

# **WYŻSZA SZKOŁA TECHNOLOGII INFORMATYCZNYCH W KATOWICACH**

---

## **KWIATOWNIK – SYSTEM INFORMACYJNY – ENCYKLOPEDIA KULINARNO OGRODNICZA O ROŚLINACH EUROPY**

**AUTOR:** MICHał ZBIGNIEW JÓZEF LASOTA

**NR ALBUMU:** 7212

**PRZEDMIOT:** PROJEKT SYSTEMU INFORMATYCZNEGO

**PROWADZĄCY:** JACEK ŹYWICZOK

# Spis treści

2. Streszczenie.....	4
3. Zamierzony cel projektu.....	4
3.1. Priorytety:.....	4
4. Wstępne założenia i uwarunkowania.....	5
5. Założone ograniczenia i możliwość ewaluacji projektu.....	5
5.1. Ograniczenia projektowe.....	5
5.2. Ewaluacja : Projekt uznam za udany i kompletny, jeśli:.....	6
<b>II. CZĘŚĆ REALIZACYJNA.....</b>	<b>6</b>
6. Specyfikacja funkcjonalna systemu.....	6
6.1. Moduł Zarządzania Użytkownikami.....	7
6.2. Moduł Cyfrowego Herbarium.....	7
6.3. Moduł Gildii Permakulturowych.....	7
6.4. Moduł Interakcji Społecznościowej.....	8
6.5. Moduł Generatorów i Kalkulatorów.....	8
6.6. Moduł Administracyjny.....	8
6.7 System Notatek Prywatnych:.....	8
7. Wymagania niefunkcjonalne.....	9
7.1. Wymagania Sprzętowe.....	9
7.2. Wymagania Systemowe i Programowe.....	9
7.4. Bezpieczeństwo i pewność danych.....	10
7.5. Ergonomia i UX.....	10
8. Wymagania dotyczące danych.....	11
8.1. Warstwa Relacyjna SQL.....	11
Tabela: User.....	11
Tabela: Comment.....	11
8.2. Szczegółowa budowa bazy wiedzy (Pliki JSON).....	12
8.3. Logi, Hasła i Bezpieczeństwo Danych.....	15
9. Metody pracy, narzędzia i techniki.....	16
9.1. Środowisko programistyczne i narzędzia.....	16
9.2. Zastosowane techniki i wzorce projektowe.....	17
9.3 Metodyka wytwarzania.....	17
10. Opis głównych klas, metod, obiektów i algorytmów.....	19
10.1. Logika Modelowania Danych SQL.....	19
10.2. Algorytm A: Heurystyka Generowania Receptur.....	20
10.3. Algorytm B: Mapowanie Geometrii Gildii.....	21
10.4. Metoda Wyszukiwania Kontekstowego.....	22
10.5. Wykorzystanie narzędzi Open Source.....	23
<b>III. DOKUMENTACJA KOŃCOWA I UŻYTKOWNIKA.....</b>	<b>24</b>
11. Przebieg uruchamiania projektu (Instrukcja wdrożeniowa).....	24
11.1. Wymagania wstępne.....	24
11.2. Przygotowanie środowiska wirtualnego.....	24

11.3. Instalacja zależności.....	24
11.4. Inicjalizacja bazy danych i plików JSON.....	24
11.5. Uruchomienie serwera deweloperskiego.....	25
11.6. Konfiguracja konta administratora.....	25
<b>12. Podręcznik użytkownika.....</b>	<b>25</b>
12.1. Rejestracja i Personalizacja profilu.....	25
12.2. Przeglądanie Encyklopedii i Wyszukiwanie Terapeutyczne.....	25
12.3. Instrukcja obsługi Karty Rośliny.....	26
A. Morfologia i surowce zielarskie.....	26
B. Kalendarz prac i zjawisk fenologicznych.....	26
C. Apteka i kuchnia.....	27
D. Wiedza i ekologia.....	27
E. Ciekawostki.....	27
F. Gildia Permakulturowa.....	28
12.4. Zarządzanie Wiedzą: Komentarze vs Prywatne Notatki.....	28
A. Publikowanie komentarzy społecznościowych.....	28
B. Tworzenie Prywatnych Notelek Zielarskich.....	28
C. Wizualna identyfikacja rodzaju wpisu.....	28
D. Edycja i zarządzanie własnymi wpisami.....	29
12.5. Praca z Generatorem Receptur.....	30
A. Inteligentne filtrowanie surowców według właściwości.....	30
B. Konfiguracja parametrów i typu preparatu.....	30
C. Analiza wyników i proporcji wagowych.....	30
D. Zbiorczy system ostrzeżeń bezpieczeństwa.....	31
<b>13. Rozwiązywanie problemów (FAQ).....</b>	<b>31</b>
<b>14. Spostrzeżenia i zalecenia do użytkowania.....</b>	<b>32</b>
<b>15. Scenariusze testowe i walidacja systemu.....</b>	<b>33</b>
15.1. Testy akceptacyjne i funkcjonalne.....	33
15.2. Weryfikacja responsywności (UX).....	33
15.3. Przykład testu wstrzyknięcia kodu SQL (SQL Injection).....	34
Scenariusz testowy:.....	34
Warianty wpisanego tekstu:.....	34
<b>16. Wykryte błędy w działaniu.....</b>	<b>35</b>
<b>17. Spisy ilustracji i rysunków.....</b>	<b>36</b>
<b>18. Bibliografia i źródła.....</b>	<b>37</b>
Literatura przedmiotu (Merytoryka).....	37
Dokumentacja techniczna (Technologie IT).....	37
Zasoby internetowe i narzędzia AI.....	37
<b>IV. ZAŁĄCZNIKI I DODATKI.....</b>	<b>37</b>
19. Nośniki elektroniczne Flash.....	37

## 2. Streszczenie

Przedmiotem pracy jest projekt i realizacja systemu informatycznego „Kwiatownik”, pełniącego funkcję interaktywnej encyklopedii roślin użytkowych. Aplikacja została wykonana w architekturze klient-serwer przy użyciu języka Python (framework Flask) oraz bazy danych SQLite.

W dokumencie przedstawiono pełny proces wytwórczy: od analizy wymagań i projektowania architektury hybrydowej (SQL/JSON), poprzez implementację algorytmów ostrzegawczych, aż po walidację bezpieczeństwa i funkcjonalności systemu.

## 3. Zamierzony cel projektu

Celem głównym było rozwiązywanie problemu rozproszenia i niskiej wiarygodności informacji o ziołolecznictwie dostępnych w Internecie. „Kwiatownik” ma służyć jako referencyjne, cyfrowe repozytorium wiedzy botanicznej, które systematyzuje dane o roślinach i zapewnia ich bezpieczną interpretację. Projekt odpowiada na potrzebę dostępu do zweryfikowanej wiedzy eksperckiej, minimalizując ryzyko błędów podczas samodzielnego stosowania fitoterapii.

### 3.1. Priorytety:

- **Weryfikacja merytoryczna (Human-Validated Knowledge):** Priorytetem było stworzenie bazy wiedzy, w której dane generowane przez SI są weryfikowane przez eksperta, co eliminuje ryzyko błędów (halucynacji) w krytycznych informacjach medycznych.
- **Hybrydowa architektura danych:** Zastosowanie modelu łączącego relacyjną bazę danych (SQL) dla użytkowników z dokumentową strukturą (JSON) dla opisów botanicznych, co zapewnia elastyczność i skalowalność.
- **Edukacja i bezpieczeństwo:** Implementacja algorytmów ostrzegawczych oraz kalkulatora proporcji, które minimalizują ryzyko błędów podczas samodzielnego przygotowywania preparatów roślinnych.

**Wniosek:** Powyższe cele zmierzają do stworzenia platformy, która w sposób nowoczesny, bezpieczny i wysoce użyteczny popularyzuje wiedzę o ekosystemach roślinnych, promując odpowiedzialne i naukowe podejście do tradycyjnego zielarstwa.

## 4. Wstępne założenia i uwarunkowania

Proces projektowania systemu „Kwiatownik” poprzedzony był sformułowaniem zestawu kluczowych założeń, które zdefiniowały ostateczną strukturę aplikacji oraz wybór technologii implementacyjnych:

- **Mobilność i powszechna dostępność:** Przyjęto założenie, że system będzie eksploatowany głównie w warunkach terenowych (ogrody, obszary leśne). W związku z tym priorytetem stało się zapewnienie pełnej responsywności interfejsu (RWD) oraz optymalizacja kodu pod kątem szybkości działania na urządzeniach przenośnych.
- **Hybrydowy model składowania danych:** Ze względu na wysoką złożoność i zmienną strukturę opisów botanicznych, zrezygnowano z ograniczającego schematu relacyjnego na rzecz elastycznych plików JSON w obszarze encyklopedii. Pozwala to na swobodną rozbudowę bazy wiedzy bez konieczności modyfikowania architektury bazy danych.
- **Standardy bezpieczeństwa i prywatności:** Ochrona danych użytkowników stanowi fundament systemu. Zastosowano profesjonalne mechanizmy kryptograficzne, w tym nieodwracalne haszowanie haseł, co gwarantuje poufność informacji nawet w przypadku próby nieautoryzowanego dostępu do bazy.
- **Interaktywność i dynamika systemu:** Aby uniknąć statycznego charakteru aplikacji, wdrożono bazę SQL dedykowaną do obsługi interakcji społecznościowych. Dzięki temu teoretyczna wiedza botaniczna może być konfrontowana z praktycznymi doświadczeniami użytkowników w formie komentarzy oraz prywatnych notatek, stanowiących osobisty dziennik rozwoju.

## 5. Założone ograniczenia i możliwość ewaluacji projektu

W procesie wytwarzyczym świadomie zdefiniowano ramy projektowe, co pozwoliło na zachowanie wysokich standardów jakościowych przy uwzględnieniu dostępnych zasobów czasowych oraz technologicznych.

### 5.1. Ograniczenia projektowe

- **Ramy czasowe i priorytetyzacja funkcji:** Ze względu na określony termin realizacji, proces deweloperski skoncentrowano na implementacji kluczowych modułów systemowych (autoryzacja, wyszukiwanie kontekstowe, system interakcji). Pozwoliło to na dostarczenie stabilnej wersji aplikacji o wysokiej użyteczności, zamiast rozbudowywania drugorzędnych funkcji społecznościowych.
- **Dobór technologii bazodanowej:** Podjęto świadomą decyzję o wykorzystaniu silnika SQLite zamiast zewnętrznych serwerów baz danych. Rozwiążanie to zapewnia lekkość aplikacji oraz upraszcza proces wdrożenia (deploymentu), będąc jednocześnie w pełni wystarczającym dla obecnej skali projektu. Architektura systemu została jednak zaprojektowana w sposób umożliwiający bezproblemową migrację do silnika PostgreSQL w przypadku przyszłej skalowalności.

- **Selekcja i weryfikacja danych:** W obszarze bazy wiedzy postawiono na jakość, a nie na ilość rekordów. Zakres danych ograniczono do wyselekcjonowanej grupy roślin, co umożliwiło przeprowadzenie pełnej, ludzkiej weryfikacji informacji (Human-Validated Knowledge). Podejście to eliminuje ryzyko powielania błędów generatywnej sztucznej inteligencji, co jest kluczowe w projektach o profilu fitoterapeutycznym.

