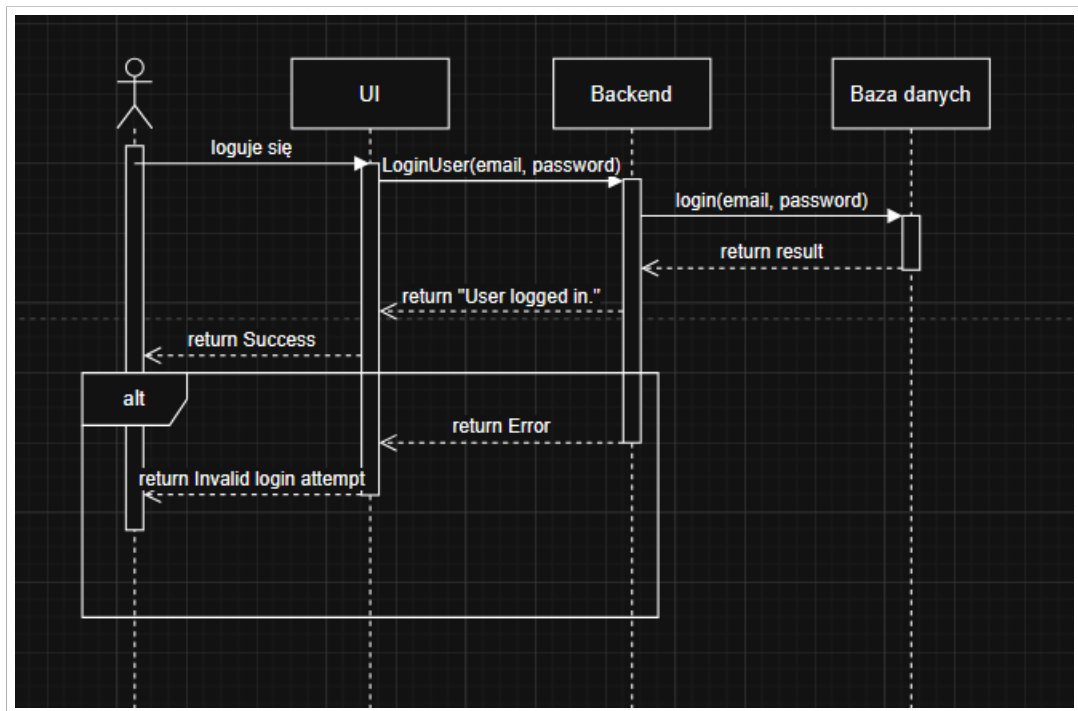
3.3 Dane

System wykorzystuje zewnętrzne API pogodowe (np. WeatherAPI) do pobierania aktualnej pogody. Uzyskane dane są przekazywane jako kontekst do modelu językowego, który – na podstawie zadanych promptów – generuje naturalnie brzmiącą rekomendację ubioru. Dane użytkownika (miasto, preferencje) są przekazywane z formularza w interfejsie i obsługiwane przez komponenty Blazor.

