

Міністерство освіти і науки України
Національний університет «Києва-Могилянська академія»

Факультет інформатики

Кафедра інформатики

Магістерська робота

на тему: **«УНІВЕРСАЛІЗАЦІЯ МОДУЛЯ ПІДКЛЮЧЕННЯ (ХАБУ)
ЗОВНІШНІХ СИСТЕМ ДО МЕСЕНЖЕРІВ»**

Виконав: студент 2-го року навчання

Спеціальності

122 Комп'ютерні науки

Базалицький Віталій Михайлович

Керівник Афонін А.О.

доцент, кандидат фізико-математичних наук

Магістерська робота захищена

з оцінкою _____

Секретар ЕК _____

«_____» _____ 2021 р.

Київ 2021

Національний університет «Києво-Могилянська академія»

Факультет інформатики

Кафедра інформатики

Освітній ступень магістра

Спеціальність 122 Комп'ютерні науки

Освітньо-наукова програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри мультимедійних систем,

_____ В. В. Бублик

“ _ ” _____ 2021 р.

ЗАВДАННЯ

ДЛЯ МАГІСТЕРСЬКОЇ РОБОТИ СТУДЕНТУ

Базалицькому Віталію Михайловичу

1. Тема роботи: Універсалізація модуля підключення (хабу) зовнішніх систем до месенджерів

керівник роботи Афонін Андрій Олександрович, кандидат фізико-математичних наук,
доцент

затверджені наказом вищого навчального закладу від

“ _ ” _____ 2021 року № _____

2. Строк подання студентом роботи 4 червня 2021 року

3. План роботи

Індивідуальне завдання

Анотація

Вступ

1. Існуюча реалізація інтеграції зовнішніх систем з WeChat, готовий функціонал

2. Мета універсалізації модуля підключення зовнішніх систем до месенджерів

3. Готові універсальні рішення для інтеграцій систем з месенджерами

4. Інтеграція зовнішніх систем: елементи інтеграції

5. Огляд API популярних месенджерів для спільної обробки обраних елементів інтеграції

6. Реалізація: стек технологій

7. Опис готової системи

Висновок

Список використаних джерел

ГРАФІК ПІДГОТОВКИ МАГІСТЕРСЬКОЇ РОБОТИ ДО ЗАХИСТУ

№	ПЕРЕЛІК РОБІТ	Термін виконання	Примітка
1.	Отримання завдання на курсову роботу.	листопад	
2.	Існуюча реалізація інтеграції зовнішніх систем з WeChat, готовий функціонал	грудень	
3.	Мета універсалізації модуля підключення зовнішніх систем до месенджерів	грудень	
4.	Готові універсальні рішення для інтеграцій систем з месенджерами	січень	
5.	Інтеграція зовнішніх систем: елементи інтеграції	лютий	
6.	Огляд API популярних месенджерів для спільної обробки обраних елементів інтеграції	лютий	
7.	Реалізація: стек технологій. Створення застосунку	березень	
8.	Тестування та випробування системи	березень	
9.	Опис готової системи.	квітень	
10.	Надання роботи керівнику для перевірки.	квітень	
11.	Коригування роботи за результатами перевірки керівником.	травень	
12.	Перед-захист магістерської роботи	13 травня 2021 р.	
12.	Захист магістерської роботи	14 червня 2021 р.	

Студент Базалицький В.М.

Керівник Афонін А.О.

“ _____ ” _____ 2021 р.

ЗМІСТ

АНОТАЦІЯ	6
ВСТУП	7
РОЗДІЛ 1: ІСНУЮЧА РЕАЛІЗАЦІЯ ІНТЕГРАЦІЇ ЗОВНІШНІХ СИСТЕМ З WECHAT. ГОТОВИЙ ФУНКЦІОНАЛ.	9
РОЗДІЛ 2: МЕТА УНІВЕРСАЛІЗАЦІЇ МОДУЛЯ ПІДКЛЮЧЕННЯ ЗОВНІШНІХ СИСТЕМ ДО МЕСЕНДЖЕРІВ.....	11
РОЗДІЛ 3: ГОТОВІ УНІВЕРСАЛЬНІ РІШЕННЯ ДЛЯ ІНТЕГРАЦІЙ СИСТЕМ З МЕСЕНДЖЕРАМИ.....	13
2.1. Respond.io	13
2.2. Sunshine Conversations	15
2.3. Порівняльна характеристика	18
РОЗДІЛ 4: ІНТЕГРАЦІЯ ЗОВНІШНІХ СИСТЕМ: ЕЛЕМЕНТИ ІНТЕГРАЦІЇ	19
3.1. Месенджер WeChat	19
3.2. Месенджер Telegram	20
РОЗДІЛ 5: ОГЛЯД АРІ ПОПУЛЯРНИХ МЕСЕНДЖЕРІВ ДЛЯ СПІЛЬНОЇ ОБРОБКИ ОБРАНИХ ЕЛЕМЕНТІВ ІНТЕГРАЦІЇ	23
5.1. WeChat	23
5.1.1. Інтеграція каналу WeChat	23
5.1.2. Отримання токена для спілкування з серверами WeChat.....	24
5.1.3. Створення меню для швидкою взаємодії користувача з каналом	26
5.1.4. Обмін повідомленнями	29
5.2. Telegram	31
5.2.1. Інтеграція з Telegram.....	33
5.2.2. Створення меню для швидкою взаємодії користувача з каналом	33
РОЗДІЛ 6: РЕАЛІЗАЦІЯ: СТЕК ТЕХНОЛОГІЙ.....	36
РОЗДІЛ 7: РЕАЛІЗАЦІЯ: ОПИС ГОТОВОЇ СИСТЕМИ.....	39
7.1 Інтеграція WeChat каналу в систему	39

7.2 Створення меню в універсальному модулі підключення зовнішніх систем до месенджерів	40
7.3 Демонстрація функціоналу меню в універсальному модулі підключення зовнішніх систем до месенджерів в WeChat	41
7.4 Інтеграція Telegram каналу в систему	45
7.5 Демонстрація функціоналу меню в універсальному модулі підключення зовнішніх систем до месенджерів в Telegram	46
7.6 Демонстрація функціоналу статистики в універсальному модулі підключення зовнішніх систем до месенджерів в WeChat	48
7.7 Демонстрація функціоналу статистики в універсальному модулі підключення зовнішніх систем до месенджерів в Telegram	49
ВИСНОВКИ	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	52

АНОТАЦІЯ

Базалицький В.М. Універсалізація модуля підключення (хабу) зовнішніх систем до месенджерів

У магістерській роботі було розглянуто доцільність та можливість універсалізації модулю підключення зовнішніх систем до месенджерів, зокрема месенджером Telegram та WeChat. Було проаналізовано спільні інтеграційні можливості які надаються API Telegram та WeChat, та реалізовано універсальну систему для інтеграції зовнішніх систем з кількома месенджерами.

У першому розділі магістерської роботи було розглянуто існуючу інтеграцію зовнішніх систем з WeChat, яка була імплементована в рамках курсової роботи, та описано реалізований функціонал.

У другому розділі магістерської роботи була розглянута мета універсалізації модуля підключення зовнішніх систем до месенджерів, висвітлено основну вигоду від розробки такої системи.

У третьому розділі магістерської роботи були розглянуті готові подібні системи, які надають можливість інтеграції зовнішніх систем з кількома месенджерами, коротко описана основна функціональність та проведено порівняння з реалізованою системою.

У четвертому розділі магістерської роботи було розглянуто та описано спільні інтеграційні елементи, які були реалізовані в системі.

У п'ятому розділі магістерської роботи було розглянуто можливості API Telegram та WeChat для імплементации потрібних інтеграційних елементів.

У шостому розділі було розглянуто стек технологій, який був обраний для реалізації даної системи, та описано основні причини обрання кожної з технологій.

У сьомому розділі магістерської роботи було розглянуто готову систему, її можливості, функціонал.

ВСТУП

Актуальність теми. У сучасному світі більшість населення нашої планети зареєстрована хоча б в одному месенджері. Характерно, що для кожної країни найпопулярнішим месенджер свій. Наприклад, WeChat найпопулярніший в Китаї, практично єдиний доступний там месенджер. В Україні найпопулярнішим месенджером є Viber. У США - це Facebook Messenger. Шаленої популярності набуває месенджер Telegram, який є популярним на міжнародній арені. Отже, для того щоб охопити більшість клієнтів на сучасному ринку, бізнесу потрібно впроваджувати інтеграції в месенджери як одному з найзручніших місця для комунікації з своїми клієнтами.

Приблизно 45% населення світу користуються соціальними мережами, в середньому проводячи там 2 години та 23 хвилини на день. Цей показник постійно збільшується, так як в 2012 році час який витрачали люди на соціальні мережі був 1 година та 30 хвилин. [9]

Отже, дійсно сучасному бізнесу потрібні сучасні рішення. Імплементація інтеграції з однією системою, універсальним модулем підключення зовнішніх систем до месенджерів, набагато простіше, ніж розробка інтеграції з десятками месенджерів, які з'являються та стають популярними за дуже короткий період.