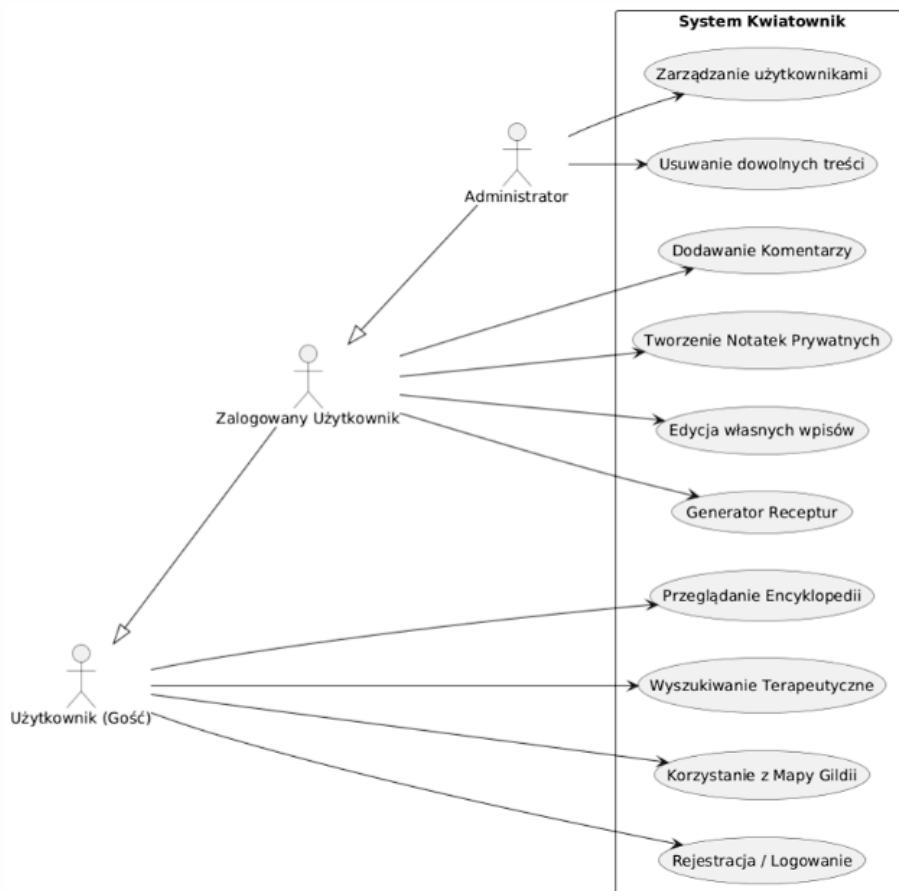
## 5.2. Ewaluacja : Projekt uznam za udany i kompletny, jeśli:

1. Użytkownik będzie mógł się bezpiecznie zarejestrować i zalogować i wylogować.
2. Algorytm wyszukiwania poprawnie znajdzie roślinę na podstawie wpisanego objawu (np. „ból gardła”) lub części roślin (np. „kora dębu”).
3. Generator receptur poprawnie przeliczy masę ziół dla zadanej objętości.
4. Mapa gildii wygeneruje się dynamicznie na podstawie relacji zapisanych w pliku JSON.
5. Karta roślin przedstawi wszystkie potwierdzone informacje zawarte w pliku JSON.

## II. CZĘŚĆ REALIZACYJNA

### 6. Specyfikacja funkcjonalna systemu

Aby obiektywnie ocenić stopień realizacji założonych celów, zdefiniowałem zestaw merytorycznych i technicznych wskaźników sukcesu. Ewaluacja projektu będzie opierać się na weryfikacji poprawności działania kluczowych modułów oraz integralności danych.



Ilustracja 1: Diagram przypadków użycia

## 6.1. Moduł Zarządzania Użytkownikami

- **Rejestracja konta:** System umożliwia założenie unikalnego profilu użytkownika (wymagany login, e-mail oraz hasło).
- **Logowanie i wylogowywanie:** Bezpieczne zarządzanie sesją użytkownika z wykorzystaniem technologii *Flask-Login*.
- **Zarządzanie uprawnieniami:** Rozróżnienie poziomów dostępu na standardowego Użytkownika (przeglądanie, komentowanie) oraz Administratora (dostęp do funkcji specjalnych - dostęp do usunięcia każdego komentarza).
- **Szyfrowanie danych wrażliwych:** Automatyczne generowanie skrótów haseł przy użyciu algorytmu PBKDF2 w trakcie zapisu do bazy SQL.

## 6.2. Moduł Cyfrowego Herbarium

- **Dynamiczny przegląd roślin:** Wyświetlanie listy wszystkich gatunków dostępnych w bazie JSON z podaniem najważniejszych cech.
- **Karta morfologiczna rośliny:** Szczegółowy widok pojedynczego gatunku, zawierający:
  - Podział na części rośliny - kora, liście, owoce.
  - Opis właściwości leczniczych i składników aktywnych - każdej głównej części rośliny.
  - Sekcję ostrzeżeń i przeciwwskazań (dane zweryfikowane pod kątem toksykologicznym) przypisanym częścią.
- **Wyszukiwarka kontekstowa:** Zaawansowany algorytm przeszukujący nie tylko nazwy, ale także opisy właściwości i objawów ukryte w strukturach JSON.

## 6.3. Moduł Gildii Permakulturowych

- **Interaktywna mapa powiązań:** Graficzne przedstawienie relacji między rośliną bazową a jej towarzyszami. Z informacjami jaki mają wpływ na gildie powiązań.
- **Obliczanie rozmieszczenia:** Algorytmiczne rozmieszczanie węzłów towarzyszących na okręgu z wykorzystaniem parametrów matematycznych przesyłanych do warstwy JS.
- **Nawigacja relacyjna:** Możliwość przejścia do karty rośliny towarzyszącej bezpośrednio z poziomu mapy gildii (o ile roślina znajduje się w bazie danych) klikając stricte w roślinę.

## 6.4. Moduł Interakcji Społecznościowej

- **System komentarzy:** Możliwość dodawania opinii i doświadczeń pod każdą kartą rośliny przez zalogowanych użytkowników. Oraz dostęp do grupy ogólnych komentarzy nie przypisanych żadnej roślinie.
- **Integracja SQL:** Automatyczne przypisywanie komentarzy z bazy relacyjnej do odpowiednich dokumentów JSON za pomocą klucza `plant_id` jeśli są tworzone na karcie roślin, w innym przypadku są ustawione jako globalne z wartością `null`
- **Znaczniki czasu i autorstwa:** Automatyczna rejestracja daty dodania komentarza oraz powiązanie go z profilem autora (relacja 1:N) oraz ustawianie prywatności komentarza tworząc z niego notatek lub komentarz.

## 6.5. Moduł Generatorów i Kalkulatorów

- **Kalkulator receptur:** Narzędzie pozwalające na wybór surowców i objętości docelowej mikstury, dzięki wybranym właściwościom moduł pozwala odfiltrować wyszukiwane surowce.
- **Automatyczne przeliczanie proporcji:** Algorytm opiera się na wagach surowców jako nalewek (1:5) lub jako naparu (1:10) wobec rozpuszczalnika wody/alkoholu
- **Agregacja przeciwskaźników:** System skanuje wybrane surowce i generuje zbiorczy komunikat bezpieczeństwa dla tworzonej receptury.

## 6.6. Moduł Administracyjny

- **Automatyczna walidacja struktur danych:** System został wyposażony w mechanizmy weryfikujące kompletność kluczy w plikach JSON. Pozwala to na skutecną obsługę wyjątków (np. `UndefinedError`) i zapobiega błędem renderowania interfejsu w przypadku napotkania niepełnych zbiorów danych.
- **Elastyczne zarządzanie treścią:** Architektura systemu umożliwia dynamiczną modyfikację bazy wiedzy poprzez bezpośrednią edycję plików źródłowych JSON. Proces ten odbywa się bez ingerencji w relacyjną strukturę bazy SQL i nie wymaga restartu serwera, co zapewnia ciągłość działania aplikacji.

## 6.7 System Notatek Prywatnych:

- **Zindywidualizowany dziennik obserwacji:** Moduł ten umożliwia użytkownikom prowadzenie w pełni poufnych zapisów, dedykowanych dokumentowaniu własnych doświadczeń. Dane te są ściśle powiązane z profilem autora i niedostępne dla pozostałych uczestników platformy.
- **Techniczna izolacja danych wrażliwych:** W przeciwieństwie do publicznego systemu komentarzy, notatki podlegają rygorystycznej separacji na poziomie logicznym i bazodanowym. Dostęp do nich jest autoryzowany i możliwy wyłącznie po pomyślnym zalogowaniu użytkownika, co gwarantuje najwyższy poziom prywatności.

## 7. Wymagania niefunkcjonalne

### 7.1. Wymagania Sprzętowe

A. Serwer aplikacji Zestaw minimalnych parametrów niezbędnych do uruchomienia logiki backendowej i obsługi bazy danych:

- **Pamięć RAM:** Minimum 512 MB (zalecane 1 GB dla zapewnienia płynnej obsługi wielu sesji jednocześnie).
- **Procesor:** Dowolna jednostka obliczeniowa o taktowaniu min. 1.0 GHz.
- **Przestrzeń dyskowa:** Ok. 100 MB na kod źródłowy i bazę danych (możliwość skalowania w miarę rozbudowy bazy plików JSON wymaga większej przestrzeni dyskowej).

B. Urządzenie użytkownika Wymagania dla sprzętu, na którym wyświetlany będzie interfejs aplikacji:

- Smartfon lub tablet. Z ekranem o minimalnej rozdzielczości ekranu: 320px

### 7.2. Wymagania Systemowe i Programowe

Ze względu na architekturę klient-serwer, wymagania zostały podzielone na środowisko uruchomieniowe aplikacji (serwer) oraz środowisko użytkownika końcowego (klient).

A. Środowisko serwerowe (**Serwer**) Niezbędne do poprawnego uruchomienia i działania logiki aplikacji oraz bazy danych:

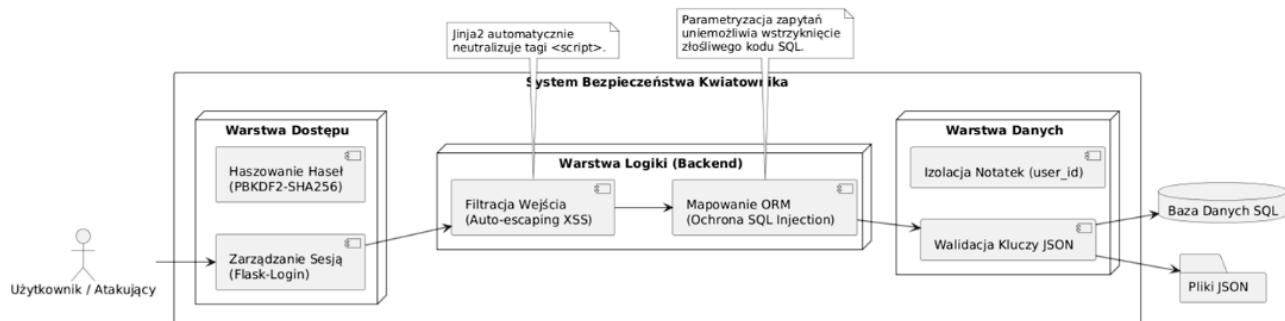
- **System operacyjny:** Wieloplatformowość – pełna kompatybilność z systemami Windows 10/11, Linux oraz macOS.
- **Interpreter:** Język Python w wersji 3.8 lub nowszej.
- **Kluczowe biblioteki i frameworki:** Cała lista zawarta w pliku requirements.txt której instalacja odbywa się automatycznie w momencie instalacji aplikacji na serwerze.

B. Środowisko użytkownika (**Klient**) Aplikacja jest w pełni responsywna (RWD), co umożliwia jej obsługę na dowolnym urządzeniu posiadającym dostęp do sieci:

- Urządzenie końcowe:
  - Komputer stacjonarny / Laptop (zalecane do prac administracyjnych).
  - Smartfon / Tablet (zalecane do pracy w ogrodzie).
- **Przeglądarka internetowa:** Dowolna nowoczesna przeglądarka wspierająca standardy HTML5, CSS3 oraz JavaScript (ES6+), np.:
  - Google Chrome (wersja desktop/mobile)
  - Mozilla Firefox
  - Safari (iOS/macOS)
  - Microsoft Edge

## 7.4. Bezpieczeństwo i pewność danych

Zapewnienie bezpieczeństwa danych w systemie „Kwiatownik” opiera się na architekturze warstwowej. Każda próba interakcji z systemem przechodzi przez szereg filtrów ochronnych, co minimalizuje ryzyko wystąpienia incydentów bezpieczeństwa.



