

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Навчально-науковий інститут атомної та теплової енергетики

Кафедра цифрових технологій в енергетиці

"На правах рукопису"
УДК _____

«До захисту допущено»
Завідувач кафедри
Наталія АУШЕВА
“ ” _____ 2024 р.

Магістерська дисертація

на здобуття ступеня другого (магістерського) рівня вищої освіти
за освітньою програмою “Цифрові технології в енергетиці”
зі спеціальності 122 “Комп’ютерні науки”

на тему Засоби автоматичної генерації тестових повідомлень HL7_____

Виконав: студент 2 курсу, групи ТР-32мп

Швагрун Андрій Васильович

(прізвище, ім’я, по батькові)

_____ (підпис)

Науковий керівник доцент каф. цифрових технологій
в енергетиці, к.т.н., доц., Ірина МИХАЙЛОВА

(посада, науковий ступінь, вчене звання, ім’я ПРІЗВИЩЕ)

_____ (підпис)

Рецензент доцент каф. теплової та альтернативної
енергетики., к.т.н., доц., Ірина ФУРТАТ

(посада, науковий ступінь, вчене звання, ім’я ПРІЗВИЩЕ)

_____ (підпис)

Н.контроль доцент каф. цифрових технологій в
енергетиці, д.ф., Артем ГУРІН

(посада, ім’я ПРІЗВИЩЕ)

_____ (підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ - 2024

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ

Кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

Рівень вищої освіти – другий (магістерський)

спеціальність 122 “Комп’ютерні науки”

Освітньо-професійна програма “Цифрові технології в енергетиці”

ЗАТВЕРДЖУЮ

Завідувач кафедри ЦТЕ

Наталія АУШЕВА

« » 2024 р.

ЗАВДАННЯ
на дипломну роботу студенту

ШВАГРУНУ Андрію Васильовичу

(прізвище, ім’я, по батькові)

1. Тема роботи Засоби автоматичної генерації тестових повідомлень HL7

керівник роботи Михайлова Ірина Юрївна к.т.н., доцент

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «11» листопада 2024 р. № 5040-с

2. Термін подання роботи студентом 06.12.2024 року

3. Вихідні дані до роботи мова програмування Python, мікрофреймворк Flask, бібліотека Faker, OpenAI API, Gemini API

4. Перелік питань, які потрібно розробити

- 1) Вивчити структуру стандарту HL7 та існуючі методи автоматизації генерації тестових даних.
- 2) Розробити алгоритми автоматизованої генерації HL7-повідомлень різних типів.
- 3) Вибрати та реалізувати технології для інтеграції зовнішніх API і бібліотек.
- 4) Створити веб-інтерфейс для налаштування параметрів генерації, перегляду результатів та експорту даних.

5) Провести тестування системи на відповідність стандартам HL7 і оцінити її ефективність.

5. Орієнтований перелік ілюстративного матеріалу “Актуальність магістерської дисертації”, “Мета та перелік поставлених задач до роботи”, “Об’єкт та предмет дослідження”, “Стандарт HL7”, “Обґрунтування обраних методів”, “Обґрунтування обраних алгоритмів”, “Базовий метод розв’язання задачі”, “Програмні засоби реалізації”, “Блок схема”, “Архітектура системи”, “Головна сторінка”, “Постановка обчислювального експерименту”, “Перевірка валідності повідомлення за допомогою сервісу HL7 Parser та HL7 Inspector Neo”, “Висновки з виконаної роботи”, “Публікації за темою магістерської дисертації”.

6. Дата видачі завдання «11» листопада 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Затвердження теми роботи.	01.09.2024	Виконано
2.	Розробка та тестування програмного продукту.	01.09.2024 – 26.10.2024	Виконано
3.	Формування вступу.	26.10.2024 – 01.11.2024	Виконано
4.	Написання першого розділу. Аналіз HL7 стандарту і огляд інструментів.	26.10.2024 – 01.11.2024	Виконано
5.	Написання другого розділу. Аналіз методів та алгоритмів.	01.11.2024 – 05.11.2024	Виконано
6.	Написання третього розділу. Проектування архітектури системи і вибір технологій.	05.11.2024 – 10.11.2024	Виконано
7.	Написання четвертого розділу. Реалізація програмної частини.	10.11.2024 – 15.11.2024	Виконано
8.	Тестування системи. Перевірка відповідності стандарту HL7.	15.11.2024 – 18.11.2024	Виконано
9.	Написання п’ятого розділу. Стартап-проект.	18.11.2024 – 20.11.2024	Виконано
10.	Підготовка до захисту.	20.11.2024 – 18.12.2024	Виконано

Студент

(підпис)

Керівник роботи

(підпис)

Андрій ШВАГРУН

(ім’я, ПРІЗВИЩЕ)

Ірина МИХАЙЛОВА

(ім’я, ПРІЗВИЩЕ)

РЕФЕРАТ

Магістерська дисертація на тему «Засоби автоматичної генерації тестових повідомлень HL7».

Робота складається зі вступу, 5 розділів, висновків до розділів, загальних висновків, списку джерел (20 джерел) та додатку. Загальний обсяг дисертації – 88 сторінок. Основний зміст викладено на 75 сторінках. Роботу проілюстровано 22 таблицями, 18 рисунками.

Актуальність теми. Автоматизація генерації HL7-повідомлень є ключовою для стандартизованого обміну медичними даними та тестування інформаційних систем у сфері охорони здоров'я. Наявні рішення часто не задовольняють потребу у створенні реалістичних повідомлень HL7, що ускладнює ефективне тестування. Розробка інструменту для автоматичної генерації великих обсягів повідомлень дозволить проводити масштабоване тестування, підвищуючи сумісність систем і надійність обміну медичною інформацією, що сприяє покращенню якості медичних послуг.

Мета роботи. Розробити систему автоматичної генерації HL7-повідомлень, яка забезпечує масштабованість, відповідність стандартам HL7 та реалістичність створених даних, спрямовану на оптимізацію процесів тестування медичних інформаційних систем.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- дослідити стандарт HL7, зокрема структуру повідомлень, правила їх формування та перевірки відповідності для визначення ключових вимог до тестових даних;
- розробити алгоритми автоматизованої генерації HL7-повідомлень різних типів;
- проаналізувати та обрати технології для реалізації компонентів системи, враховуючи інструменти для роботи з HL7, генерації даних та створення інтерфейсу;

- реалізувати програмний продукт із зручним веб-інтерфейсом для автоматизованої генерації HL7-повідомлень, забезпечивши можливість налаштування параметрів, перегляду результатів і експорту даних;
- провести тестування системи, перевірити відповідності згенерованих даних стандартам HL7, оцінити коректність роботи алгоритмів та загальну ефективність.

Об’єкт дослідження. Програмне забезпечення для автоматичної генерації HL7-повідомлень.

Предмет дослідження. Алгоритми та програмне забезпечення для автоматичної генерації HL7-повідомлень різних типів, здатних забезпечити тестування медичних систем на відповідність стандарту HL7.

Методи дослідження. Для виконання поставлених завдань були використані методи експериментального моделювання, порівняльного аналізу технологій генерації даних, а також розробка програмного забезпечення з інтеграцією зовнішніх API та бібліотек.

Апробація результатів роботи. Основні результати дослідження представлено на конференції та опубліковано у фахових виданнях.

Структура роботи. Дисертація складається зі вступу, п’яти розділів, висновків, списку використаних джерел та додатків.

Публікації:

Швагрун А. В. Засоби автоматичної генерації тестових повідомлень HL7 : матер. VII всеукр. наук.-практ. інтернет-конф. молодих вчених та студентів (м. Київ, 29.11.2024) Київ-Херсон, 2024.

Ключові слова. HL7-повідомлення, автоматизована генерація, медичні стандарти, тестування медичних систем, медичні дані.

ABSTRACT

Master's Thesis on the topic «Tools for Automatic Generation of HL7 Test Messages».

The thesis consists of an introduction, five chapters, conclusions for each chapter, general conclusions, a list of references (20 sources), and an appendix. The total volume of the thesis is 88 pages, with the main content presented on 75 pages. The work is illustrated with 22 tables and 18 figures.

Relevance of the topic. Automating the generation of HL7 messages is key to standardized medical data exchange and testing information systems in healthcare. Existing solutions often fail to meet the need for creating realistic HL7 messages, complicating efficient testing. Developing a tool for automatically generating large volumes of messages will enable scalable testing, enhancing system compatibility and the reliability of medical information exchange, thereby improving the quality of healthcare services.

Objective of the work. To develop a system for the automatic generation of HL7 messages that ensures scalability, compliance with HL7 standards, and realism of the generated data, aimed at optimizing the testing processes of medical information systems.

To achieve the goal, the following tasks need to be completed:

- study the HL7 standard, including message structure, rules for their formation, and compliance verification, to define key requirements for test data;
- develop algorithms for the automated generation of various types of HL7 messages;
- analyze and select technologies for implementing system components, considering tools for working with HL7, data generation, and interface creation;
- implement a software product with a user-friendly web interface for automated HL7 message generation, providing options for parameter configuration, results preview, and data export;

– test the system, verify the compliance of generated data with HL7 standards, evaluate the correctness of algorithms, and assess overall efficiency.

Object of research. Програмне забезпечення для автоматичної генерації HL7-повідомлень.

Subject of research. Algorithms and software for automatic generation of various types of HL7 messages to ensure testing of medical systems for compliance with the HL7 standard.

Research methods. The research employs methods of experimental modeling, comparative analysis of data generation technologies, and software development with integration of external APIs and libraries.

Practical application. The main results of the research were presented at a conference and published in scientific journals.

Structure of the work. The thesis consists of an introduction, five chapters, conclusions, a list of references, and appendices.

Publications:

Shvagrun A. V. Tools for Automatic Generation of HL7 Test Messages: materials of the VII All-Ukrainian scientific-practical Internet conference of young scientists and students (Kyiv, November 29, 2024). Kyiv-Kherson, 2024.

Keywords. HL7 messages, automated generation, medical standards, testing of medical systems, medical data.

ЗМІСТ

ВСТУП.....	9
1 ПОСТАНОВКА ЗАДАЧІ ТА ОГЛЯД МЕТОДІВ ТА ПРОГРАМНИХ ЗАСОБІВ АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ ТЕСТОВИХ ПОВІДОМЛЕНЬ HL7.....	11
1.1 Постановка задачі автоматичної генерації тестових повідомлень HL7.....	11
1.2 Огляд засобів автоматичної генерації тестових повідомлень HL7.....	13
1.2.1 Мова програмування Python.....	14
1.2.2 Мікрофреймворк Flask.....	15
1.2.3 Бібліотека hl7apy.....	15
1.2.4 Сервіс OpenAI API.....	16
1.2.5 Сервіс Gemini API.....	16
1.2.6 Бібліотека Faker.....	17
1.2.7 Система контролю версій Git.....	17
1.2.8 Бібліотека Dotenv.....	17
1.3 Огляд методів генерації тестових повідомлень HL7.....	18
1.3.1 Ручне створення HL7-повідомлень.....	18
1.3.2 Використання бібліотек для роботи з HL7.....	18
1.3.3 Генератори тестових даних.....	19
1.3.4 Інтеграція штучного інтелекту.....	20
2 ОПИС СТАНДАРТУ HL7 ТА ОБРАНИХ МЕТОДІВ ГЕНЕРАЦІЇ ПОВІДОМЛЕНЬ.....	22
2.1 Опис стандарту HL7.....	22
2.2 Методи та алгоритми обробки HL7-повідомлень.....	29
2.3 Методи генерації тестових даних.....	30
2.4 Алгоритми оптимізації генерації HL7-повідомлень.....	33
2.5 Характеристики програмних засобів для реалізації.....	35
2.6 Постановка задачі обчислювального експерименту.....	36
3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ ТЕСТОВИХ ПОВІДОМЛЕНЬ HL7.....	38
3.1 Архітектура програмної системи.....	38
3.2 Модель представлення даних.....	41
3.3 Опис клієнтської частини.....	43
3.4 Опис серверної частини.....	45
4 ТЕСТУВАННЯ ТА ПОРІВНЯННЯ МЕТОДІВ, ДЕМОНСТРАЦІЯ ФУНКЦІОНАЛЬНОСТІ.....	47
4.1 Інсталяція програмного забезпечення.....	47
4.2 Робота користувача з системою.....	48

4.3 Порівняння результатів генерації.....	54
4.4 Перевірка валідності HL7 повідомлень.....	57
5 РОЗРОБКА СТАРТАП-ПРОЄКТУ.....	61
5.1 Опис ідеї стартап-проекту.....	61
5.2 Технологічний аудит ідеї стартап–проекту.....	63
5.3 Аналіз ринкових можливостей запуску стартап–проекту.....	64
5.4 Розроблення ринкової стратегії стартап–проекту.....	75
5.5 Розроблення маркетингової програми стартап–проекту.....	78
ВИСНОВКИ.....	84
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	85
ДОДАТОК А.....	87

ВСТУП

Сучасна медицина активно використовує інформаційні технології для покращення якості та оперативності надання медичних послуг. Особливу роль у цьому процесі відіграє забезпечення коректного обміну медичними даними між різними системами, включаючи електронні медичні записи, системи управління медичними установами, лабораторні системи та інші інформаційні інструменти. Стандартизація обміну медичними даними стає ключовою вимогою, що обумовлена як зростанням обсягу медичних даних, так і підвищеними вимогами до надійності та безпеки передачі інформації.

Одним із найбільш поширених стандартів для забезпечення такого обміну є HL7 (Health Level 7), який широко використовується для структурування даних, що циркулюють між різними системами охорони здоров'я. Стандарт HL7 дозволяє уніфікувати інформаційні процеси, знижуючи ризик помилок і підвищуючи ефективність комунікацій між медичними системами. Водночас, зростаючий обсяг медичних даних, потреба у їхній швидкій обробці та необхідність підтримки сумісності між різними системами обумовлюють потребу в масштабованому тестуванні медичних інформаційних систем. З огляду на це, особливо важливо забезпечити коректну та масштабовану генерацію HL7-повідомлень, що може бути досягнуто завдяки автоматизованим інструментам для їх створення.

На сьогодні автоматизація процесу генерації HL7-повідомлень є критичною для забезпечення надійного та стандартизованого обміну медичними даними, зокрема для тестування інформаційних систем, що використовуються в охороні здоров'я [1]. Існуючі рішення не завжди повною мірою задовольняють потребу в автоматизованій генерації реалістичних повідомлень HL7, що ускладнює ефективне тестування медичних систем. Розробка інструменту, здатного автоматично генерувати великі обсяги повідомлень, дозволяє здійснювати масштабоване тестування на відповідність стандартам HL7, покращуючи сумісність систем і надійність обміну медичною інформацією. Це є важливим

кроком до підвищення якості обслуговування пацієнтів та оптимізації процесів у сфері охорони здоров'я.

Робота складається зі вступу, 5 розділів, висновків до розділів, загальних висновків, списку джерел (20 джерел) та додатку. Загальний обсяг дисертації – 88 сторінок. Основний зміст викладено на 75 сторінках. Роботу проілюстровано 22 таблицями, 18 рисунками.

1 ПОСТАНОВКА ЗАДАЧІ ТА ОГЛЯД МЕТОДІВ ТА ПРОГРАМНИХ ЗАСОБІВ АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ ТЕСТОВИХ ПОВІДОМЛЕНЬ HL7

Автоматизація процесів у медичних інформаційних системах є ключовою умовою для їх ефективного функціонування та впровадження. Однією з найважливіших задач у цій сфері є стандартизований обмін даними між медичними установами, що забезпечується завдяки використанню стандарту HL7. Цей розділ присвячений постановці задачі створення засобів автоматичної генерації тестових HL7-повідомлень та огляду засобів та методів, які можуть бути використані для її вирішення.

1.1 Постановка задачі автоматичної генерації тестових повідомлень HL7

Однією з головних проблем впровадження медичних систем є необхідність масштабного тестування таких систем перед їх інтеграцією в робочі процеси. Це тестування потребує великих обсягів реалістичних даних у форматі HL7. Однак створення таких даних вручну є трудомістким і ресурсозатратним процесом, що може спричинити затримки у розробці та впровадженні інформаційних систем.

Таким чином актуальною є задача створення програмної системи для автоматичної генерації тестових даних у форматі HL7 [2].