Метою магістерської роботи є розробка універсальної система, яка б надавала можливість впровадити якомога простіше інтеграцію з кількома месенджерами. Дана система надає можливість керувати усіма інтеграційними елементами та подіями з одного місця, що є дуже зручно та економічно. Також клієнтам з точки зору UX набагато краще у кожному з месенджерів, на який би вони не переходили користуватись каналами зв'язку з бізнесом у схожий спосіб. Отже, метою є спростити спосіб інтеграції зовнішніх систем з месенджерами, та надавати зручний функціонал для управління контентом та взаємодією з користувачами.

Структура магістерської роботи. Робота складається із анотації, вступу, семи розділів, висновків, списку використаних джерел (у кількості 8 найменувань).

РОЗДІЛ 1: ІСНУЮЧА РЕАЛІЗАЦІЯ ІНТЕГРАЦІЇ ЗОВНІШНІХ СИСТЕМ З WECHAT. ГОТОВИЙ ФУНКЦІОНАЛ.

В рамках курсової роботи першого курсу магістратури, було розроблено систему інтеграції зовнішніх систем з месенджером WeChat. Система надавала можливість швидко та просто додавати меню різних типів за допомогою яких можна взаємодіяти з користувачами месенджер каналу. Показ стандартних повідомлень під час першого входу користувача на месенджер канал або повідомлення при помилково введеному повідомленні від користувача легко налаштовувалось в реалізованій системі. Також існувала можливість збирати статистику дій користувачів у каналі, таких як натиснення меню. Статистика взаємодій користувача з меню зберігалась навіть з архівними меню (якщо меню в системі було замінено іншим меню).

При виконанні даної роботи було проаналізовано API WeChat, в результаті чого можна було стверджувати про величезний функціонал, який можна було реалізувати в подальшому. Прикладом такого функціоналу може бути: розширена статистика про користувачів каналу, така як вік, стать, місто проживання і інше. Але в той же час потрібно врахувати, що доцільно реалізовувати тільки той функціонал, який є спільним для більшості популярних месенджерів на сьогоднішній день, тому потрібно провести аналіз функціоналу, який варто додавати в новостворену універсальну систему (хаб).

Однією з найважливіших функцій, яка не була реалізована в курсовій роботі, є спілкування між користувачами та зовнішніми системами за допомогою інтеграційної системи. За допомогою цього функціоналу, користувачеві не потрібно виходити з улюбленого месенджеру задля того, щоб дізнатись статус свого замовлення, дізнатись вартість нового замовлення чи отримати оновлення від зовнішньої системи. Все це в одному місці, безпечно та зручно.

Реалізована система розроблялась використовуючи SOLID[1], принципи та архітектурою при якій можливе розширення функціоналу в майбутньому, що означає що універсализація готової системи може пройти просто, та без суттєвих витрат часу на рефакторинг системи.

Отже, за основу універсального модулю підключення зовнішніх систем до месенджерів було взято готову систему, у якій реалізована інтеграція з WeChat.

Таким чином, реалізація нової системи яка буде дозволяти інтегрувати декілька месенджерів з зовнішніми системами в одному місці відбудеться більш ефективно та, в результаті, більш якісно, що в свою чергу дозволяє зосередитись на доданні більшої кількості функціоналу у систему, аніж переробленням системи для можливості внесення змін та розширення функціоналу.

РОЗДІЛ 2: МЕТА УНІВЕРСАЛІЗАЦІЇ МОДУЛЯ ПІДКЛЮЧЕННЯ ЗОВНІШНІХ СИСТЕМ ДО МЕСЕНДЖЕРІВ

Основною метою універсалізації модуля підключення зовнішніх систем до месенджерів є створення системи, яка могла б інтегруватись до будь якого месенджера з невисокими затратами на розробку нових інтеграцій з новими месенджерами. Майже усі месенджери мають дуже схожий функціонал. Це означає, що за допомогою універсальної системи інтеграцій, можливо розробити таке програмне забезпечення, яке б надавало можливість використовувати зовнішній системі кілька месенджерів, зробивши інтеграцію тільки з одною універсальною системою, яка може виступити як адаптер до кожного з месенджерів, які вона підтримує.

Великим бізнес системам недостатньо обмежити співпрацю зі своїми клієнтами до спілкування за допомогою email, внутрішніх сповіщень системи чи SMS повідомленнями на телефон. У сучасному світі, кожна людина зареєстрована в кількох соціальних мережах та месенджерах, все більшу популярність набирають такі месенджери як Telegram, WhatsApp та інші. Люди проводять по кілька годин на день у кожному із месенджерів. Тому це дуже важливо, надати можливість своїм клієнтам тримати під контролем свої замовлення, дізнаватись вартість послуг, отримувати останні новинки саме у месенджері, місці в якому люди і так проводять багато часу. Єдине місце в якому користувач керує усіма процесами та співпрацює із системою.

Але більшість сучасних людей користуються різними месенджерами, на сьогоднішній день їх існує кілька десятків, і в кожного з месенджерів величезна аудиторія, а витратити час на інтеграцію з кожним, та підтримувати контент та список функціоналу, який може надаватись у кожному з месенджерів є доволі трудомісткою справою для великої бізнес системи.

Рішення є – універсальний модуль підключення зовнішніх систем, який буде надавати можливість керувати усіма месенджерами з одного місця, при чому час на інтеграцію є мінімальний.

Отже, мета універсалізації, полягає в тому, щоб створити такий застосунок, з яким буде існувати можливість інтеграції, яка забезпечує доступ до кількох месенджерів, з якими вже проведена інтеграція. Ця інтеграція буде дозволяти створення меню, яке буде бачити користувач, для зручної співпраці з каналом в месенджері. Створивши меню тільки один раз в універсальному модулі підключення зовнішніх систем до месенджерів, воно буде з'являтися у кожному месенджері. Також, в системі буде показано скільки користувачів і за який проміжок часу натискали на кожне з меню у кожному з месенджерів. Ця статистика є надзвичайно важливою для бізнесу. І однією з найочікуваніших функцій, яка не була реалізований у минулій версії системи інтеграції зовнішніх систем з месенджерами, є спілкування між клієнтами месенджеру та зовнішніми системами (використання системи як хабу між месенджером та зовнішньою системою).

Для досягнення мети буде реалізований такий функціонал: створити інтеграцію з новим месенджером, та впевнитись що додавання нових месенджерів не буде потребувати великих затрат людино-годин. Для цього потрібно буде переглянути архітектуру та проаналізувати час який буде затрачено для додавання нового месенджеру, тому спочатку в систему будуть внесені певні архітектурні зміни, а потім додана нова інтеграція. Розділення статистики для кожного месенджеру.

РОЗДІЛ 3: ГОТОВІ УНІВЕРСАЛЬНІ РІШЕННЯ ДЛЯ ІНТЕГРАЦІЙ СИСТЕМ З МЕСЕНДЖЕРАМИ

У цьому розділі розглядається рішення для інтеграції зовнішніх систем з месенджерами для кращого розуміння систем з подібним функціоналом. Це допоможе проаналізувати корисний функціонал який потрібно додати до системи, яка буде реалізованою, або ж навпаки позбутись, так як це не приносить ніякої цінності.

2.1. Respond.io

Одним з найпопулярніших застосунків які надають послуги для інтеграції популярних месенджерів в одне місце є Respond.io. Respond.io - це платформа обміну повідомленнями для бізнесу, яка зв'язує контакти з будь-якого каналу до кожної команди[2].

Respond.io позиціонує свою місію як покращити відносини між бізнесом та їхніми клієнтами. Аргументують вони це тим, що на сьогоднішній день існує понад 4 мільярди облікових записів для обміну повідомленнями у всьому світі, люди проводять у них більше часу, ніж у будь-яких інших застосунках в Інтернеті. У компанії переконані, що збираючи всі повідомлення в одному місці для бізнесу, створюється можливість зробити взаємодію з брендами простішою, швидшою та, зрештою, більш продуктивною.[2]

Оскільки кількість споживачів які вимагають особистого спілкування з найрізноманітніших каналів зростають, бізнес шукає програмного забезпечення, які б надали можливість надавати ці послуги якомога ефективніше.[2]

Отже, розглянемо головну функціональність яку висвітлює Respond.io. Дана система дозволяє клієнтам надсилати повідомлення через будь-який канал (месенджери, поштові скриньки і тд.) і забезпечує повну картину спілкування з клієнтами для кожної команди компанії.

Respond.io хоче допомогти краще керувати контактами та розмовами з різних каналів та полегшити вашій команді надавати ефективнішу відповідь вашим контактам. Платформа також забезпечує спосіб співпраці між співробітниками вашої компанії.[2]

Основний функціонал Respond.io:

1. Спільна папка "Вхідні"

Маючи спільну поштову скриньку, ваша команда можете співпрацювати з колегами та швидко відповідати на питання клієнтів. Таким чином, колеги по команді можуть швидше співпрацювати в розмовах та надавати вражаючі послуги клієнтам.

2. 360° Customer View

Більше не потрібно входити на кілька платформ - знайдіть усі чати клієнтів в одному місці. Незалежно від каналу спілкування, існує можливість знайти одну історію повідомлень для кожного клієнта з одного джерела. За допомогою користувацьких полів і тегів буде простіше досягти своїх потреб у сегментації клієнтів.

