

## **ІНТЕГРАЦІЯ МУЛЬТИПЛАТФОРМНИХ МЕСЕНДЖЕРІВ З ВИКОРИСТАННЯМ FASTAPI ТА WEBSOCKETS ДЛЯ СТВОРЕННЯ УНІФІКОВАНОЇ КОМУНІКАЦІЙНОЇ ПЛАТФОРМИ**

У цифрову епоху, що стрімко розвивається, фрагментація спілкування між різними платформами обміну повідомленнями, такими як Telegram, Viber та Facebook Messenger, становить значну проблему як для приватних осіб, так і для бізнесу, що прагне до ефективної взаємодії з клієнтами. Для вирішення проблеми було розроблено новаторський проєкт, який використовує фреймворк FastAPI на основі Python та WebSockets для спілкування в режимі реального часу для створення уніфікованого багатоплатформного додатку чат-бота [1]. Інноваційне рішення має на меті подолати розриви між службами обміну повідомленнями за допомогою складної модульної архітектури, впорядковуючи цифрову комунікацію та покращуючи користувацький досвід. В основі системи лежать критично важливі компоненти: модуль вебхуків і специфічний модуль обробки подій. Модуль вебхуків слугує наріжним каменем програми, функціонуючи як основний шлюз для інтеграції декількох платформ обміну повідомленнями. Модуль чудово налаштовує вебхуки для кожного сервісу, полегшуючи автоматичні HTTP-відповіді, які запускаються при отриманні повідомлень. Цей механізм має вирішальне значення для здатності системи оперативного обробляти вхідні повідомлення, починаючи з ідентифікації платформи походження повідомлення, за якою слідує перевірка повідомлення на автентичність і попередня обробка, щоб підготувати повідомлення до подальших дій в системі.

Після первинної обробки модулем вебхуків провідну роль відіграє модуль обробки подій, що залежить від платформи, який забезпечує індивідуальну взаємодію між різними службами обміну повідомленнями. Компонент складається з низки підмодулів, кожен з яких призначений для управління унікальними функціями та вимогами конкретної платформи, забезпечуючи таким чином повне використання специфічних API та функціональних можливостей кожного сервісу. Модульна та розширювана конструкція підкреслює здатність додатку до індивідуальної взаємодії, підкреслюючи ретельну інтеграцію сучасних веб-технологій, таких як FastAPI та WebSockets, для вирішення складнощів багатоплатформної комунікації.

Бездоганна співпраця між модулем вебхуків та модулем обробки подій для конкретної платформи є прикладом комплексного вирішення проблем інтеграції декількох платформ обміну повідомленнями, пропонуючи масштабовану та надійну архітектуру. Проєкт не лише покращує досвід обміну повідомленнями для користувачів, використовуючи сильні сторони кожної платформи, але й демонструє гнучкість і масштабованість системи, наближаючи нас до майбутнього, в якому цифрова комунікація між різними сервісами буде спрощеною і ефективною. Таким чином, уніфікована система обміну повідомленнями є значним кроком вперед у подоланні бар'єрів фрагментації цифрової комунікації, розширюючи можливості користувачів і компаній у зусиллях щодо цифрової взаємодії.

У прагненні створити цілісну, уніфіковану платформу обміну повідомленнями стратегічна інтеграція різноманітних служб обміну повідомленнями за допомогою модулів обробки подій, специфічних для конкретної платформи, знаменує собою значний стрибок уперед. Інтеграція забезпечує ефективність системи в обслуговуванні унікальних екосистем різних платформ обміну повідомленнями, модулі спеціально розроблені для управління нюансами кожної платформи, від автентифікації та валідації запитів до обробки різноманітних типів повідомлень і вкладень. Підхід є важливим для системи, яка має на меті забезпечити узгоджену та адаптовану взаємодію з широким спектром цифрових каналів зв'язку.

Породжувальний патерн «Фабричний метод» [2], що використано в проєкті, який дозволяє легко додавати підтримку нових месенджерів, при цьому не змінюючи жодним чином будь-який написаний раніше код. Таким чином, система дозволяє писати єдину бізнес-логіку не тільки для месенджерів що підтримується на даний момент, а й для месенджерів, що будуть підтримуватися в майбутньому.