Метою дослідження є розробка засобів автоматичної генерації тестових HL7-повідомлень для масштабного тестування медичних інформаційних систем, що дозволяє забезпечити їх стабільність, функціональність та відповідність стандартам. Для досягнення цієї мети необхідно вирішити такі завдання:

- дослідити стандарт HL7, зокрема структуру повідомлень, правила їх формування та перевірки відповідності для визначення ключових вимог до тестових даних;
- розробити алгоритми автоматизованої генерації HL7-повідомлень різних типів;
- проаналізувати та обрати технології для реалізації компонентів системи, враховуючи інструменти для роботи з HL7, генерації даних та створення інтерфейсу;
- реалізувати програмний продукт із зручним веб-інтерфейсом для автоматизованої генерації HL7-повідомлень, забезпечивши можливість налаштування параметрів, перегляду результатів і експорту даних;
- провести тестування системи, перевірити відповідності згенерованих даних стандартам HL7, оцінити коректність роботи алгоритмів та загальну ефективність.

Об'єктом дослідження є програмне забезпечення для автоматичної генерації HL7-повідомлень.

Предметом дослідження є алгоритми та програмне забезпечення для автоматичної генерації HL7-повідомлень різних типів, здатних забезпечити тестування медичних систем на відповідність стандарту HL7.

Для досягнення поставленої мети були використані наступні методи:

1. Аналітичний метод для дослідження та вибору ефективних підходів до автоматизації генерації HL7-повідомлень;
2. Програмний метод для реалізації алгоритмів генерації та створення веб-застосунку;
3. Експериментальний метод для тестування розробленого інструменту на обсягах повідомлень різної складності;
4. Метод порівняльного аналізу для оцінки ефективності різних підходів до генерації тестових даних, зокрема за допомогою AI.

Розроблений інструмент дозволяє автоматизувати процес генерації HL7-повідомлень, що забезпечує масштабоване тестування медичних

інформаційних систем. Використання цього інструменту допоможе медичним установам та ІТ-компаніям, що займаються розробкою інформаційних систем для охорони здоров'я, проводити тестування на відповідність стандартам HL7, підвищуючи надійність і стабільність обміну даними.

Таким чином, виконання цих завдань дозволить створити програмний продукт, який стане ефективним інструментом для тестування медичних інформаційних систем.

1.2 Огляд засобів автоматичної генерації тестових повідомлень HL7

Під час розробки веб-застосунку були використані наступні мови програмування, фреймворки та технології:

1. Мова програмування Python, яка обрана завдяки багатій екосистемі, зрозумілому синтаксису та широким можливостям у веб-розробці, автоматизації й аналізі даних.
2. Flask використовується як основний фреймворк для створення веб-додатків завдяки його легкості, гнучкості та можливості швидкої розробки API.
3. HTML (HyperText Markup Language) використовується для створення структури веб-сторінок, забезпечуючи правильну організацію контенту, включаючи текст, зображення та мультимедіа.
4. CSS (Cascading Style Sheets) дозволяє оформлювати веб-сторінки, додаючи стиль, кольори, шрифти та адаптивний дизайн, який робить сторінки зручними для різних пристроїв.
5. JavaScript [3] додає інтерактивність до веб-сторінок, дозволяючи створювати динамічні елементи, анімації та виконувати дії у відповідь на дії користувача.

6. HL7apy служить для роботи з HL7-повідомленнями у сфері охорони здоров'я, забезпечуючи створення, парсинг та перевірку цих даних у зручному форматі.
7. OpenAI API забезпечує доступ до потужних мовних моделей, що дозволяє автоматизувати генерацію текстового контенту, створювати відповіді на запити та виконувати складні завдання.
8. Gemini API пропонує інструменти для роботи зі структурованими даними, забезпечуючи можливості генерації та обробки даних для інтеграції з різноманітними системами та додатками.
9. Faker допомагає створювати тестові дані, такі як імена, адреси, номери телефонів, що значно спрощує тестування додатків і сервісів.
10. Git використовується як система контролю версій, що дозволяє відстежувати зміни в коді, співпрацювати над проєктами і зберігати історію розробки.
11. dotenv використовується для безпечного зберігання конфіденційної інформації, такої як ключі API чи налаштування бази даних, у спеціальних файлах конфігурації.

1.2.1 Мова програмування Python

Python [4] є однією з найпопулярніших мов програмування, відомою своєю простотою та універсальністю. Завдяки інтуїтивно зрозумілому синтаксису Python дозволяє розробникам швидко створювати складні додатки з мінімальними витратами часу. Його популярність пояснюється широкою підтримкою бібліотек і фреймворків, які охоплюють численні сфери, від обробки даних і автоматизації до розробки веб-додатків і систем машинного навчання.

Python відзначається багатою екосистемою бібліотек, яка включає інструменти для текстового, числового й структурованого аналізу даних. Завдяки підтримці багатопоточності та багатопроцесорності Python здатний обробляти великі обсяги даних, що є важливим для сучасних програм. Ще однією перевагою

мови є її кросплатформеність: програми на Python можуть працювати на Windows, Linux та macOS без значних змін у коді. Окрім технічних переваг, Python приваблює розробників своєю спільнотою. Активне співтовариство створює і підтримує величезну кількість бібліотек і документації, що робить мову доступною навіть для початківців.

1.2.2 Мікрофреймворк Flask

Flask [5] — це мікрофреймворк для створення веб-застосунків на Python. Він забезпечує основу для розробки, зосереджену на простоті та гнучкості. Flask є легким у використанні, що дозволяє розробникам швидко створювати прототипи або повноцінні веб-застосунки, не перевантажуючи код непотрібними компонентами.

Його структура базується на мінімалістичному підході: розробник отримує базовий функціонал для створення додатків і має свободу додавати лише ті компоненти, які необхідні. Це робить Flask особливо корисним для проєктів, які вимагають кастомізації або експериментальних рішень.

Flask підтримує створення RESTful API, обробку HTTP-запитів та інтеграцію з базами даних. Він широко використовується у веб-розробці завдяки своїй легкій інтеграції з іншими технологіями Python, такими як бібліотеки для обробки даних або інструменти автоматизації.

1.2.3 Бібліотека hl7apy

hl7apy [6] — це спеціалізована бібліотека для мови Python, яка дозволяє працювати з повідомленнями стандарту HL7. Її функціональність спрямована на автоматизацію створення, перевірки та редагування HL7-повідомлень. Інструмент надає готові шаблони для формування стандартних сегментів і структур, що значно полегшує роботу з медичними даними.

Бібліотека дозволяє адаптувати повідомлення до різних сценаріїв використання. Зокрема, hl7apy забезпечує динамічне формування полів і перевірку валідності створених повідомлень відповідно до вимог стандарту. Це дозволяє автоматизувати процеси, які раніше вимагали значного втручання з боку розробника. Завдяки своїй інтуїтивно зрозумілій архітектурі бібліотека hl7apy широко використовується у сфері медичних інформаційних систем.

1.2.4 Сервіс OpenAI API

OpenAI API [7] є передовим інструментом для інтеграції функціональності мовних моделей у додатки. Цей інтерфейс надає можливість автоматизувати створення тексту, генерувати відповіді на запити користувачів та виконувати складні завдання. Однією з головних переваг API є здатність генерувати текст, який адаптується до конкретного контексту. Це робить його ідеальним для створення текстових конструкцій, таких як рекомендації, описові дані чи автоматизовані відповіді на часті запитання. OpenAI API також забезпечує високу якість результатів, що дозволяє використовувати його в проєктах, де потрібна точність і реалістичність тексту.

1.2.5 Сервіс Gemini API

Gemini API [8] пропонує можливості для роботи з великими обсягами структурованих даних, надаючи розробникам інструменти для масштабованого створення й обробки інформації. Цей API є зручним рішенням для задач, що вимагають автоматизації, інтеграції або оптимізації процесів у проєктах, пов'язаних із даними.

Серед основних можливостей Gemini API — створення складних структур даних, моделювання сценаріїв інтеграції та масове генерування даних для тестування. Його швидкодія та адаптивність роблять його популярним у сфері розробки масштабованих систем.

1.2.6 Бібліотека Faker

Faker [9] — це бібліотека для Python, яка використовується для створення фіктивних (тестових) даних. Вона дозволяє генерувати реалістичні значення для різних сценаріїв, таких як імена, адреси, дати народження, номери телефонів або унікальні ідентифікатори.

Ця бібліотека є незамінною у проєктах, де потрібно швидко створювати тестові набори даних. Faker підтримує локалізацію, що дозволяє генерувати дані з урахуванням регіональних особливостей. Це робить бібліотеку універсальним інструментом для задач тестування й розробки.

1.2.7 Система контролю версій Git

Git [10] є стандартною системою контролю версій, яка забезпечує управління змінами у програмному коді. Її використання дозволяє розробникам вести спільну роботу, фіксувати зміни в історії проєкту та відновлювати попередні версії у разі потреби.

Серед переваг Git — можливість створювати гілки для тестування нових функцій і збереження стабільності основної версії програми. Це дозволяє ефективно організовувати процес розробки навіть у великих командах. Git також забезпечує інтеграцію з популярними платформами, такими як GitHub або GitLab.

1.2.8 Бібліотека Dotenv

Dotenv [11] є інструментом для роботи зі змінними середовища. Він дозволяє зберігати конфіденційну інформацію, таку як ключі API, паролі чи налаштування підключення до баз даних, у спеціальних файлах конфігурації. Цей інструмент є надзвичайно корисним для розмежування налаштувань між різними середовищами, такими як розробка, тестування та продакшн. Dotenv допомагає уникнути випадкового розголошення конфіденційної інформації, що робить його важливим компонентом у забезпеченні безпеки додатків.

1.3 Огляд методів генерації тестових повідомлень HL7

Автоматизація створення HL7-повідомлень є важливим етапом тестування медичних інформаційних систем. Це дозволяє забезпечити швидке й масштабоване формування даних, необхідних для перевірки коректності роботи інтеграційних модулів. Сучасна практика пропонує кілька основних підходів до реалізації цього завдання. Розглянемо детальніше основні методи, що використовуються в цій сфері.

1.3.1 Ручне створення HL7-повідомлень

Ручне створення HL7-повідомлень є найстарішим але найбільш трудозатратним підходом. У цьому випадку інженери самостійно формують структуру повідомлень, заповнюючи необхідні сегменти відповідно до специфікацій стандарту. Цей метод часто використовується на етапах навчання або для створення одиничних тестових повідомлень, де важливо забезпечити максимальну точність і контроль за процесом.

Попри свою простоту, ручний метод має суттєві обмеження. Основною проблемою є низька продуктивність, адже кожне повідомлення потребує ретельної перевірки та значного часу на створення. Висока ймовірність помилок, зумовлена людським фактором, також робить цей підхід менш надійним. Крім того, створення великого обсягу тестових даних вручну є практично неможливим через значну трудомісткість процесу. Саме тому ручний підхід поступово витісняється автоматизованими методами.

1.3.2 Використання бібліотек для роботи з HL7

З появою спеціалізованих бібліотек для роботи з HL7-повідомленнями процес автоматизації значно спростився. Бібліотеки, такі як hl7apy для Python, забезпечують програмне створення повідомлень, використовуючи готові шаблони

та структури. Це дозволяє ефективно формувати необхідні сегменти, такі як MSH, PID, OBR, з мінімальними зусиллями.

Такий підхід є особливо корисним для великих проєктів, де важливо забезпечити точність і відповідність повідомлень специфікаціям HL7. Бібліотеки дозволяють автоматично перевіряти правильність структури даних і дотримання стандарту, що значно знижує ризик помилок. Крім того, програмне налаштування полів дає змогу швидко адаптувати повідомлення до різних сценаріїв, наприклад, тестування інтеграції з різними системами.

Однією з переваг використання бібліотек є можливість інтеграції з іншими інструментами, такими як генератори тестових даних. Це відкриває нові можливості для автоматизації й масштабування процесу створення HL7-повідомлень.

1.3.3 Генератори тестових даних

Сучасні генератори тестових даних стали важливим інструментом для автоматизації створення реалістичних значень полів HL7-повідомлень. Інструменти на зразок Faker дозволяють створювати випадкові дані, що відповідають реальним сценаріям. Це можуть бути імена пацієнтів, дати народження, адреси, номери телефонів або навіть медичні ідентифікатори.

Однією з ключових переваг генераторів є можливість створювати значний обсяг унікальних даних за короткий проміжок часу. Такі інструменти також підтримують локалізацію, що дозволяє генерувати дані, адаптовані до специфіки конкретної країни або регіону. Це робить їх ідеальними для тестування інформаційних систем, орієнтованих на міжнародний ринок.

Генератори тестових даних можуть використовуватися як окремо, так і в комбінації з бібліотеками для роботи з HL7, що значно розширює їх функціональні можливості.

1.3.4 Інтеграція штучного інтелекту

Штучний інтелект пропонує нові можливості для автоматизації створення складних полів HL7-повідомлень [12]. Інструменти, такі як OpenAI API або подібні системи генерації тексту, дозволяють створювати реалістичні текстові поля, наприклад, клінічні замітки, описи діагнозів чи рекомендації. Це особливо актуально для тестування систем, де важливо оцінити, як вони обробляють великі обсяги текстової інформації.

Штучний інтелект також використовується для моделювання специфічних сценаріїв, наприклад, генерації даних для тестування інтеграції між різними медичними установами. Однак такі методи потребують значних обчислювальних ресурсів, що може обмежувати їх використання в проєктах із великим обсягом даних.

Висновок до розділу 1

Основна проблема полягає у створенні великих обсягів реалістичних тестових даних, необхідних для перевірки медичних інформаційних систем. Ручне створення таких даних є неефективним, що робить автоматизацію процесу ключовим завданням.

Завдяки аналізу було сформульовано мету — створення програмного інструменту, який забезпечує високу швидкість генерації даних, відповідність стандартам HL7 і можливість адаптації до різних сценаріїв використання. Для цього розглянуто сучасні бібліотеки, як hl7apy, генератори тестових даних Faker, інструменти штучного інтелекту OpenAI API та Gemini API, а також засоби управління конфігураціями (dotenv) та контролю версій (Git).

Огляд методів виявив, що автоматизовані рішення значно перевищують ефективність ручних підходів, забезпечуючи масштабованість, точність і гнучкість. Використання бібліотек для HL7 дозволяє знизити ризик помилок, тоді як генератори даних прискорюють процес тестування. Інтеграція штучного

інтелекту додає можливість створення реалістичного контенту для текстових полів.

Отже, аналіз підтвердив доцільність розробки програмного забезпечення для автоматичної генерації HL7-повідомлень, а також визначив ключові компоненти, які необхідно використати для створення функціонального та продуктивного інструменту.

2 ОПИС СТАНДАРТУ HL7 ТА ОБРАНИХ МЕТОДІВ ГЕНЕРАЦІЇ ПОВІДОМЛЕНЬ

Цей розділ описує ключові аспекти стандарту HL7, включаючи його основну структуру, типи повідомлень і механізми передачі даних між медичними інформаційними системами. Особлива увага приділяється деталізації форматів, роздільників і специфікацій сегментів HL7-повідомлень. Наведено реалістичні приклади повідомлень різних типів (ADT, ORM, DFT, ORU), їхні сегменти та значення полів. Розглянуто методи та алгоритми обробки HL7-повідомлень, а також підходи до їхньої оптимізації та генерації. Це створює базу для подальших обчислювальних експериментів і розробки системи автоматизованої генерації HL7-повідомлень.

2.1 Опис стандарту HL7

Стандарт HL7 (Health Level 7) — це міжнародний протокол для обміну даними між медичними інформаційними системами [13]. Його використання забезпечує уніфікований підхід до передачі даних у системах охорони здоров'я. HL7 використовується для відправлення даних про пацієнтів, передачі результатів лабораторних тестів, запису про госпіталізацію, виписку рецептів тощо.

Структура HL7-повідомлення складається з послідовності структурованих сегментів, кожен з яких містить визначений набір полів. Поля можуть бути розбиті на підполя, а ті — на компоненти. Розділення структурних елементів здійснюється за допомогою спеціальних символів:

- роздільник полів “|”;
- роздільник підполів “^”;
- роздільник повторюваних елементів “~”;
- роздільник компонентів “&;”;

Кожне HL7-повідомлення починається із заголовка (MSH), який містить основну інформацію про відправника, отримувача, дату передачі й тип повідомлення.