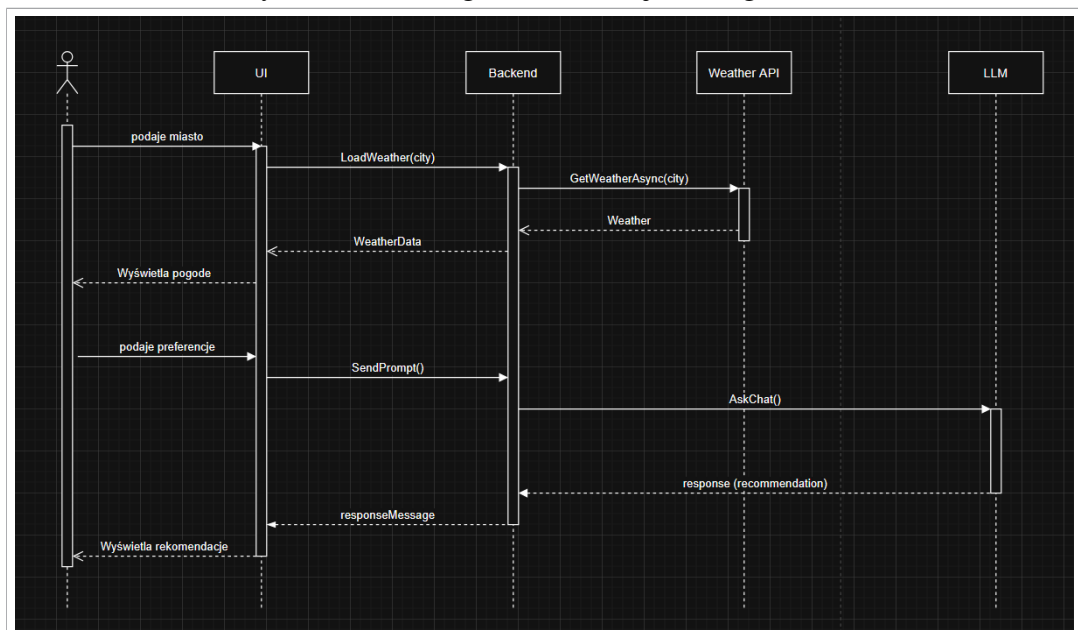
3.4 Architektura systemu



Rysunek 3.1: Diagram przypadków użycia



Rysunek 3.2: Diagram sekwencji dla logowania



Rysunek 3.3: Diagram sekwencji dla pogody oraz LLM

Rozdział 4

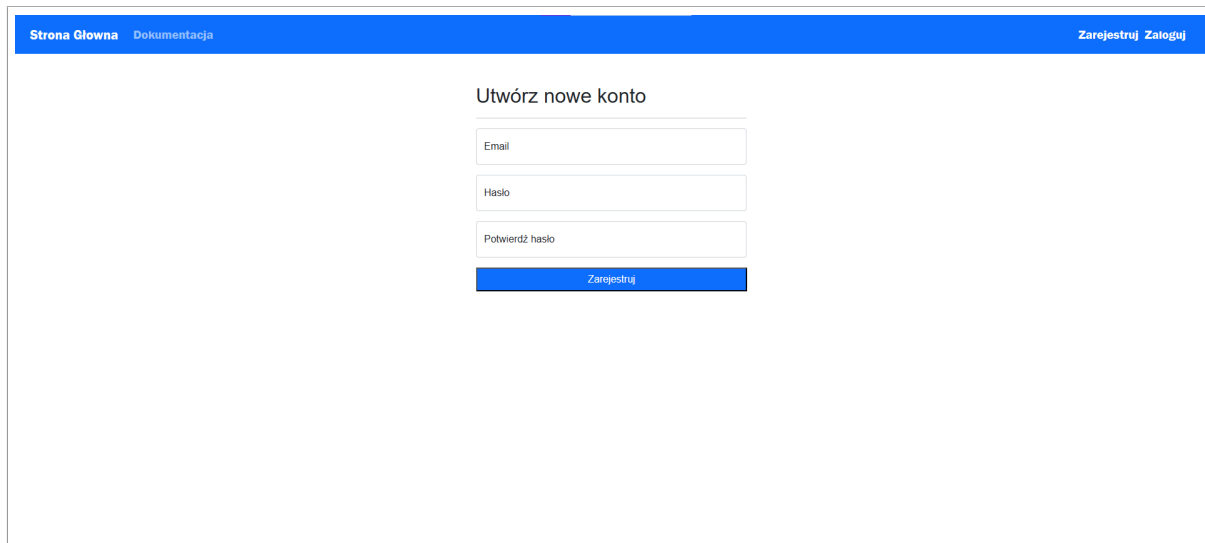
Szczegółowy opis funkcjonalności i działania programu z perspektywy użytkownika

4.1 Interfejs użytkownika

W tym punkcie szczegółowo opisano ekrany, na które może trafić użytkownik w trakcie korzystania z asystenta. Uwzględniono wszystkie etapy korzystania z aplikacji, od utworzenia konta, przez korzystanie z asystenta aż po edycję profilu.

4.1.1 Ekran rejestracji

"Asystent pogodowy ubioru" jest wyposażony w moduł logowania użytkowników, co zabezpiecza przed nieautoryzowanym korzystaniem z aplikacji. Ekran rejestracji użytkownika prezentuje się następująco:



Strona Główna Dokumentacja Zarejestruj Zaloguj

Utwórz nowe konto

Email

Hasło

Potwierdź hasło

Zarejestruj

Rysunek 4.1: Ekran rejestracji

Jeżeli rejestracja się nie powiedzie, użytkownik zostanie poinformowany co błędnie wypełnił:

The screenshot shows a web page titled "Utwórz nowe konto" (Create new account). The page has a blue header with "Strona Główna" and "Dokumentacja" on the left, and "Zarejestruj" and "Zaloguj" on the right. The main content area contains a form with three input fields: "Email", "Hasło", and "Potwierdź hasło". Each field has a red border and a red error message below it. The "Email" field has the message "Nieprawidłowy adres email." The "Hasło" field has the message "Hasło musi mieć co najmniej 6 i maksymalnie 100 znaków." The "Potwierdź hasło" field has the message "Hasło i potwierdzenie nie są zgodne." There is a blue "Zarejestruj" button at the bottom of the form.

