



GreenLeaf Guide - Архітектура програмного проекту

ЗАГАЛЬНА ІНФОРМАЦІЯ

Тема проекту: База даних отруйних рослин з аналітичними функціями

Способ використання: Графічний інтерфейс з можливістю аналізу даних

Технології: Python 3.9+, Java 8+, Swing, JSON, REST API

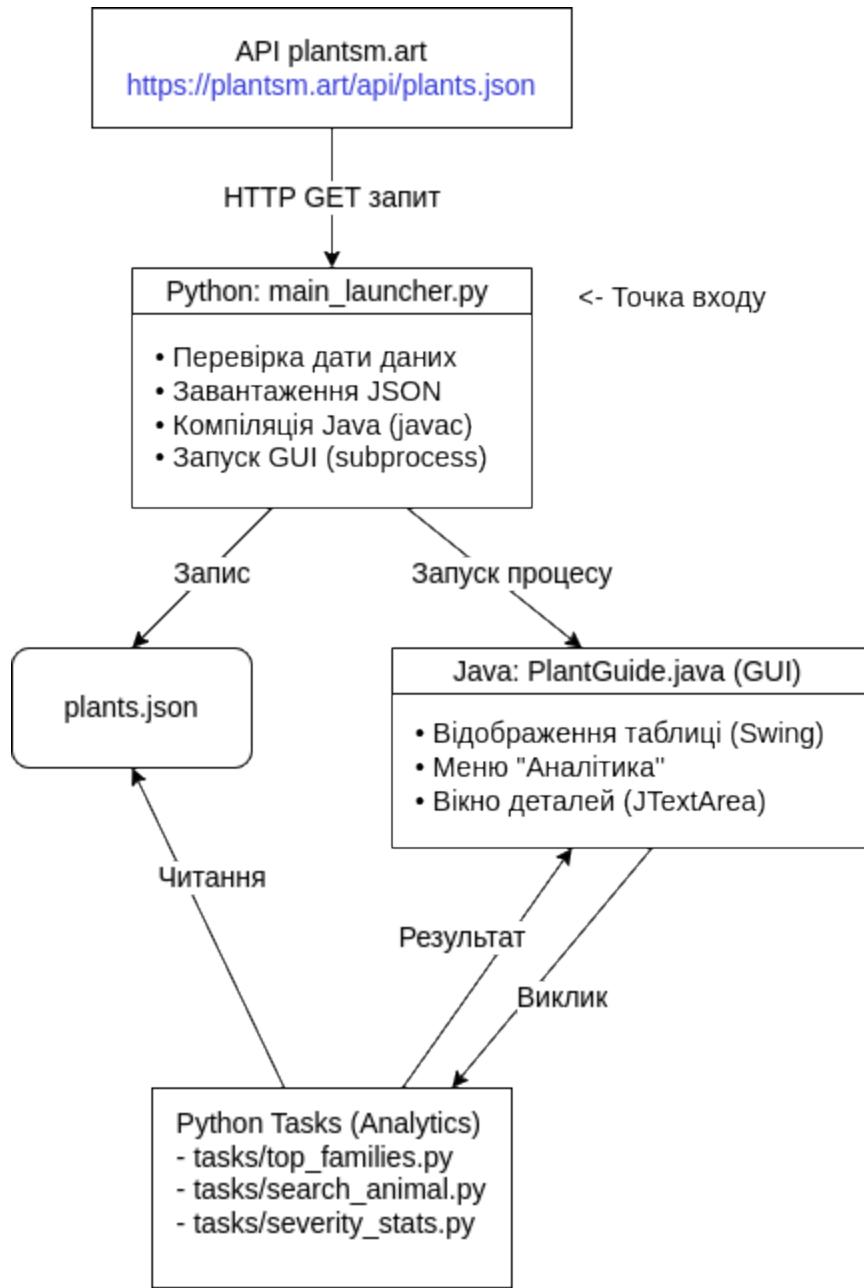
Роботу виконали:

Альона Саєнко (ПМІ-43):