Приклад HL7-повідомлення зображений на рисунку 2.1.

```
MSH|^~\&|||||20111215114509.682+0100||ORU^R01^ORU_R01|88c3|P|2.6.1|||NE|AL|||||
PID|||005.13226488954651^^^&1.2.40.0.10.1.6.1.0.1.100.1.1&ISO||Testpatient^Hage
nberg||19310320000000+0100|M|||
OBR|
OBX||NM|150021^MDC_PRESS_BLD_NONINV_SYS^MDC|1.0.1.1|125|266016^MDC_DIM_MMHG^MCD
||||F|||||||20111215114505.824+0100
OBX||NM|150022^MDC_PRESS_BLD_NONINV_DIA^MDC|1.0.1.2|85|266016^MDC_DIM_MMHG^MCD|
||||F|||||||20111215114505.824+0100
```

Рисунок 2.1 – Приклад HL7 повідомлення

Розшифрування сегментів:

- MSH (Message Header): заголовок повідомлення, що містить технічну інформацію.
- PID (Patient Identification): містить ідентифікаційну інформацію про пацієнта.
- OBR (Observation Request): інформація про госпіталізацію або візит пацієнта.
- OBX (Observation Result): містить результати спостереження або тестування

Стандарт HL7 передбачає використання різних типів повідомлень залежно від контексту:

1. ADT (Admit, Discharge, Transfer) — використовується для обробки даних про госпіталізацію пацієнтів, їх виписку або переведення між медичними установами чи відділеннями.
2. ORM (Order) — призначене для створення медичних замовлень, наприклад, на лабораторні тести, процедури чи лікарські засоби.
3. ORU (Observation Result) — забезпечує передачу результатів спостережень, таких як лабораторні аналізи, діагностичні висновки чи інші клінічні дані.

4. ACK (Acknowledgment) — використовується для підтвердження отримання та обробки повідомлень іншими системами.
5. MDM (Medical Document Management) — дозволяє керувати медичними документами, такими як виписки, клінічні записи чи довідки.
6. SCH (Schedule Information) — використовується для передачі інформації про розклади процедур, консультацій чи інших подій у медичних установах.
7. SIU (Scheduling Information Unsolicited) — забезпечує передачу незапланованих змін у розкладі, таких як скасування або перенесення процедур чи прийомів.
8. DFT (Detail Financial Transactions) — дозволяє деталізувати фінансові транзакції, пов'язані з оплатою медичних послуг чи процедур.
9. BAR (Billing Account) — застосовується для управління обліковими записами, пов'язаними з медичними рахунками.
10. VXU (Unsolicited Vaccination Record Update) — використовується для передачі інформації про щеплення, зокрема для оновлення записів у базах даних.
11. RDE (Pharmacy/Treatment Encoded Order) — забезпечує передачу зашифрованих замовлень на медикаменти чи лікувальні процедури.
12. QBP (Query by Parameter) — використовується для виконання запитів за заданими параметрами, наприклад, для пошуку інформації про пацієнта чи процедуру.
13. MFN (Master File Notification) — передає інформацію про зміни в базах даних, наприклад, оновлення довідкової інформації про лікарські засоби чи медичне обладнання.
14. RDS (Pharmacy/Treatment Dispense) — використовується для передачі інформації про відпуск медикаментів чи виконання процедур.
15. RDT (Display-based Response) — застосовується для передачі результатів на запити, які відображаються у вигляді структурованих даних.
16. UDI (Unique Device Identifier) — забезпечує передачу унікальних ідентифікаторів медичних пристроїв та обладнання.

Кожен тип повідомлення має унікальну структуру, яка описується специфікаціями стандарту. Для кожного типу повідомлення буде надано детальний опис основних сегментів, їхнього змісту та значення полів. Це дозволить краще зрозуміти, як повідомлення використовуються для передачі різноманітної медичної інформації.

Приклад повідомлення ADT^A01 (госпіталізація або виписки пацієнта) зображений на рисунку 2.2.

```
MSH|^~\&|EHRSystem|GeneralHospital|EHR|GeneralHospital|202411291200||ADT^A01|123456|P|
2.3 PID|||78901^^^GeneralHospital||Johnson^Michael^A||19800515|M|||
123 Elm St^^Metropolis^NY^12345^USA||(123)456-7890|(123)456-7890|S||101234567|987-65-4321|
||EN PV1|1||12W^110^1^GeneralHospital^^^Room12||Smith^John^A^^MD|ORTH|||ADM|SELF|||
|123456789^InsurancePlan|123456^AccountNum|VIP|GeneralHospital
```

Рисунок 2.2 – Приклад HL7 повідомлення типу ADT^A01

- 1) MSH (Message Header):
 - тип повідомлення: ADT^A01;
 - дата створення: 202411291200 (29 листопада 2024 року, 12:00);
 - відправник: EHRSystem;
 - отримувач: GeneralHospital;
- 2) PID (Patient Identification):
 - ідентифікатор пацієнта: 78901^^^GeneralHospital;
 - ім'я: Johnson^Michael^A;
 - дата народження: 19800515 (15 травня 1980 року);
 - стать: M (чоловіча);
 - адреса: 123 Elm St^^Metropolis^NY^12345^USA;
 - телефон: (123)456-7890;
 - номер соціального страхування: 987-65-4321;
- 3) PV1 (Patient Visit):
 - тип візиту: I (госпіталізація);
 - палата: 12W^110^1;

- лікар: Smith^John^A^^MD;
- спеціалізація: ORTH;
- статус пацієнта: VIP;
- план страхування: 123456789^InsurancePlan;
- номер рахунку: 123456^AccountNum;

Приклад повідомлення ORM^O01 (запит на лабораторне дослідження) зображений на рисунку 2.3.

```
MSH|^~\&|LabSystem|GeneralHospital|EHR|GeneralHospital|202411291230||ORM^O01|987654|P|
2.3 PID|||23456^^^GeneralHospital||Smith^Sarah^L||19901210|F|||
456 Maple St^^Springfield^IL^62704^USA||(217)123-4567||(217)987-6543|M||204567890|
987-12-3456|||EN ORC|NW|LAB123456|||R|||202411291230|Smith^John^A^^MD||
OrderDept^Lab OBR|1||LAB123||CBC|||202411291300|||Technician^Jane^Q|||Fasting|
202411291000|202411291200|Dr. Brown^Ethan^R
```

Рисунок 2.3 – Приклад HL7 повідомлення типу ORM^O01

- 1) MSH (Message Header):
 - тип повідомлення: ORM^O01.
 - дата створення: 202411291230 (29 листопада 2024 року, 12:30).
- 2) PID (Patient Identification):
 - ідентифікатор пацієнта: 23456^^^GeneralHospital.
 - ім'я: Smith^Sarah^L.
 - дата народження: 19901210 (10 грудня 1990 року).
 - стать: F (жіноча).
 - адреса: 456 Maple St^^Springfield^IL^62704^USA.
 - телефон: (217)123-4567.
 - номер соціального страхування: 987-12-3456.
- 3) ORC (Order Control):
 - ідентифікатор запиту: LAB123456.
 - статус: NW (новий запит).
 - лікар: Smith^John^A^^MD.

4) OBR (Observation Request):

- тип тесту: CBC (Complete Blood Count).
- дата та час: 202411291300 (29 листопада 2024 року, 13:00).
- особливі вимоги: fasting (натщесерце).

Приклад повідомлення DFT^P03 (фінансова транзакція) зображений на рисунку 2.3.

```
MSH|^~\&|BillingSystem|GeneralHospital|EHR|GeneralHospital|202411291245||DFT^P03|654789|P|
2.3 PID|||34567^^^GeneralHospital||Brown^Emily^R||19950620|F|||
789 Oak St^^Gotham^NY^10001^USA||(212)555-0199||M||305678912|987-65-4321||EN FT1|1|
202411291200|202411291215|CH|||350.00|USD||MRI^Scan||Brown^Ethan^R
```

Рисунок 2.4 – Приклад HL7 повідомлення типу DFT^P03

1) MSH (Message Header):

- тип повідомлення: DFT^P03.
- дата створення: 202411291245 (29 листопада 2024 року, 12:45).

2) PID (Patient Identification):

- ідентифікатор пацієнта: 34567^^^GeneralHospital.
- ім'я: Brown^Emily^R.
- дата народження: 19950620 (20 червня 1995 року).
- стать: F (жіноча).
- адреса: 789 Oak St^^Gotham^NY^10001^USA.

3) FT1 (Financial Transaction):

- тип транзакції: CH (оплата).
- сума: 350.00 USD.
- опис послуги: MRI^Scan.

Приклад повідомлення ORU^R01 (результати лабораторного дослідження) зображений на рисунку 2.5.

```
MSH|^~\&|LabSystem|GeneralHospital|EHR|GeneralHospital|202411291330||ORU^R01|567890|P|
2.3 PID|||45678^^^GeneralHospital||Taylor^Chris^M||19781122|M|||
789 Birch St^^StarCity^CA^94016^USA||(408)555-6789|||M|406789123|987-11-2233|||EN OBR|1||
LAB789||LiverPanel|||202411291315|||Technician^Anna^T OBX|1|NM|Glucose^Blood||5.2|mmol/L|
4.0-6.0|N|||F OBX|2|NM|ALT^Liver||25|U/L|10-40|N|||F
```

Рисунок 2.5 – Приклад HL7 повідомлення типу ORM^O01

- 1) MSH (Message Header):
 - тип повідомлення: ORU^R01.
 - дата створення: 202411291330 (29 листопада 2024 року, 13:30).
- 2) PID (Patient Identification):
 - ідентифікатор пацієнта: 45678^^^GeneralHospital.
 - ім'я: Taylor^Chris^M.
 - дата народження: 19781122 (22 листопада 1978 року).
 - стать: M (чоловіча).
- 3) OBR (Observation Request):
 - тип тесту: LiverPanel (печінковий профіль).
 - дата проведення: 202411291315.
- 4) OBX (Observation Result):
 - результат тесту: Glucose – 5.2 ммоль/л (норма: 4.0–6.0).
 - результат тесту: ALT – 25 U/L (норма: 10–40).

Ці приклади показують гнучкість та універсальність HL7-повідомлень, які використовуються для передачі різних видів медичної інформації, забезпечуючи стандартизований обмін між інформаційними системами.

2.2 Методи та алгоритми обробки HL7-повідомлень

Парсинг є базовим етапом обробки, який включає розбиття повідомлення на сегменти, поля та підполя відповідно до специфікації HL7. Зазвичай використовуються алгоритми розбору тексту, що враховують символи-роздільники (|, ^, ~, &).

Алгоритм парсингу полягає у виконанні наступних етапів:

1. Розділити повідомлення на сегменти за символом \r (роздільник рядків).
2. Для кожного сегмента розбити дані за символом | (роздільник полів).
3. Додатково розділити кожне поле на підполя за символами ^, ~, &.

Парсинг забезпечує доступ до будь-якого елемента повідомлення, що є необхідним для подальших операцій.

Модифікація передбачає зміну значень полів або підполів у HL7-повідомленнях. Це може бути коригування пацієнтських даних, оновлення кодів тестів або додавання нових сегментів.

Алгоритм модифікації полягає у виконанні наступних етапів:

1. Визначити сегмент та поле, які потребують змін.
2. Замінити значення, дотримуючись правил формату HL7.
3. Виконати перевірку коректності змін.

HL7-повідомлення можуть містити дані в різних форматах, що потребує їх нормалізації. Наприклад, дати у форматі MM/DD/YYYY конвертуються у формат YYYYMMDD. Нормалізація також може включати видалення зайвих символів або виправлення кодів.

Алгоритм нормалізації полягає у виконанні наступних етапів:

1. Приведення дат до єдиного формату.
2. Видалення зайвих пробілів і символів.
3. Перевірка значень полів на відповідність дозволеним кодам.

Генерація повідомлень включає створення сегментів і заповнення їх даними відповідно до специфікації HL7. Цей процес автоматизується за допомогою шаблонів або програмних алгоритмів.

Алгоритм генерації полягає у виконанні наступних етапів:

1. Ініціалізувати шаблон повідомлення з базовими сегментами (MSH, PID, OBR).
2. Заповнити поля даними, отриманими від користувача або з бази.
3. Перевірити валідність створеного повідомлення.

Валідація перевіряє відповідність HL7-повідомлень специфікації. Це включає перевірку структури, обов'язкових полів, допустимих значень і правильності формату.

Алгоритм валідації:

1. Перевірити наявність усіх обов'язкових сегментів (наприклад, MSH).
2. Верифікувати значення полів за стандартами HL7.
3. Переконатися, що структура повідомлення відповідає специфікаціям.

Обробка великих обсягів HL7-повідомлень потребує оптимізації. Основні підходи включають:

1. Кешування
2. Багатопоточність
3. Оптимізацію алгоритмів

2.3 Методи генерації тестових даних

Генерація тестових даних є важливим етапом для перевірки роботи інформаційних систем. Це дозволяє створити реалістичні сценарії використання, виявити можливі помилки в системі та оцінити її продуктивність. У сучасній практиці застосовуються різноманітні методи для автоматизованого створення тестових наборів даних, які забезпечують точність, масштабованість та відповідність реальним умовам.

Одним із найпростіших методів є використання алгоритмів випадкової генерації даних. Цей підхід дозволяє створювати значення, що формально відповідають потрібним форматам, але не мають прив'язки до реальних сценаріїв.

Наприклад, випадкові числові послідовності можуть бути використані для ідентифікаторів, а символьні рядки — для імен чи кодів. Основна перевага цього методу полягає в його швидкості та простоті реалізації. Він добре підходить для первинного тестування, де важливіша перевірка технічної відповідності, ніж контекстуальна реалістичність. Проте випадково згенеровані дані часто не враховують специфіку реальних сценаріїв, що обмежує їх застосування в інтеграційному та функціональному тестуванні.

Генерація даних на основі шаблонів є більш структурованим підходом, який передбачає створення наборів даних, що відповідають визначеним правилам і форматам. У цьому методі використовуються заздалегідь підготовлені шаблони, які визначають структуру даних. Наприклад, у повідомленнях HL7 шаблони можуть задавати такі сегменти, як MSH, PID чи OBR, де кожне поле має певний формат і тип даних. Цей підхід дозволяє зберігати відповідність стандартам і вимогам до структури даних. Завдяки автоматизації процесу заповнення полів можна значно скоротити час на створення даних. Однак для складних сценаріїв генерація на основі шаблонів потребує великої кількості заздалегідь підготовлених прикладів, що може ускладнювати її масштабування.

Бібліотека Faker є одним із найпоширеніших інструментів для генерації реалістичних тестових даних. Вона надає широкий спектр можливостей для створення фіктивних імен, адрес, номерів телефонів, дат, ідентифікаційних номерів та інших даних. Faker підтримує локалізацію, що дозволяє адаптувати згенеровані дані до специфіки конкретної країни або регіону. Перевагою використання Faker є його здатність створювати дані, які виглядають реальними, але не містять справжньої конфіденційної інформації. Наприклад, для тестування медичних систем Faker може генерувати дані пацієнтів, що включають імена, дати народження та адреси. Це дозволяє створювати великі обсяги унікальних даних із мінімальними зусиллями. Крім того, бібліотека забезпечує можливість налаштування форматів вихідних даних, що робить її універсальним інструментом для різних сценаріїв. Faker є ідеальним рішенням для створення структурованих і реалістичних даних, які відповідають специфікаціям системи.

OpenAI API надає унікальні можливості для генерації текстових даних завдяки використанню мовних моделей. Цей підхід дозволяє створювати складні текстові конструкції, такі як описи, рекомендації чи замітки. Генерація даних за допомогою OpenAI API відзначається високим рівнем реалістичності завдяки здатності моделі адаптувати текст до заданого контексту. Наприклад, у сценаріях, де потрібне заповнення складних текстових полів, OpenAI API може створити реалістичний текст на основі заданих параметрів. Це особливо корисно для тестування систем, які працюють із текстовою інформацією, наприклад, систем підтримки рішень або інтеграційних платформ. Переваги OpenAI API включають адаптивність до різних мов і сценаріїв, а також здатність генерувати нестандартні текстові дані. Проте використання цього методу потребує певних обчислювальних ресурсів і може бути менш ефективним для сценаріїв із великим обсягом даних.

Gemini API є ще одним потужним інструментом для генерації структурованих даних. Його основна перевага полягає у здатності обробляти великі обсяги даних і генерувати їх у форматах, необхідних для інтеграції з різними інформаційними системами. Завдяки своїй високій швидкодії Gemini API дозволяє створювати великі набори даних для стрес-тестування чи симуляції масових запитів. Наприклад, за допомогою цього інструменту можна автоматизувати створення тисяч повідомлень із заданими параметрами, що є важливим для перевірки продуктивності інформаційних систем. Gemini API також підтримує налаштування форматів вихідних даних і адаптацію до специфічних вимог.

Перевага використання Gemini API полягає у його точності та масштабованості, що робить його ідеальним вибором для складних сценаріїв генерації даних.

2.4 Алгоритми оптимізації генерації HL7-повідомлень

Оптимізація генерації HL7-повідомлень є важливим завданням для забезпечення ефективності, швидкодії та масштабованості системи. Це особливо актуально в умовах роботи з великими обсягами даних або потреби у швидкому створенні повідомлень у реальному часі.

