

Politechnika Wrocławска
Wydział Informatyki i Telekomunikacji

Kierunek: Informatyka Techniczna (ITE)
Specjalność: Inżynieria systemów informatycznych (INS)

**PRACA DYPLOMOWA
INŻYNIERSKA**

**Zastosowanie języka Python do realizacji bazodanowej
aplikacji webowej wspomagającej obsługę stacji
meteorologicznej**

Opiekun pracy
dr inż. Robert Wójcik, K30W04ND03

Słowa kluczowe: aplikacja webowa, stacja meteorologiczna, MySQL, Python, Flask

WROCŁAW, 2024

Geometric Complexity Theory and Algebraic Complexity Theory

Manuel Kauers
Technische Universität Wien

$$\text{Complexity} = \text{Efficiency}$$

Complexity theory studies the computational complexity of problems.
The goal is to find efficient algorithms.

$$\text{Efficiency} = \text{Time} + \text{Space}$$

$$\text{Time} = \Theta(n^k)$$

$$\text{Space} = \Theta(n^k)$$

$$\text{Efficiency} = \Theta(n^k)$$

$$\text{Efficiency} = \Theta(n^k)$$

$$\text{Efficiency} = \Theta(n^k)$$

$$\text{Efficiency} = \Theta(n^k)$$

Streszczenie

W pracy dyplomowej przedstawiono projekt oraz implementację systemu webowego, który realizuje funkcjonalności prostej aplikacji pogodowej, wspomagającej zarządzanie danymi uzyskanymi z wybranych stacji meteorologicznych, tj. internetowych serwisów meteorologicznych. Aplikacja umożliwia wybór lokalizacji serwisu z danymi, komunikację z serwisem z wykorzystaniem interfejsu REST API, a także przetwarzanie i przechowywanie informacji uzyskanych z serwisu w bazie danych. System zrealizowano przy użyciu języka Python, mikroframeworka Flask, bazy danych MySQL oraz środowiska wirtualizacji Docker. W pracy zawarto wymagania funkcjonalne i niefunkcjonalne aplikacji, a także wykorzystane technologie i narzędzia. W części projektowej przedstawiono analizę struktur danych, diagram związków encji, model fizyczny bazy danych oraz opis architektury systemu i interfejsu użytkownika. Zawarto również opis mechanizmów bezpieczeństwa aplikacji. Implementacja obejmuje kluczowe funkcjonalności systemu: dodawanie, usuwanie, modyfikowanie, analizę i prezentację danych meteorologicznych uzyskanych z określonego serwisu, a także walidację i filtrację danych. Przeprowadzono również testy jednostkowe, funkcjonalne oraz integracyjne aplikacji, które potwierdziły poprawność jej działania. W podsumowaniu zaprezentowano osiągnięcia pracy oraz propozycje dalszego rozwoju zrealizowanego systemu.

Abstract

The thesis presents the design and implementation of a web-based system that implements the functionalities of a simple weather application to support the management of data obtained from selected weather stations, i.e., online weather services. The application allows for the selection of the location of the data service, communication with the service using the REST API, and processing and storing the information obtained from the service in a database. The system was realized using the Python language, the Flask microframework, the MySQL database, and the Docker virtualization environment. The paper includes the functional and non-functional requirements of the application, as well as the technologies and tools used. The design section presents an analysis of data structures, an entity relationship diagram, a physical database model, and a description of the system architecture and user interface. A description of the application's security mechanisms is also included. The implementation covers the key functionalities of the system: adding, deleting, modifying, analyzing, and presenting meteorological data obtained from a specific service, as well as validating and filtering the data. Unit, functional, and integration tests of the application were also carried out, confirming its operation's correctness. In conclusion, the achievements of the work and proposals for further development of the realized system were presented.

Spis treści

1. Wstęp	8
1.1. Cel pracy	8
1.2. Zakres i układ pracy	9
2. Analiza wymagań systemu.....	11
2.1. Przegląd wybranych API i aplikacji pogodowych	12
2.1.1. API pogodowe	12
2.1.2. PWSMonitor.com	13
2.1.3. OpenWeather	15
2.2. Wnioski dotyczące koncepcji działania własnej aplikacji	16
2.3. Opis działania i schemat architektury systemu	17
2.4. Wymagania funkcjonalne	18
2.4.1. Diagram przypadków użycia	19
2.4.2. Scenariusze przypadków użycia	20
2.5. Wymagania niefunkcjonalne	21
2.6. Wykorzystywane technologie	22
2.6.1. Frontend	23
2.6.2. Backend	23
2.6.3. Środowisko wirtualizacji projektu	24
2.7. Używane narzędzia	25
2.8. Integracja z zewnętrznymi API	26
3. Projekt bazy danych.....	27
3.1. Analiza rzeczywistości	28
3.2. Diagram konceptualny ERD.....	30
3.3. Struktura tabel i schemat fizyczny	30
4. Projekt aplikacji	33
4.1. Architektura aplikacji	34
4.2. Interfejs użytkownika (frontend)	35
4.3. Struktura menu	38
4.4. Działanie strony serwerowej (backend)	39
4.5. Komunikacja z bazą danych	42
4.6. Operacje systemu	43
4.7. Mechanizmy bezpieczeństwa aplikacji	48
5. Implementacja systemu.....	52
5.1. Struktura danych i modele	52
5.2. Realizacja bazy danych	53
5.3. Wybór stacji meteorologicznych – serwisów danych	55
5.4. Dodawanie i usuwanie danych	57