### Opis mechanizmów przedstawionych na diagramie:

- **Warstwa Dostępu:** Pierwsza linia obrony, która dba o to, by użytkownik był tym, za kogo się podaje. Hasła są nieodwracalnie haszowane, a sesje podpisywane cyfrowo.
- **Warstwa Logiki:** Tutaj system „czyści” dane. Dzięki silnikowi Jinja2, wszelkie próby wpisania kodu JavaScript w komentarzach są zamieniane na bezpieczny tekst (XSS protection). Z kolei SQLAlchemy pilnuje, by zapytania do bazy były odporne na próby włamań (SQL Injection).
- **Warstwa Danych:** Ostatni etap weryfikacji. System sprawdza, czy pliki JSON są kompletne oraz czy użytkownik próbuje odczytać notatkę, która faktycznie należy do niego (izolacja na poziomie `user_id`).

## 7.5. Ergonomia i UX

Priorytetem w projektowaniu interfejsu (GUI) była maksymalna użyteczność oraz estetyka spójna z tematyką przyrodniczą. Przyjęto następujące założenia :

- **Intuicyjna nawigacja:** Układ elementów interfejsu został zaprojektowany zgodnie z powszechnymi wzorcami projektowymi (breadcrumbs, jasne etykiety), co eliminuje konieczność długiego procesu uczenia się obsługi systemu.
- **Stylistika "Modern Botanical":** Zastosowano paletę barw inspirowaną naturą oraz minimalistyczną typografię, co ma na celu redukcję obciążenia poznawczego użytkownika i budowanie odpowiedniego nastroju aplikacji.
- **Dostępność:** Zachowano wysoki kontrast tekstu względem tła oraz odpowiednią wielkość elementów interaktywnych, co umożliwia wygodną obsługę aplikacji na urządzeniach mobilnych w warunkach silnego nasłonecznienia (np. w ogrodzie).

## 8. Wymagania dotyczące danych

System wykorzystuje architekturę hybrydową, separując dane transakcyjne od bazy wiedzy. Zastosowano podział na warstwę relacyjną (SQL) oraz dokumentową (JSON), co optymalizuje wydajność aplikacji.

### 8.1. Warstwa Relacyjna SQL

Warstwa ta odpowiada za przechowywanie danych ustrukturyzowanych, wymagających zachowania ścisłej integralności i relacji.

- **Tabela: User**

Służy do unikalnej identyfikacji użytkowników oraz obsługi procesu uwierzytelniania.

**Tabela 1: Klasa - User**

Nazwa pola	Typ danych	Rozmiar	Opis i ograniczenia
ID	Integer	-	Klucz główny (Primary Key), autoinkrementacja
USERNAME	String	80 znaków	Unikalna nazwa użytkownika (Unique)
EMAIL	String	120 znaków	Unikalny adres e-mail (Unique)
PASSWORD_HASH	String	128 znaków	Skrót hasła (PBKDF2). Nie przechowujemy jawnego tekstu
IS_ADMIN	Boolean	-	Flaga uprawnień (Default: False). Pierwszy zarejestrowany użytkownik dostaje prawa administratora

- **Tabela: Comment**

Przechowuje interakcje użytkowników, wiążąc treść komentarza z autorem (relacja klucza obcego) oraz konkretną rośliną.

**Tabela 2: Klasa - Comment**

Nazwa pola	Typ danych	Rozmiar	Opis i ograniczenia
ID	Integer	-	Klucz główny
USER_ID	Integer	-	Klucz obcy (Foreign Key) powiązany z USER.ID
CONTENT	Text	500+ znaków	Treść komentarza
TIMESTAMP	DateTime	-	Automatyczna data i godzina dodania (UTC)
IS_PRIVATE	Boolean	-	True dla notatki prywatnej, False dla publicznego komentarza
PLANT_ID	String	100 znaków	Klucz łączący komentarz z odpowiednim plikiem JSON

## 8.2. Szczegółowa budowa bazy wiedzy (Pliki JSON)

Każdy gatunek rośliny opisany jest w osobnym pliku JSON o ustandaryzowanej strukturze.

Kluczowe sekcje pliku obejmują:

- **Identyfikacja taksonomiczna:** Unikalny identyfikator (ID), pełna nazwa polska i łacińska oraz przynależność do rodziny botanicznej.
- **Charakterystyka botaniczna:** Opis morfologii rośliny, jej cyklu życiowego oraz roli w ekosystemie.

```
"id": "dab_szypulkowy",
"nazwa_pl": "Dąb szypułkowy",
"nazwa_lat": "Quercus robur",
"rodzina": "Bukowate",
"opis": "Potężne drzewo liściaste, długowieczne, o szerokiej koronie. Jest najważniejszym gatunkiem lasotwórczym w Europie Środkowej.",
```

Ilustracja 2: ID, nazwa, opis - JSON

- **Surowce zielarskie (czesci\_rosliny):** Szczegółowy wykaz surowców (kora, liście, owoce) wraz z ich składem chemicznym, właściwościami farmakologicznymi oraz optymalnym terminem zbioru. Oraz zastrzezeniami związanymi z używaniem poszczególnych części.

```
"czesci_rosliny": {
  "kora": {
    "nazwa_surowca": "Cortex Quercus",
    "opis_biotyczny": "Gładka i lśniąca u młodych okazów, z czasem staje się grubą, głęboko spękaną skorupą.",
    "skladniki_aktywne": ["Garbniki taninowe (do 20%)", "Kwasy fenolowe", "Flawonoidy", "Sole mineralne"],
    "wlasciwosci": "Silnie ściągające, przeciwzapalne, hamujące drobne krwawienia.",
    "czas_zbioru": "Wiosna (marzec-kwiecień), zanim pojawią się liście, z młodych, gładkich gałęzi.",
    "ostrzezenia": "Nie stosować wewnętrznie odwarów z kory przy dużych uszkodzeniach skóry oraz w ostrych stanach niewydolności serca."
  },
  "lisicie": {
    "nazwa_surowca": "Folium Quercus",
    "opis_biotyczny": "Krótkogonkowe, o nieregularnie klapowanych brzegach, skórzaste, z charakterystycznymi uszkami u nasady.",
    "skladniki_aktywne": ["Garbniki", "Kwercetyna", "Związki żywiczone"],
    "wlasciwosci": "Antyseptyczne, ściągające, stosowane głównie zewnętrznie w medycynie ludowej.",
    "czas_zbioru": "Czerwiec - Lipiec (młode liście)."
  },
  "owoce": {
    "nazwa_surowca": "Glandes Quercus (żołędzie)",
    "opis_biotyczny": "Owalne orzeczy osadzone w miseczkach na długich szypułkach (stąd nazwa gatunku).",
    "skladniki_aktywne": ["Skrobia", "Tłuszcze", "Białka", "Garbniki (wymagają wyługowania)"],
    "wlasciwosci": "Odżywcze, po obróbce termicznej i wodnej działają przeciwbiegunkowo i wzmacniająco.",
    "czas_zbioru": "Wrzesień - Październik (gdy same opadają).",
    "ostrzezenia": "Nadmier garbników hamuje wchłanianie żelaza i witamin."
  }
},
```

Ilustracja 3: Części i składniki rośliny - JSON

- **Zastosowanie w praktyce:** Podział na zastosowanie medyczne, kulinarne oraz funkcje w architekturze krajobrazu.

```
"zastosowanie": {
    "medyczne": "Odwar z kory: stany zapalne jamy ustnej, gardła, odmrożenia i oparzenia. Żołędzie: biegunki, rekonalizacja.",
    "kulinarne": "Kawa żołędziowa, mąka do wypieku chleba (po żugowaniu), dodatki do pasztetów.",
    "ogrodowe": "Gatunek kluczowy dla bioróżnorodności, siedlisko dla setek gatunków owadów, produkcja wysokiej jakości próchnicy."
},
```

Ilustracja 4: Zastosowanie - JSON

- **Bezpieczeństwo i toksykologia:** Krytyczna sekcja zawierająca ostrzeżenia, przeciwwskazania oraz informacje o interakcjach z lekami syntetycznymi.

```
"ostrzezenia": [
    "Nie stosować wewnętrznie odwarców z kory przy dużych uszkodzeniach skóry oraz w ostrych stanach niewydolności serca.",
    "Nadmier gębników hamuje wchłanianie żelaza i witamin."],
```

Ilustracja 5: Ostrzeżenia - JSON

- **Permakultura i Gildie:** Definicja funkcji ekologicznych oraz lista roślin towarzyszących (gildii), wspierających wzrost danego gatunku poprzez allelopatię pozytywną.

```
"permakultura": {
    "funkcje": [
        "Pompa mineralna (głębokie korzenie wyciągają minerały na powierzchnię)",
        "Naturalny wiatrochron",
        "Producent żołędzi dla systemów wypasu (silvopasture)"
    ],
    "gildie": [
        {
            "nazwa": "Leszczyna pospolita",
            "rola": "Piętro niższe, produkcja biomasy i orzechów"
        },
        {
            "nazwa": "Głów jednoszyjkowy",
            "rola": "Strażnik (bariera kolczasta) i wsparcie zapylaczy"
        },
        {
            "nazwa": "Śliwa tarnina",
            "rola": "Zarośla ochronne i stabilizacja gleby"
        },
        {
            "nazwa": "Poziomka pospolita",
            "rola": "Żywy mulcz (okrywa runa)"
        },
        {
            "nazwa": "Marzanka wonna (Przytulia wonna)",
            "rola": "Runo leśne i bioindykator gleby"
        },
        {
            "nazwa": "Trufle",
            "rola": "Symbioza mikoryzowa (strefa korzeniowa)"
        }
    ]
},
```

Ilustracja 6: Permakultura - JSON

- **Kalendarz ogrodnika:** To Twój osobisty przewodnik po pracach sezonowych. Zawiera konkretne czynności (siew, zbiór, obserwacja) wraz z optymalną porą dnia, pogodą, a nawet wpływem faz księżyca na witalność rośliny.

```

"kalendarez_ogrodnika": {
  "zadania": [
    {
      "czynosc": "Siew (bezpośredni)",
      "miesiace": [10, 11],
      "opis": "Siew dojrzałych żołędzi bezpośrednio do gruntu na głębokość 5-7 cm.",
      "faza_ksiezyca": "Książyc ubywający (faza korzenia)",
      "pora_dnia": "Poranek",
      "pogoda": "Dzień bezmroźny, wilgotna gleba",
      "wplyw_ksiezyca": "Sprzyja szybkiemu ukorzenianiu i zapobiega gniciu nasion."
    },
    {
      "czynosc": "Zbiór kory",
      "miesiace": [3, 4],
      "opis": "Pozyskiwanie młodej, gładkiej kory z gałęzi przeznaczonych do przycinków.",
      "faza_ksiezyca": "Nów lub pierwszy kwartał",
      "pora_dnia": "Przedpołudnie",
      "pogoda": "Słonecznie, sucho (ważne dla procesu suszenia)",
      "wplyw_ksiezyca": "Wzrost soków w górnych partiach rośliny zwiększa zawartość garbników."
    },
    {
      "czynosc": "Obserwacja kwitnienia",
      "miesiace": [5],
      "opis": "Monitorowanie rozwoju kwiatów męskich (kotek) i żeńskich.",
      "faza_ksiezyca": "Pełnia",
      "pora_dnia": "Południe",
      "pogoda": "Lekki wiatr (sprzyja zapylaniu krzyżowemu)",
      "wplyw_ksiezyca": "Maksymalna witalność rośliny i emisja pyłku."
    },
    {
      "czynosc": "Zbiór owoców (żołędzi)",
      "miesiace": [9, 10],
      "opis": "Zbiór w pełni dojrzałych orzechów opadłych na ziemię.",
      "faza_ksiezyca": "Książyc ubywający"
    }
  ]
}

```

Ilustracja 7: Kalendarz ogrodnika - JSON

- **Domowa receptura (Przepisy):** Gotowe instrukcje krok po kroku, jak przygotować leki (odwary, płukanki) oraz potrawy (np. mąkę lub kawę z żołądżi).

```

"przepisy_medyyczne": [
  {
    "czesc_uztya": "kora",
    "nazwa": "Odwar do nasiadówek i płukanek",
    "skladniki": "2 łyżki kory, 1 litr wody",
    "przygotowanie": "Gotować 15 minut, odstawić na 10 minut. Stosować zewnętrznie."
  },
  {
    "czesc_uztya": "owoce",
    "nazwa": "Mąka żołądżiowa",
    "skladniki": "Dojrzałe żołądzie, woda, popiół drzewny (opcjonalnie)",
    "przygotowanie": "Żołądzie obrać, moczyć w wodzie zmieniając ją przez 3-4 dni (ługowanie garbników), wysuszyć i zmielić."
  }
],
"przepisy_kulinarne": [
  {
    "czesc_uztya": "owoce",
    "nazwa": "Staropolska kawa żołądżiowa",
    "skladniki": "Uprzążone żołądzie, korzeń cykorii, goździki",
    "przygotowanie": "Zmielone składniki gotować 2-3 minuty. Podawać z miodem i mlekiem."
  }
]

```

Ilustracja 8: Przepisy - JSON

### 8.3. Logi, Hasła i Bezpieczeństwo Danych

W tej sekcji opisano techniczne aspekty implementacji mechanizmów chroniących dane, ze szczególnym uwzględnieniem izolacji notatek prywatnych.