Кешування є одним із найпоширеніших методів оптимізації, що дозволяє зменшити кількість обчислень за рахунок повторного використання результатів, отриманих раніше. У контексті генерації HL7-повідомлень це може включати збереження часто використовуваних значень, таких як шаблони сегментів або попередньо згенеровані фрагменти даних. Кешування допомагає уникнути повторної обробки однакових запитів, що значно скорочує час виконання. Наприклад, сегменти, які залишаються незмінними у багатьох повідомленнях (MSH або PID), можуть бути попередньо збережені та повторно використані для швидкого формування нових повідомлень.

Багатопоточність дозволяє одночасно обробляти кілька задач, що значно підвищує продуктивність системи. У випадку генерації HL7-повідомлень це означає створення декількох повідомлень одночасно. Алгоритми багатопотокової обробки розподіляють завдання між потоками, що працюють паралельно, використовуючи ресурси процесора максимально ефективно. Для реалізації багатопоточності важливо забезпечити синхронізацію доступу до загальних ресурсів, наприклад, до шаблонів або баз даних. Це дозволяє уникнути конфліктів і забезпечити стабільність роботи системи навіть за високого навантаження.

Попередня генерація даних є ще одним ефективним методом оптимізації. Цей підхід передбачає створення необхідного набору HL7-повідомлень заздалегідь, до моменту, коли вони будуть потрібні. Заздалегідь згенеровані дані можуть зберігатися у файлах або базах даних і бути доступними для миттєвого використання. Цей метод особливо корисний для сценаріїв, де потрібна велика кількість повідомлень із типовими значеннями. Наприклад, для стрес-тестування

системи можна підготувати набір із тисяч повідомлень, що відповідають заданим параметрам, і використовувати їх для симуляції масових запитів.

Ефективна реалізація алгоритмів є ключовим фактором для оптимізації генерації HL7-повідомлень. Це включає мінімізацію кількості операцій, використання ефективних структур даних і оптимізацію роботи з пам'яттю.

Наприклад, замість створення кожного сегмента з нуля можна використовувати бібліотеки, такі як hl7apy, які забезпечують готові моделі для формування сегментів. Це дозволяє зменшити складність коду, прискорити процес розробки та мінімізувати ризик помилок.

Поєднання різних підходів до генерації даних також сприяє оптимізації. Наприклад, випадково згенеровані дані можна поєднувати з заздалегідь підготовленими шаблонами, щоб створювати повідомлення, які відповідають специфічним сценаріям. Це дозволяє досягти балансу між гнучкістю і швидкістю. Комбінування різних методів також може включати інтеграцію штучного інтелекту для генерації складних текстових полів і використання алгоритмів кешування для повторного використання шаблонів. Такий підхід забезпечує як масштабованість, так і відповідність згенерованих даних реальним умовам.

Ефективне використання ресурсів є ще одним важливим аспектом оптимізації. Це включає управління пам'яттю, уникнення надлишкових операцій вводу/виводу та оптимізацію роботи з базами даних. Наприклад, використання індексів у базах даних дозволяє значно прискорити доступ до збережених даних.

Крім того, варто мінімізувати використання ресурсів на стороні клієнта. Це можна реалізувати через передачу лише необхідної інформації замість повних повідомлень або використання компактних форматів даних.

2.5 Характеристики програмних засобів для реалізації

Для успішної реалізації автоматизованої системи генерації даних необхідно правильно підібрати програмні засоби, які забезпечують гнучкість, продуктивність та відповідність вимогам. Вибір інструментів ґрунтується на їх функціональності, зручності інтеграції та масштабованості.

Характеристика основних програмних засобів:

1) Python є однією з найбільш універсальних і популярних мов програмування, яка забезпечує швидкий і ефективний процес розробки. Завдяки своїй простій синтаксичній структурі Python значно скорочує час розробки програмного забезпечення. Мова має широку екосистему бібліотек і фреймворків, що охоплюють усі аспекти розробки — від роботи з даними до побудови веб-додатків і систем автоматизації. Python підтримує багатопоточність і багатопроцесорність, що забезпечує оптимізацію продуктивності для задач, пов'язаних із генерацією великих обсягів даних. Крім того, гнучкість Python дозволяє легко інтегрувати нові технології, зокрема зовнішні API та бібліотеки для роботи зі специфічними стандартами.

2) Flask — це мікрофреймворк для створення веб-додатків, який широко використовується завдяки своїй простоті та модульності. Він дозволяє реалізовувати RESTful API, що забезпечує ефективну взаємодію між клієнтською та серверною частинами системи. Flask дозволяє легко налаштовувати серверну частину для обробки запитів, інтеграції з базами даних та зовнішніми інструментами. Його легкість у використанні робить Flask ідеальним вибором для проєктів, де потрібна швидка реалізація і гнучка адаптація до вимог.

3) Бібліотека Faker забезпечує генерацію реалістичних тестових даних, таких як імена, адреси, номери телефонів, дати народження.

4) Бібліотека hl7apy спеціалізується на створенні, парсингу та перевірці HL7-повідомлень, дозволяючи працювати з медичними даними відповідно до стандартів.

5) Сучасні API, такі як OpenAI API та Gemini API, забезпечують інтеграцію функцій штучного інтелекту в програмні рішення. Вони дозволяють створювати складні текстові конструкції, які адаптуються до контексту, та генерувати масштабовані набори даних. OpenAI API ефективно використовується для автоматизації генерації текстових даних, включаючи сценарії, звіти та описові поля. Gemini API, зі свого боку, забезпечує швидку генерацію великих обсягів структурованих даних із можливістю інтеграції в різні інформаційні системи.

2.6 Постановка задачі обчислювального експерименту

Проведення обчислювального експерименту є ключовим етапом для оцінки ефективності та надійності системи автоматизованої генерації HL7-повідомлень. Основна мета експерименту полягає у перевірці відповідності створених даних стандарту HL7, аналізі продуктивності системи, а також оцінці реалістичності згенерованих повідомлень у різних умовах експлуатації.

Експеримент охоплює кілька важливих аспектів. Перевірка відповідності HL7-повідомлень передбачає аналіз правильності структури сегментів та полів у створених повідомленнях. Це дозволяє переконатися, що система генерує дані, які можуть бути успішно інтегровані у медичні інформаційні системи.

Продуктивність системи оцінюється через вимірювання часу генерації різних обсягів HL7-повідомлень. Це дозволяє зрозуміти, як система поводить себе за умов інтенсивного використання та великих навантажень.

Реалістичність даних є ще одним важливим параметром експерименту. Оцінюється правдоподібність текстових полів, таких як імена пацієнтів, клінічні записи та рекомендації. Аналіз унікальності й випадковості створених даних гарантує, що вони відповідають реальним сценаріям використання.

Крім того, експеримент охоплює перевірку масштабованості та стабільності роботи системи. Це включає генерацію великих обсягів повідомлень і аналіз її стійкості до тривалих навантажень.

Методика проведення експерименту передбачає підготовку різноманітних сценаріїв тестування, які охоплюють широкий спектр умов: від невеликого обсягу даних до високонавантажених систем. Для кожного сценарію генеруються відповідні набори HL7-повідомлень та аналізуються отримані результати.

Очікувані результати експерименту включають коректну відповідність HL7-стандартам, високу продуктивність навіть за великих обсягів даних, значне прискорення роботи завдяки багатопоточності, а також створення реалістичних і стабільних даних, які легко інтегруються з іншими системами.

Успішне проведення обчислювального експерименту дозволяє переконатися у функціональності, ефективності та готовності системи до використання в реальних умовах медичних інформаційних систем. Це також забезпечує її відповідність високим стандартам продуктивності та стабільності.

Висновок до розділу 2

У цьому розділі проведено аналіз стандарту HL7, його структури та принципів побудови повідомлень, що забезпечує ефективний обмін даними між медичними системами. Розглянуто типи повідомлень і їхні сегменти, що дозволяє розуміти їхнє призначення та контекст застосування. Описані методи парсингу, модифікації та нормалізації HL7-повідомлень забезпечують коректну обробку даних, тоді як алгоритми оптимізації гарантують високу продуктивність системи.

Отримані результати є основою для подальшої розробки системи автоматизованої генерації HL7-повідомлень, її тестування та впровадження в практичну діяльність медичних закладів.

3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМИ АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ ТЕСТОВИХ ПОВІДОМЛЕНЬ HL7

Розробка ефективної системи автоматизованої генерації HL7-повідомлень потребує продуманої архітектури, яка забезпечує гнучкість, масштабованість та інтеграцію з сучасними технологіями. В основу системи покладено клієнт-серверну архітектуру, що включає клієнтську частину для взаємодії з користувачем, серверну частину для обробки запитів і генерації даних, а також зовнішні API та бібліотеки для забезпечення високої якості та реалістичності створених даних. Такий підхід дозволяє ефективно розподілити функціональність між компонентами системи, що сприяє її продуктивності та зручності використання.

3.1 Архітектура програмної системи

Розроблена система автоматизованої генерації HL7-повідомлень побудована на основі клієнт-серверної архітектури. Вона складається з клієнтської частини, серверної частини та зовнішніх API, які забезпечують генерацію, збереження та експорт даних.

Основні компоненти системи:

1) клієнтська частина:

- веб-інтерфейс, який дозволяє користувачу взаємодіяти із системою.
- забезпечує введення параметрів генерації, вибір типу HL7-повідомлення та методу генерації, а також відображення результатів з можливістю експорту.

2) серверна частина:

- реалізована на Flask, відповідає за обробку запитів від клієнта.

- здійснює генерацію HL7-повідомлень за допомогою інтеграції з бібліотеками та API.
- забезпечує експорт даних у потрібному форматі.

3) зовнішні API та бібліотеки:

- hl7apy: Використовується для роботи зі стандартом HL7.
- faker: Генерує випадкові, але реалістичні дані для полів повідомлень.
- OpenAI API: Використовується для контекстуальної генерації складних текстових даних.
- Gemini API: Забезпечує реалістичну генерацію медичних даних та сценаріїв.

На діаграмі архітектури (рисунок 3.1) відображено клієнтську частину, сервер, API та їх взаємодію через мережеві запити.

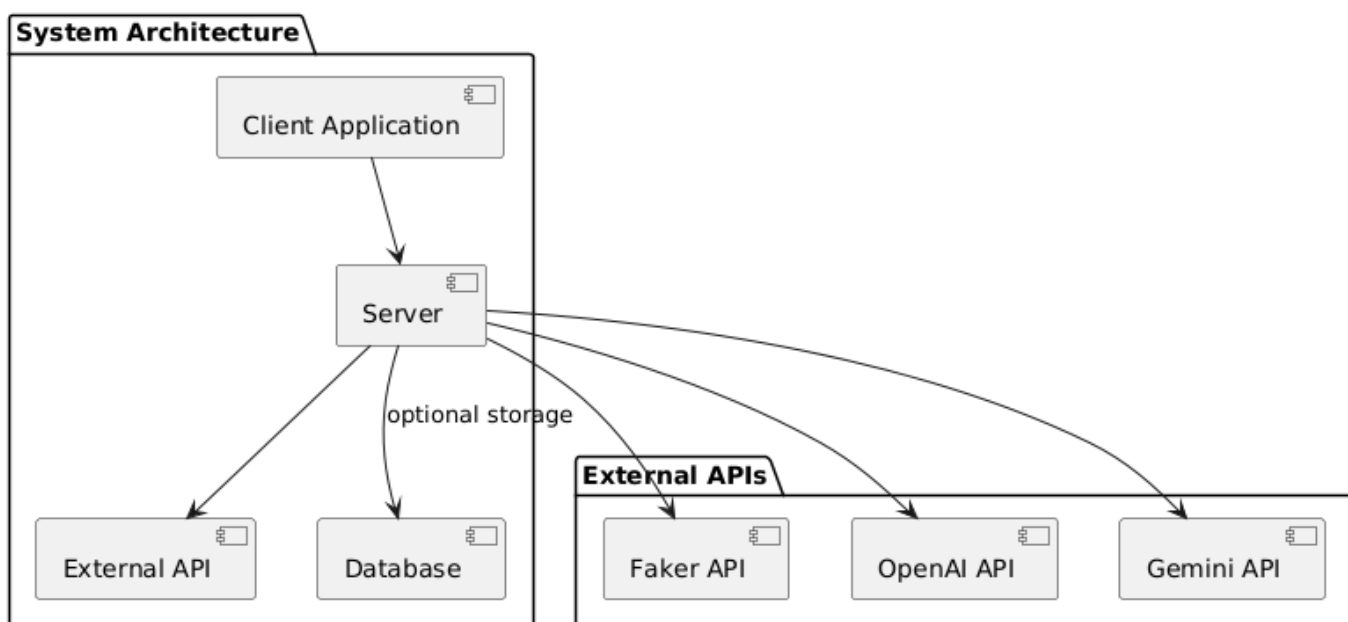


Рисунок 3.1 – Діаграма архітектури системи

Блок-схема роботи програми (рисунок 3.2) відображає послідовність виконання ключових операцій, необхідних для створення стандартизованих медичних даних.

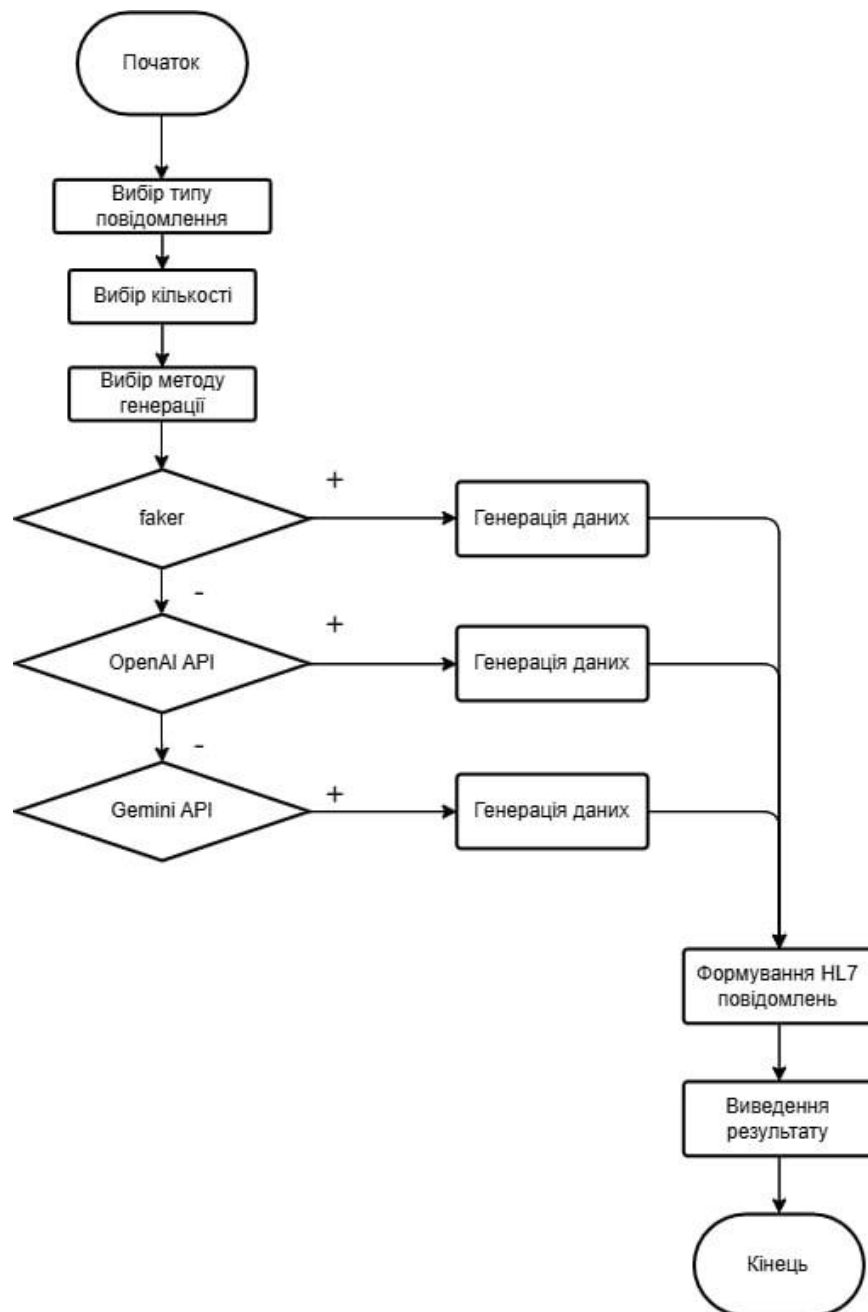


Рисунок 3.2 – Блок схема роботи програми

Блок-схема демонструє основні етапи роботи програми для автоматичної генерації HL7-повідомлень.