5.5. Implementacja testów jednostkowych	59
6. Testowanie systemu	62
6.1. Konfigurowanie środowiska testowego.....	62
6.2. Weryfikacja systemu z użyciem testów jednostkowych	64
6.3. Automatyczne wykonywanie testów jednostkowych	67
6.4. Testy funkcjonalne	68
6.4.1. Logowanie administratora.....	70
6.4.2. Dodanie lokalizacji	71
6.4.3. Usunięcie lokalizacji	72
6.4.4. Zmiana hasła	73
6.4.5. Wybór lokalizacji i stacji	75
6.5. Dodawanie danych do bazy	77
6.6. Wnioski z testów	78
7. Podsumowanie	80
A. Instrukcja wdrożeniowa.....	82
A.1. Wymagania wstępne	82
A.2. Konfiguracja środowiska	82
A.2.1. Backend	83
A.2.2. Frontend	84
A.3. Przed pierwszym uruchomieniem	84
A.3.1. Zmienne środowiskowe	84
A.3.2. Tworzenie kont na serwisach pogodowych.....	85
A.4. Pierwsze uruchomienie.....	85
Literatura	88
Spis rysunków	90
Spis tabel.....	91
Spis listingów.....	92

1. Wstęp

Współczesna meteorologia nieustannie ewoluje, w dużej mierze dzięki zaawansowanym technologiom informatycznym, które umożliwiają precyzyjne monitorowanie, analizowanie i prognozowanie warunków pogodowych. Dane meteorologiczne gromadzone przez stacje meteorologiczne są kluczowe dla wielu sektorów, takich jak rolnictwo, transport, energetyka oraz zarządzanie kryzysowe [5]. Efektywne zarządzanie tymi danymi wymaga nowoczesnych narzędzi, które pozwalają na ich przechowywanie, przetwarzanie oraz udostępnianie w czasie rzeczywistym.

W dobie cyfryzacji i powszechnej dostępności Internetu, aplikacje webowe stają się niezwykle popularnym rozwiązaniem w różnych dziedzinach, w tym również w pogodoznawstwie. Systemy te oferują użytkownikom łatwy dostęp do danych z dowolnego miejsca na świecie, co jest nieocenione w kontekście globalnego monitorowania i analizy warunków atmosferycznych [10]. Istotne znaczenie w ich praktycznym wykorzystaniu mają takie cechy jak przejrzystość, dostępność oraz użyteczność.

Niniejsza praca dyplomowa koncentruje się na projekcie i implementacji bazodanowej aplikacji webowej realizującej funkcjonalności i obsługę prostej stacji pogodowej o nazwie „Weather App”. Aplikacja została zrealizowana przy użyciu języka Python [1] oraz mikroframeworka Flask [3], co pozwala na efektywne zarządzanie danymi meteorologicznymi. System umożliwia monitorowanie, przechowywanie oraz prezentację danych pogodowych, a także integrację z serwisami pogodowymi [11], [12] w Internecie za pomocą interfejsu REST/API [13]. W pracy przedstawiono zarówno teoretyczne podstawy projektu jak i praktyczne aspekty jego realizacji.

1.1. Cel pracy

Celem niniejszej pracy dyplomowej jest opracowanie bazodanowej aplikacji webowej dedykowanej do obsługi prostej stacji pogodowej zrealizowanej w języku Python [1], [4] z wykorzystaniem mikroframeworka Flask [3] oraz bazy danych MySQL [2], [17], [32].

Projekt ma na celu stworzenie nowoczesnego i użytecznego systemu umożliwiającego pobieranie, przetwarzanie, prezentację oraz przechowywanie danych meteorologicznych, a także ich udostępnianie za pomocą interfejsu REST/API, co pozwoli na integrację z innymi aplikacjami i platformami pogodowymi.

Głównym aspektem technicznym projektu jest efektywne wykorzystanie bazy danych MySQL, zainstalowanej w środowisku wirtualizacji Docker [19], [20], [32], umożliwiającej przechowywanie dużych ilości danych meteorologicznych w sposób uporządkowany i łatwo dostępny. Dodatkowo aplikacja zostanie wyposażona w funkcjonalności wspierające analizę oraz wizualizację danych, co pozwoli na ich intuicyjne przedstawienie użytkownikom końcowym. Kluczowym elementem systemu będzie także podział na część główną oraz administracyjną, co zapewni rozdzielenie funkcjonalności dostępnych dla użytkowników i administratorów, a tym samym zwiększy bezpieczeństwo danych poprzez wdrożenie mechanizmów uwierzytelniania i autoryzacji.