3. Автоматизація чату

Менше роботи, більше повідомлень для більш продуктивної робочої сили. Існує можливість створити робочі процеси, щоб допомогти команді краще керувати своїми зусиллями щодо вирішення проблем із клієнтами.

4. Відповідь на ходу

За допомогою мобільного додатка response.io існує можливість відповідати клієнтам на мобільному пристрої у зручний спосіб. Мобільний додаток має необхідне для виконання завдань, щоб існувала можливість просто працювати, навіть в дорозі.

Отже, функціонал, який надає Respond.io, не є подібним до того який буде розроблений в системі інтеграції систем управління бізнес-процесами з месенджерами, але концепції є подібними. В Respond.io можливо інтегрувати практично всі популярні месенджери та працювати з повідомленнями в одному місці. Але у зовнішніх систем немає можливості доступу до цих повідомлень, керування меню для швидкого доступу та формування статистики. Також відсутня можливість через API відповідати користувачам фіксованими повідомленнями. Система націлена на задоволення абсолютно інших цілей, але концепція схожа.

2.2. Sunshine Conversations

Sunshine Conversations - це програмна платформа, яка дозволяє компаніям спілкуватися зі своїми клієнтами через усі популярні програми обміну повідомленнями.[3]

Розробники можуть використовувати хмару Sunshine Conversations Cloud для додавання можливостей обміну повідомленнями та розмов до свого програмного забезпечення. Розширені API-інтерфейси Sunshine Conversations дозволяють керувати розмовами, обмінюватися повідомленнями, збирати метадані користувачів, керувати обліковим записом тощо.[3]

Підприємства можуть використовувати хмару Sunshine Conversations Cloud для підключення своїх улюблених бізнес-систем, ботів та інших інструментів до своїх клієнтів за допомогою обміну повідомленнями. Використовуючи комплексний набір інтеграцій Sunshine Conversations, компанія може підключити свою довідкову службу до різноманітних каналів обміну повідомленнями, автоматизувати розмови за допомогою ботів та фіксувати потенційних клієнтів за допомогою CRM - без написання рядка коду.[3]

За допомогою Sunshine Conversations існує можливість використання REST API, щоб додати функції спілкування до вашого продукту та покращити його набір функцій, а саме:

- Програмне забезпечення служби підтримки може вийти за рамки традиційного управління електронною поштою та миттєво додати підтримку для всіх популярних каналів обміну повідомленнями
- Програмне забезпечення для чату в режимі реального часу може запропонувати бізнесу можливість спілкуватися зі своїми клієнтами, використовуючи бажаний канал.
- Програмне забезпечення для управління соціальними медіа може забезпечити можливості обміну повідомленнями one to one, які допомагають командам соціальних мереж піклуватися про своїх клієнтів.
- Бот-платформи можуть зосередитись на побудові розумної розмовної логіки та штучного інтелекту, не турбуючись про управління повідомленнями, розмову та функціональність ідентифікації користувача[3].

Також існує можливість використання вбудованих інтеграції Sunshine Conversations для спілкування з клієнтами з ваших існуючих інструментів:

- Залучайте клієнтів до розмов за різними каналами обміну повідомленнями та керуйте цими контактами за допомогою вашої улюбленої бізнес-системи.
- Тримайте стейкхолдерів у курсі, гарантуючи, що повідомлення, надіслані клієнтом вашому бізнесу, відображаються у різних інструментах, що використовуються у вашому бізнесі.
- Додайте ботів, щоб автоматизувати типові завдання, зберігаючи людину в курсі розмови, щоб контролювати спілкування бота. [3]

Список месенджерів доступних до інтеграції:

- Android SDK
- Apple Business Chat
- Facebook Messenger

- Instagram
- iOS SDK
- LINE
- Mailgun
- MessageBird
- Telegram
- Twilio
- Twitter DM
- Viber
- Web Messenger
- WeChat
- WhatsApp

Sunshine Conversations дозволяє у повній мірі скористатися переліком структурованих і розширених типів повідомлень, які ці месенджери можуть запропонувати, завдяки одному простому у використанні API. Коли месенджери випускають нові функції, Sunshine Conversations залишається в курсі оновлень і допомагає скористатися новою функціональністю, практично не змінюючи існуючий код.[3]

Отже, Sunshine Conversations має подібний функціонал до того, що має розробитись, а саме, надсилання різних типів повідомлень у різні месенджери з одного місця, використовуючи зовнішні системи або підключаючи людей чи ботів для спілкування з клієнтом. Але відсутній такий функціонал як налаштування меню для зручнішого користування каналом, відсутні стандартні повідомлення, немає статистики по відправлених повідомленнях та використуванням меню. Але це дуже хороший приклад системи яку можна використовувати для порівняння і аналізу системи яка планується розробляти.

2.3. Порівняльна характеристика

Функціонал	Sunshine Conversations	Respond.io	Реалізована система
Меню	-	-	+
Стандартна відповідь	-	-	+
Статистика	-	-	+
Надсилання повідомлень	+	+	+
Інтеграція зовнішніх систем	+	-	+
Інтеграція з кількома месенджерами	+	+	+

РОЗДІЛ 4: ІНТЕГРАЦІЯ ЗОВНІШНІХ СИСТЕМ: ЕЛЕМЕНТИ ІНТЕГРАЦІЇ

Інтеграція зовнішніх систем буде проводитись з месенджером який у попередній роботі був обраний за основу – WeChat. Новим месенджером який був використаний у даній роботі є Telegram, так як це лідер по збільшенню кількості користувачів сьогодні. Також Telegram надає дуже зручне API для інтеграції з іншими системами. Отже, в цьому розділі буде розглянуто елементи інтеграції, які є спільними для WeChat та Telegram та демонструють можливості універсального модулю підключення зовнішніх систем до месенджерів.

3.1. Месенджер WeChat

WeChat – найпопулярніший додаток у Китаї, який використовується не тільки для обміну повідомленнями, а й для оплати послуг, розваг, знайомств, читання новин і так далі.

Розглянемо елементи інтеграції, які зможуть використовувати зовнішні системи, інтегрувавши універсальний модуль підключення зовнішніх систем до месенджерів у свою систему.

1. Меню для швидкої взаємодії користувача з каналом.

У WeChat воно може вміщувати до трьох пунктів, кожен з яких містить до 5 підменю.

Меню може бути різних типів, тобто виконувати різну роботу. Буде надаватись можливість використовувати 4 типи меню для WeChat каналів:

- Текст – у відповідь на натискання меню користувачем буде отримано простий текст, наприклад контакти компанії, або ціна на бажану послугу користувачем.
- Новини – тип меню, яке може в собі комбінувати картинку, текст та при натисканні на таке повідомлення переходити за посиланням.
- Посилання – при натисканні на таке меню користувач буде одразу переміщений за посиланням.

- Mini-program – тип меню, натиснувши на який, користувач буде переходити у міні-програму WeChat; даний тип меню є доволі унікальним для WeChat.

2. Обмін повідомленнями

Повідомлення, які будуть надіслані користувачем, будуть автоматично пересилатись у зовнішню систему, якщо стандартної поведінки не буде передбачено в універсальному модулі.

Також зовнішня система буде мати можливість розсилати broadcast повідомлення або індивідуальні повідомлення користувачам чи групі користувачів через зручне API універсального модуля.

3. Стандартна поведінка

Універсальний модуль надає можливість налаштувати відсилання стандартних повідомлень за деяких сценаріїв, один з яких – повідомлення при підписуванні на канал або вітальне повідомлення.

4. Збирання статистики

Універсальний модуль надає можливість відслідкувати кількість натискань на кожне з меню. Також відслідковується кількість повідомлень, які були надіслані до зовнішньої системи (за допомогою готових темплейтів, на який API надсилати повідомлення користувача), та кількість повідомлень, які були надіслані зовнішньою системою користувачу/користувачам.

3.2. Месенджер Telegram

Telegram - один з найпопулярніших месенджерів на сьогоднішній день, він є популярним у багатьох країнах, на відміну від WeChat. Telegram знаходиться в активній стадії додавання нового функціоналу, як от оплата за допомогою месенджера, відеоконференції та додатків типу ігор або інших міні-програм.

Існує багато елементів для інтеграції, але буде обрано максимально подібні до WeChat.

1. Меню для швидкої взаємодії користувача з каналом.

У Telegram меню для швидкої взаємодії користувача виглядає трішки інакше. Існує два типи меню для таких цілей. Перше - більш схоже на меню як у WeChat, але воно має набагато менше функціоналу, як от відсутність посилань при натисканні на меню, також таке меню не буде надсилати ніякої інформації про те, що користувач натиснув його, що в свою чергу є поганим тому, що це унеможливорює збирання статистики. Другий тип меню є більш функціональним та має всі функції, як і у WeChat, але таке меню як повідомлення, потрібно надіслати системою, тобто воно ніде не закріплене. Таке меню може вміщувати в собі до 12 пунктів.