1. Початок: процес починається з ініціації роботи програми.
2. Вибір типу повідомлення: користувач обирає тип HL7-повідомлення, наприклад, ADT, ORM чи інший, який необхідно згенерувати.

3. Вибір кількості повідомлень: користувач задає кількість HL7-повідомлень, які необхідно створити.

4. Вибір методу генерації: на цьому етапі користувач визначає, який метод буде використовуватися для генерації даних:

- Faker
- OpenAI API
- Gemini API

5. Перевірка вибору методу генерації:

- якщо вибрано Faker, система переходить до генерації даних цим методом.
- якщо вибрано OpenAI API, здійснюється генерація за допомогою відповідного API.
- якщо вибрано Gemini API, система використовує цей інструмент для створення даних.

6. Генерація даних: на цьому етапі відповідно до вибраного методу виконується автоматизована генерація даних.

7. Формування HL7-повідомлень: згенеровані дані об'єднуються у структуру відповідно до стандарту HL7.

8. Виведення результату: програма формує готові HL7-повідомлення та надає їх у вигляді результату, який можна переглянути або експортувати.

9. Кінець: робота програми завершується після успішного виведення результатів.

Ця блок-схема ілюструє послідовність дій у програмі, починаючи від вибору параметрів користувачем і закінчуючи отриманням готового результату у форматі HL7.

3.2 Модель представлення даних

Модель представлення даних у системі автоматизованої генерації HL7-повідомлень розроблена для забезпечення ефективної організації, передачі та

зберігання інформації між компонентами системи. Особливу увагу приділено підтримці структурованих форматів даних, сумісних із сучасними стандартами обміну інформацією.

JSON [14] є основним форматом обміну даними між клієнтською і серверною частинами системи. Він забезпечує простоту використання, високу швидкодію та легкість інтеграції з веб-інтерфейсом.

У форматі JSON передаються наступні елементи:

- параметри, задані користувачем у клієнтському інтерфейсі (тип повідомлення, кількість, метод генерації).
- згенеровані дані HL7 для попереднього перегляду у веб-інтерфейсі.
- статус і час виконання запиту.

HL7 використовується як цільовий формат для експорту даних, відповідно до вимог медичних стандартів. Згенеровані повідомлення формуються відповідно до специфікацій HL7 і містять такі компоненти:

- сегменти (MSH, EVN, PID, PV1 тощо).
- структуровані поля й підполя із валідацією на відповідність стандарту.
- вказівки щодо обробки, наприклад, тип події чи інформація про пацієнта.

XML [15] надається як додатковий формат для експорту даних, що підходить для систем, які підтримують обробку структурованих даних у форматі XML.

Основні особливості XML:

- чітка ієрархічна структура, яка відображає взаємозв'язок сегментів HL7.
- зручність інтеграції з іншими системами, що підтримують обмін XML-документами.

Обмін даними між компонентами системи відбувається наступним чином:

- 1) Клієнтська частина передає запит на сервер у форматі JSON, який містить вибрані користувачем параметри, такі як тип HL7-повідомлення, метод генерації та кількість повідомлень.

2) Серверна частина обробляє отримані дані, формує HL7-повідомлення за допомогою бібліотек HL7apy, Faker або API, і повертає результат клієнту у вигляді JSON.

3) Згенеровані дані можуть бути перетворені на інші формати (HL7 або XML) для експорту чи інтеграції з іншими системами.

Таким чином, модель представлення даних побудована на основі сучасних стандартів і забезпечує гнучкість у роботі з різними форматами, що підвищує ефективність і сумісність системи.

3.3 Опис клієнтської частини

Клієнтська частина реалізована як веб-інтерфейс, що дозволяє користувачам взаємодіяти з застосунком через браузер. Вона відповідає за збір даних, ініціалізацію запитів до серверної частини та відображення результатів.

Веб-інтерфейс створено з використанням HTML для структури сторінок, CSS для візуального оформлення та JavaScript для інтерактивності. Дизайн забезпечує зручний доступ до функцій додатка та адаптивний вигляд для різних пристроїв.

На веб-сторінці реалізовані форми, які дозволяють користувачеві вибрати тип HL7-повідомлення, кількість повідомлень для генерації та метод їх створення.

JavaScript забезпечує взаємодію користувача зі сторінкою, обробляючи кліки, вибір параметрів та відправку форми. Завдяки цьому сторінка не перезавантажується, а дані обробляються асинхронно через AJAX-запити [16].

На рисунку 3.3 зображена діаграма активності, яка демонструє ключові етапи процесу взаємодії користувача з клієнтською частиною системи автоматизованої генерації HL7-повідомлень. Діаграма ілюструє послідовність дій користувача, а також логіку роботи клієнтської частини під час обміну даними із сервером.

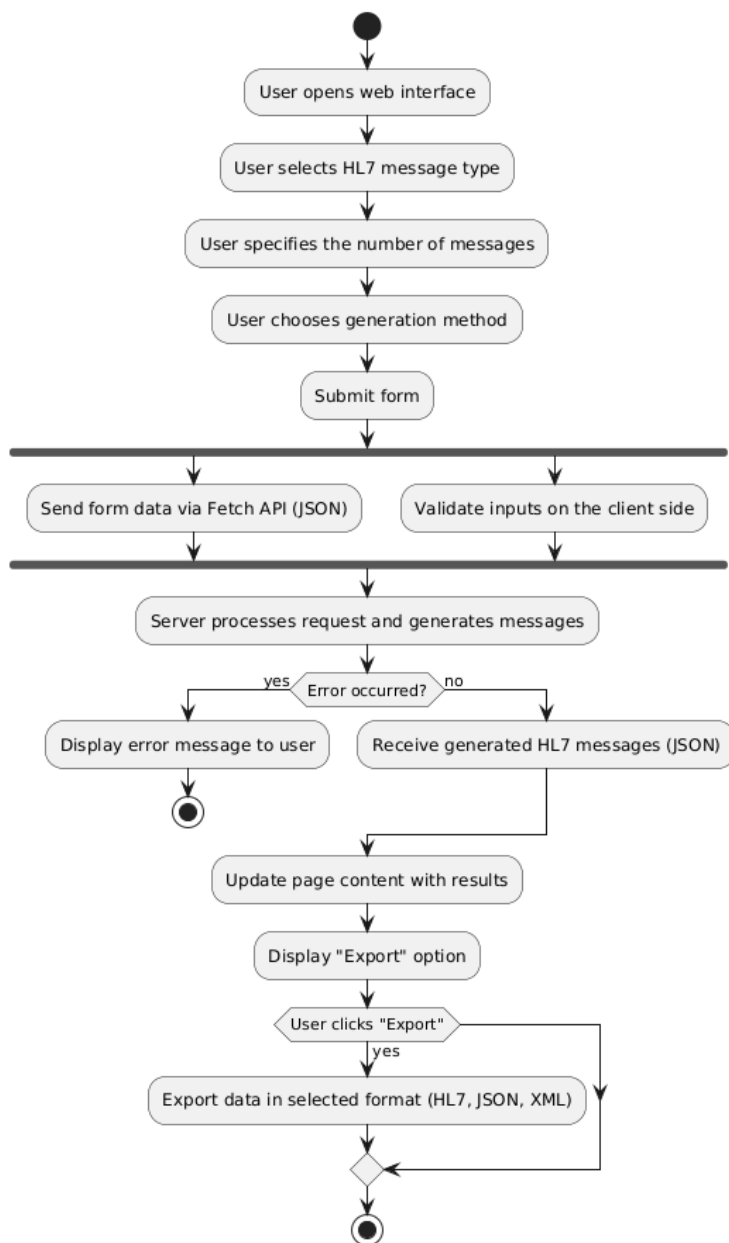


Рисунок 3.3 – Діаграма активностей

Процес починається з відкриття користувачем веб-інтерфейсу системи через браузер. Користувач обирає тип HL7-повідомлення, наприклад, ADT, ORM чи ORU, залежно від своїх потреб, задає кількість повідомлень для генерації та вибирає метод створення даних (Faker, Gemini API, OpenAI API). Після заповнення всіх необхідних параметрів користувач натискає кнопку "Генерувати", яка ініціює запит до серверної частини.

Дані з форми передаються на сервер за допомогою Fetch API у форматі JSON. Перед відправленням виконується валідація введених параметрів на клієнтському боці, щоб уникнути можливих помилок. Сервер отримує запит, обробляє його та повертає результат — згенеровані HL7-повідомлення.

Після отримання відповіді клієнтська частина перевіряє її на наявність помилок. У разі виникнення помилки (наприклад, некоректного запиту чи внутрішньої серверної помилки) користувач отримує відповідне повідомлення про проблему. Якщо ж запит було успішно оброблено, отримані HL7-повідомлення завантажуються на веб-сторінку.

Після успішної генерації користувач має можливість експортувати згенеровані дані у вибраному форматі (HL7, JSON чи XML) за допомогою кнопки "Експортувати". Це забезпечує просту інтеграцію створених даних у сторонні системи для тестування або інших цілей.

3.4 Опис серверної частини

Серверна частина системи побудована на основі Flask, що забезпечує швидку та гнучку організацію обробки запитів, генерації даних та інтеграції з зовнішніми сервісами. Основу роботи серверної частини складають кілька ключових маршрутів, які отримують запити від клієнта та повертають відповідні результати. Головний маршрут, /generate, приймає параметри, які надходять із клієнтської частини, обробляє їх і формує відповідь у форматі JSON.

Для обробки запитів сервер отримує дані через POST-запити, що містять інформацію про тип повідомлення, метод генерації та кількість повідомлень. Перед початком обробки сервер проводить валідацію вхідних даних, щоб запобігти помилкам та некоректним запитам.

Процес генерації HL7-повідомлень реалізується за допомогою кількох підходів. Використовуючи бібліотеку HL7ару, сервер створює структуровані HL7-повідомлення, які відповідають стандарту HL7 і дозволяють легко

модифікувати дані. Додатково, для генерації текстових HL7-повідомлень, що адаптовані до контексту, застосовується API OpenAI, який забезпечує створення динамічного контенту. Інтеграція з Gemini API надає можливість отримувати медичні шаблони та додаткові параметри, необхідні для конкретних типів повідомлень. Для створення випадкових тестових даних, таких як імена пацієнтів, дати народження, адреси та контактна інформація, використовується бібліотека Faker. Це дозволяє автоматизувати заповнення повідомлень та зберегти реалістичність тестових даних.

Серверна частина забезпечує високий рівень безпеки та конфіденційності. Для захисту конфіденційних даних, таких як ключі до API OpenAI та Gemini, застосовується бібліотека dotenv, яка дозволяє зберігати ці дані у захищеному файлі .env, недоступному для сторонніх.

Усі запити, що обробляються сервером, захищені обробниками виключень. Це дозволяє у разі виникнення помилок, таких як збій інтеграції з API, формувати детальні повідомлення про проблему та надсилати їх до клієнта. Таким чином, користувач отримує точну інформацію про характер помилки та може вчасно виправити проблему.

Результати генерації HL7-повідомлень повертаються клієнтській частині у форматі JSON. Кожне згенероване повідомлення містить статус успішності виконання та додаткову інформацію, що дозволяє клієнту легко використовувати отримані дані для подальшої роботи чи тестування.

Висновок до розділу 3

Система автоматизованої генерації HL7-повідомлень, побудована на клієнт-серверній архітектурі, забезпечує зручний інтерфейс для користувача та ефективну обробку запитів сервером. Інтеграція з HL7apy, Faker, OpenAI API і Gemini гарантує реалістичність і відповідність даних стандарту HL7. Також реалізований експорт у формати HL7, JSON, XML дозволяє зручно працювати з згенерованими повідомленнями.

4 ТЕСТУВАННЯ ТА ПОРІВНЯННЯ МЕТОДІВ, ДЕМОНСТРАЦІЯ ФУНКЦІОНАЛЬНОСТІ

Цей розділ присвячено аналізу роботи розробленої системи автоматизованої генерації HL7-повідомлень через проведення тестування та порівняння застосованих методів. У ньому описується процес взаємодії користувача із системою, порівнюються результати, отримані за допомогою різних методів генерації даних, а також виконується перевірка валідності створених HL7-повідомлень.

4.1 Інсталяція програмного забезпечення

Для розгортання системи автоматичної генерації HL7-повідомлень необхідно виконати кілька кроків, які забезпечать налаштування серверної та клієнтської частини, а також інтеграцію з зовнішніми бібліотеками та API. Нижче наведено докладний процес інсталяції програмного забезпечення.

Перед початком інсталяції переконайтесь, що ваш пристрій відповідає наступним мінімальним вимогам:

- операційна система: Windows, macOS або Linux;
- python: версія 3.8 або вище;
- браузер: сучасний браузер із підтримкою JavaScript (наприклад, Chrome, Firefox, Edge);
- інтернет-з'єднання: для доступу до зовнішніх API;

Кроки інсталяції:

1. На початку користувач повинен переконатися, що на його системі встановлено Python версії 3.8 або вище. У разі відсутності Python, його можна завантажити та встановити з офіційного сайту.

2. Завантажте архів із вихідним кодом застосунку.

3. У каталозі проєкту виконайте команду для створення віртуального середовища:

- `python -m venv env`

4. Активація середовища:

- `source env/bin/activate` # для Linux/macOS `env\Scripts\activate` # для Windows

5. Після активації середовища виконайте команду:

- `pip install -r requirements.txt`

Це встановить усі необхідні бібліотеки для роботи застосунку.

6. Створіть файл `.env` у кореневому каталозі проєкту та додайте до нього наступні параметри:

- `OPENAI_API_KEY=ваш_ключ`

- `GEMINI_API_KEY=ваш_ключ`

Ключі OpenAI API та Gemini API можна отримати після реєстрації у відповідних сервісах.

7. Для запуску сервера виконайте команду:

- `flask run`

Після запуску сервера користувач може відкрити веб-інтерфейс у браузері, перейшовши за адресою `http://127.0.0.1:5000`.

Після завершення інсталяції система готова до роботи, і ви можете використовувати її для генерації HL7-повідомлень та тестування медичних інформаційних систем.

4.2 Робота користувача з системою

Відправною точкою роботи з системою, є головна сторінка, вона зображена на рисунку 4.1.

The image shows a web application titled "HL7 ГЕНЕРАТОР ПОВІДОМЛЕНЬ". It features a vertical list of 15 light blue buttons, each containing the name of an HL7 message type and its full name in parentheses. The buttons are arranged in a single column and are separated by thin white lines. The background is a light gray.

HL7 Message Type
ADT (Admit, Discharge, Transfer)
ORM (Order)
ORU (Observation Result)
ACK (Acknowledgment)
MDM (Medical Document Management)
SCH (Schedule Information)
SIU (Scheduling Information Unsolicited)
DFT (Detail Financial Transactions)
BAR (Billing Account)
RDE (Pharmacy/Treatment Encoded Order)
VXU (Unsolicited Vaccination Record Update)
QBP (Query by Parameter)
MFN (Master File Notification)
RDS (Pharmacy/Treatment Dispense)
RDT (Display-based Response)
UDI (Unique Device Identifier)

Рисунок 4.1 – Головна сторінка системи

На цій сторінці користувач має змогу обрати тип HL7-повідомлення, яке він бажає генерувати. Система пропонує широкий вибір типів повідомлень, що відповідають різним стандартам HL7, зокрема:

- ADT (Admit, Discharge, Transfer): обробка даних про госпіталізацію, виписку або переведення пацієнтів.
- ORM (Order): створення медичних замовлень, таких як лабораторні тести або процедури.
- ORU (Observation Result): передача результатів досліджень і спостережень.
- ACK (Acknowledgment): підтвердження отримання або обробки повідомлень.
- MDM (Medical Document Management): керування медичними документами, наприклад, виписками чи записами.
- SCH (Schedule Information): інформація про розклади процедур і консультацій.
- SIU (Scheduling Information Unsolicited): передачі змін у розкладі, таких як перенесення або скасування подій.
- DFT (Detail Financial Transactions): деталізація фінансових транзакцій.
- BAR (Billing Account): облік медичних рахунків.
- VXU (Unsolicited Vaccination Record Update): оновлення інформації про щеплення.
- RDE (Pharmacy/Treatment Encoded Order): замовлення на медикаменти чи процедури.
- QBP (Query by Parameter): запити за заданими параметрами.
- MFN (Master File Notification): повідомлення про зміни в базах даних.
- RDS (Pharmacy/Treatment Dispense): інформація про відпуск медикаментів.
- RDT (Display-based Response): передача результатів на запити.
- UDI (Unique Device Identifier): передача унікальних ідентифікаторів медичних пристроїв.