Додавання підтримки нових месенджерів у цьому випадку є дуже простим. При цьому, використовуючи у якості прикладу задокументований код, що додає підтримку решти месенджерів, найбільша складність полягає у вивченні документації API нового месенджера для отримання всієї необхідної для забезпечення роботи застосунку інформації від сторонньої системи. Доповнюючи архітектуру системи, головний прикладний модуль стає операційним ядром, організовуючи прийом, обробку та передачу повідомлень у реальному часі. Цей модуль є ключовим, керуючи кінцевими точками HTTP і WebSocket, що дозволяє ефективно обробляти зворотні виклики вебхуків і полегшує взаємодію в реальному часі, відіграє важливу роль у забезпеченні отримання, автентифікації та належної маршрутизації повідомлень з різних платформ у системі. Зокрема, кінцева точка WebSocket необхідна для підтримки динамічного та інтерактивного користувацького інтерфейсу, що дозволяє здійснювати постійний обмін інформацією, який збагачує взаємодію з користувачами. Крім того, інтеграція головного прикладного модуля з іншими компонентами системи, такими як модулі обробки подій для конкретної платформи та модуль взаємодії з базами даних, підкреслює модульність та гнучкість системи. Цей взаємозв'язок не лише полегшує безперервний потік даних у системі, але й гарантує, що уніфікована платформа обміну повідомленнями залишається адаптивною і швидко реагує на мінливі потреби багатоплатформної комунікації.

Завдяки складній інтеграції модулів система уніфікованих повідомлень демонструє комплексний підхід до вирішення проблем цифрової комунікації у фрагментованому середовищі. Модулі обробки подій, орієнтовані на конкретні платформи, та головний прикладний модуль, що виконує роль комунікаційного центру, разом

ілюструють прагнення проекту створити універсальну, ефективну та уніфіковану систему обміну повідомленнями. Система не лише долає розрив між різними платформами обміну повідомленнями, але й прокладає шлях у майбутнє, де цифрова комунікація буде більш згуртованою, інтуїтивно зрозумілою та орієнтованою на користувача.

У сфері уніфікованих систем обміну повідомленнями ефективно зберігання та управління даними розмов є критично важливими стовпами, що підтримують інтелект, швидкість реагування та масштабованість платформ. Розвиток систем був полегшений завдяки інтеграції мультиплатформних месенджерів за допомогою FastAPI і WebSockets, що створило комплексну комунікаційну платформу, яка характеризується модульною архітектурою. Поточна структура складається з п'яти ключових модулів: модуля вебхуків, модуля обробки подій, специфічних для платформи, модуля API, модуля основного додатку та модуля взаємодії з базами даних, кожен з яких відіграє унікальну роль у функціональності та ефективності системи.

Центральне місце в ефективній роботі систем займає модуль взаємодії з базами даних, який використовує SQLAlchemy, складний інструмент об'єктно-реляційного відображення (ORM) для Python. Цей модуль має першорядне значення для забезпечення безперешкодної взаємодії між додатком і базовою базою даних, керуючи такими важливими сутностями, як користувачі, чати та повідомлення. Стратегічне використання SQLAlchemy забезпечує гнучкий, простий та інтуїтивно зрозумілий засіб навігації складними SQL-запитами, тим самим покращуючи процес розробки та забезпечуючи підтримку системи. Функціональність модуля взаємодії з базами даних виходить за рамки простого зберігання даних; він є незамінним для таких функцій, як пошук, отримання, аналіз повідомлень та сповіщення в режимі реального часу, що значно підвищує операційну ефективність та зручність роботи користувачів. Крім того, модуль відіграє вирішальну роль у забезпеченні розширених функціональних можливостей системи, таких як діалогова аналітика, персоналізований користувацький досвід та інтеграція інсайтів на основі штучного інтелекту. Систематично організовуючи та зберігаючи дані розмов, він підтримує здатність системи генерувати аналітику щодо поведінки користувачів і тенденцій обміну повідомленнями, персоналізувати взаємодію на основі історичних даних і підтримувати можливості ШІ, такі як автоматичні відповіді та аналіз настроїв, що ілюструє незамінність складного управління даними при розробці уніфікованих систем обміну повідомленнями, які не тільки швидко реагують і є ефективними, але й інтелектуальними та орієнтованими на користувача.

Насамкінець, інтеграція і детальний розгляд цих модулів в рамках системи уніфікованих повідомлень підкреслює значний прогрес у цифровій комунікації. Система не лише долає розбіжності між різними платформами обміну повідомленнями, але й забезпечує безпрецедентний рівень інтеграції, реагування в режимі реального часу та інтелектуальні можливості, що базуються на даних. Представлена робота, висвітлює потенціал використання FastAPI та WebSockets для створення універсальної, ефективної та масштабованої платформи, що є важливою віхою в еволюції цифрової комунікації. Забігаючи наперед, можна сказати, що розглянуті методології та технології закладають міцний фундамент для майбутніх інновацій, обіцяючи більш взаємопов'язаний і гармонійний ландшафт цифрових комунікацій. Завдяки дослідженню було продемонстровано доцільність, ефективність та результативність інтегрованого підходу в сучасних комунікаційних системах.

#### Список використаних джерел

1. FastAPI [Електронний ресурс] – Режим доступу: <https://web.archive.org/web/20240310015827/https://fastapi.tiangolo.com/>.
2. Фабричний метод [Електронний ресурс] – Режим доступу: <https://web.archive.org/web/20240112140625/https://refactoring.guru/uk/design-patterns/factory-method>.