Так як і у WeChat таке меню може бути різних типів. Було обрано 4 типи меню подібних до WeChat:

- Текст – у відповідь на натискання меню буде отримано простий текст
- Новини – тип меню, яке може в собі комбінувати картинку, текст та при натисканні на таке повідомлення переходити за посиланням. Цікавий факт, що в такому типові повідомлення більший акцент робиться на картинці, а не на описі. В WeChat – навпаки.
- Посилання – при натисканні на таке меню користувач буде одразу переміщений за посиланням, натискання на таке меню не буде відсилати данні про цю подію в універсальний модуль.
- Mini-program – тип меню натиснувши на який користувач буде переходити у міні-,програму Telegram.

2. Обмін повідомленнями

Аналогічно WeChat повідомлення надіслані користувачем будуть автоматично пересилатись у зовнішню систему через універсальний модуль, якщо стандартної поведінки не буде передбачено у системі.

Як і в WeChat зовнішня система буде мати можливість розсилати індивідуальні, групові або broadcast повідомлення користувачам через єдине для усіх месенджерів API універсального модуля.

3. Стандартна поведінка

Так як і для WeChat буде існувати можливість налаштувати стандартних повідомлень за деяких сценаріїв, наприклад вітальне повідомлення.

4. Збирання статистики

Універсальний модуль буде збирати кількість взаємодій між користувачем та кожним з месенджерів індивідуально, та надаватиме комфортний інтерфейс для переглядання/аналізування статистики, її скачування у зручному форматі.

РОЗДІЛ 5: ОГЛЯД API ПОПУЛЯРНИХ МЕСЕНДЖЕРІВ ДЛЯ СПІЛЬНОЇ ОБРОБКИ ОБРАНИХ ЕЛЕМЕНТІВ ІНТЕГРАЦІЇ

У цьому розділі розглянемо як інтегрувати WeChat та Telegram в зовнішню систему. Зробимо огляд API WeChat та Telegram, а саме розглянемо елементи інтеграції які були обрані.

5.1. WeChat

5.1.1. Інтеграція каналу WeChat

Для початку, у документації WeChat пропонується створити сервіс, власне зовнішню систему, з якою можна інтегрувати WeChat. Отже, мінімальна вимога – створити публічний ендпоінт, який буде повертати успішну відповідь – 200.

Ендпоінт має виглядати наступним чином: `http://public network IP/wx`

Наступним кроком буде створення офіційного акаунту WeChat. Для цього нам потрібно перейти за адресою <https://mp.weixin.qq.com/cgi-bin/loginpage> та пройти всі етапи реєстрації. Після успішного створення каналу, ми можемо перейти у розділ Basic Configuration та налаштувати інтеграцію. На сторінці налаштувань ми маємо заповнити такі поля:

- URL: `http://public network IP/wx`
- Token – використовується для забезпечення безпеки інтеграції, щоб ніхто інший, крім WeChat не міг надсилати запит на публічні ендпоінти, тільки WeChat сервери.
- EncodingAESKey – необов'язкове поле, може бути використане для кодування інформації, яка буде відправлена з WeChat серверів або на WeChat сервери.

5.1.2. Отримання токена для спілкування з серверами WeChat

Наступним кроком є отримання токена, який буде використовуватись при запитах до серверів WeChat, для оновлення меню, відправки повідомлень та інше.

На сторінці конфігурації можна знайти два важливих поля, які називаються appid та secret. Вони будуть використовуватись при отриманні access_token.

Формат запиту API:

```
HTTP request method: GET
https://api.weixin.qq.com/cgi-bin/token?grant\_type=client\_credential&appid=APPID&secret=APPSECRET [4]
```

Параметри:

Parameter	Required	Description
grant_type	Yes	Enter client_credential to get access_token
appid	Yes	Unique credential of third-party users
secret	Yes	The AppSecret. It is the key of the unique credential of third-party users.[4]

Опис відповіді:

Для успішного запиту сервери WeChat повертають наступний JSON:

```
{"access_token":"ACCESS_TOKEN","expires_in":7200} [4]
```


Parameters

Parameter	Description
access_token	The obtained credential
expires_in	The time when the credential expires (in sec) [4]

Коли виникає помилка, сервери WeChat повертають код помилки та відповідну інформацію. Приклад JSON (для недійсного AppID) показано нижче:[4]

```
{"errcode":40013,"errmsg":"invalid appid"}
```

Error codes

Error Code	Description
-1	System is busy. Try again later.
0	Request successful
40001	Incorrect AppSecret or mismatched AppSecret. Check whether AppSecret is correct.
40002	Ensure that the value of grant_type field is client_credential.

Error Code	Description
40164	The IP address used to call this API is not in the whitelist. Set it in the API IP whitelist. [4]

5.1.3. Створення меню для швидкою взаємодії користувача з каналом

Перейдемо до створення елементів інтеграції за допомогою WeChat API. Перший елемент - створення меню для швидкої взаємодії користувача з каналом.

Розглянемо типи кнопок, які були використанні в створенні універсального модулю підключення зовнішніх систем до месенджерів:

1. click

Надсилає подію, як тільки користувач клікнув на меню. Після того, як користувач натискає кнопку, сервер WeChat надсилає розробнику структуру з подією разом із значенням ключа, введеним розробником у кнопку. Розробник може взаємодіяти з користувачами за допомогою спеціального ключа.[5]

2. view

Переспрямовує на URL. Після того, як користувач натискає кнопку перегляду, програма WeChat відкриває URL-адресу веб-сайту, введену в кнопку розробником, для доступу до API. Це дозволить отримання базової інформації користувача через авторизацію веб-сторінки для того, щоб отримати основну інформацію користувача.[5]

3. media_id

Надсилає повідомлення (крім статей). Після того, як користувач натискає кнопку media_id, сервер WeChat надсилає користувачеві медіа, що відповідає постійному ідентифікатору медіа, введеному

розробником. Медіа, яка може бути включена до такого типу кнопки: зображення, аудіо, відео та статті. Примітка: постійним ідентифікатором мультимедіа повинен бути дійсний ідентифікатор, отриманий після завантаження носія за допомогою API попередньо на WeChat сервер[5]

Формат запиту API:

HTTP request method: POST (HTTPS protocol should be used) https://api.weixin.qq.com/cgi-bin/menu/create?access_token=ACCESS_TOKEN

Приклад запиту для click та view

```
{
  "button": [
    {
      "type": "click",
      "name": "Today's songs",
      "key": "V1001_TODAY_MUSIC"
    },
    {
      "name": "Menu",
      "sub_button": [
        {
          "type": "view",
          "name": "Search",
          "url": "http://www.soso.com/"
        },
        {
          "type": "miniprogram",
          "name": "wxa",
          "url": "http://mp.weixin.qq.com",
          "appid": "wx286b93c14bbf93aa",
          "pagepath": "pages/lunar/index"
        },
        {
          "type": "click",
          "name": "Give us a like",
          "key": "V1001_GOOD"
        }
      ]
    }
  ]
}
```

Параметри

Parameter	Required	Description
button	Yes	The array of primary menus. The number of menus should be between 1 and 3.
sub_button	No	The array of secondary menus. The number of menus should be between 1 and 5.
type	Yes	Response action type of the menu. view: Web page; click: Click; miniprogram: Mini Program
name	Yes	Menu name, with a length of up to 16 bytes. The maximum length of a sub-menu is 60 bytes.
key	Required for click type	Menu key, which is up to 128 bytes and used for pushing messages via the Message API.
url	Required for view and miniprogram types	The web page URL, which can be opened when a user taps the menu. It should not exceed 1024 bytes. When type is miniprogram, this URL will be opened in the older versions of the app that does not support Mini Program.
media_id	Required for media_id and view_limited types	The valid media_id returned by calling the New Permanent Media API
appid	Required for miniprogram type	The Mini Program's appid (only supported for verified Official Accounts)
pagepath	Required for miniprogram type	Path to Mini Program screen

[5]

Відповідь

Повернутий JSON для успішного запиту:

```
{"errcode":0,"errmsg":"ok"}
```

5.1.4. Обмін повідомленнями

Коли користувач Weixin відправляє повідомлення в офіційний обліковий запис, сервер WeChat відправляє пакет даних XML повідомлення на URL-адресу, введену розробником при налаштуванні каналу.[6]

Пакети даних XML для різних типів повідомлень структуровані таким чином:

Text Messages

```
<xml>
  <ToUserName><![CDATA[toUser]]></ToUserName>
  <FromUserName><![CDATA[fromUser]]></FromUserName>
  <CreateTime>1348831860</CreateTime>
  <MsgType><![CDATA[text]]></MsgType>
  <Content><![CDATA[this is a test]]></Content>
  <MsgId>1234567890123456</MsgId>
</xml> [6]
```

Parameter	Description
ToUserName	Developer's Weixin ID
FromUserName	Sender's account (an OpenID)
CreateTime	Message creation time (integer)
MsgType	Message type. It is "text" for text messages.
Content	Text message content
MsgId	Message ID (64-bit integer)

[6]

Для надсилання певному списку користувачів певне повідомлення, використовується наступний запит: [6]

```
HTTP request method: POST
https://api.weixin.qq.com/cgi-bin/message/mass/send?access_token=ACCESS_TOKEN
```

[6]

Приклад body для надсидаання media (статей):

```
{
  "touser": [
    "OPENID1",
    "OPENID2"
  ],
  "mpnews": {
    "media_id": "123dsdajkasd231jhksad"
  },
  "msgtype": "mpnews",
  "send_ignore_reprint": 0
}
```

media_id – попередньо завантажена стаття на сервери WeChat.