**A. Bezpieczeństwo haseł i uwierzytelnianie** System nie gromadzi haseł w formie tekstu jawnego. Zastosowano standardy uniemożliwiające odtworzenie hasła nawet w przypadku wycieku bazy danych:

- **Algorytm haszowania:** Wykorzystano bibliotekę Werkzeug do generowania jednokierunkowych skrótów haseł przy użyciu algorytmu PBKDF2-SHA256. Jest to standard przemysłowy odporny na ataki typu *rainbow table*.
- **Zarządzanie tożsamością:** Moduł Flask-Login odpowiada za bezpieczne utrzymanie sesji, weryfikując tożsamość użytkownika przy każdym żądaniu dostępu do zasobów chronionych.

**B. Izolacja i ochrona danych prywatnych** Kluczowym elementem architektury bezpieczeństwa jest logiczna separacja danych w warstwie zapytań SQL. Gwarantuje ona poufność notatek osobistych:

- **Filtrowanie autoryzacyjne:** Dostęp do notatek prywatnych jest ograniczany bezpośrednio w zapytaniu do bazy danych (klauzula WHERE). System wymusza zgodność user\_id notatki z ID aktualnie zalogowanego użytkownika. Dzięki temu pobranie cudzej notatki jest technicznie niemożliwe na poziomie silnika bazy.

- **Weryfikacja własności przy edycji:** Przed wykonaniem operacji modyfikacji (UPDATE) lub usunięcia (DELETE), backend ponownie weryfikuje, czy obiekt należy do osoby wysyłającej żądanie, co chroni przed manipulacją parametrami URL (IDOR).

C. Ochrona plików i obsługa błędów Wdrożono mechanizmy zabezpieczające fizyczne zasoby serwera oraz ukrywające szczegóły techniczne przed użytkownikiem końcowym:

- **Tryb produkcyjny:** W finalnej wersji aplikacji wyłączono tryb debugowania, co zapobiega wyświetlaniu stosu błędów (stack trace) i ujawnianiu ścieżek systemowych w przypadku awarii.
- **Ochrona bazy danych SQLite:** Plik bazy danych (`kwiatownik.db`) został umieszczony w strukturze katalogów w sposób uniemożliwiający jego bezpośrednie pobranie poprzez wpisanie adresu w przeglądarce.
- **Integralność plików JSON:** Pliki encyklopedii otwierane są przez aplikację wyłącznie w trybie odczytu (`read-only`), co eliminuje ryzyko ich przypadkowego nadpisania lub uszkodzenia przez błędy w logice programu.

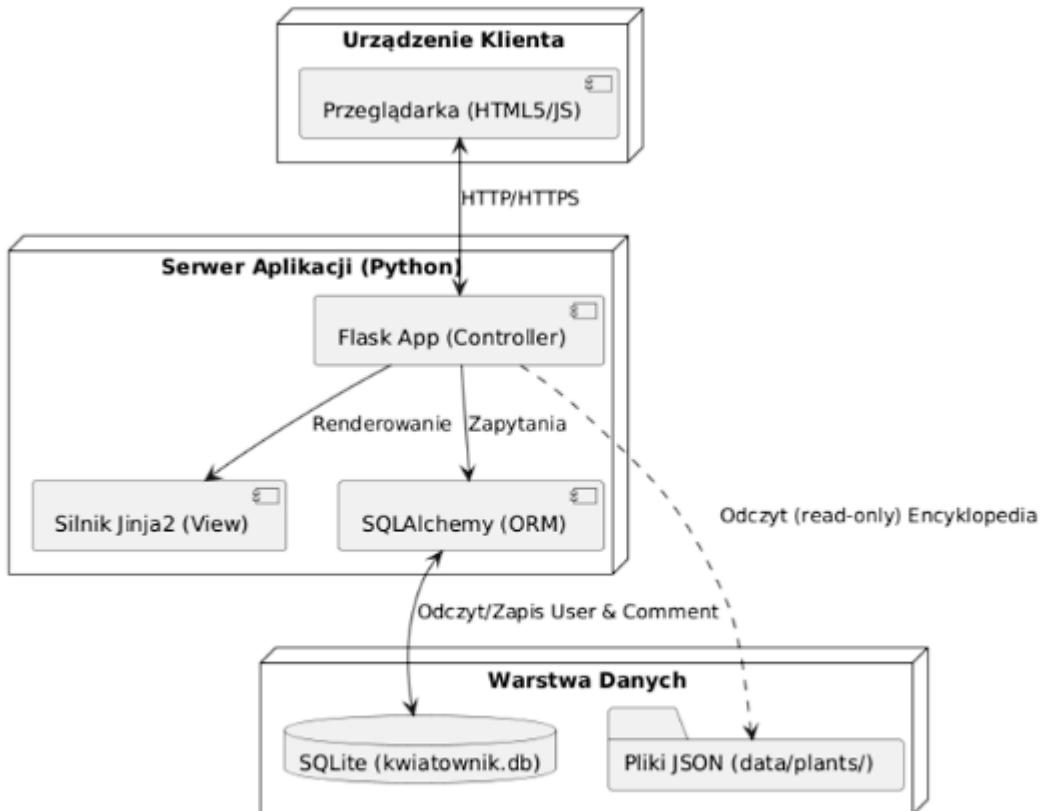
## 9. Metody pracy, narzędzia i techniki

Wytworzenie aplikacji „Kwiatownik” wymagało zastosowania nowoczesnego stosu technologicznego oraz metodyki pracy zorientowanej na czystość kodu i bezpieczeństwo danych.

### 9.1. Środowisko programistyczne i narzędzia

Do budowy i testowania aplikacji wykorzystano następujące narzędzia:

- **Język programowania:** Python 3.8 – wybrany ze względu na doskonałą obsługę formatów danych (JSON) oraz bogatą bibliotekę frameworków webowych.
- **Framework:** Flask – mikro-framework umożliwiający szybkie budowanie skalowalnych aplikacji webowych przy zachowaniu pełnej kontroli nad strukturą projektu.
- **Technologie frontendowe:** HTML5, CSS3, JavaScript oraz silnik szablonów `jinja2`
- **Edytor kodu:** PyCharm – wykorzystanie profesjonalnych IDE z debuggerami pozwoliło na szybką eliminację błędów w logice generatora receptur.



*Ilustracja 9: Diagram architektury wdrożenia*

- **Zarządzanie bazą danych:** xampp – narzędzie służące do podglądu i manualnej weryfikacji integralności tabel użytkowników i komentarzy.

## 9.2. Zastosowane techniki i wzorce projektowe

Architektura systemu opiera się na sprawdzonych wzorcach inżynierii oprogramowania:

- **Wzorzec MVC:** Model-View-Controller- Rozdzielenie warstwy danych (modele SQLAlchemy), logiki sterującej (trasy w Flask) oraz prezentacji (szablony Jinja2) zapewnia łatwą rozbudowę systemu.
- **Hybrydowe przechowywanie danych:** Połączenie relacyjnej bazy SQLite z dokumentową strukturą JSON, co optymalizuje czas dostępu do informacji.
- **Mapowanie Obiektowo-Relacyjne:** Wykorzystanie SQLALCHEMY do komunikacji z bazą danych.

## 9.3 Metodyka wytwarzania

Praca nad projektem przebiegała w sposób iteracyjny, kładąc szczególny nacisk na wiarygodność bazy wiedzy zielarskiej:

1. **Gromadzenie danych:** Wstępne szkielety plików JSON zostały wygenerowane przy użyciu zaawansowanych modeli sztucznej inteligencji, co pozwoliło na szybkie ustrukturyzowanie ogromnych ilości danych botanicznych i medycznych.

**2. Weryfikacja wiarygodności:** Każdy wygenerowany plik JSON (np.*DAB\_SZYPULKOWY.JSON*) został poddany manualnej weryfikacji merytorycznej w oparciu o literaturę przedmiotu oraz atlasy roślin leczniczych. Proces ten miał na celu wyeliminowanie potencjalnych halucynacji SI, szczególnie w krytycznych sekcjach dotyczących dawkowania i ostrzeżeń.

**Tabela 3: Raport z walidacji i korekty danych systemowych.**

Cecha danych	Pierwotna propozycja SI	Korekta po teście weryfikacji	Cel i uzasadnienie modyfikacji
Nazewnictwo	Dąb pospolity	Dąb szypułkowy ( <i>Quercus robur</i> )	Precyza botaniczna; odróżnienie od dębu bezszypułkowego.
Nazewnictwo	Marzanka leśna	Marzanka wonna ( <i>Galium odoratum</i> )	Zastosowanie oficjalnej nomenklatury farmaceutycznej i zielarskiej.
Nazewnictwo	Hazelnut Bush	Leszczyna pospolita ( <i>Corylus avellana</i> )	Polonizacja i standaryzacja nazw łacińskich dla spójności wyszukiwarki.
Toksyczność	"Owoce tarniny są jadalne od razu."	"Wymagają przemrożenia; pestki zawierają amigdalinę (usuwać!)."	Bezpieczeństwo: Eliminacja ryzyka spożycia toksycznych glikozydów.
Toksyczność	"Marzanka jest bezpieczna w każdej ilości."	"Zawiera kumarynę. Przedawkowanie powoduje bóle głowy."	Bezpieczeństwo: Dodanie kluczowych ostrzeżeń o dawkowaniu terapeutycznym.
Ekologia (Gildie)	Sugestia sadzenia Lawendy pod Dębem.	Zamiana na Marzankęwoną i Poziomkę.	Logika: Dostosowanie do warunków świątecznych (cień dębu vs. słońce lawendy).
Ekologia (Gildie)	"Trufle rosną na liściach dębu."	"Trufle rosną pod ziemią."	Logika: Poprawa błędu biologicznego dotyczącego mechanizmu symbiozy.
Ekologia (Gildie)	"Głów służy tylko jako ozdoba."	"Głów: strażnik (bariera), pokarm dla ptaków, wabik zapylaczy."	Logika: Zdefiniowanie konkretnych funkcji permakulturowych rośliny.
Struktura IT	Dane w formie długiego akapitu.	Podział na zagnieżdżone obiekty: czesci_rosliny, funkcje.	Technologia: Umożliwienie dynamicznego renderowania w Jinja2.
Struktura IT	Daty zbiorów jako tekst "wiosną".	Zastosowanie list numerycznych [3, 4, 5] dla miesięcy.	Technologia: Umożliwienie logicznego filtrowania w Dashboardzie Ogrodnika.
Struktura IT	Brak unikalnych kluczy ID.	Wprowadzenie ustandaryzowanych kluczy typu slug (np. <i>dab_szypulkowy</i> ).	Technologia: Zapewnienie stabilnego linkowania między podstronami (brak błędów 404).
Toksyczność	Brak uwag przy korze dębu.	"Może hamować wchłanianie żelaza i leków."	Bezpieczeństwo: Ostrzeżenie przed interakcjami farmakologicznymi garbników.