Після того, як користувач обрав тип повідомлення, система перенаправляє його на сторінку генерації повідомлення обраного типу (рисунок 4.2).

Головна

ADT - HL7 Message Type

ADT (Admit, Discharge, Transfer) - повідомлення для передачі інформації про дії з пацієнтами.

Генерація ADT повідомлень

Виберіть субтип:

ADT_A01 – Прийом пацієнта

Виберіть метод генерації:

AI Gemini

Кількість повідомлень:

1

Генерувати

Рисунок 4.2 – Сторінка генерації повідомлення

На сторінці генерації зверху вказано, для чого використовується повідомлення даного типу. Також користувач має змогу налаштувати параметри для створення відповідного HL7-повідомлення. При цьому поле підтип є доступний відповідно до того, чи має його даний тип повідомлення. Вибір підтипу повідомлення зображений на рисунку 4.3.

Генерація ADT повідомлень

Виберіть субтип:

ADT_A02 – Переведення пацієнта
ADT_A01 – Прийом пацієнта
ADT_A02 – Переведення пацієнта
ADT_A03 – Виписка пацієнта
ADT_A04 – Реєстрація пацієнтів
ADT_A05 – Попередній прийом пацієнта
ADT_A08 – Оновлення інформації про пацієнта
ADT_A11 – Скасування прийому пацієнта
ADT_A12 – Скасування переведення пацієнта

Генерувати

Рисунок 4.3 – Вибір підтипу повідомлення

Після того, як підтип було обрано користувач обирає метод генерації, як на рисунку 4.4 та натискає кнопку “Генерувати”.

Генерація ADT повідомлень

Виберіть субтип:

ADT_A02 – Переведення пацієнта

Виберіть метод генерації:

AI Gemini
AI Gemini
AI ChatGPT
Faker

Генерувати

Рисунок 4.4 – Вибір методу генерації даних

Після обробки запиту система повертає згенеровані HL7 повідомлення та час, який знадобився для їх генерації, приклад на рисунку 4.5.

ADT - HL7 Message Type

ADT (Admit, Discharge, Transfer) - повідомлення для передачі інформації про дії з пацієнтами.

Генерація ADT повідомлень

Виберіть субтип:

ADT_A08 – Оновлення інформації про пацієнта

Виберіть метод генерації:

Faker

Кількість повідомлень:

20

Генерувати

PID|1|0c4eed49-b371-4db0-81c2-5a767c21a897||Kristen Allen|1982-02-08|F|062 Brandon Roads, South Amanda, GA 60891|78561

- MSH|^~\&|Park and Sons|Scott Ltd|Webb and Sons|Sanchez-Murphy|20240101000000||ADT^A08^ADT_A08|1|P|2.5
PID|1|4d8555d0-95ce-4e0b-9fc4-be5175900358||Erin Newman|1972-03-08|F|2066 Soto Curve, New Brandonview, KS 15995|277311
- MSH|^~\&|Boyer-McConnell|Soto, Livingston and Moore|Stevens Inc|Marshall, Green and Sweeney|20240101000000||ADT^A08^ADT_A08|1|P|2.5
PID|1|de88d825-3cde-4c35-a23a-5a1b87995417||Michelle Middleton|1963-12-18|M|891 Jay Coves Suite 132, Port Brian, CT 12
- MSH|^~\&|Neal, Davis and Mitchell|Johnson PLC|Rogers Group|Lee-Chavez|20240101000000||ADT^A08^ADT_A08|1|P|2.5
PID|1|9eda6d20-8de9-4811-ad09-1fad0775e0e7||Caleb Cervantes|1936-06-03|M|32705 Scott Skyway Suite 975, Marshalltown, IA 24336|812-418
- MSH|^~\&|King PLC|Richard-Gonzalez|Clark Group|Bryant Group|20240101000000||ADT^A08^ADT_A08|1|P|2.5
PID|1|b5c99f21-3f00-44f1-8a32-7818586b09a2||Andre Pineda|1960-08-31|M|56171 Ramirez Creek, Jasonfurt, IA 24336|812-418
- MSH|^~\&|Welch-Murphy|Jones, Turner and Gonzalez|Erickson, Ford and Benson|Cook Ltd|20240101000000||ADT^A08^ADT_A08|1|P|2.5
PID|1|2980fcfe-4ac4-4799-908d-3aef9bddc4a1||Frederick Evans|1989-01-13|F|328 Thomas Shoal, Ayalafurt, VT 76985|237.839
- MSH|^~\&|Greene, Perkins and Lawrence|Smith-Morgan|Clarke, Moore and Brown|Smith, Smith and Adams|20240101000000||ADT^A08^ADT_A08|1|P|2.5
PID|1|520dfcf4-c48e-47a1-b6b7-d311f983cb94||Renee Rogers|1997-05-07|F|965 McBride Lodge, Matthewmouth, PR 65079|(459)5
- MSH|^~\&|Lewis-Green|Gillespie, Garcia and Rocha|Phillips, Hancock and Myers|Martinez Inc|20240101000000||ADT^A08^ADT_A08|1|P|2.5
PID|1|31cf3e82-752a-422d-91d9-a91641be392d||Matthew Green|1926-04-13|M|0996 Jane Burgs Apt. 046, South Amandashire, SC

Time Elapsed: 0.02

hl7

json

xml

Виберіть формат: hl7 Експортувати

Рисунок 4.5 - Відображення результатів генерації повідомлень

Після отримання результатів з'являється кнопка “Експортувати” із вибором зручного формату, Результати можна зручно скопіювати або експортувати у зручний формат.

4.3 Порівняння результатів генерації

Для оцінки ефективності системи автоматизованої генерації HL7-повідомлень було проведено обчислювальний експеримент, метою якого стало порівняння трьох різних методів генерації даних: Faker, Gemini API, та OpenAI API. Основними критеріями для оцінки стали продуктивність (час, потрібний на генерацію заданої кількості повідомлень) та якість створених даних (реалістичність, наповненість, відповідність контексту).

Користувач вибрав генерацію повідомлень типу ADT_A02 – Переведення пацієнта. Для кожного методу генерації було задано однакову кількість повідомлень.

Кожен метод оцінювався окремо, результати порівнювалися на основі наступних параметрів:

- 1) Час, необхідний для створення 10 повідомлень.
- 2) Наповненість та реалістичність згенерованих даних.
- 3) Відповідність створених даних специфікаціям HL7.

Результати експерименту:

- 1) Метод Faker (рисунок 4.6)

Час генерації склав 0.01 секунди. В результаті відображаються базові текстові дані з використанням шаблонів. Поля заповнені простими випадковими даними (імена, адреси, дати). Основні переваги такого методу це швидкість генерації та простота реалізації. Згенеровані дані виглядають менш реалістичними через одноманітність та відсутність логічного контексту.

Метод Faker є оптимальним для задач, де важлива швидка генерація великих обсягів простих тестових даних, однак його обмеження роблять цей підхід менш придатним для сценаріїв, де потрібна реалістичність та контекстна логіка.

ADT - HL7 Message Type

ADT (Admit, Discharge, Transfer) - повідомлення для передачі інформації про дії з пацієнтами.

Генерація ADT повідомлень

Виберіть субтип:

ADT_A02 – Переведення пацієнта

Виберіть метод генерації:

Faker

Кількість повідомлень:

10

Генерувати

```

PID|1|e5d230f0-ccbfc-4817-8929-90813773e19b||Jesse Schultz|1984-04-17|F|9913 Jeremy Point Suite 745, Richardfort, GU 93.
• MSH|^~\&|Baker, Cook and Cowan|Martinez, Robles and Schaefer|Jordan, Henson and Rodgers|Conley LLC|20240101000000||ADT^v
PID|1|b082e59f-527d-4853-bba6-e325e49bfe18||Cody Mack|1931-05-05|F|446 Wells Lock Suite 834, Alexanderberg, MT 76476|0
• MSH|^~\&|Woods, Smith and Bush|Hunt Inc|Bennett-Huffman|Wise and Sons|20240101000000||ADT^A02^ADT_A02|1|P|2.5
PID|1|e95cc95f-b28e-41ce-9861-3f9fb4901a07||Anna Alvarado|1981-10-02|H|171 Foster Street, Meltonfurt, MN 21630|719-214
• MSH|^~\&|Mason and Sons|Donovan, Berry and Wilson|Fowler-Shields|King-Jones|20240101000000||ADT^A02^ADT_A02|1|P|2.5
PID|1|c90ec21e-6459-4af4-b0b4-b0823e8e00c5||Linda Taylor|1947-03-14|F|65303 Paul Port Suite 657, Lake Maryland, MP 321
• MSH|^~\&|Murphy-Jennings|Williams Inc|Moreno, Wright and Shaw|Turner-golden|20240101000000||ADT^A02^ADT_A02|1|P|2.5
PID|1|f86e470c-9dd3-430a-9614-fcdf5a7e0a16||Mr. David Miller|1941-10-26|F|0618 Liu Stravenue, Clayport, WZ 42867|70667.
• MSH|^~\&|Brown-Jones|Martin-Wheeler|Berry-Grant|Barr, Strickland and Boyd|20240101000000||ADT^A02^ADT_A02|1|P|2.5
PID|1|e811f570-0c90-4a7e-9f2f-f634cbe863f0||Bryan Branch|1985-07-16|F|Unit 6419 Box 3406, DPO AA 80007|001-836-481-0634
• MSH|^~\&|Jackson, Roberts and Johns|Griffin-Cook|Hall, Mitchell and Bush|Garrett-Diaz|20240101000000||ADT^A02^ADT_A02|1
PID|1|b13f644d-9ed5-4a8d-942c-8f2a0ff2f4ae||Jennifer Smith|1973-04-14|M|USS Hamilton, FPO AA 22281|939-246-8704|74354|
• MSH|^~\&|Johnson, Price and Jones|Moses, Lee and Pugh|Williams-Henderson|Patel-Newman|20240101000000||ADT^A02^ADT_A02|1
PID|1|c64eaf3-cff8-4f08-90e6-37e43ff8939c||Debbie Anderson|1967-06-07|F|0591 Aguilar Stream Suite 928, Sharonborough,

```

Time Elapsed: 0.01 seconds

Виберіть формат:

hl7

Експортувати

Рисунок 4.6 - Генерація 10 повідомлень за допомогою Faker

2) Метод Gemini API

Результати генерації зображені на рисунку 4.7. Час генерації склав 24.36 секунди. В результаті система генерує дані високого рівня реалістичності, враховуючи логічні зв'язки між полями. Поля заповнені детальними та контекстно обґрунтованими даними, такими як імена пацієнтів, дати народження, адреси, медичні ідентифікатори та інформація про переведення пацієнтів.

Основні переваги цього методу — це реалістичність створених даних, відповідність стандарту HL7 та логічна узгодженість між полями. Згенеровані дані ідеально підходять для функціонального тестування, але час обробки значно триваліший порівняно з іншими методами.

Метод Gemini API є ефективним для створення тестових даних, які потребують високого рівня реалістичності та відповідності реальним сценаріям використання в медичних системах. Він підходить для функціонального тестування інтеграції систем, де якість даних має критичне значення. Однак, через тривалий час обробки, його ефективність може знижуватися при необхідності генерувати великі обсяги даних у короткий термін.

Генерація ADT повідомлень

Виберіть субтип:

ADT_A02 – Переведення пацієнта

Виберіть метод генерації:

AI Gemini

Кількість повідомлень:

10

Генерувати

•

MSH|^~\&|AthenaHealth|Northside Community Hospital|NextGen|Eastside Medical Center|20240101000000||ADT^A02^A0201|P112233445|||Mary Johnson|1978-04-15|F|789 Pine St, Metropolis, NY 10001|+14165553456|Dr. James Le

•

MSH|^~\&|Cerner Millennium|Westside Hospital|EpicCare|Lakeside Clinic|20240101000000||ADT^A01^ADT_A01|1|PID|1|P223344556|||Robert Brown|1963-11-02|M|321 Oak St, Gotham, MA 02111|+15165554567|Dr. Sarah Kim|INS

•

MSH|^~\&|McKesson Paragon|Central Hospital|GE Centricity|Springfield Health Center|20240101000000||ADT^A02^A0201|P334455667|||Linda Davis|1980-02-20|F|654 Maple St, River City, CA 90210|+16175555678|Dr. William

•

MSH|^~\&|eClinicalWorks|South Valley Hospital|eMDs|North Valley Medical Center|20240101000000||ADT^A01^A0101|P445566778|||James Wilson|1995-07-30|M|987 Birch St, Sunnyvale, TX 75001|+17175556789|Dr. Karen Ma

•

MSH|^~\&|NextGen Healthcare|Eastside Hospital|Allscripts Sunrise|Westside Health Clinic|20240101000000||ADT^A02^A0201|P556677889|||Patricia Taylor|1975-12-14|F|210 Cedar St, Pleasantville, FL 33001|+18175557890|Dr. F

•

MSH|^~\&|Greenway Health|Oak Ridge Hospital|Practice Fusion|Hilltop Medical Center|20240101000000||ADT^A01^A0101|P667788990|||Michael Miller|1990-03-18|M|159 Walnut St, Greenfield, WI 53129|+19175558901|Dr. Jess

•

MSH|^~\&|CareCloud|Maplewood Clinic|Cerner PowerChart|Downtown Hospital|20240101000000||ADT^A01^ADT_A01|1|P778899001|||Elizabeth Martinez|1988-06-22|F|753 Willow St, Oldtown, AZ 85001|+11075559012|Dr. Ste

•

MSH|^~\&|Medhost|Lakeside Hospital|Meditech Expanse|Riverside Medical Center|20240101000000||ADT^A01^ADT_A01|1|P889900112|||Christopher Garcia|1983-09-09|M|369 Birchwood Ln, Lakeview, OR 97630|+12075550123|Dr.

Time Elapsed: 24.36 seconds

<

>

Виберіть формат:

hl7

Експортувати

Рисунок 4.7 - Генерація 10 повідомлень за допомогою Gemini API

3) Метод OpenAi API

Результати генерації зображені на рисунку 4.8. Час генерації склав 29.14 секунди. Результатом є дані найвищого рівня реалістичності, згенеровані за допомогою мовної моделі OpenAI API. Поля наповнені контекстуально

пов'язаною інформацією, включаючи імена пацієнтів, дати народження, адреси та детальну інформацію про медичні заклади. Генерація враховує логічний контекст, реалістичність зв'язків між полями та додаткові текстові описи.

Основними перевагами цього методу є глибока деталізація й адаптивність до специфічних сценаріїв. Однак триваліший час генерації робить цей метод менш підходящим для масових запитів, але ідеальним для сценаріїв, де потрібні висока якість та контекстуальна точність даних.

Генерація ADT повідомлень

Виберіть субтип:
ADT_A02 – Переведення пацієнта

Виберіть метод генерації:
AI ChatGPT

Кількість повідомлень:
10

Генерувати

- MSH|^~\&|MediFlow|Eastside Hospital|PatientPortal|Central Clinic|20240101000000||ADT^A01^ADT_A01|1|P|2.5
PID|1|PAT789012||Alice Johnson|1978-12-11|F|789 Pine Road, Denver, CO 80203|+13035556789|Dr. Sarah Lee|
- MSH|^~\&|HealthConnect|Lakeside Health|MedLink|North Point Clinic|20240101000000||ADT^A01^ADT_A01|1|P|2.
PID|1|PAT345678||Robert Brown|1963-07-19|M|234 Elm Street, Miami, FL 33101|+17862345678|Dr. James Wilson|
- MSH|^~\&|CareManager|Sunrise Healthcare|PatientCare|East End Clinic|20240101000000||ADT^A01^ADT_A01|1|P|
PID|1|PAT456789||Emily Davis|1989-11-30|F|567 Birch Lane, Seattle, WA 98104|+12065559876|Dr. Laura Green|
- MSH|^~\&|MedConnect|Riverbend Hospital|HealthSync|Westlake Clinic|20240101000000||ADT^A01^ADT_A01|1|P|2.
PID|1|PAT567890||Michael Thompson|1975-02-14|M|890 Cherry Street, Austin, TX 73301|+15125556789|Dr. Kevin White|
- MSH|^~\&|HealthTracker|Pine Valley Hospital|CareLink|Downtown Medical Clinic|20240101000000||ADT^A01^ADT
PID|1|PAT678901||Sophia Martinez|1982-03-22|F|321 Cedar Avenue, Los Angeles, CA 90001|+13105551234|Dr. Angela Brown|
- MSH|^~\&|PatientManager|Mountain View Medical Center|HealthNet|Southern Clinic|20240101000000||ADT^A01^A
PID|1|PAT789012||James Anderson|1990-08-05|M|654 Spruce Drive, Chicago, IL 60601|+13125552345|Dr. Angela Brown|
- MSH|^~\&|MedPortal|Central City Hospital|CareStream|Eastside Health Clinic|20240101000000||ADT^A01^ADT_A
PID|1|PAT890123||Olivia Wilson|1995-05-12|F|987 Maplewood Road, Phoenix, AZ 85001|+16025551234|Dr. Thomas Green|
- MSH|^~\&|HealthSuite|Seaside Medical Center|PatientFirst|Coastal Clinic|20240101000000||ADT^A01^ADT_A01|
PID|1|PAT901234||Liam Garcia|1987-09-18|M|111 Ocean Drive, San Diego, CA 92101|+16195559876|Dr. William Lee|

Time Elapsed: 29.14 seconds

Виберіть формат: hl7 **Експортувати**

Рисунок 4.8 - Генерація 10 повідомлень за допомогою OpenAi API

4.4 Перевірка валідності HL7 повідомлень

Для перевірки валідності згенерованого повідомлення використаємо незалежний ресурс HL7 Parser [17] та Inspector Neo [18] (рисунок 4.9-4.10).