[6]

Приклад body для надсилання текстових повідомлень:

```
{
  "touser": [
    "OPENID1",
    "OPENID2"
  ],
  "msgtype": "text",
  "text": { "content": "hello from boxer."}
}
```

[6]

5.2. Telegram

Інтеграцією зовнішньої системи і Telegram месенджера можна назвати просто – telegram-bot. Каналом керує певний сервіс. Що він вміє та як інтегруватись - описується в даному розділі.

В документації Telegram API сказано що телеграм боти - це сторонні програми, які працюють всередині Telegram. Користувачі можуть взаємодіяти з ботами, надсилаючи їм повідомлення, команди та вбудовані запити. Зовнішній сервер керує своїми ботами, використовуючи запити HTTPS до Telegram API ботів.[7]

Що вміє робити Bot:

- Користувачі можуть отримувати індивідуальні сповіщення та новини. Бот може виступати як розумна газета, надсилаючи відповідний вміст відразу після його публікації. [7]
- Інтеграція з іншими службами. Бот може збагатити чати Telegram вмістом із зовнішніх служб. Бот Gmail, бот GIF, бот IMDB, бот Wiki, музичний бот, бот Youtube, GitHubBot[7]
- Приймати платежі від користувачів Telegram. Бот може запропонувати платні послуги або працювати як віртуальна вітрина. [7]
- Створювати персональні інструменти. Бот може надавати попередження, прогнози погоди, переклади, форматування чи інші послуги. [7]
- Побудова соціальних послуг. Бот міг би зв'язати людей, які шукають партнерів для бесіди на основі спільних інтересів або відстаній на якій вони знаходяться один від одного. [7]

Як працюють боти Telegram?

По суті, Telegram-боти - це спеціальні облікові записи, для налаштування яких не потрібен додатковий номер телефону. Користувачі можуть взаємодіяти з ботами двома способами:

- Надсилати повідомлення та команди ботам, відкриваючи з ними чат або додаючи їх до груп.
- Надсилайте запити безпосередньо з поля введення, ввівши @username бота та запит. Це дозволяє надсилати вміст від вбудованих ботів безпосередньо в будь-який чат, групу чи канал. [7]

Повідомлення, команди та запити, надіслані користувачами, передаються програмному забезпеченню, що працює на серверах зовнішніх систем. Telegram сервер, який працює як посередник, виконує усі функції шифрування та зв'язку з API Telegram. Існує простий HTTPS-інтерфейс для простого спілкування з цим сервером, який пропонує спрощену версію API Telegram. По-іншому цей інтерфейс можна назвати API для ботів. [7]

Бот створюється в кілька кліків за допомогою іншого бота, який називається BotFather. Після створення бота, де можна заповнити такі данні, як: назва бота, його нікнейм, опис і інше, буде отримано токен. Цей токен буде використовуватись при інтеграції.

Також, дуже важливим є пункт про те, що телеграм надає зручну бібліотеку для роботи з усіма елементами, які доступні по API, що пришвидшує та полегшує розробку ботів.

При розробці боту інтеграції було використано бібліотеку Telegram для Java.

5.2.1. Інтеграція з Telegram

Кожному боту при його створенні надається унікальний маркер автентифікації. Маркер виглядає приблизно так: 123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11

Отже, щоб зробити будь який запит, формат буде наступним:
https://api.telegram.org/bot<token>/METHOD_NAME[7]

5.2.2. Створення меню для швидкою взаємодії користувача з каналом

Для меню швидкої взаємодії з користувачем було обрано - `InlineKeyboardMarkup`. Вона має наступні параметри:

Field	Type	Description
<code>inline_keyboard</code>	Array of Array of InlineKeyboardButton	Array of button rows, each represented by an Array of InlineKeyboardButton objects[7]

Це меню, яке містить в собі масив кнопок. `InlineKeyboardButton` – преставлення цих кнопок, має наступні параметри:

Field	Type	Description
<code>text</code>	String	Label text on the button
<code>url</code>	String	<i>Optional.</i> HTTP or tg:// url to be opened when button is pressed
<code>login_url</code>	LoginUrl	<i>Optional.</i> An HTTP URL used to automatically authorize the user. Can be used as a replacement for the Telegram Login Widget .

Field	Type	Description
callback_data	String	<i>Optional.</i> Data to be sent in a callback query to the bot when button is pressed, 1–64 bytes
switch_inline_query	String	<i>Optional.</i> If set, pressing the button will prompt the user to select one of their chats, open that chat and insert the bot's username and the specified inline query in the input field. Can be empty, in which case just the bot's username will be inserted.
switch_inline_query_current_chat	String	<i>Optional.</i> If set, pressing the button will insert the bot's username and the specified inline query in the current chat's input field. Can be empty, in which case only the bot's username will be inserted.
callback_game	CallbackGame	<i>Optional.</i> Description of the game that will be launched when the user presses the button. NOTE: This type of button must always be the first button in the first row.
pay	Boolean	<i>Optional.</i> Specify True, to send a Pay button . NOTE: This type of button must always be the first button in the first row. [7]

Отже, як і у WeChat є 3 основні типи кнопок, це click, просто інформація про те що користувач хоче виконати якусь дію, наприклад, переглянути список доступних послуг. Далі, view – а саме кнопка з прямим посиланням на іншу сторінку. І остання, media – у відповідь надсилається картинка з описом та лінком на іншу сторінку.

Для реалізації `view` – заповнюємо поле `url`. Для реалізації інших двох – заповнюємо поле `callback_data` – сюди можна розмістити наприклад `id` кнопки, а при обробці даної події (коли сервер Telegram сповістить про те, що користувач натиснув на неї) будуть використані данні про те, який це тип кнопки і в залежності від цього надсилати відповідну відповідь.

Для типу кнопки `click` – це що завгодно, а найпростіше - це відіслати текст. Для цього було використано `SendMessage` об'єкт, в який просто передається `text` повідомлення.

Для типу кнопки `media` – використовується `SendPhoto` об'єкт, який в собі має `InlineKeyboardMarkup`, який містить одну кнопку з посиланням (наприклад, читати додатково), має текстове поле `description` та саме фото.

РОЗДІЛ 6: РЕАЛІЗАЦІЯ: СТЕК ТЕХНОЛОГІЙ

Для реалізації даної задачі ідеально підходила об'єктно орієнтована мова програмування з використанням принципів інкапсуляції, поліморфізму та наслідування. З огляду на це та попередній стек технологій, який використовувався для минулої версії системи, було обрано мову програмування Java та найпопулярніші технології для розробки веб-застосунку за допомогою Java.[9]

Отже, на back-end було використано Java 8. Вона посідає друге місце у рейтингу найпопулярніших мов програмування. Java є кросплатформеною, тому може запускатись на сервері з будь-якою операційною системою або і навіть без неї. Данна мова програмування має величезне ком'юніті, що в свою чергу означає, що існує дуже багато готових бібліотек для вирішення будь-яких завдань на Java, які дозволяють швидко та зручно розробляти практично будь-який продукт. [9]

Традиційно для Java було обрано Spring Framework. Найпопулярніший фреймворк для розробки Web застосувань на мові програмування Java. Вона надає багато можливостей та готових рішень для розробки продуктів будь-якої складності. Один з підпроектів Spring Framework – Spring Data. Він дозволяє легко працювати з різними типами баз даних одночасно як SQL так і NoSQL, що є ключовою вимогою для реалізації даної задачі, так як при проектуванні системи було вирішено, що буде використовуватись дві бази даних, SQL для зберігання даних про користувачів, компанії і ін. А NoSQL буде використана для роботи з статистикою.[9]

Для SQL бази даних було використано найпопулярніший ORM для Java - Nibernate, фактична імплементація Java Persistence API (JPA) специфікації. Він значно полегшує роботу розробника з базою даних, дозволяючи спростити роботу з Java Database Connection (JDBC), не витратити час на писання запитів в ручну, та працювати з сутностями, використовуючи повний потенціал OOP: успадкування,

поліморфізм, композицію та інше. Також Hibernate надає особливу sql подібну мову запитів, за допомогою якої, можливо писати запити до сутностей Java, а не таблиць SQL. [9]

Для збирання проекту в JAR та додавання залежностей для проекту було використано Maven – це один з найпопулярніших збирачів пакетів залежностей. Він дозволяє автоматизувати build процес, надає можливість унікально ідентифікувати проект та його версію, допомагає працювати із залежностями ефективніше. [9]

Як вже зазначалось вище, MySQL буде використано як система управління базою даних. MySQL - це одна з найпопулярніших SQL баз даних від компанії Oracle. Дана система буде використана для зберігання меню, зберігання стандартних відповідей, та відповідей по ключам меню і іншою подібною інформацією. [9]