**3. Standaryzacja struktury:** Sprawdzone dane zostały sformatowane zgodnie z opracowanym schematem, zapewniając pełną kompatybilność z modułami generatora receptur oraz wyszukiwarki terapeutycznej.

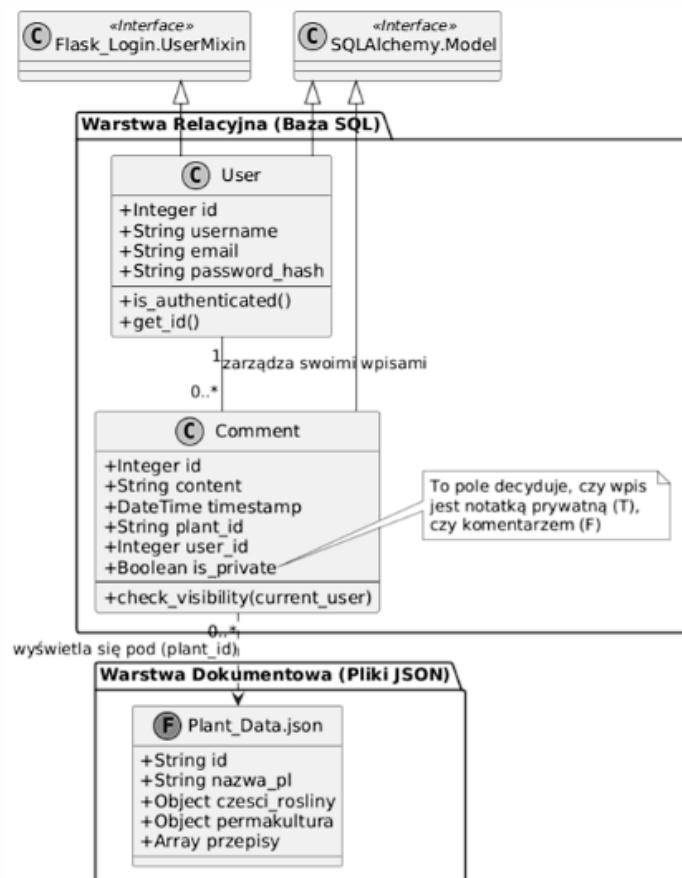
**4. Implementacja i testy:** Gotowe dokumenty zostały zintegrowane z aplikacją Flask, gdzie testowano poprawność ich renderowania na kartach roślin.

## 10. Opis głównych klas, metod, obiektów i algorytmów

W tej sekcji przedstawiono kluczowe komponenty logiki backendowej. Zastosowanie programowania obiektowego (OOP) pozwoliło na odwzorowanie rzeczywistych encji w strukturze kodu.

### 10.1. Logika Modelowania Danych SQL

- **Klasa User:** Dziedziczy po `db.Model` oraz `UserMixin`. Odpowiada za proces uwierzytelniania, zarządzanie sesją oraz relacjami z obiektami podrzędnymi (komentarze).
- **Klasa Comment:** Reprezentuje pojedynczy wpis w systemie. Kluczowym atrybutem jest `plant_id`, który pełni rolę logicznego wskaźnika wiążącego rekord SQL z plikiem JSON.
- **Algorytm widoczności (Prywatność):** Podczas pobierania danych system weryfikuje flagę `is_private`. Rekordy z wartością `True` są filtrowane i wyświetlane wyłącznie autorowi wpisu, co realizuje funkcjonalność prywatnego dziennika obserwacji.



## 10.2. Algorytm A: Heurystyka Generowania Receptur

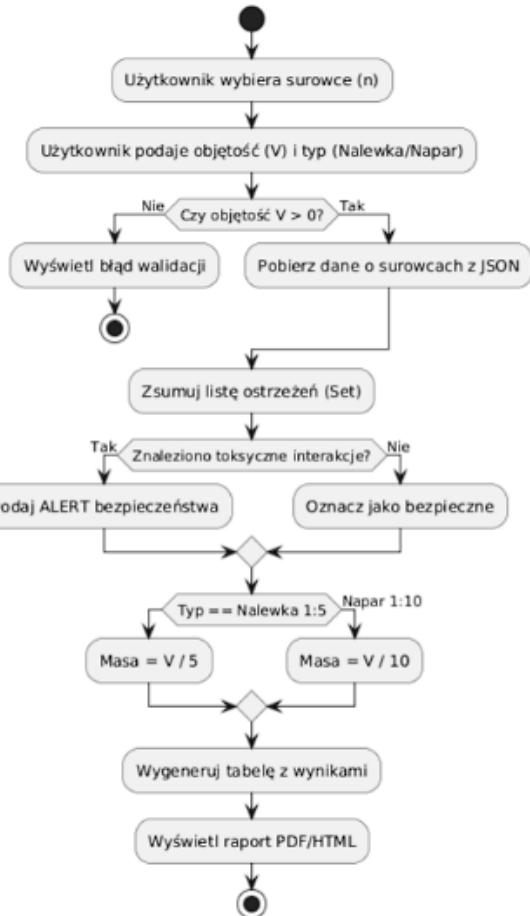
Ten algorytm odpowiada za automatyczne wyliczanie proporcji składników w recepturach zielarskich.

- Dane wejściowe:** n – ilość wybranych surowców, V - objętość docelowa (ml)
- Równanie:** Algorytm implementuje wzór całkowitej masy surowców a następnie oblicza wagową wage dla każdego surowca:

$$M_{wszystkich\ zi\u0142\u00f3w} = \frac{V}{n}$$

$$M_{poszczeg\u0144nego\ surowca} = \frac{M_{wszystkich\ zi\u0142\u00f3w}}{n}$$

- Walidacja:** Metoda dołącza listę ostrzeżeń z przypisanych wobec wszystkich zawartych surowcach i eliminuje duplikaty. Zwracając szereg ostrzeżeń by zachować bezpieczeństwo użycia poszczególnego surowca.
- Efekt:**



Ilustracja 11: Schemat blokowy algorytm receptur

Protok\u0105l Spor\u0144dzenia: NALEWKA	
Sturowiec (Pharmacopoeia)	Ilosc (masa)
Liscie D\u0105b szypu\u0144kowy (Folium Quercus) Działanie: Antyseptyczne, \u0144ciagaj\u0144ce, stosowane g\u0142ownie zewn\u0144trznie w medycynie ludowej.	25.0g
Owoce G\u0142\u0105 jednoszyjkowy (Fructus Crataegi) Działanie: Wzmacniaj\u0144ce naczynia krwiono\u0144ne, witaminowe.	25.0g
Owoce Leszczyna pospolita (Nux Coryli (Orzechy)) Działanie: Wysokoenergetyczne, bogate w witamin\u0144 E i kwasy nienasycone.	25.0g
Ziele Marzanka wonna (Przytulia wonna) (Herba Asperulae (Herba Galii odorati)) Działanie: Uspokajaj\u0144ce, przeciwskurczowe, poprawiaj\u0144ce kr\u0144jenie \u0144ylne, przeciwzapalne.	25.0g
Rozpuszczalnik (Alkohol/Woda)	500ml
<b>\u25bc Przeciwwskazania dla u\u0144tych cz\u0142ci:</b> <ul style="list-style-type: none"> <li>D\u0105b szypu\u0144kowy (liscie): Nie stosowa\u0144 wewn\u0144trznie odwarów z kory przy duzych uszkodzeniach sk\u0142y oraz w ostrych stanach niewydolno\u0144ci serca. Nadmiar garbników hamuje wchlanianie \u0144elaza i witamin.</li> <li>Marzanka wonna (Przytulia wonna) (ziele): Nie nale\u0144y przekracza\u0144 dawk terapeutycznych. Kumaryna w duzych ilo\u0144ciach jest toksyczna dla w\u0144roby. Unika\u0144 cia\u0144y.</li> </ul>	
Wygenerowano przez system Kwiato\u0144nik dnia 17.1.2026	

Ilustracja 12: Protok\u0105l spor\u0144dzania mikstur

### 10.3. Algorytm B: Mapowanie Geometrii Gildii

Algorytm ten służy do wizualizacji powiązań między roślinami w modelu permakulturowym.

- **Obiekt wejściowy:** Lista  $[plant].gilda$  której ilość pozycji jest przypisanych argumentowi  $n$ .
- **Logika pozycjonowania:** Zamiast statycznych obrazów, system oblicza kąt rozmieszczenia dla każdego „towarzysza” rośliny centralnej.
- **Implementacja:** Wykorzystano pętlę w JavaScript, która dla każdego  $i$ -tego elementu listy oblicza zmienną  $angle$  na podstawie wzoru:

$$\text{angle} = i \cdot \left( \frac{360}{n} \right)$$

- **Wynik:** Dane te są przekazywane do arkusza stylów, który za pomocą funkcji transform: rotate() rozmieszcza ikony roślin na okręgu w sposób dynamiczny.
- **Efekt:**



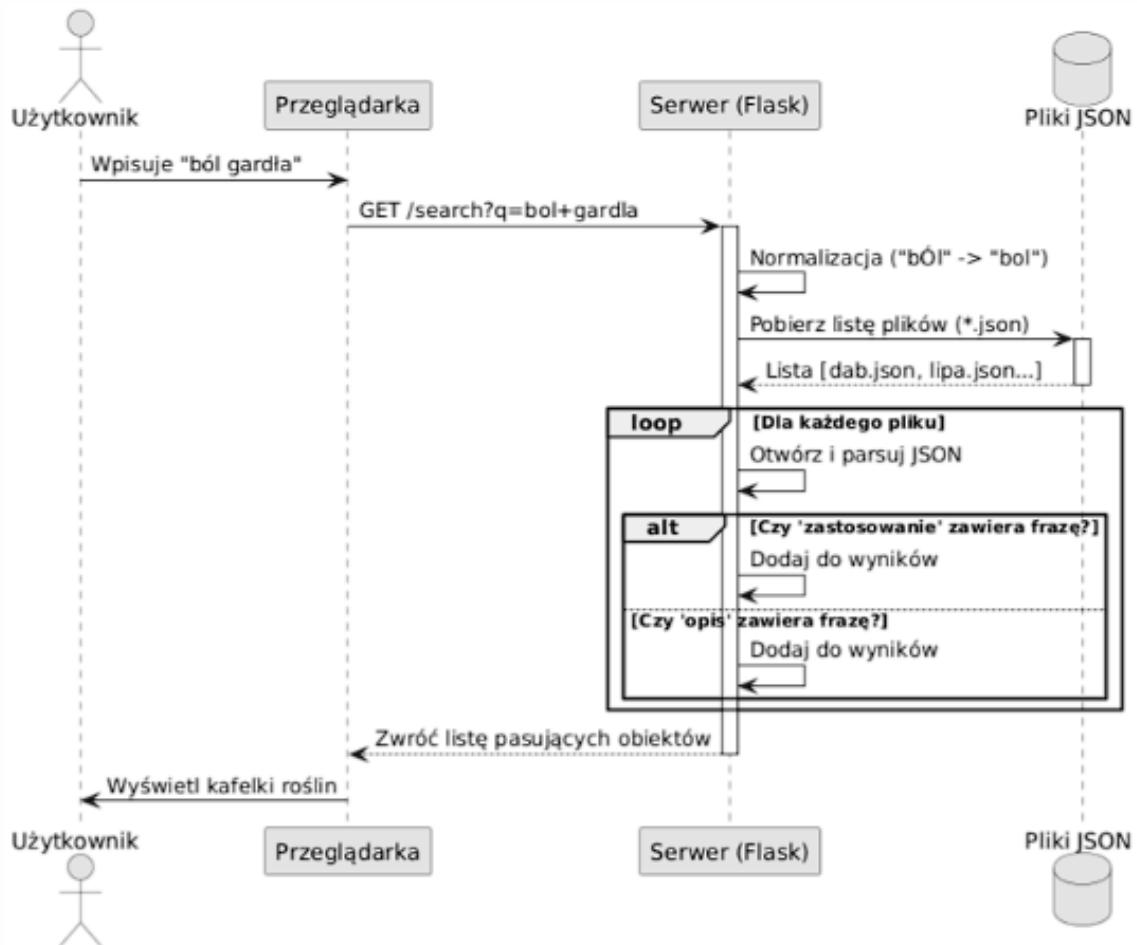
Ilustracja 13: Rozmieszczenie sąsiadów w gildii - Permakultura

## 10.4. Metoda Wyszukiwania Kontekstowego

W przeciwieństwie do prostych zapytań SQL, wyszukiwarka w Kwiatowniku musi przeszukiwać niestrukturalne pliki tekstowe.