- розробка графічного інтерфейсу користувача на Java (Swing), який реалізує відображення таблиці рослин з сортуванням, кольорове кодування рівнів небезпеки та вікна детальної інформації;
- реалізація функцій для визначення найпоширеніших ботанічних родин (top_families.py), функції пошуку загроз для конкретних тварин (search_animals.py), збору статистики розподілу рівнів токсичності (severity_stats.py)

Олег Зозуля (ПМІ-42):

- створення основного контролера системи (main_launcher.py), який забезпечує завантаження даних з API, валідацію JSON, компіляцію Java-компонентів та автоматичну збірку релізної версії (створення .jar, генерація скриптів запуску);
- реалізація функцій генерації інструкцій першої допомоги з визначенням терміновості (first_aid.py), системи пошуку рослин за симптомами отруєння (search_symptoms.py) та підбору безпечних альтернатив токсичним рослинам (safe_alternatives.py).



1. JSON-файл

Веб-адреса: <https://plantsm.art/api/plants.json>

Формат: JSON

Опис: База даних токсичних рослин з детальною інформацією про токсичність, симптоми отруєння та групи ризику для тварин

2. Python Backend:

Файл: main_launcher.py.

Роль: Основний контролер системи

Функції:

- Перевірка актуальності локальних даних
 - Завантаження оновлених даних з API
 - Створення тестових даних при відсутності інтернету
 - Компіляція Java-коду
 - Запуск графічного інтерфейсу
 - Автоматична збірка релізу (створення папки Release, пакування .jar файлу)
 - Генерація скриптів запуску (run.bat, run.sh) для Windows/Linux
-

3. Java GUI:

Файл: PlantGuide.java (компілюється в PlantGuide.class).

Роль: Візуальний інтерфейс користувача

Функції:

- Відображення таблиці рослин з сортуванням
 - Кольорове кодування рівня небезпеки
 - Детальна інформація про вирану рослину
 - Кнопки для запуску аналітичних задач
 - Відображення результатів Python-скриптів
-

4. Аналітичні задачі (Python Scripts)

Папка: tasks/ (містить скрипти .py).

ЗАДАЧА 1. Топ найнебезпечніших рослин

Файл: tasks/top_families.py

Призначення: Визначає найпоширеніші ботанічні родини серед токсичних рослин.

Алгоритм: Аналізує поле family, підраховує частоту появи кожної родини та формує рейтинг у порядку спадання.

ЗАДАЧА 2. Пошук загроз для конкретних тварин

Файл: tasks/search_animals.py

Призначення: Пошук рослин, що становлять загрозу для конкретного виду тварин.

Алгоритм: Приймає назву тварини (напр. "Cat"), фільтрує базу даних та повертає список небезпечних рослин разом із симптомами отруєння.

ЗАДАЧА 3. Статистика рівнів небезпеки

Файл: tasks/severity_stats.py

Призначення: Аналіз розподілу рослин за ступенем токсичності.

Алгоритм: Агрегує дані за полем severity (рівень загрози), підраховує кількість рослин у кожній категорії та формує зведену статистику.

ЗАДАЧА 4. Пошук за симптомами

Файл: tasks/search_symptoms.py

Призначення: Знаходить усі рослини, що викликають конкретний фізіологічний симптом (наприклад, "блювота", "тремор").

Алгоритм: Виконує пошук введеного ключового слова серед списків симптомів усіх рослин.

ЗАДАЧА 5. Пошук безпечних альтернатив

Файл: tasks/safe_alternatives.py

Призначення: Пропонує безпечні кімнатні рослини для заміни токсичних видів, враховуючи наявних у користувача домашніх тварин.

Алгоритм: Аналізує запит користувача (небезпечна рослина + список тварин вдома).

ЗАДАЧА 6. Інструкція першої допомоги

Файл: tasks/first_aid.py

Призначення: Надає екстрені рекомендації, оцінку терміновості та список дій при отруєнні конкретною рослиною.

Алгоритм: Ідентифікує рослину за назвою та визначає рівень її небезпеки (severity).