NoSQL база даних – MongoDB. Найпопулярніша NoSQL база даних, яка буде використовуватись для зберігання статистики. Статистика, яка буде збиратись з месенджерів, займає досить багато місця; також буде багато нероздільної текстової інформації, тобто інформації, яка має бути збережена в одне поле. А як відомо, SQL бази даних погано справляються із такою задачею. [9]

Для front-end використовувались доволі прості та стандартні інструменти, такі як front-end є доволі простим. HTML і CSS – це стандартні технології, які використовуються для розмітки та стилізування Front-end застосунків. [9]

jQuery – це одна з найпопулярніших JavaScript бібліотек, яка часто використовується для застосунків з неважкою логікою, що ідеально підходить для реалізації потрібного функціоналу, а саме надання можливості створити меню, відповіді за замовчуванням та відповіді за ключами меню, відображати статистику і інші подібні задачі. [9]

Також при розробці використовувалась система управління версіями файлів GIT. Як віддалений сервер для зберігання файлів було використано Bitbucket. [9]

Застосунок хоститься на серверах Amazon Web Services – це одна з найпопулярніших та найкращих платформ хмарних обчислень. [9]

Операційна система серверу – Windows Server, є легкою для налаштування, так як до цього серверу можна підключитись за допомогою Remote Desktop Connection та працювати з сервером, наче на локальній машині. [9]

РОЗДІЛ 7: РЕАЛІЗАЦІЯ: ОПИС ГОТОВОЇ СИСТЕМИ

У цьому розділі розглянемо готовий функціонал системи для двох месенджерів WeChat та Telegram. Буде продемонстрований функціонал для інтеграції обох каналів, створення меню різних типів, спілкування між зовнішніми системами та користувачем. Попередньо було створено два канали: у WeChat та Telegram.

7.1 Інтеграція WeChat каналу в систему

WeChat тестовий канал було створено попередньо, він виглядає наступним чином:



Для інтеграції нам потрібен appId та appsecret, який ми маємо додати в налаштування інтеграції на сторінці.

Dashboard

Menu
Rich Messages
Default reply messages
Mini site editing
Template messages
Template messages V2

API Credentials
API Alias of terms
API Integration Specifications

Enter this link in WeChat developer panel: <https://2.43bc446e3b.ngrok.io/api/v1/wechat/>
We recommend You to fill up this form with a developer

Service API domain name ?
<https://dashboard.cwauthors.com.cn>

WeChat
Telegram

WeChat app data ?

WeChat app ID

WeChat app secret

Save

Заповнюємо ці данні та зберігаємо. Під час збереження буде виконаний запит на сервер WeChat, де буде виконана спроба отримати `access_token`.

Якщо ця дія буде виконана успішно, то ми отримаємо повідомлення про успішне отримання токена.

Наступним кроком, потрібно додати домен, до якого буде звертатись WeChat для відправлення інформації на сервер універсального модулю:

На попередньому скріншоті можна знайти даний лінк, який потрібно додати у налаштуваннях каналу:

Enter this link in WeChat developer panel: <https://2.43bc446ebe3b.ngrok.io/api/v1/wechat/>

При збереженні лінки на сервер універсального модулю підключення WeChat спробує до нього доступитись, та, при успішній операції, виведе відповідне повідомлення.

Якщо всі кроки пройшли успішно, то інтеграція відбулась. На наступному кроці розглянемо як створити меню швидкої взаємодії з користувачем.

7.2 Створення меню в універсальному модулі підключення зовнішніх систем до месенджерів

Створити меню, яке буде використовуватись для всіх інтегрованих месенджерів можна на головній сторінці Dashboard:

Dashboard

Menu Rich Messages Default reply messages Mini site editing Template messages Template messages V2

API Credentials API Alias of terms API Integration Specifications

WeChat menu item

first button

Menu name link to google

Menu type Custom menu

Menu response type Link

Link URL https://google.com

Menu name simple text content

Menu type Custom menu

Menu response type Text

Content text

Template message select

Add menu item

За допомогою цього меню можна додавати головні меню та підменю. В WeChat будуть відображатись і головні меню і підменю. В Telegram є тільки підменю, так як вони функціональні. Можливості додати головні меню для групування підменю в Telegram поки що не є доступними.

Для створення rich menu існує окрема сторінка, де можна створити темплейти та потім підставляти їх при створенні меню. Сторінка виглядає наступним чином:

Dashboard

rich message

Title rich message

Description rich message description

URL to picture https://media.macphun.com/img/uploads/customer/how-to/579/15531840725c93b5489d84e9.43781620.jp

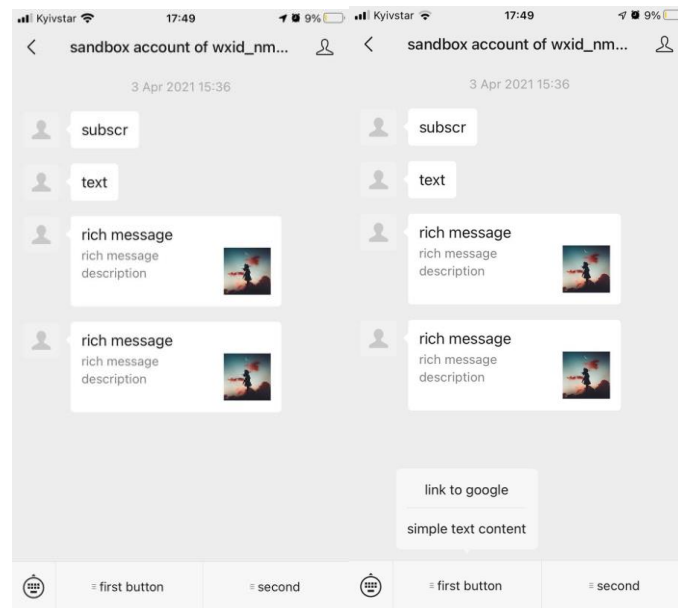
Website https://sinoptik.ua/

Add rich message

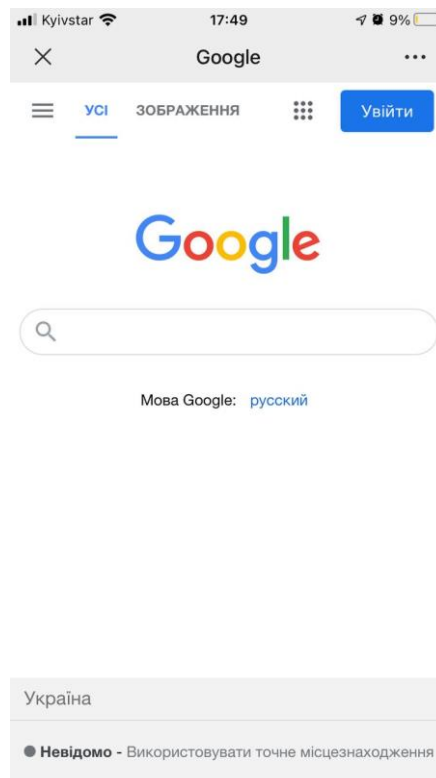
7.3 Демонстрація функціоналу меню в універсальному модулі підключення зовнішніх систем до месенджерів в WeChat

Перейдемо до завантаження створеного меню в WeChat. В кінці сторінки є кнопка “Upload menu to WeChat chat”. При натисненні на неї отримаємо повідомлення про успішне завантаження меню.

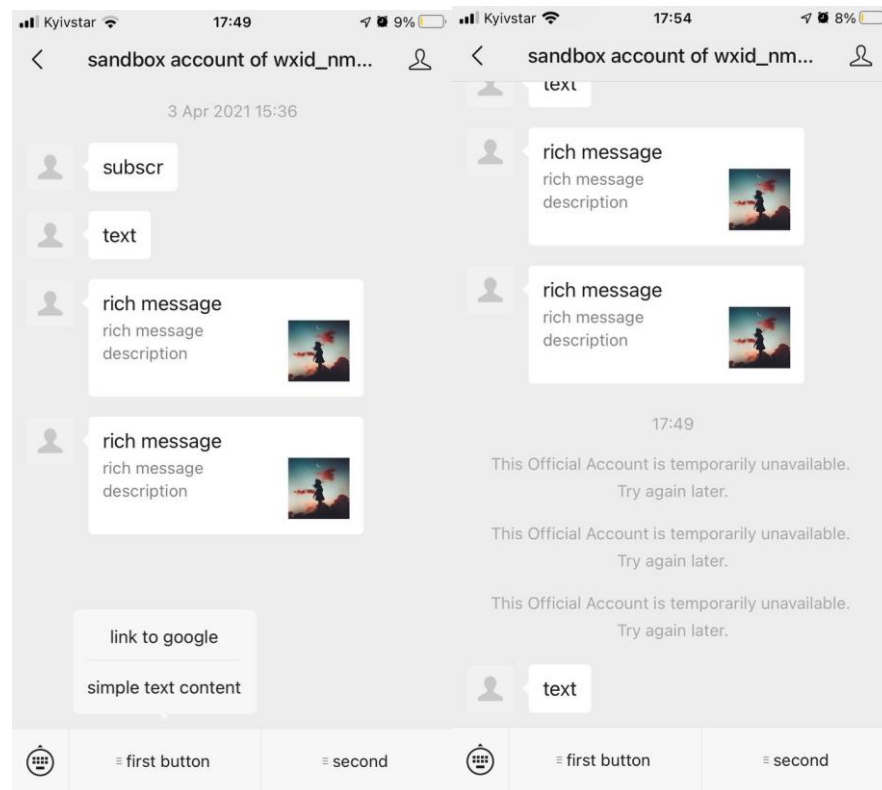
У данному меню було додані усі типи кнопок для тестування функціоналу.