- **Metoda search\_plants:**

1. Ładuje wszystkie pliki z folderu data/plants/.
2. Normalizuje zapytanie (małe litery, usuwanie znaków specjalnych).
3. Przeprowadza iterację przez pola: nazwa, właściwości, składniki\_aktywne oraz zastosowanie.medyczne.
4. Zwraca listę obiektów, które spełniają warunek częściowego dopasowania (tzw. *fuzzy matching*).



Ilustracja 14: Diagram sekwencji wyszukiwania

## 10.5. Wykorzystanie narzędzi Open Source

W projekcie świadomie zrezygnowano z pisania własnych systemów niskopoziomowych na rzecz sprawdzonych rozwiązań:

- **Flask-Bcrypt:** Wykorzystany do bezpiecznego, jednostronnego haszowania haseł.
- **Flask-SQLAlchemy:** Zapewnia bezpieczną komunikację z bazą danych, chroniąc przed SQL Injection.
- **Flask-Login:** Kluczowa biblioteka zarządzająca sesjami użytkowników, odpowiadająca za bezpieczne uwierzytelnianie i separację dostępu do notatek prywatnych.
- **Jinja2 Filters:** Zastosowanie wbudowanych filtrów (np. `|truncate`, `|safe`) pozwala na bezpieczne renderowanie treści HTML oraz dynamiczne formatowanie podglądu opisów botanicznych.
- **Python JSON Library:** Standardowa biblioteka wykorzystana do szybkiego i bezbłędnego mapowania struktury plików JSON na obiekty języka Python wewnętrz logiki serwera.

## **III. DOKUMENTACJA KOŃCOWA I UŻYTKOWNIKA**

### **11. Przebieg uruchamiania projektu (Instrukcja wdrożeniowa)**

Sekcja ta opisuje kroki niezbędne do przygotowania środowiska lokalnego oraz poprawnej inicjalizacji aplikacji „Kwiatownik”.

#### **11.1. Wymagania wstępne**

Przed przystąpieniem do instalacji należy upewnić się, że w systemie zainstalowane są:

- Python 3.8: Główny interpreter języka.
- Menedżer pakietów *pip*: Do instalacji bibliotek zewnętrznych.
- Dostęp do terminala/wiersza poleceń: Systemy Windows (*CMD/PowerShell*), macOS lub Linux.

#### **11.2. Przygotowanie środowiska wirtualnego**

W celu uniknięcia konfliktów między bibliotekami różnych projektów, zaleca się stosowanie izolowanego środowiska wirtualnego:

1. Otwórz terminal w folderze głównym projektu.
2. Utwórz środowisko komendą: PYTHON -M VENV VENV.
3. Aktywuj środowisko:
  - Windows: VENV\SCRIPTS\ACTIVATE
  - macOS/Linux: SOURCE VENV/BIN/ACTIVATE

#### **11.3. Instalacja zależności**

Wszystkie niezbędne biblioteki (FLASK, SQLALCHEMY, FLASK-LOGIN, WERKZEUG) zostały wyszczególnione w pliku REQUIREMENTS.TXT. Ich instalacja odbywa się za pomocą jednej komendy:

Bash:

```
PIP INSTALL -R REQUIREMENTS.TXT
```

#### **11.4. Inicjalizacja bazy danych i plików JSON**

Aplikacja została zaprojektowana tak, aby maksymalnie uprościć pierwsze uruchomienie:

- Baza SQL: Przy pierwszym starcie system automatycznie tworzy plik KWiatownik.DB w folderze głównym oraz generuje niezbędne tabele (USER, COMMENT).
- Baza NoSQL (JSON): Należy upewnić się, że w folderze DATA/PLANTS/ znajdują się pliki opisu roślin. Jeśli folder nie istnieje, aplikacja utworzy go automatycznie przy pierwszym uruchomieniu.

## **11.5. Uruchomienie serwera deweloperskiego**

Aby wystartować aplikację, należy wykonać komendę:

Bash:

```
PYTHON APP . PY
```

Po poprawnym uruchomieniu, terminal wyświetli adres lokalny: `HTTP://127.0.0.1:5000`.

Należy go skopiować i wkleić do przeglądarki internetowej.

## **11.6. Konfiguracja konta administratora**

Aplikacja posiada wbudowaną logikę bezpieczeństwa: pierwszy zarejestrowany użytkownik w systemie automatycznie otrzymuje uprawnienia Administratora (`IS_ADMIN = TRUE`).

Pozwala to na pełne zarządzanie komentarzami i testowanie funkcji moderacyjnych bezpośrednio po instalacji.

# **12. Podręcznik użytkownika**

Niniejszy poradnik opisuje standardowe scenariusze korzystania z aplikacji Kwiatownik i ma na celu ułatwienie korzystanie z systemu a także ułatwienie w nawigacji po jej modułach.

## **12.1. Rejestracja i Personalizacja profilu**

Dostęp do podstawowej encyklopedii jest otwarty, jednak pełna funkcjonalność wymaga założenia konta.

- Rejestracja: Użyj zakładki „Zarejestruj się”, podając unikalną nazwę użytkownika.
- Logowanie: Po zalogowaniu system aktywuje Twój Osobisty Dziennik Zielarski, umożliwiając zapisywanie notatek oraz korzystanie z działu komentarzy.

## **12.2. Przeglądanie Encyklopedii i Wyszukiwanie Terapeutyczne**

Kwiatownik oferuje dwa tryby odnajdywania wiedzy:

1. Lista Alfabetyczna: Widoczna na stronie głównej, pozwala na szybki wgląd w dostępne gatunki.
2. Wyszukiwarka Terapeutyczna: Kluczowe narzędzie dla zielarzy. Wpisz objaw (np. *bezsenność*, *stany zapalne*) lub właściwość (np. *wykrzutne*). System przeszuka całą strukturę bazy i wyświetli rośliny, których składniki odpowiadają Twojemu zapytaniu.

## 12.3. Instrukcja obsługi Karty Rośliny

Karta rośliny to interaktywne kompendium wiedzy, do którego dostęp uzyskujemy po kliknięciu nazwy wybranego gatunku w encyklopedii lub wynikach wyszukiwania. Poniżej znajduje się opis poszczególnych modułów informacyjnych:

# Dąb szypułkowy

*Quercus robur / Rodzina: Bukowate*

Potężne drzewo liściaste, długowieczne, o szerokiej koronie. Jest najważniejszym gatunkiem lasotwórczym w Europie Środkowej.

Ilustracja 15: Tytuł rośliny - Karta rośliny

### A. Morfologia i surowce zielarskie

Sekcja ta prezentuje szczegółową budowę botaniczną rośliny wraz z wykazem części użytkowych, ich składem chemicznym oraz właściwościami surowca.

#### Morfologia i Surowce Zielarskie

KORA	LISIE	OWOCE
<p><b>Cortex Quercus</b></p> <p><b>Cechy:</b> Gładka i lśniąca u młodych okazów, z czasem staje się grubą, głęboko spekaną skorupą.</p> <p><b>Właściwości:</b> Silnie ściągające, przeciwzapalne, hamujące drobne krwawienia.</p> <p><b>Składniki:</b> Garbniki taninowe (do 20%), Kwasy fenolowe, Flavonoidy, Sole mineralne</p> <p><b>Zbiór:</b> Wiosna (marzec-kwiecień), zanim pojawią się liście, z młodych, gładkich gałęzi.</p>	<p><b>Folium Quercus</b></p> <p><b>Cechy:</b> Krótkoogonkowe, o nieregularnie klapowanych brzegach, skórzaste, z charakterystycznymi uszkiem u nasady.</p> <p><b>Właściwości:</b> Antyseptyczne, ściągające, stosowane głównie zewnętrznie w medycynie ludowej.</p> <p><b>Składniki:</b> Garbniki, Kwercyna, Związki żywicze</p> <p><b>Zbiór:</b> Czerwiec - Lipiec (młode liście).</p>	<p><b>Glandes Quercus (Żołędzie)</b></p> <p><b>Cechy:</b> Owalne orzechy osadzone w miseczkach na długich szypulkach (stąd nazwa gatunku).</p> <p><b>Właściwości:</b> Odżywcze, po obróbce termicznej i wodnej działają przeciwbiegunkowo i wzmacniająco.</p> <p><b>Składniki:</b> Skrobia, Tłuszcze, Bialka, Garbniki (wymagają wylugowania)</p> <p><b>Zbiór:</b> Wrzesień - Październik (gdy same opadają).</p>

Ilustracja 16: Morfologia i surowce Karta rośliny

### B. Kalendarz prac i zjawisk fenologicznych

Moduł ten zawiera harmonogram działań ogrodniczych i terminów zbiorów, uwzględniający fazy fenologiczne rośliny oraz zalecenia biodynamiczne.

#### Kalendarz Prac i Zjawisk Fenologicznych

Harmonogram uwzględniający fazy księżyca oraz warunki atmosferyczne.

<p><b>Siew (bezpośredni)</b></p> <p>Siew dojrzałych żołędzi bezpośrednio do gruntu na głębokość 5-7 cm.</p> <p><b>Księżyc:</b> Księżyca ubijającego (faza korzenia)</p> <p><b>Pora dnia:</b> Poranek</p> <p><b>Pogoda:</b> Dzień bezmroźny, wilgotna gleba</p> <p><b>Wpływ kosmiczny:</b> Sprzyja szybkiemu ukorzenianiu i zapobiega gniciu nasion.</p>	<p><b>Zbiór kory</b></p> <p>Pozyskiwanie młodej, gładkiej kory z gałęzi przeznaczonych do przycinki.</p> <p><b>Księżyc:</b> Nów lub pierwszy kwartal</p> <p><b>Pora dnia:</b> Przedpołudnie</p> <p><b>Pogoda:</b> Słonecznie, sucho (ważne dla procesu suszenia)</p> <p><b>Wpływ kosmiczny:</b> Wzrost soków w górnych partiach rośliny zwiększa zawartość garbników.</p>	<p><b>Obserwacja kwitnienia</b></p> <p>Monitorowanie rozwoju kwiatów męskich (kotek) i żeńskich.</p> <p><b>Księżyc:</b> Pełnia</p> <p><b>Pora dnia:</b> Południe</p> <p><b>Pogoda:</b> Lekki wiatr (sprzyja zaplanowaniu krzyżowemu)</p> <p><b>Wpływ kosmiczny:</b> Maksymalna vitalność rośliny i emisja pyiku.</p>
---	---	--

Ilustracja 17: Kalendarz Prac - Karta rośliny

## C. Apteka i kuchnia

W tej sekcji użytkownik odnajdzie gotowe receptury na preparaty lecznicze oraz inspiracje kulinarne wykorzystujące jadalne części danej rośliny.

### Apteka i Kuchnia

#### Przepisy Medyczne

**Odwar do nasiadówek i płukanek** (Surowiec: kora)

*Składniki:* 2 łyżki kory, 1 litr wody

Gotować 15 minut, odstawić na 10 minut. Stosować zewnętrznie.

**Mąka żolędziowa** (Surowiec: owoce)

*Składniki:* Dojrzałe żolędzie, woda, popiół drzewny (opcjonalnie)

Żolędzie obrać, moczyć w wodzie zmieniając ją przez 3-4 dni (lugowanie garbników), wysuszyć i zmielić.