Rysunek 4.2: Błąd przy rejestracji

Jeżeli rejestracja się powiedzie, konto zostanie utworzone.

The screenshot shows a web page titled "Potwierdzenie rejestracji" (Registration confirmation). The page has a blue header with "Strona Główna" and "Dokumentacja" on the left, and "Zarejestruj" and "Zaloguj" on the right. The main content area contains a green button labeled "Potwierdź konto". Below the button, there is a small text: "Kliknij przycisk aby potwierdzić rejestrację konta".

Rysunek 4.3: Potwierdzenie rejestracji

4.1.2 Ekran logowania

Jeżeli użytkownik posiada już konto, może się na nie zalogować. :

The screenshot shows a web page titled "Zaloguj się" (Log in). The page has a blue header with "Strona Główna" and "Dokumentacja" on the left, and "Zarejestruj" and "Zaloguj" on the right. The main content area contains a form with two input fields: "Email" and "Hasło". Below the "Hasło" field, there is a checkbox labeled "Zapamiętaj mnie". There is a blue "Zaloguj" button at the bottom of the form. Below the button, there is a link "Zarejestruj nowe konto".

Rysunek 4.4: Ekran logowania

Jeżeli logowanie się nie powiedzie, otrzyma on stosowną informację:

The screenshot shows a web application interface with a blue header bar. On the left, there are links: "Strona Główna" and "Dokumentacja". On the right, there are links: "Zarejestruj" and "Zaloguj". The main content area has a light green error message box at the top that says "Błąd. Nieprawidłowa próba logowania." Below this is a heading "Zaloguj się". There are two input fields: "Email" with the value "test123@gmail.com" and "Hasło" with masked characters "*****". Below the password field is a checkbox labeled "Zapamiętaj mnie". At the bottom of the form is a blue button labeled "Zaloguj". Below the button is a link "Zarejestruj nowe konto".

Rysunek 4.5: Błąd przy logowaniu

4.1.3 Ekran zarządzania profilem

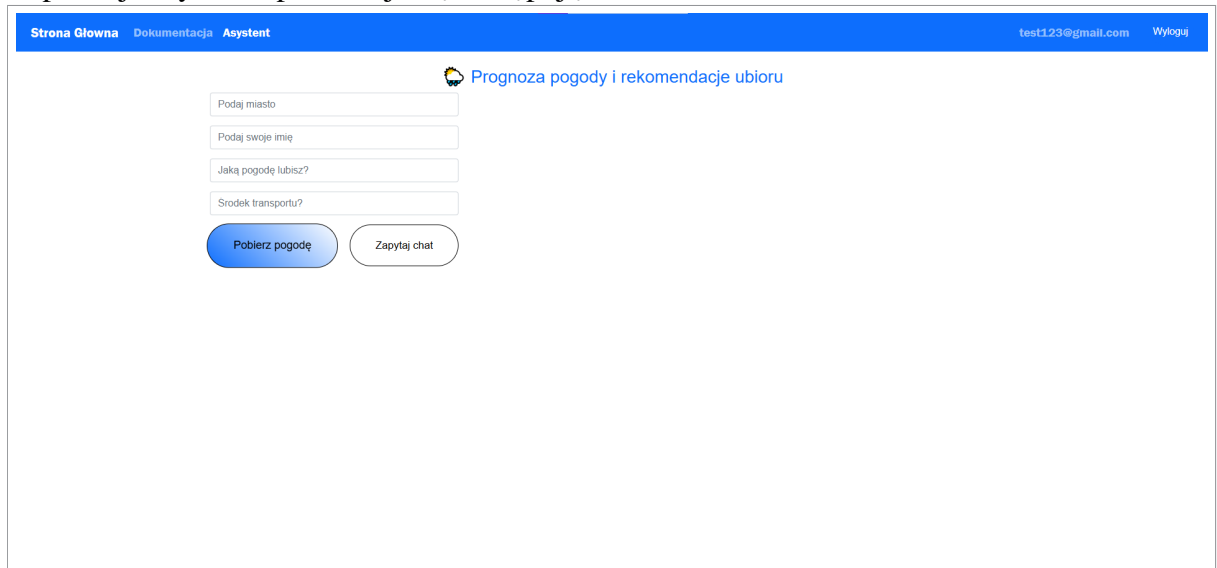
Zalogowany użytkownik w każdej chwili może dodać, zedytować lub całkowicie usunąć swoje dane lub profil z panelu:

The screenshot shows a web application interface with a blue header bar. On the left, there are links: "Strona Główna", "Dokumentacja", and "Asystent". On the right, there are links: "test123@gmail.com" and "Wyloguj". The main content area has a heading "Zarządzanie kontem" and a sub-heading "Zmiana danych". There is a sidebar with a menu: "Profil" (highlighted in blue), "Email", "Hasło", "Autentykacja dwuetapowa", and "Informacje". The main content area has a form titled "Profil". It contains two input fields: "Nazwa użytkownika" with the value "test123@gmail.com" and "Numer telefonu". At the bottom of the form is a blue button labeled "Zapisz".

Rysunek 4.6: Edycja profilu

4.1.4 Ekran aplikacji

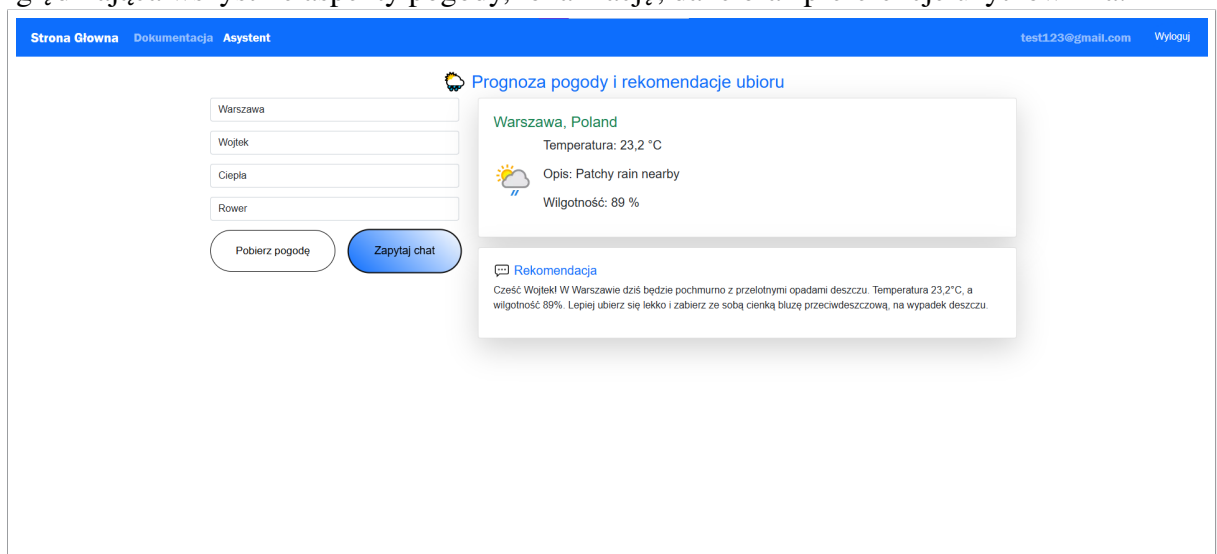
Sama aplikacja asystenta prezentuje się następująco:



The screenshot shows the 'Asystent' page of the application. The header is blue with links to 'Strona Główna', 'Dokumentacja', and 'Asystent'. On the right, it shows a user email 'test123@gmail.com' and a 'Wyloguj' button. The main content area has a title 'Prognoza pogody i rekomendacje ubioru' with a weather icon. Below the title are four input fields: 'Podaj miasto', 'Podaj swoje imię', 'Jaką pogodę lubisz?', and 'Środek transportu?'. At the bottom of these fields are two buttons: 'Pobierz pogodę' (highlighted in blue) and 'Zapytaj chat'.

Rysunek 4.7: Aplikacja asystent

Użytkownik może wpisać nazwę dowolnego miasta, i jeżeli jest ono obsługiwane przez API, to do użytkownika zostanie zwrócona informacja z obecną pogodą w danym mieście. Kiedy poda resztę swoich danych i kliknie przycisk "Zapytaj chat", zostanie mu zwrócona odpowiedź od podłączonego LLM-a, uwzględniająca wszystkie aspekty pogody, lokalizację, dane oraz preferencje użytkownika.