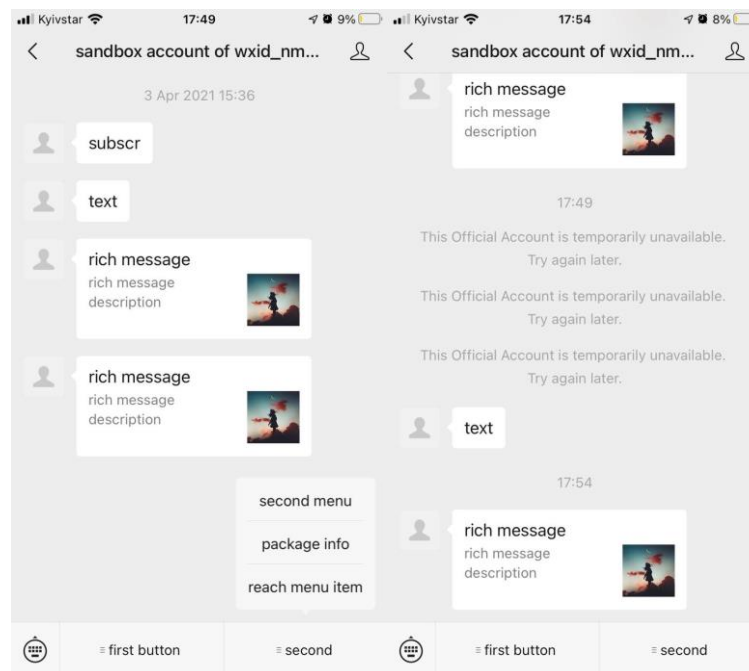
Перша кнопка, яка називається “link to google”, є кнопкою типу view – переходить за посиланням:



Наступною кнопкою є кнопка типу click – виконує будь-що, в даному випадку відсилає просто текстовий контент:

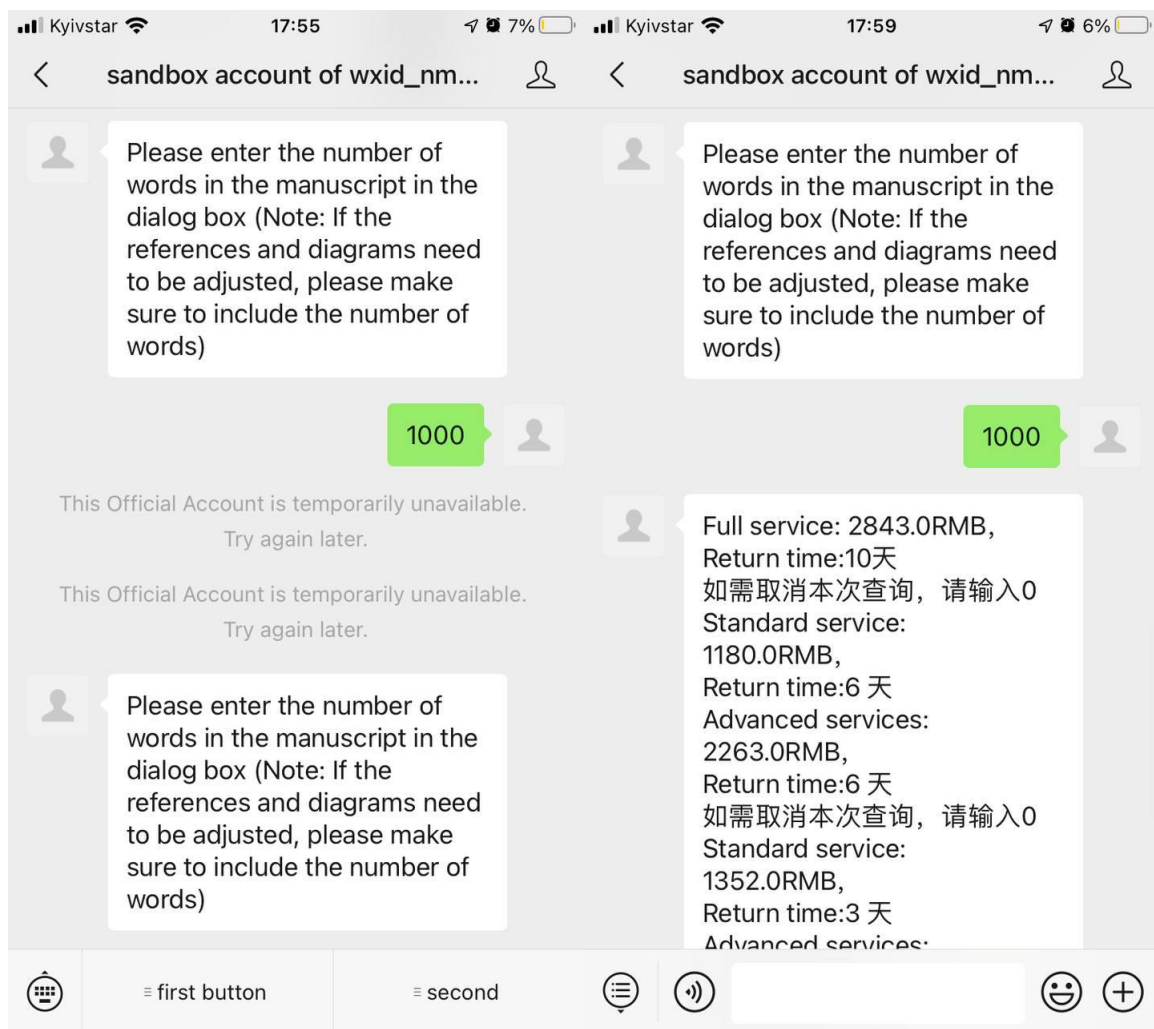


Наступна група кнопок. Почнемо з “second menu”. Це кнопка типу rich_menu, яка надсилає текстовий контент, картинку і посилання:



Остання кнопка “package info” є найцікавішою, її місія – це ще на початку діалогу надіслати повідомлення користувачеві для уточнення (наприклад, кнопка

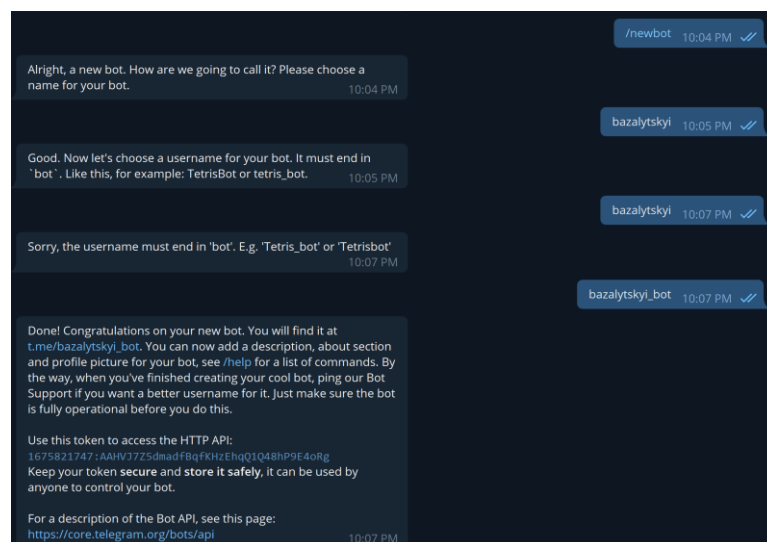
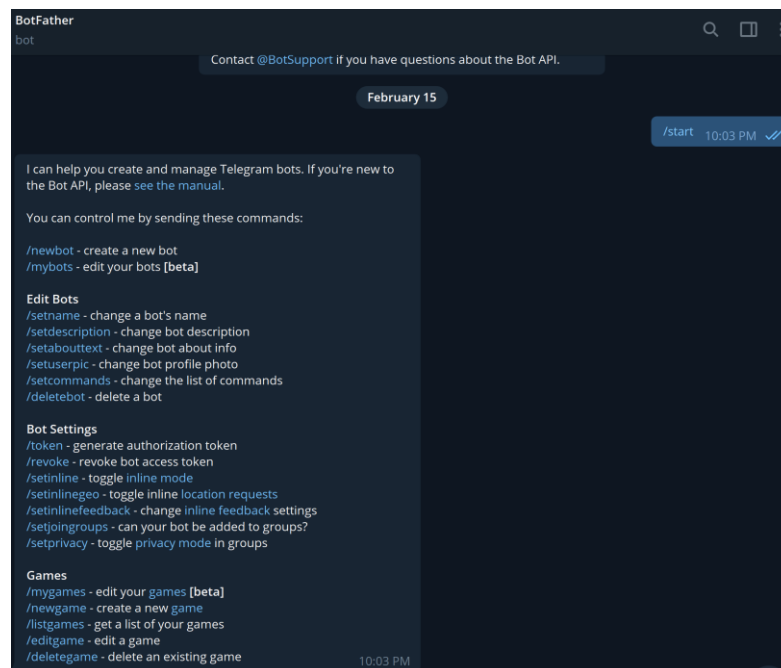
«скільки буде коштувати замовлення»), дана кнопка надсилає текстовий контент для введення даних від користувача (в прикладі кількість слів, яку потрібно обробити), після введення користувачем запитуваної інформації надсилає цю інформацію в зовнішню систему, отримує відповідь на неї та відправляє користувачу:



7.4 Інтеграція Telegram каналу в систему

В цьому розділі розглянемо все те, що було продемонстровано для інтеграційного процесу для WeChat, тільки для Telegram.