#### Przepisy Kulinarne

##### Staropolska kawa żolędziowa

Użyta część: owoce

Zmielone składniki gotować 2-3 minuty. Podawać z miodem i mlekiem.

*Ilustracja 18: Przepisy medyczne i kulinarne - Karta rośliny*

## D. Wiedza i ekologia

Tutaj znajdują się kluczowe informacje o wymaganiach uprawowych (gleba, słońce) oraz niezbędne ostrzeżenia dotyczące bezpieczeństwa i przeciwwskazań w stosowaniu.

### Wiedza i Ekologia

#### ⚠ Ostrzeżenia

Nie stosować wewnętrznie odwarów z kory przy dużych uszkodzeniach skóry oraz w ostrych stanach niewydolności serca. Nadmiar garbników hamuje wchłanianie żelaza i witamin.

## E. Ciekawostki

Ostatnia sekcja zawiera zbiór faktów historycznych, etymologicznych i ludowych, które wzbogacają ogólną wiedzę o danym gatunku.

### Ciekawostki

- Dęby są jednymi z najbardziej miododajnych drzew nie przez kwiaty, lecz przez spadź dębową.
- Kora dębu była historycznie głównym surowcem w garbarstwie do wyprawiania skór.
- W legendach dąb był drzewem kosmicznym, łączącym świat podziemny z niebem.

*Ilustracja 20: Ciekawostki - Karta rośliny*

## F. Gildia Permakulturowa

Moduł wizualizuje system naczyń połączonych w ogrodzie, wskazując rośliny towarzyszące i ich funkcje wspierające w ramach gildii ekologicznej.

 **Gildia Permakulturowa**

**Rośliny wspierające i ich rola:**

<b>Leszczyna pospolita</b> Piętro niższe, produkcja biomasy i orzechów	<b>Głów jednoszyjkowy</b> Strażnik (bariera kolczasta) i wsparcie zapylaczy
<b>Śliwa tarnina</b> Zarośla ochronne i stabilizacja gleby	<b>Poziomka pospolita</b> Żywy mulcz (okrywa runa)
<b>Marzanka wonna (Przytulia wonna)</b> Runo leśne i bioindykator gleby	<b>Trufle</b> Symbioza mikoryzowa (strefa korzeniowa)

**Funkcje systemowe:**

- Pompa mineralna (głębokie korzenie wyciągają minerały na powierzchnię)
- Naturalny wiatrochron
- Producent żołędzi dla systemów wypasu (silvopasture)

Ilustracja 21: Opis gildii permakulturowych - Karta rośliny

## 12.4. Zarządzanie Wiedzą: Komentarze vs Prywatne Notatki

Moduł interakcji pozwala na personalizację bazy danych poprzez dodawanie własnych treści, które dzielą się na publiczne opinie oraz całkowicie poufne zapiski.

### A. Publikowanie komentarzy społecznościowych

Użytkownik może brać udział w publicznej dyskusji, dzieląc się swoimi doświadczeniami z uprawy lub zbioru roślin, co buduje wspólną bazę wiedzy wszystkich użytkowników Kwiatownika.

### B. Tworzenie Prywatnych Noteatek Zielarskich

Poprzez zaznaczenie opcji „Zapisz jako prywatną notatkę”, system tworzy chroniony wpis widoczny wyłącznie dla zalogowanego autora, idealny do zapisywania lokalizacji stanowisk czy dat nastawienia własnych preparatów.

### C. Wizualna identyfikacja rodzaju wpisu

Dla zapewnienia przejrzystości i poczucia bezpieczeństwa, Twoje prywatne notatki są wyróżnione w systemie charakterystycznym żółtym tłem, co natychmiast odróżnia je od białych komentarzy publicznych.

#### Interakcje i Notatki

 Zapisz jako prywatną notatkę (widoczna tylko dla Ciebie) B

Opublikuj wpis 2026-01-15 10:32

Tomasz Nowak MOJA NOTATKA 2026-01-15 10:32

Kawa żołędziowa dobrze smakuje z dodatkiem masla.

[Edytuj] [Usuń] C

Tomasz Nowak 2026-01-15 10:31

Na północy

[Edytuj] [Usuń] D

Adam Kowalski 2026-01-15 10:30

Gdzie można znaleźć tą roślinę w Chorzowskim parku?

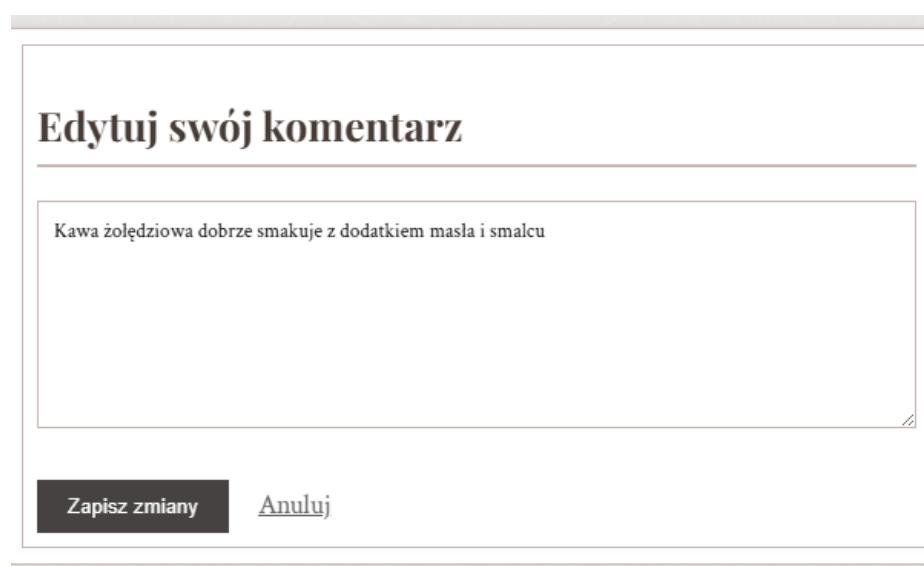
A B

Ilustracja 22: Wygląd sekcji komentarze - Komentarze

## D. Edycja i zarządzanie własnymi wpisami

Aplikacja zapewnia pełną kontrolę nad wprowadzonymi treściami, umożliwiając ich modyfikację lub trwałe usunięcie w dowolnym momencie przez autora.

1. Wybór wpisu do edycji Każdy komentarz lub notatka Twojego autorstwa posiada dedykowane przyciski funkcyjne, które pozwalają na błyskawiczne przejście do trybu poprawiania treści.
2. Formularz modyfikacji treści Po kliknięciu przycisku edycji, system otwiera dedykowany formularz, w którym możesz zaktualizować tekst swojego spostrzeżenia, zachowując jego dotychczasową datę dodania lub status prywatności.



Ilustracja 23: Edycja komentarza/notatki - Komentarze

## Interakcje i Notatki



Tomasz Nowak MOJA NOTATKA

2026-01-15 10:32

Kawa żołędziowa dobrze smakuje z dodatkiem masła i smalcu

[Edytuj] [Usuń]

## 12.5. Praca z Generatorem Receptur

Generator receptur to zaawansowane narzędzie obliczeniowe, które na podstawie parametrów wprowadzonych przez użytkownika automatyzuje proces tworzenia bezpiecznych mieszanek ziołowych.

### A. Inteligentne filtrowanie surowców według właściwości

Zintegrowana wyszukiwarka z systemem podpowiedzi pozwala na dynamiczne filtrowanie listy dostępnych surowców poprzez wpisanie pożądanego działania terapeutycznego (np. „wykrztuśne”).

### B. Konfiguracja parametrów i typu preparatu

Użytkownik wybiera bazę receptury (np. nalewka, napar) oraz docelową objętość płynu, co aktywuje algorytm dobierający odpowiednie proporcje masy surowca do rozpuszczalnika.

### C. Analiza wyników i proporcji wagowych

Na podstawie wybranych komponentów system generuje tabelę z dokładnymi wagami każdego surowca w gramach, uwzględniając podział na części składowe mieszanki.

# Zaawansowany Kreator Receptur

Typ mikstury:

Objętość docelowa (ml):

Wyszukaj po właściwościach (np. uspokajające, przeciwzapalne):

A

Leszczyna pospolita

owoce (wysokoenergetyczne, bogate w witaminę e i kwasy nienasycone.)

Poziomka pospolita

owoce (odżywcze, bogate w witaminę c i żelazo.)

B

Generuj Protokół Farmakopealny

Ilustracja 25: Wybór surowców - Zaawansowany Kreator Receptur

## D. Zbiorczy system ostrzeżeń bezpieczeństwa

Protokół Sporządzenia: NALEWKA	
Surowiec (Pharmacopoeia)	Ilość (masa)
Liscie Dąb szypułkowy (Folium Quercus) Działanie: Antyseptyczne, ściągające, stosowane głównie zewnętrznie w medycynie ludowej.	C 30.0g
Kwiaty Głog jednoszyjkowy (Inflorescentia Crataegi) Działanie: Nasercowe, obniżające ciśnienie krwi.	30.0g
Liscie Leszczyna pospolita (Folium Coryli) Działanie: ściągające, przeciwzapalne, uszczelniające naczynia krwionośne.	30.0g
Ziele Marzanka wonna (Przytulia wonna) (Herba Asperulae (Herba Galii odorati)) Działanie: Uspokajające, przeciwskurczowe, poprawiające krażenie żylnie, przeciwzapalne.	30.0g
Liscie Śliwa tarnina (Folium Pruni spinosae) Działanie: Moczopędne, pomocniczo w stanach zapalnych dróg moczowych.	30.0g
Rozpuszczalnik (Alkohol/Woda)	750ml

**⚠️ Przeciwwskazania dla użytych części:**

**D**

- Marzanka wonna (Przytulia wonna) (ziele): Nie należy przekraczać dawek terapeutycznych. Kumaryna w dużych ilościach jest toksyczna dla wątroby. Unikać w ciąży.
- Dąb szypułkowy (liscie): Nie stosować wewnętrznie odwarów z kory przy dużych uszkodzeniach skóry oraz w ostrych stanach niewydolności serca. Nadmiar garbników hamuje wchłanianie żelaza i witamin.
- Śliwa tarnina (liscie): Pestki tarniny zawierają amigdalinę (kwas pruski) i nie powinny być spożywane. Przetwory z owoców należy zawsze drylować.

Wygenerowano przez system Kwiatownik dnia

Ilustracja 26: Wygenerowany protokół przepisu - Zaawansowany Kreator Receptur

W dolnej części wyniku generator automatycznie agreguje i wyświetla wszystkie przeciwwskazania dotyczące wybranych roślin, co pozwala na natychmiastową ocenę bezpieczeństwa skomponowanej mieszanki.

## 13. Rozwiązywanie problemów (FAQ)

- Nie widzę opcji dodawania komentarzy: Upewnij się, że jesteś zalogowany.
- Wyszukiwarka nie znajduje rośliny: Spróbuj wpisać nazwę bez polskich znaków lub użyj wyszukiwarki terapeutycznej, wpisując jedynie fragment nazwy.
- Czy moje notatki są bezpieczne? Tak, system filtry dane po ID użytkownika bezpośrednio w bazie SQL, więc nikt postronny nie ma do nich wglądu.

- Brak wyświetlania Mapy Gildii dla konkretnego gatunku: Jeśli sekcja gildii jest pusta, oznacza to, że dany plik JSON nie posiada jeszcze zdefiniowanych relacji ekologicznych lub struktura klucza "permakultura" została naruszona.
- Komunikat: „Nieprawidłowy login lub hasło” System wyświetla to ostrzeżenie, gdy wprowadzone dane uwierzytelniające nie zgadzają się z zaszyfrowanym skrótem (hash) zapisanym w bazie danych SQLite.
- Brak wyników w wyszukiwarce terapeutycznej W przypadku wpisania frazy, której system nie odnajdzie w polach właściwości plików JSON, użytkownik otrzymuje powiadomienie z sugestią użycia ogólniejszych terminów lub sprawdzenia pisowni.