This screenshot shows the same interface as Figure 4.7, but with data entered in the input fields: 'Warszawa', 'Wojtek', 'Ciepła', and 'Rower'. The 'Zapytaj chat' button is now highlighted in blue. A modal window is displayed on the right side of the screen, showing the results of the chat query. The modal has a title 'Warszawa, Poland' and contains the following information: 'Temperatura: 23,2 °C', 'Opis: Patchy rain nearby' (with a sun and cloud icon), and 'Wilgotność: 89 %'. Below this, there is a section titled 'Rekomendacja' with a speech bubble icon, containing the text: 'Cześć Wojtek! W Warszawie dziś będzie pochmurno z przelotnymi opadami deszczu. Temperatura 23,2°C, a wilgotność 89%. Lepiej ubierz się lekko i zabierz ze sobą cienką bluzę przeciwdeszczową, na wypadek deszczu.'

Rysunek 4.8: Aplikacja asystent po odpowiedzi

Po kliknięciu przycisku „Pobierz pogodę” aplikacja sprawdza, czy wszystkie pola są uzupełnione. Jeśli nie, pojawia się komunikat: „Uzupełnij wszystkie pola.”

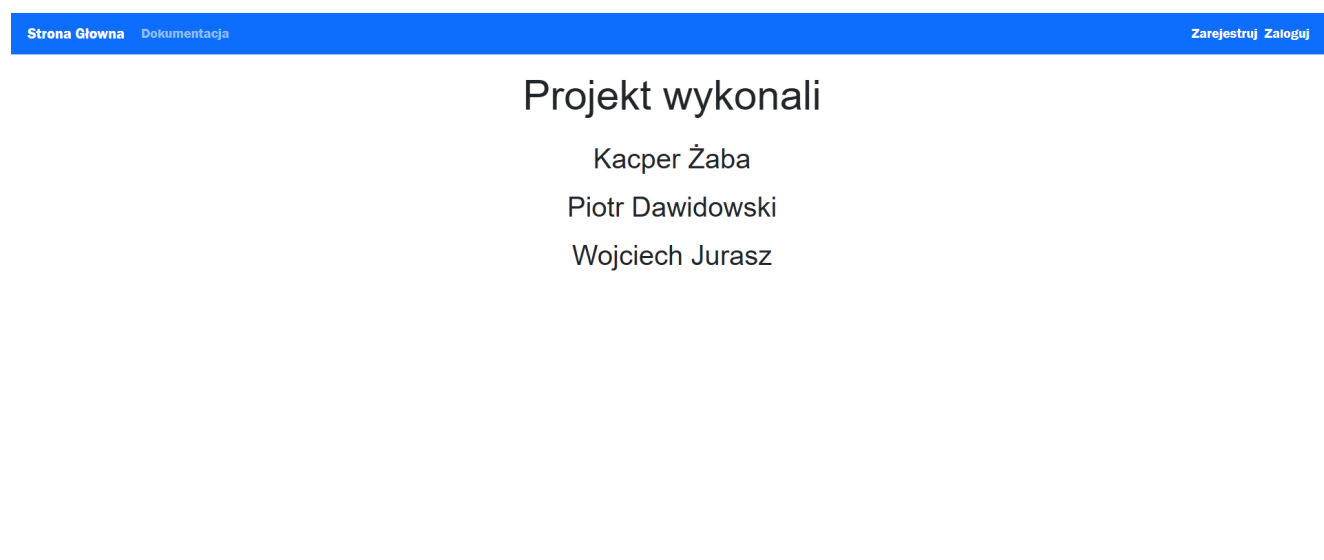
The screenshot shows a web application interface with a blue header bar containing the text "Strona Główna", "Dokumentacja", and "Asystent". On the right side of the header, the email "test123@gmail.com" and a "Wyloguj" link are visible. The main content area has a title "Prognoza pogody i rekomendacje ubioru" with a weather icon. Below the title, there are four input fields: "Podaj miasto", "Podaj swoje imię", "Jaka pogodę lubisz?", and "Środek transportu?". At the bottom of these fields are two buttons: "Pobierz pogodę" and "Zapytaj chat". A white modal box is displayed over the input fields, containing the text "Rekomendacja" with a speech bubble icon and "Uzupełnij wszystkie pola." with a red 'X' icon.