Повідомлення, яке будемо перевіряти:

```
MSH|^~\&|HospitalSystem|MainHospital|ReceivingSystem|SecondaryHospital|202411271030||ADT^A02|MSG123456|P|2.5
EVN|A02|202411270930|202411270900||123456
PID|1|123456|789123^^^MainHospital^MR|789123^^^MainHospital^MR|Johnson^Emily^L||19840321|F||123
Street^^Denver^CO^80203||(303)123-4567|(303)987-6543||S|Caucasian|987654321|||U||
||20231101
PV1|1|I|2000^2050^01|R|1234^Smith^John^D^^Dr.||||Cardiology|||3||123456||567890|||||
|||||||202411260800|202411270900
PV2|||Regular Check-Up||||N|||||||Private Insurance|||||202411270900
```

HL7 ParserParse Message

Parsed HL7v2 Message

Original Message

```
MSH|^~\&|HospitalSystem|MainHospital|ReceivingSystem|SecondaryHospital|202411271030||ADT^A02|MSG123456|P|^
EVN|A02|202411270930|202411270900||123456
PID|1|123456|789123^^^MainHospital^MR|789123^^^MainHospital^MR|Johnson^Emily^L||19840321|F||123 Elm Street
PV1|1|I|2000^2050^01|R|1234^Smith^John^D^^Dr.||||Cardiology|||3||123456||567890|||||202411260800|
PV2|||Regular Check-Up||||N|||||||Private Insurance|||||202411270900
```

Reparse

Parsed Message

```

└─ADT_A02
  └─MSH
    ├──MSH.1 - Field Separator - |
    ├──MSH.2 - Encoding Characters - ^~\&
    ├──MSH.3 - Sending Application
    │   └─MSH.3.1 - Namespace ID - HospitalSystem
    ├──MSH.4 - Sending Facility
    │   └─MSH.4.1 - Namespace ID - MainHospital
    ├──MSH.5 - Receiving Application
    │   └─MSH.5.1 - Namespace ID - ReceivingSystem
    ├──MSH.6 - Receiving Facility
    │   └─MSH.6.1 - Namespace ID - SecondaryHospital
    ├──MSH.7 - Date/Time Of Message
    │   └─MSH.7.1 - Time of an event - 202411271030
    ├──MSH.9 - Message Type
    │   ├──MSH.9.1 - Message type - ADT
    │   └─MSH.9.2 - Trigger event - A02
    ├──MSH.10 - Message Control ID - MSG123456
    ├──MSH.11 - Processing ID
    │   └─MSH.11.1 - Processing ID - P
    └─MSH.12 - Version ID
        └─MSH.12.1 - Version ID - 2.5

└─EVN
└─PID
└─PV1
└─PV2
```

Рисунок 4.9 - Перевірка HL7 ADT-A02 повідомлення за допомогою HL7 Parser

Повідомлення пройшло перевірку, тепер для впевненості перевіримо повідомлення за допомогою Inspector Neo.

HL7 Inspector Neo Paste or drop messages Current profile: HL7 V2.5.1 ▾

Tree View Editor Compare

Name: ADT^A02 Message Length: 564 Characters

	Value
+	MSH ^~\& HospitalSystem MainHospital ReceivingSystem SecondaryHospital 202411271030 ADT^A02 MSG123456 P 2.5
✎	1:
✎	2: ^~\&
➤	3: HospitalSystem
➤	4: MainHospital
➤	5: ReceivingSystem
➤	6: SecondaryHospital
➤	7: 202411271030
➤	9: ADT^A02
➤	10: MSG123456
➤	11: P
➤	12: 2.5
➤	EVN A02 202411270930 202411270900 123456
➤	PID 1 123456 789123^^^MainHospital^MR 789123^^^MainHospital^MR Johnson^Emily^L 19840321 F 123 Elm Street^^Denver^CO^80203 ...
➤	PV1 1 2000^2050^01 R 1234^Smith^John^D^^Dr. Cardiology 3 123456 567890 202411260800 20...
➤	PV2 Regular Check-Up N Private Insurance 202411270900

Рисунок 4.10 – Перевірка HL7 ADT-A02 повідомлення за допомогою Inspector Neo

Як бачимо, обидва сервіси коректно розпізнали структуру HL7-повідомлення типу ADT_A02 (Переведення пацієнта). Повідомлення було розкладене на сегменти (MSH, EVN, PID, PV1, PV2), де кожен сегмент був відповідно інтерпретований і представлений у деревоподібній структурі.

Сегмент MSH (Message Header): усі ключові поля, такі як відправник (HospitalSystem), отримувач (ReceivingSystem), дата та час повідомлення (202411271030), тип повідомлення (ADT^A02), та версія стандарту (2.5), були

коректно розпізнані. Відсутність помилок у форматуванні підтверджує, що сегмент відповідає вимогам стандарту HL7.

- 1) Сегмент EVN (Event Type): тип події (A02) і час події (202411270930) були розпізнані без помилок. Поле з пов'язаним ідентифікатором (123456) також відповідає специфікаціям.
- 2) Сегмент PID (Patient Identification): дані пацієнта, зокрема ім'я (Johnson^Emily^L), дата народження (19840321), адреса (123 Elm Street^^Denver^CO^80203), та номери телефонів, були коректно зчитані та розподілені по полях. Структура і значення цього сегмента відповідають специфікаціям HL7.
- 3) Сегмент PV1 (Patient Visit): Місцезнаходження пацієнта (2000^2050^01) і лікар, який перевів пацієнта (1234^Smith^John^D^^Dr.), були коректно оброблені. Дати госпіталізації (202411260800) та переведення (202411270900) були правильно розпізнані.
- 4) Сегмент PV2 (Patient Visit Additional Information): поля, що містять інформацію про причину переведення (Regular Check-Up) та тип страхування (Private Insurance), були успішно ідентифіковані.

Висновок до розділу 4

Обчислювальний експеримент оцінює три методи генерації: Faker, Gemini API і OpenAI API. Faker забезпечує найшвидшу генерацію, але менш реалістичні дані. Gemini API генерує якісніші дані з логічними зв'язками, а OpenAI API забезпечує найвищу реалістичність, хоча й з більшим часом генерації. Вибір методу залежить від пріоритетів користувача: швидкість чи реалістичність.

Перевірка валідності за допомогою незалежних парсерів HL7 підтверджує відповідність повідомлень стандарту HL7. Всі сегменти, такі як MSH, EVN, PID, PV1 і PV2, сформовані коректно, що свідчить про здатність системи інтегруватися в медичні інформаційні системи.

5 РОЗРОБКА СТАРТАП-ПРОЄКТУ

Цей розділ присвячений детальному розгляду ключових аспектів стартап-проєкту, спрямованого на створення інноваційного програмного забезпечення для автоматизованої генерації HL7-повідомлень.

Деталізовані аспекти, включаючи опис ідеї проєкту, технологічний аудит, ринковий аналіз та розроблення маркетингової стратегії, висвітлюються в цьому розділі. Основна увага приділяється аналізу потреб клієнтів, конкурентному середовищу, визначенню сильних та слабких сторін продукту, а також розробці ринкових та комунікаційних стратегій, що забезпечать успішне впровадження продукту на ринок.

5.1 Опис ідеї стартап-проєкту

Головною метою даного стартап-проєкту є створення програмного забезпечення для автоматизованої генерації HL7-повідомлень, яке дозволяє швидко і ефективно тестувати медичні інформаційні системи. Основна ідея полягає в розробці інструменту, що забезпечує масштабованість, відповідність стандартам HL7 та реалістичність створених тестових даних.

Проєкт об'єднує сучасні технології автоматизації, генерації даних і веб-інтерфейсу, що сприяє покращенню процесів тестування медичних систем. У таблиці 5.1 наведено опис змісту ідеї, напрямки застосування та вигоди для користувача.

Таблиця 5.1 – Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
------------	-----------------------	------------------------

Кінець таблиці 5.1

Розробка програмного забезпечення для автоматичної генерації HL7-повідомлень з використанням сучасних технологій.	1. Тестування медичних інформаційних систем.	1. Скорочення часу на створення тестових даних.
	2. Стандартизація обміну медичною інформацією.	2. Висока точність та відповідність стандартам HL7.
	3. Масштабованість тестування різних сценаріїв.	3. Створення великих обсягів унікальних даних для тестування.

Аналіз потенційних техніко–економічних переваг ідеї порівняно із пропозиціями конкурентів наведено в таблиці 5.2

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик

No п/п		(потенційні) товари/концепції конкурентів		
		Мій проєкт	HL7.cc [19]	Caristix [20]
1	W сторона слабка	Обмежена функціональність порівняно з більш складними продуктами.	Базовий функціонал, який обмежує використання для складних сценаріїв тестування.	Висока вартість ліцензії, яка ускладнює доступ для малих компаній.
2		Потреба у налаштуванні для специфічних сценаріїв інтеграції.	Відсутність розширеної автоматизації процесів інтеграції HL7.	Складність освоєння для початківців через великий обсяг функцій та налаштувань.

Кінець таблиці 5.2

3	S сильна сторона	Простота використання та інтуїтивний інтерфейс	Безкоштовний доступ до основних інструментів (з обмеженнями)	Потужні інструменти автоматизації процесів генерації та тестування HL7-повідомлень.
4		Доступна ціна	Простота інтеграції з іншими платформами завдяки відкритим API.	Широкий спектр функцій для інтеграції та забезпечення якості даних.

5.2 Технологічний аудит ідеї стартап–проекту

Технологічний аудит — це ключовий етап оцінки можливостей реалізації стартапу з автоматичної генерації HL7-повідомлень. Цей аналіз дозволяє визначити наявні технології, оцінити їхню відповідність поставленим задачам і доступність для команди проекту. Технологічна здійсненність ідеї проекту розглянута в таблиці 5.3.

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

No п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Генерація HL7-повідомлень	hl7apy	Наявна	Доступна безкоштовно

Кінець таблиці 5.3

2	Генерація текстових полів	OpenAI API	Наявна	Доступна за підпискою
3	Генерація реалістичних структурованих даних	Faker	Наявна	Доступна безкоштовно
4	Масштабована генерація даних	Gemini API	Наявна	Доступна за підпискою
5	Зберігання і управління змінними середовища	Dotenv	Наявна	Доступна безкоштовно
6	Управління версіями коду	Git	Наявна	Доступна безкоштовно

Висновок: Проект реалізувати можливо.

Обрані технології реалізації ідеї проекту: бібліотека hl7apy для роботи з HL7-повідомленнями, інструменти Faker, OpenAI API та Gemini API для генерації реалістичних даних. Dotenv для управління змінними середовища, та Git для контролю версій.

5.3 Аналіз ринкових можливостей запуску стартап–проекту

Аналіз ринкових можливостей є важливим етапом для визначення конкурентних переваг стартапу. Він дозволяє оцінити перспективи продукту, виявити ключові ринкові чинники, які сприяють його впровадженню, і визначити ризики, що можуть перешкодити успішній реалізації.

Для стартапу, орієнтованого на автоматичну генерацію HL7-повідомлень, рентабельність досягається завдяки чіткому розумінню потреб клієнтів, унікальним функціональним можливостям продукту та конкурентній ціновій

політиці. Попередня характеристика потенційного ринку стартап-проєкту наведена в таблиці 5.4.

Таблиця 5.4. – Попередня характеристика потенційного ринку стартап–проєкту

№ п/п	Показники стану ринку	Характеристика
1	Кількість головних гравців	2 (HL7.cc, Caristix)
2	Загальний обсяг продаж	2 млн доларів на рік за оцінкою конкурентів
3	Динаміка ринку	Зростаюча, 10–15% щорічно
4	Наявність обмежень для входу	Висока конкуренція, потреба у відповідності стандартам
5	Специфічні вимоги до стандартизації	Відповідність стандарту HL7
6	Середня норма рентабельності	30%

Аналіз ринку підтверджує, що попит на автоматизовані рішення для генерації HL7-повідомлень залишається стабільно високим завдяки зростаючій інтеграції медичних систем. Ринок зростає приблизно на 10–15% щорічно, що створює нові можливості для впровадження інноваційних рішень.

За результатами аналізу таблиці ринок є привабливим для входження за попереднім оцінюванням.

Потенційні групи клієнтів, їх характеристики та орієнтовний перелік вимог до товару для кожної групи наведено в таблиці 5.5

Таблиця 5.5. Характеристика потенційних клієнтів стартап–проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Генерація HL7-повідомлень для тестування медичних систем	Великі лікарні, медичні стартапи, ІТ-компанії, що займаються розробкою медичних інформаційних систем	Великі лікарні укладають довготривалі договори для комплексного обслуговування, медичні стартапи потребують дешевих і швидких рішень, ІТ-компанії вимагають високої масштабованості для інтеграції	Стабільність роботи, відповідність стандартам HL7, можливість інтеграції, доступна ціна
2	Тестування продуктивності інтеграційних рішень	Провайдери електронних медичних записів (EMR/EHR), дослідницькі центри	Провайдери EHR/EMR прагнуть до довготривалого співробітництва з технічною підтримкою, дослідницькі центри зацікавлені в можливостях кастомізації	Надійність у роботі, підтримка кастомізації, технічна підтримка, документація

Кінець таблиці 5.5

3	Автоматизація створення великих масивів тестових даних	Розробники медичних додатків, освітні установи, що проводять навчання для медичних програмістів	Розробники додатків зацікавлені у гнучких інструментах із можливістю експорту, освітні установи надають перевагу простим і зручним інтерфейсам	Простий інтерфейс, масштабованість, навчальні матеріали, підтримка експорту даних
---	--	---	--	---

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають таблиці 5.6–5.7.

Таблиця 5.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Залежність від зовнішніх API	Генерація HL7-повідомлень частково залежить від роботи сторонніх API (наприклад, OpenAI API, Gemini API).	Впровадження механізму кешування результатів генерації та використання резервних API для критичних задач.
2	Вимоги до відповідності стандартам	Невідповідність повідомлень HL7-стандартам може призвести до втрати довіри клієнтів.	Регулярне оновлення бібліотек (наприклад, hl7apy) та проведення автоматизованого тестування валідності.

Кінець таблиці 5.6

3	Високі технічні вимоги	Робота з великими обсягами даних може перевищити обчислювальні можливості серверів.	Оптимізація алгоритмів і розгортання масштабованих хмарних рішень для обробки великих даних.
---	------------------------	---	--

Таблиця 5.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Зростання попиту на автоматизацію	Розробники медичних систем активно шукають інструменти для спрощення процесів тестування.	Активна промоція продукту через спеціалізовані платформи та участь у виставках медичних ІТ-рішень.
2	Швидкий розвиток технологій	Використання сучасних інструментів, таких як OpenAI API, Gemini API, розширює можливості продукту.	Постійне вдосконалення продукту шляхом інтеграції нових технологій, проведення аналізу трендів у сфері автоматизації.

Кінець таблиці 5.7

3	Розвиток хмарних технологій	Хмарні сервіси дозволяють масштабувати рішення та забезпечувати стабільну роботу з великими обсягами даних.	Використання хмарної інфраструктури для розгортання продукту та пропонування клієнтам SaaS-рішення.
---	-----------------------------	---	---

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку. Аналіз пропозиції необхідно виконати аналізуючи існуючі види конкуренцій.