## 14. Spostrzeżenia i zalecenia do użytkowania

- **Krytyczna weryfikacja danych**

Przed przystąpieniem do sporządzania preparatów należy bezwzględnie zweryfikować sekcję ostrzeżeń medycznych, gdyż automatyzacja algorytmiczna stanowi jedynie wsparcie dla profesjonalnej wiedzy farmaceutycznej.

- **Synergia wyszukiwania i projektowania ekosystemów**

Najwyższą efektywność pracy z systemem zapewnia łączenie wyszukiwarki terapeutycznej z modułem gildii, co pozwala na jednoczesne znalezienie leku i zaprojektowanie wspierającego go ogrodu.

- **Kalibracja bazy poprzez Notatki Prywatne**

Prowadzenie osobistych zapisków umożliwia precyzyjne korygowanie ogólnych przepisów o realną moc surowców pozyskanych z konkretnych, lokalnych stanowisk i zbiorów.

- **Konserwacja techniczna i integralność danych**

Dla zachowania pełnej sprawności aplikacji wymagana jest walidacja struktury plików JSON przy dodawaniu nowych gatunków oraz systematyczne tworzenie kopii zapasowych bazy danych SQLite.

- **Wiarygodność wiedzy weryfikowanej przez człowieka**

Mimo wykorzystania sztucznej inteligencji do gromadzenia danych, każda informacja przeszła proces manualnej weryfikacji, co czyni system bezpiecznym punktem wyjścia do domowej fitoterapii

## 15. Scenariusze testowe i walidacja systemu

W celu zapewnienia niezawodności aplikacji przeprowadziłem szereg testów funkcjonalnych, mających na celu weryfikację poprawności działania kluczowych modułów oraz odporności systemu na błędy użytkownika.

### 15.1. Testy akceptacyjne i funkcjonalne

**Tabela 4: testy akceptacyjne i funkcjonalne**

ID	Nazwa testu	Opis działań	Oczekiwany rezultat	Status
T.01	Uwierzytelnianie	Próba logowania przy użyciu błędego hasła.	System odrzuca dostęp i wyświetla komunikat: „Nieprawidłowy login lub hasło”.	Pozytywny
T.02	Wyszukiwarka	Wpisanie frazy „antybiotyk” w polu wyszukiwania.	System poprawnie filtuje pliki JSON i wyświetla listę roślin o działaniu antybiotycznym (np. Czosnek).	Pozytywny
T.03	Generator (Walidacja)	Wpisanie wartości „-500” w polu objętości płynu.	Formularz blokuje wysyłkę danych, żądając podania wartości dodatniej.	Pozytywny
T.04	Generator (Obliczenia)	Wybór nalewki (1:5) dla objętości 250 ml.	Algorytm zwraca wynik „50.0 g” surowca oraz agreguje odpowiednie ostrzeżenia.	Pozytywny
T.05	Mapa Gildii	Kliknięcie w węzeł rośliny towarzyszącej na grafie.	System przekierowuje użytkownika do szczegółowej karty wybranej rośliny (nawigacja relacyjna).	Pozytywny
T.06	Prywatność notatek	Próba odczytu notatki użytkownika A przez użytkownika B.	System izoluje rekordy w bazie SQL; notatki są niewidoczne dla osób postronnych.	Pozytywny
T.07	Integralność JSON	Usunięcie wymaganego klucza „surowce” z pliku rośliny.	Serwer przechwytuje błąd (UndefinedError) i wyświetla stronę błędu zamiast awarii całej aplikacji.	Pozytywny

### 15.2. Weryfikacja responsywności (UX)

Przeprowadzono testy na urządzeniach o różnych rozdzielczościach ekranu (Desktop, Tablet, Mobile). Wykorzystano narzędzia deweloperskie przeglądarki do symulacji widoków.

- Wynik:** Menu nawigacyjne oraz interaktywna mapa gildii poprawnie adaptują się do szerokości ekranu, zachowując czytelność tekstów i dostępność przycisków akcji.



Ilustracja 27: Responsywność - Gilda Permakulturowych

### 15.3. Przykład testu wstrzyknięcia kodu SQL (SQL Injection)

W aplikacji, gdzie komentarze są zapisywane w bazie SQLite za pomocą SQLAlchemy, test polega na próbie „oszukania” zapytania SQL poprzez wprowadzenie znaków sterujących (takich jak średnik, apostrof czy komentarz SQL - -).

#### Scenariusz testowy:

- **Cel:** Sprawdzenie, czy system poprawnie neutralizuje (escapuje) znaki specjalne i nie wykonuje ich jako poleceń bazy danych.
- **Miejsce testu:** Pole tekstowe „Treść komentarza” na karcie rośliny.

#### Warianty wpisanego tekstu:

1. Próba usunięcia tabeli (Destrukcyjny):

- **Wpisana treść:** BARDZO POMOCNY OPIS'); DROP TABLE USERS; --
- **Mechanizm ataku:** Napastnik liczy na to, że system domknie zapytanie INSERT apostrofem i nawiasem, a następnie wykona nowe polecenie usuwające tabelę użytkowników.

Tomasz Nowak

2026-01-17 10:34

Bardzo pomocny opis'); DROP TABLE users; --

[Edytuj] [Usuń]

Ilustracja 28: Test - SQL Injection

#### 2. Próba obejścia logiki (Tautologia):

- **Wpisana treść:** OR 1=1 --
- **Mechanizm ataku:** Stosowany głównie w polach logowania lub wyszukiwania, aby wymusić na bazie zwrócenie wszystkich rekordów (ponieważ  $1=1$  jest zawsze prawda).

Tomasz Nowak

2026-01-17 10:35

OR 1=1 --

[Edytuj] [Usuń]

Ilustracja 29: Test - Tautologia

**Wynik testu:** Dzięki zastosowaniu SQLAlchemy ORM, system automatycznie sparametryzował zapytanie. Ciąg znaków nie został potraktowany jako instrukcja systemowa, lecz jako zwykły tekst.

1. Tabela comments nie została usunięta.
2. W bazie danych zapisał się dokładnie taki tekst, jaki wpisano: 'ŚWIĘTNA ROŚLINA'); *DROP TABLE COMMENTS; --*.
3. Na stronie wyświetlił się pełny komentarz wraz z „groźnym” kodem, co potwierdza, że został on w pełni zneutralizowany.

## 16. Wykryte błędy w działaniu

W procesie testowania oprogramowania zidentyfikowano poniższe błędy, które zostały wyeliminowane w finalnej wersji systemu:

- **Wrażliwość wyszukiwarki na wielkość liter (Case-sensitivity)**
  - **Problem:** System nie zwracał wyników, gdy użytkownik wpisał nazwę rośliny wielką literą (np. „Dąb” zamiast „dąb”), mimo że rekord istniał w bazie.
  - **Rozwiążanie:** Zaimplementowano funkcję normalizującą ciągi znaków. Przed przeszukaniem bazy zarówno zapytanie użytkownika, jak i treść plików JSON są konwertowane do małych liter (metoda `.lower()`), co zapewnia skuteczność wyszukiwania niezależnie od użytej pisowni.
- **Formatowanie wyników w Generatorze Receptur**
  - **Problem:** Algorytm obliczeniowy zwracał wartości z nadmierną precyzją zmiennoprzecinkową (np. `16.666667 g`), co było nieczytelne i niepraktyczne dla użytkownika.
  - **Rozwiążanie:** Wprowadzono funkcję zaokrąglającą wyniki końcowe do jednego miejsca po przecinku. Dodatkowo, w celu standaryzacji procesów laboratoryjnych, ograniczono możliwość wyboru objętości docelowej preparatu do trzech stałych wartości: 500 ml, 750 ml oraz 1000 ml.
- **Kodowanie znaków diakrytycznych (Encoding)**
  - **Problem:** W opisach pobieranych z plików zewnętrznych występuły błędy w wyświetlaniu polskich znaków narodowych (tzw. "krzaki").
  - **Rozwiążanie:** Zmodyfikowano moduł obsługi plików, wymuszając standard kodowania UTF-8 (`encoding='utf-8'`) podczas operacji odczytu plików JSON przez serwer Flask.

## 17. Spisy ilustracji i rysunków

Ilustracja 1: Diagram przypadków użycia.....	6
Ilustracja 2: ID, nazwa, opis - JSON.....	12
Ilustracja 3: Części i składniki rośliny - JSON.....	12
Ilustracja 4: Zastosowanie - JSON.....	13
Ilustracja 5: Ostrzeżenia - JSON.....	13
Ilustracja 6: Permakultura - JSON.....	13
Ilustracja 7: Kalendarz ogrodnika - JSON.....	14
Ilustracja 8: Przepisy - JSON.....	15
Ilustracja 9: Diagram architektury wdrożenia.....	17
Ilustracja 10: Diagram klas - Bazy Danych.....	19
Ilustracja 11: Schemat blokowy algorytm receptur.....	20
Ilustracja 12: Protokół sporządzania mikstur.....	20
Ilustracja 13: Rozmieszczenie sąsiadów w gildii - Permakultura.....	21
Ilustracja 14: Diagram sekwencji wyszukiwania.....	22
Ilustracja 15: Tytuł rośliny - Karta rośliny.....	26
Ilustracja 16: Morfologia i surowce Karta rośliny.....	26
Ilustracja 17: Kalendarz Prac - Karta rośliny.....	26
Ilustracja 18: Przepisy medyczne i kulinarnie - Karta rośliny.....	27
Ilustracja 19: Ostrzeżenia - Karta rośliny.....	27
Ilustracja 20: Ciekawostki - Karta rośliny.....	27
Ilustracja 21: Opis gildii permakulturowych - Karta rośliny.....	28
Ilustracja 22: Wygląd sekcji komentarze - Komentarze.....	28
Ilustracja 23: Edycja komentatorza/notatki - Komentarze.....	29
Ilustracja 24: Wygląd sekcji komentarzy - Komentarze.....	29
Ilustracja 25: Wybór surowców - Zaawansowany Kreator Receptur.....	30
Ilustracja 26: Wygenerowany protokół przepisu - Zaawansowany Kreator Receptur.....	31
Ilustracja 27: Responsywność - Gildia Permakulturowych.....	33
Ilustracja 28: Test - SQL Injection.....	34
Ilustracja 29: Test - Tautologia.....	34

## **18. Bibliografia i źródła**

- **Literatura przedmiotu (Merytoryka)**

1. Farmakopea Polska (wydanie aktualne) – podstawowe źródło standardów dotyczących przygotowania preparatów roślinnych i proporcji surowców.
2. Ożarowski A., Jaroniewski W., "Rośliny lecznicze i ich zastosowanie" – klasyczne opracowanie wykorzystane do weryfikacji właściwości terapeutycznych opisanych w plikach JSON.

- **Dokumentacja techniczna (Technologie IT)**

3. Dokumentacja frameworka Flask  
(<https://www.google.com/search?q=flask.palletsprojects.com>) – źródło dotyczące implementacji tras (routes), obsługi sesji oraz systemu szablonów Jinja2.
4. SQLAlchemy Documentation – wytyczne dotyczące budowania relacyjnych modeli danych i mapowania obiektowo-relacyjnego (ORM).
5. SQLite Standard Library – dokumentacja silnika bazy danych w zakresie integralności danych i operacji SQL.
6. Werkzeug Security Helpers – specyfikacja dotycząca bezpiecznego haszowania haseł za pomocą algorytmu PBKDF2.

- **Zasoby internetowe i narzędzia AI**

7. OpenAI / Anthropic Language Models – wykorzystane jako narzędzia wspierające proces agregacji i ustrukturyzowania danych w formacie JSON.
8. DB Browser for SQLite ([sqlitebrowser.org](http://sqlitebrowser.org)) – narzędzie wykorzystane do projektowania i manualnej weryfikacji tabel bazy danych.

## **IV. ZAŁĄCZNIKI I DODATKI**

### **19. Nośniki elektroniczne Flash**

Opis zawartości nośnika: kod źródłowy, pliki bazy danych, listing kodu w PDF, dokumentacja w formie edytowalnej.