Rysunek 4.9: Aplikacja asystent po odpowiedzi

Rozdział 5

Przykłady użycia i demonstracja działania aplikacji

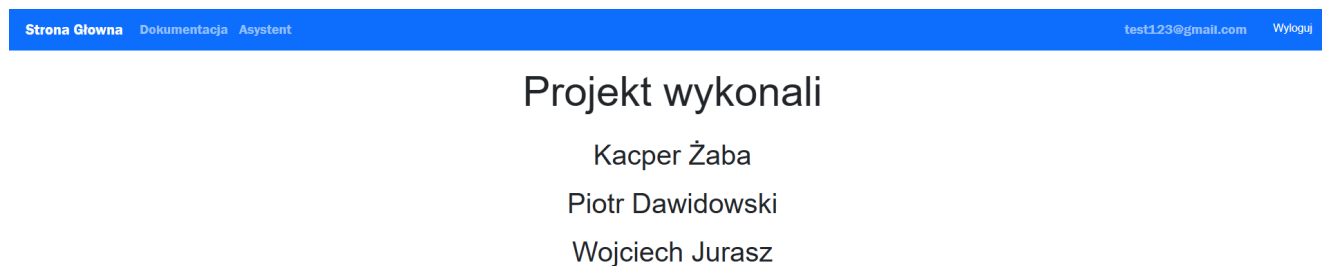
5.1 Widok po uruchomieniu aplikacji



Rysunek 5.1: Strona główna widoczna dla niezalogowanego użytkownika.

Po uruchomieniu aplikacji użytkownik zostaje przeniesiony na stronę główną. Aby uzyskać dostęp do funkcjonalności asystenta pogodowego, konieczne jest zalogowanie się na istniejące konto lub założenie nowego.

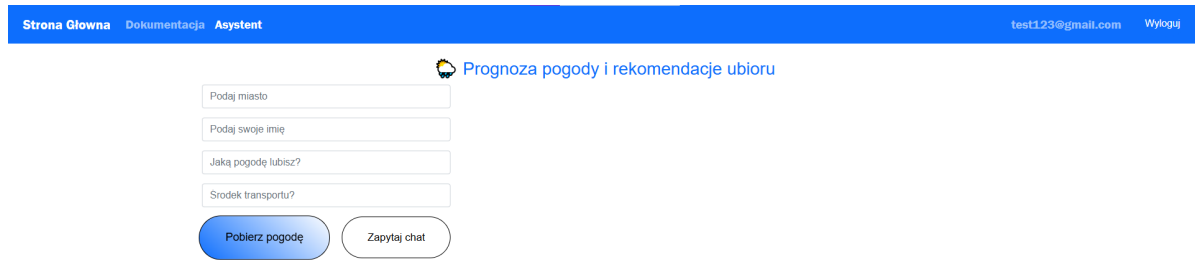
5.2 Widok po zalogowaniu



Rysunek 5.2: Strona główna widoczna po zalogowaniu.

Po zalogowaniu użytkownik uzyskuje dostęp do dodatkowych funkcji, w tym zakładki **Asystent**, która umożliwia uzyskanie prognozy pogody oraz rekomendacji dotyczących ubioru.

5.3 Zakładka „Asystent”



Strona Główna Dokumentacja Asystent test123@gmail.com Wyloguj

Prognoza pogody i rekomendacje ubioru

Podaj miasto

Podaj swoje imię

Jaką pogodę lubisz?

Środek transportu?

Pobierz pogodę Zapytaj chat

Rysunek 5.3: Widok zakładki **Asystent**.

Zakładka **Asystent** umożliwia interakcję z systemem w dwojaki sposób:

- Użytkownik może podać jedynie nazwę miasta, aby uzyskać aktualną prognozę pogody.
- Wprowadzając dodatkowe dane, takie jak imię, preferencje pogodowe i środek transportu, użytkownik otrzyma spersonalizowaną rekomendację dotyczącą ubioru.

Strona Główna

Dokumentacja

Asystent

test123@gmail.com Wyloguj

Rzeszów

Podaj swoje imię

Jaką pogodę lubisz?

Środek transportu?

Pobierz pogodę

Zapytaj chat

Prognoza pogody i rekomendacje ubioru

Rzeszów, Poland

Temperatura: 26,3 °C

☀️

Opis: Sunny

Wilgotność: 58 %

Rysunek 5.4: Prezentacja wyników działania asystenta pogodowego.