Пропозиції повинні відповідати на питання “Як просувати продукт”. Аналіз пропозицій зображено у таблиці 5.8.

Таблиця 5.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції – монополія/олігополія/монополістична/чиста	Чиста конкуренція, обмежена кількома ключовими гравцями.	Прямі договори з медичними стартапами, активна участь у виставках для демонстрації продукту.
2. Рівень конкурентної боротьби – локальний/національний/...	Національний рівень із можливістю розширення на міжнародний.	Публікація наукових і технічних статей, участь у конференціях та форумах.

Кінець таблиці 5.8

3. Галузева ознака – міжгалузева/внутрішньогалузева	Внутрішньогалузева конкуренція між компаніями, що розробляють подібні рішення.	Розробка функцій, яких бракує конкурентам, активне вивчення потреб цільової аудиторії.
4. Конкуренція за видами товарів – товарно-родова/товарно-видова/між бажаннями	Товарно-видова конкуренція, спрямована на медичні інструменти для автоматизації обміну даними.	Демонстрація переваг продукту: автоматизація, масштабованість, відповідність стандартам HL7.
5. Характер конкурентних переваг – цінова/нецінова	Нецінова конкуренція завдяки функціональності, масштабованості, інноваційному підходу.	Інтеграція функцій, які відсутні у конкурентів, підвищення рівня автоматизації та зручності.
6. Інтенсивність – марочна/не марочна	Марочна конкуренція через наявність відомих брендів у сфері автоматизації HL7.	Просування власного бренду через сертифікацію продукту, співпрацю з відомими медичними установами.

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі (таблиця 5.9).

Таблиця 5.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари–замінники
	HL7.cc, Caristix – часткова автоматизація	Великі IT-компанії, які інтегрують HL7	hl7apy, OpenAPI, Faker – основні бібліотеки	Лікарні, IT-компанії, медичні стартапи	Ручне створення даних, інші тестові інструменти
Висновки:	Помірна конкуренція	Існує ризик появи нових конкурентів	Постачальники надійні та доступні	Попит та клієнти зростає, але менш вимогливі	Замінники ефективні

На основі аналізу конкуренції, проведеного в п. 3.5 таблиця 5.9, а також із урахуванням характеристик ідеї проекту таблиця 5.2, вимог споживачів до товару таблиця 5.5 та факторів маркетингового середовища таблиця 5.6–5.7 визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлюється за таблицею 5.10.

Таблиця 5.10. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
-------	-------------------------------	---

Кінець таблиці 5.10

1	Швидкість роботи	Система підтримує багатопоточність і обробляє великі обсяги даних, у той час як HL7.cc має обмежену продуктивність при великих навантаженнях.
2	Низька вартість обслуговування	На відміну від Caristix, наше рішення пропонує доступну вартість ліцензії та мінімальні вимоги до технічної підтримки.
3	Інтерфейс користувача	Наш інструмент має інтуїтивно зрозумілий інтерфейс, тоді як Caristix вимагає більше часу для освоєння та налаштування.

За визначеними факторами конкурентоспроможності таблиця 5.10 проводиться аналіз сильних та слабких сторін стартап–проекту таблиця 5.11.

Таблиця 5.11. Порівняльний аналіз сильних та слабких сторін

No п/п	Фактор конкурентоспроможності	Бали 1–20	Рейтинг товарів–конкурентів у порівнянні з програмою оцінки рівнів електромагнітного випромінювання (даним продуктом)						
			–3	–2	–1	0	1	2	3
1	Швидкість роботи	20			+				
2	Інтерфейс користувача	15			+				
3	Низька вартість обслуговування	10	+						

Кінець таблиці 5.11

4	Відповідність стандартам HL7	5				+			
5	Масштабованість системи	5				+			

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT–аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) таблиця 5.12 на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін таблиця 5.11.

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

Таблиця 5.12. SWOT–аналіз стартап–проекту

Сильні сторони: Застосування сучасних засобів програмування. Висока продуктивність із підтримкою багатопоточності.	Слабкі сторони: Відсутність готових інтеграцій із популярними медичними платформами та сервісами.
---	---

Кінець таблиці 5.12

Можливості: Зростання попиту на автоматизацію генерації тестових даних у медичних системах. Розширення функціональності та супроводу продукту через оновлення та інтеграцію новітніх технологій.	Загрози: Конкуренція з боку великих компаній, які можуть запропонувати подібні рішення з кращим фінансуванням.
---	--

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

Таблиця 5.13. Альтернативи ринкового впровадження стартап–проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Орієнтація програмного продукту на інтеграцію з медичними інформаційними системами	90 %	1.5 роки

Кінець таблиці 5.13

2	Розробка спрощеної версії продукту для малих медичних стартапів	70 %	1 рік
3	Зосередження на створенні хмарного сервісу для автоматизації генерації HL7-повідомлень	80 %	2 роки

Альтернатива з найвищою ймовірністю отримання ресурсів:

- №1, що передбачає інтеграцію з медичними інформаційними системами, оскільки така стратегія відповідає зростаючому попиту на автоматизацію в цій галузі.

Альтернатива з найкоротшим строком реалізації:

- №2, орієнтована на створення спрощеної версії продукту для малих стартапів, що дозволяє швидко вийти на ринок і залучити початкову аудиторію.

5.4 Розроблення ринкової стратегії стартап–проєкту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (таблиця 5.14).

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку.

Таблиця 5.14. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Великі лікарні	Висока	Високий	Середня	Середня
2	Медичні стартапи	Висока	Середній	Низька	Висока
Які цільові групи обрано: Великі лікарні					

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (таблиця 5.15).

Таблиця 5.15. Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентос–пром ожні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Орієнтація на ринок великих лікарень	Стратегія концентрованого маркетингу	Простоту використання, масштабованість продукту та можливість швидкого налаштування під потреби клієнтів.	Стратегія спеціалізації з орієнтацією на потреби великих лікарень

Кінець таблиці 5.15

2	Розробка гнучких інструментів для медичних стартапів	Стратегія нішевого маркетингу	Низька ціна, простота у використанні, доступ до навчальних матеріалів	Стратегія зниження витрат для залучення стартапів
---	--	-------------------------------	---	---

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту.

Наступним кроком є вибір стратегії конкурентної поведінки (таблиця 5.16).

Таблиця 5.16. Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Ні	Шукати нових споживачів	Ні	Стратегія заняття конкурентної ніші

З обраних сегментів до постачальника (стартап–компанії) та до продукту розробляється стратегія позиціонування (таблиця 5.17). що полягає у формуванні ринкової позиції, за яким споживачі мають ідентифікувати торговельну марку чи проект.

Таблиця 5.17. Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто–спром ожні позиції власного стартап–проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Стабільність роботи Невисока ціна Наявність пробного періоду Наявність документації Висока продуктивність	Стратегія концентрованог о маркетингу	Висока продуктивність, Інтуїтивний інтерфейс, низька вартість обслуговування	Продуктивність, надійність, масштабованість

5.5 Розроблення маркетингової програми стартап–проекту

Для цього у таблиці 5.18 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 5.18. Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Простота використання	Підходить для користувачів різного рівня підготовки	Інші рішення вимагають специфічних знань і додаткового навчання.
2	Швидка генерація даних	Миттєве створення великих обсягів даних для тестування	Продуктивність перевищує аналоги завдяки підтримці багатопоточності.
3	Масштабованість	Здатність адаптуватися до зростаючих обсягів даних та нових вимог	Відсутність обмежень на обсяги створюваних даних, що є у конкурентів.
4	Низькі вимоги до обслуговування	Мінімальні витрати на технічну підтримку та обслуговування продукту	Інші продукти вимагають регулярного втручання технічних спеціалістів, що підвищує витрати.

Надалі розробляється трирівнева маркетингова модель товару: уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (таблиця 5.19).

Таблиця 5.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Програмне забезпечення для автоматизованої генерації HL7-повідомлень, яке забезпечує високу продуктивність, відповідність стандартам HL7 та простоту інтеграції в існуючі медичні інформаційні системи.		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	Можливість оптимізації витрат часу	М	Тл
	можливість оптимізації витрат коштів	М	Вр
	відповідність актуальним технологіям	Нм	Тх
	Відповідає міжнародним стандартам HL7		
	Пакування: веб сервіс		
	Марка: HL7 AutoGen		
III. Товар із підкріпленням	Потенційні користувачі можуть ознайомитись з продуктом через наукові публікації у фахових виданнях та конференціях.		

Кінець таблиці 5.19

За рахунок чого потенційний товар буде захищено від копіювання: Назва та контент захищені ліцензією МІТ; захист інтелектуальної власності через реєстрацію авторських прав.

М/Нм – монотонні або немонотонні;

Вр/Тх/Тл/Е/Ор – вартісні, технічні, технологічні, ергономічні або органолептичні (останній – для продуктів харчування)

Після формування маркетингової моделі товару слід особливо відмітити – чим саме проект буде захищено від копіювання. Захист може бути організовано за рахунок захисту ідеї товару (захист інтелектуальної власності), або ноу–хау, чи комплексне поєднання властивостей і характеристик, закладене на другому та третьому рівнях товару.

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (таблиця 5.20).

Таблиця 5.20. Визначення меж встановлення ціни

№ п/п	Рівень цін на товари–замінники	Рівень цін на товари–аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	0 грн (ручна генерація, внутрішні ресурси)	250 000 – 500 000 грн (Caristix, HL7.cc)	500 000 – 1 200 000 грн/рік	300 000 – 600 000 грн

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (таблиця 5.21):

- проводити збут власними силами або залучати сторонніх посередників (власна або залучена система збуту);
- вибір та обґрунтування оптимальної глибини каналу збуту;
- вибір та обґрунтування виду посередників.

Таблиця 5.21. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Клієнти потребують демонстрації функціоналу через тріал-версію. Оплата здійснюється після тестування.	Надання документації, підтримка інтеграції, забезпечення легкого доступу до продукту через веб-інтерфейс	4: Розробник даного продукту – Веб-сайт – Користувач.	Проводити своїми силами через онлайн-ресурси

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 5.22).

Таблиця 5.22. Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідом - лення	Концепція рекламного звернення
----------	--	---	---	--	--------------------------------------

Кінець таблиці 5.22

1	Купівля програмного комплексу через офіційний сайт або після консультації з представником .	Веб-сайти, електронна пошта, соціальні мережі	Швидкість, доступність, відповідність стандартам HL7	Показати ефективність продукту у скороченні часу генерації тестових даних HL7	Продукт забезпечує масштабованість і відповідність HL7, спрощуючи процес інтеграції.
---	---	---	--	---	--

Висновок до розділу 5

У цьому розділі висвітлено основні аспекти запуску стартапу, орієнтованого на автоматизовану генерацію HL7-повідомлень. Проєкт демонструє високий потенціал завдяки використанню сучасних технологій, таких як OpenAI API, Gemini API та HL7apy, які забезпечують відповідність стандартам HL7, масштабованість і простоту інтеграції.

Аналіз ринку підтвердив перспективність продукту серед великих лікарень, медичних стартапів і IT-компаній. Розроблена стратегія охоплення ринку акцентує увагу на перевагах продукту: висока продуктивність, зручний інтерфейс і доступна ціна.

Проведений SWOT-аналіз дозволяє виокремити можливості для розширення ринку за рахунок нових функцій і інтеграцій, а також визначити ризики, пов'язані із залежністю від зовнішніх API та конкуренцією в галузі. Запропоновані стратегії розвитку передбачають створення ефективних рішень для різних сегментів ринку.

Таким чином, стартап має всі передумови для успішного впровадження та задоволення актуальних потреб у сфері автоматизації медичних даних.

ВИСНОВКИ

У магістерській дисертації розглянуто проблему автоматизації генерації HL7-повідомлень для забезпечення оптимізації процесів тестування медичних інформаційних систем.

Вивчено структуру, типи та специфіку HL7-повідомлень, а також сучасні методи їх обробки, модифікації та валідації. Аналіз існуючих підходів підкреслив актуальність розробки рішень, які забезпечують відповідність стандарту HL7, гнучкість і масштабованість системи.

Для реалізації розробленої системи були використані сучасні технології, які забезпечують надійну обробку та генерацію даних відповідно до стандарту HL7. Використання бібліотек для роботи з даними та інтеграції API дозволило створити функціональну та гнучку платформу, здатну адаптуватися до потреб медичних інформаційних систем. Обрані технології сприяють ефективному виконанню завдань, підтримуючи стандартизацію та масштабованість.

Запропоновано інтегроване рішення, яке базується на клієнт-серверній архітектурі. Розроблена система дозволяє автоматично генерувати HL7-повідомлення. Структура системи включає клієнтську частину, яка надає інтуїтивний інтерфейс для налаштування параметрів генерації, і серверну частину, що відповідає за обробку запитів, перевірку даних та інтеграцію з зовнішніми сервісами.

Результати дослідження свідчать, що розроблена система може бути використана для тестування та оптимізації медичних інформаційних систем, відповідаючи високим стандартам продуктивності, стабільності та відповідності HL7.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Швагрун А. В Михайлова І.Ю Засоби автоматичної генерації тестових повідомлень HL7 : матер. VII всеукр. наук.-практ. інтернет-конф. молодих вчених та студентів (м. Київ, 29.11.2024) Київ-Херсон, 2024.
2. HL7 test message generator | InterSystems Ideas. InterSystems Ideas. URL: <https://ideas.intersystems.com/ideas/DPI-I-464> (date of access: 26.10.2024).
3. JavaScript | MDN. MDN Web Docs. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (date of access: 26.10.2024).
4. Welcome to Python.org. Python.org. URL: <https://www.python.org/> (date of access: 26.10.2024).
5. Welcome to Flask – Flask Documentation (3.1.x). Welcome to Flask – Flask Documentation (3.1.x). URL: <https://flask.palletsprojects.com/en/stable/> (date of access: 26.10.2024).
6. HL7apy – HL7apy - a lightweight Python library to parse, create and handle HL7 v2.x messages. Site not found · GitHub Pages. URL: <https://crs4.github.io/hl7apy/> (date of access: 26.10.2024).
7. OpenAI API. URL: <https://openai.com/index/openai-api/> (date of access: 26.10.2024).
8. Gemini Developer API | Gemma open models | Google AI for Developers. Google AI for Developers. URL: <https://ai.google.dev/> (date of access: 26.10.2024).
9. Welcome to Faker's documentation! – Faker 33.1.0 documentation. Welcome to Faker's documentation! – Faker 33.1.0 documentation. URL: <https://faker.readthedocs.io/en/master/> (date of access: 26.10.2024).
10. GitHub · Build and ship software on a single, collaborative platform. GitHub. URL: <https://github.com/> (date of access: 26.10.2024).
11. dotenv. npm. URL: <https://www.npmjs.com/package/dotenv> (date of access: 26.10.2024).

12. Bray D. A. Artificial Intelligence and HL7 International: Proposed Immediate and Longer-Term Actions. SSRN Electronic Journal. 2024. 25 p. URL: <https://doi.org/10.2139/ssrn.4745490> (date of access: 26.10.2024).
13. HL7/FHIR = Fast Healthcare Interoperability Resources. Cegraf UFG, 2022. 58 p. URL: <https://doi.org/10.5216/hl7.ebook.978-85-495-0497-5/2022> (date of access: 26.10.2024).
14. JSON. JSON. URL: <https://www.json.org/json-en.html> (date of access: 26.10.2024).
15. XML introduction - XML: Extensible Markup Language | MDN. MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Web/XML/XML_introduction (date of access: 26.10.2024).
16. jQuery API Documentation. jQuery API Documentation. URL: <https://api.jquery.com/> (date of access: 26.10.2024).
17. Online HL7 Message Parser - Online HL7 Message Parser. Online HL7 Message Parser - Online HL7 Message Parser. URL: <https://hl7messageparser.azurewebsites.net/> (date of access: 26.10.2024).
18. HL7 Inspector Neo - Free web based HL7 tool for analyzing and editing of HL7 messages. HL7 Inspector Neo - Free web based HL7 tool for analyzing and editing of HL7 messages. URL: <https://www.hl7inspector.com/> (date of access: 26.10.2024).
19. HL7.cc Tools. HL7.cc Tools. URL: <https://hl7.cc/> (date of access: 26.10.2024).
20. Caristix HL7-Definition V2. URL: <https://hl7-definition.caristix.com/v2/> (date of access: 26.10.2024).

ДОДАТОК А

ЗАСОБИ АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ ТЕСТОВИХ ПОВІДОМЛЕНЬ NL7

VII Всеукраїнська науково-практична інтернет-конференція молодих вчених та студентів «Сучасні інформаційні системи та технології»