Praca uwzględnia również aspekt integracji systemu z innymi aplikacjami i serwisami pogodowymi za pośrednictwem interfejsu API. Ze względu na komunikację z zewnętrznym API aplikacja będzie wymagała stałego nadzoru administratora nad API, ponieważ ewentualne zmiany w jego strukturze mogą wpływać na integralność i sposób przechowywanych danych. Istotnym elementem projektu jest również zapewnienie

efektywnego przetwarzania danych, szczególnie w przypadku ich dużych wolumenów, co umożliwia płynne i niezawodne działanie systemu.

Podsumowując, niniejsza praca ma na celu nie tylko opracowanie funkcjonalnej aplikacji webowej dedykowanej obsłudze prostej stacji meteorologicznej, lecz także analizę i ocenę jej możliwości integracyjnych, bezpieczeństwa oraz wydajności. Realizacja tego projektu dostarczy istotnych wniosków, które mogą posłużyć jako podstawa do realizacji bardziej zaawansowanych systemów o podobnym zastosowaniu.

1.2. Zakres i układ pracy

Zakres pracy obejmuje pełny cykl projektowania, implementacji oraz testowania systemu, a także szczegółową specyfikację wymagań systemu oraz wniosków wynikających z analizy własności dostępnych rozwiązań rynkowych [21].

Celem dokumentu jest nie tylko przedstawienie teoretycznych podstaw budowy aplikacji, ale także szczegółowy opis procesu jej tworzenia, począwszy od analizy problemu, poprzez projektowanie architektury, aż po wdrożenie i testowanie.

Zakres pracy

Praca obejmuje następujące kluczowe aspekty:

- **Analiza wymagań systemu:** Badanie istniejących rozwiązań, identyfikacja potrzeb użytkowników, określenie wymagań funkcjonalnych i niefunkcjonalnych systemu.
- **Projekt bazy danych:** Opracowanie struktury bazy danych, w tym diagramów konceptualnych i fizycznych, które zapewnią spójne przechowywanie oraz efektywne przetwarzanie danych meteorologicznych.
- **Projekt aplikacji:** Opis architektury systemu, uwzględniający podział na frontend i backend oraz interakcje między komponentami aplikacji.
- **Implementacja systemu:** Szczegółowy opis kluczowych operacji, takich jak zarządzanie użytkownikami, obsługa danych meteorologicznych, integracja z zewnętrznymi API oraz implementacja mechanizmów bezpieczeństwa.
- **Testowanie systemu:** Konfigurowanie środowiska testowego, weryfikacja działania aplikacji poprzez testy jednostkowe, funkcjonalne i integracyjne.
- **Podsumowanie:** Ocena osiągniętych rezultatów, wnioski dotyczące wykorzystanych technologii i narzędzi, identyfikacja obszarów dalszego rozwoju aplikacji.

Układ pracy

Praca została podzielona na siedem rozdziałów, które szczegółowo opisują poszczególne etapy realizacji projektu:

1. **Wstęp:** Wprowadzenie do tematyki pracy, cel i zakres projektu.
2. **Analiza wymagań systemu:** Opis istniejących rozwiązań, analiza wymagań oraz przedstawienie logiki działania aplikacji.
3. **Projekt bazy danych:** Prezentacja modelu danych i struktury tabel.
4. **Projekt aplikacji:** Szczegółowy opis architektury oraz interfejsu użytkownika, z podziałem na warstwę frontend i backend.

- 5. Implementacja systemu:** Omówienie szczegółów technicznych związanych z realizacją aplikacji, w tym zastosowanych mechanizmów komunikacji, przetwarzania danych oraz bezpieczeństwa systemu.
- 6. Testowanie systemu:** Opis środowiska testowego oraz wyników przeprowadzonych testów działania aplikacji.
- 7. Podsumowanie:** Ocena rezultatów, wnioski oraz propozycje dalszego rozwoju aplikacji.

Każdy rozdział został szczegółowo opisany i uzupełniony rysunkami, tabelami oraz listingami kodu źródłowego, co umożliwia czytelnikowi kompleksowe zrozumienie zagadnień związanych z projektem. Praca zawiera również spisy literatury, tabel, rysunków i listingów, które ułatwiają nawigację oraz szybkie odnalezienie istotnych informacji.

Podjęte działania oraz zaprezentowane wyniki mają na celu dostarczenie praktycznego narzędzia, które może być wykorzystane zarówno w celach edukacyjnych, a także jako podstawa do dalszych badań i rozwoju w obszarze zarządzania danymi meteorologicznymi.