Strona Główna

Dokumentacja

Asystent

test123@gmail.com Wyloguj

Warszawa

Wojtek

Ciepła

Rower

Pobierz pogodę

Zapytaj chat

Prognoza pogody i rekomendacje ubioru

Warszawa, Poland

Temperatura: 23,2 °C

☁️

Opis: Patchy rain nearby

Wilgotność: 89 %

Rekomendacja

Cześć Wojtek! W Warszawie dziś będzie pochmurno z przelotnymi opadami deszczu. Temperatura 23,2°C, a wilgotność 89%. Lepiej ubierz się lekko i zabierz ze sobą cienką bluzę przeciwdeszczową, na wypadek deszczu.

Rysunek 5.5: Prezentacja wyników działania asystenta pogodowego.

17

Rozdział 6

Opis organizacji pracy zespołu

6.1 Backend

- **Integracja z API pogodowym:** Wojciech Jurasz skoncentrował się na implementacji serwisu WeatherService, który pobierał dane pogodowe z zewnętrznego API. Używał do tego HttpClient oraz konfiguracji API w pliku konfiguracyjnym, by dynamicznie łączyć się z odpowiednim serwisem i przetwarzać odpowiedzi w postaci danych o pogodzie (temperatura, wilgotność, warunki).
- **Integracja z API Groq:** Kacper Żaba pracował nad klientem GroqApiClient, który łączył aplikację z zewnętrznymi narzędziami analitycznymi, takimi jak transkrypcje i tłumaczenia audio, umożliwiając rozszerzenie funkcjonalności aplikacji o bardziej zaawansowane funkcje przetwarzania tekstu i dźwięku.
- **Zarządzanie danymi o pogodzie i komunikatach:** Piotr Dawidowski tworzył klasy odpowiedzialne za strukturę danych, takie jak WeatherData, CurrentWeather, Location, i Condition, umożliwiając łatwe przechowywanie i manipulowanie informacjami o pogodzie oraz komunikatami użytkowników za pomocą klasy ChatMessage.

6.2 Frontend

- **Interfejs użytkownika:** Każdy członek zespołu pracował nad frontendem aplikacji, tworząc interaktywne formularze do wprowadzania danych przez użytkownika, takie jak miasto, preferencje dotyczące pogody, oraz środek transportu. Frontend umożliwia użytkownikowi szybkie pobieranie prognoz pogody i rekomendacji odzieżowych, a także wyświetlanie odpowiednich komunikatów o błędach lub statusie ładowania.
- **Wykorzystanie danych API:** Zespół wykorzystywał dane z WeatherService do generowania prognoz pogody w czasie rzeczywistym, oraz integracji z komunikatami dostosowanymi do preferencji użytkownika, tworząc elastyczny system rekomendacji ubioru.

6.3 Dokumentacja

- **Opis architektury systemu i opis organizacji pracy zespołu:** Piotr Dawidowski uczestniczył w opisaniu ogólnej architektury projektu, uwzględniając role i odpowiedzialności poszczególnych komponentów backendowych i frontendowych. Przygotował również szczegółowy opis organizacji pracy zespołu, w tym podziału obowiązków oraz metodologii pracy.
- **Opis funkcjonalności oraz wnioski:** Kacper Żaba napisał szczegółowy opis funkcjonalności i działania programu z perspektywy użytkownika. Skupił się na tym, jak aplikacja działa na poziomie interfejsu użytkownika, omawiając kluczowe interakcje, takie jak wprowadzanie danych, pobieranie prognozy pogody oraz rekomendacje odzieżowe.
- **Przykłady użycia i demonstracja wyników działania oraz podsumowanie:** Wojciech Jurasz przygotował przykłady użycia aplikacji, prezentując różne scenariusze użytkownika, w tym konfigurację, sposób korzystania z aplikacji oraz interpretację wyników. Wojciech również opracował podsumowanie całego projektu, uwzględniając osiągnięte cele, napotkane trudności oraz rekomendacje na przyszłość.

Rozdział 7

Podsumowanie i wnioski

7.1 Podsumowanie osiągniętych rezultatów

W ramach projektu "Pogodowy asystent ubioru" zrealizowano kluczowe cele dotyczące stworzenia aplikacji pogodowej, która dostarcza użytkownikowi prognozy pogody oraz rekomendacje ubioru w zależności od preferencji i warunków atmosferycznych. Zespół skutecznie zintegrował zewnętrzne API pogodowe oraz Large Language Model. Projekt obejmował zarówno frontend (w technologii Blazor), jak i backend, który odpowiedzialny był za pobieranie danych pogodowych oraz przetwarzanie komunikatów użytkowników.

7.2 Najważniejsze wnioski

- **Techniczne:** Zespół nauczył się skutecznej integracji z zewnętrznymi API, co wymagało opanowania wielu aspektów komunikacji z serwisami webowymi. Implementacja WeatherService i GroqApiClient wykazała znaczenie dobrego zarządzania błędami oraz odpowiednim formatowaniem danych przy pracy z zewnętrznymi źródłami informacji.
- **Proces projektowy:** Praca zespołowa opierała się na jasnym podziale obowiązków, co pozwoliło na sprawną realizację zadań, mimo iż każdy członek zespołu angażował się zarówno w backend, frontend, jak i dokumentację. Ustalony proces pracy umożliwił efektywne wykorzystanie czasu i zasobów.
- **Zespołowe:** Kluczowym aspektem współpracy było regularne komunikowanie się i wzajemne wspieranie się przy rozwiązywaniu trudności technicznych. Każdy członek zespołu wniósł istotny wkład do projektu, zarówno w zakresie kodu, jak i dokumentacji.

7.3 Potencjalne ograniczenia

- **Zależność od zewnętrznych API:** Projekt w dużej mierze opierał się na zewnętrznych źródłach danych, co stwarza ryzyko związane z ich niedostępnością lub zmianą polityki dostępu. W przyszłości warto rozważyć implementację lokalnych źródeł danych lub mechanizmów cache'ujących.
- **Skalowalność:** Aplikacja została zaprojektowana na początkową wersję z podstawową funkcjonalnością, jednak w miarę wzrostu liczby użytkowników, może wystąpić potrzeba optymalizacji systemu backendowego i rozważenia architektury chmurowej.
- **Ograniczenia w personalizacji:** Choć aplikacja oferuje podstawowe rekomendacje ubioru, personalizacja tych rekomendacji mogłaby zostać rozszerzona o dane użytkowników (np. historia preferencji pogodowych) i integrację z bardziej zaawansowanymi modelami predykcji.

7.4 Cechy szczególne i innowacyjne aspekty produktu

Aplikacja wyróżnia się unikalnym podejściem do integracji prognoz pogody z rekomendacjami odzieżowymi, dostosowanymi do indywidualnych preferencji użytkownika. Dzięki temu, użytkownicy nie tylko uzyskują informację o prognozie pogody, ale także konkretne sugestie dotyczące ubioru, co stanowi innowację w porównaniu do tradycyjnych aplikacji pogodowych dostępnych na rynku.

7.5 Sugestie dotyczące dalszego rozwoju

- **Rozszerzenie personalizacji:** Można rozważyć wprowadzenie algorytmów uczenia maszynowego do analizy preferencji użytkowników i prognozowania ich potrzeb na podstawie historii danych.
- **Rozbudowa frontendu:** W przyszłości aplikacja mogłaby zostać rozbudowana o dodatkowe widoki, jak np. interaktywne mapy pogody, wykresy i analizy długoterminowe.

Podsumowując, projekt stanowi solidną bazę dla dalszego rozwoju aplikacji, której przyszłość zależy od dalszej personalizacji oraz wprowadzenia bardziej zaawansowanych technologii predykcyjnych i integracyjnych.

Bibliografia

- [1] Wikipedia, *Weather API Documentation*, dostępne na: <https://www.weatherapi.com/docs/>
- [2] Groq, *Groq API Documentation*, dostępne na: <https://www.groq.com/api>
- [3] Microsoft, *Blazor Documentation*, dostępne na: <https://learn.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-6.0>
- [4] OpenWeather, *Current Weather Data API*, dostępne na: <https://openweathermap.org/api>
- [5] Mozilla Developer Network, *JSON*, dostępne na: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>
- [6] Groq, *Groq API for Speech Transcription*, dostępne na: <https://www.groq.com/api/audio-transcription>
- [7] Fielding, R. T., *Architectural Styles and the Design of Network-based Software Architectures*, 2000.
- [8] Gamma, E., Helm, R., Johnson, R., Vlissides, J. M., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1994.
- [9] Martin, R. C., *Clean Architecture: A Craftsman's Guide to Software Structure and Design*, Prentice Hall, 2017.
- [10] Newman, S., *Building Microservices: Designing Fine-Grained Systems*, O'Reilly Media, 2015.
- [11] Turnbull, J., *The Docker Book: Containerization is the new virtualization*, James Turnbull, 2014.
- [12] Nakamoto, S., *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008, dostępne na: <https://bitcoin.org/bitcoin.pdf>