Отже, почнемо. Для початку потрібно створити новий Telegram бот. Він уже був попередньо створений. Виглядає процес наступним чином. У BotFather створюємо бот. Це, насправді, набагато простіше, ніж створити тестовий акаунт в WeChat.



В дві команди створюємо бот та отримуємо токен, який заповнюємо в системі для інтеграції.

На сторінці інтеграції зберігаємо отриманий токен. При збереженні токена буде виконаний тестовий запит, щоб отримати access_token.

Integration Specifications

Enter this link in WeChat developer panel: <https://2.43bc446ebe3b.ngrok.io/api/v1/wechat/>
We recommend You to fill up this form with a developer

Service API domain name ?

<https://dashboard.cwauthors.com.cn>

WeChat

Telegram

Telegram app data

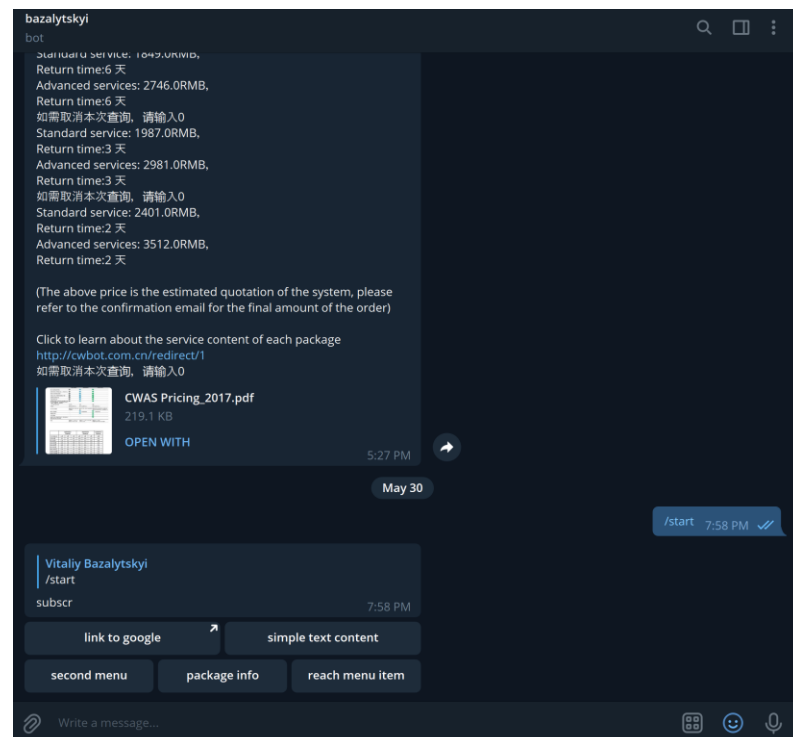
Telegram Bot token

1675821747:AAHVJ7Z5dmdfBqfKHzEhqQ1Q48hP9E4oRg

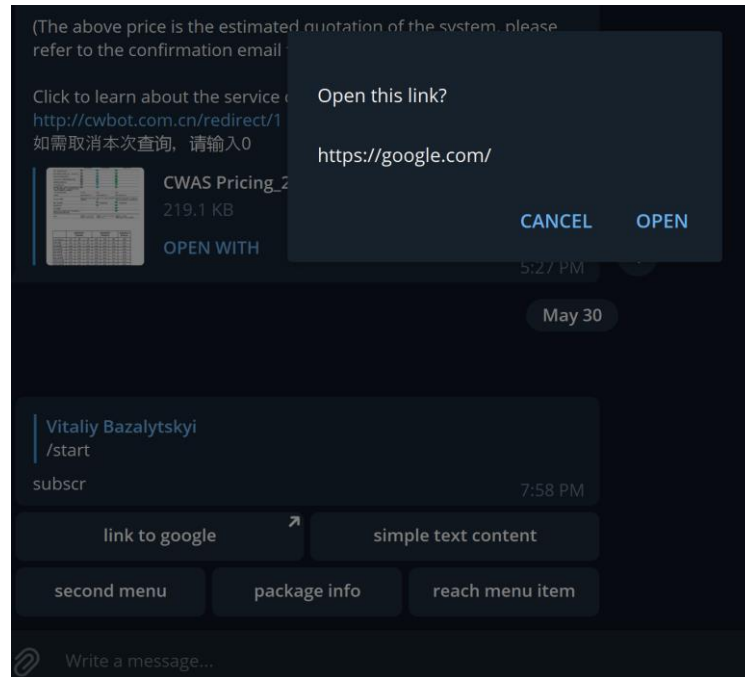
Save

7.5 Демонстрація функціоналу меню в універсальному модулі підключення зовнішніх систем до месенджерів в Telegram

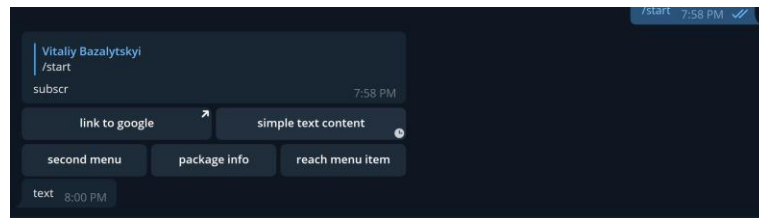
Поглянемо як таке ж меню буде виглядати в Telegram.



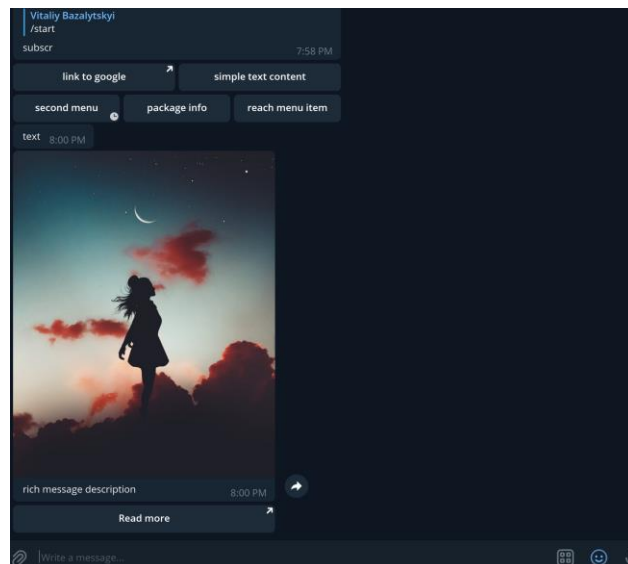
Демонстрація view кнопки:



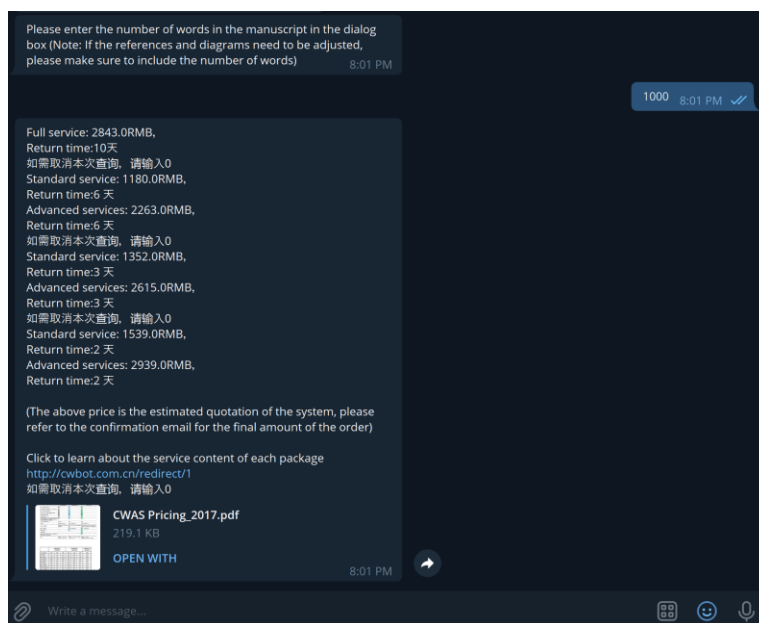
Демонстрація click кнопки:



Демонстрація reach menu кнопки:

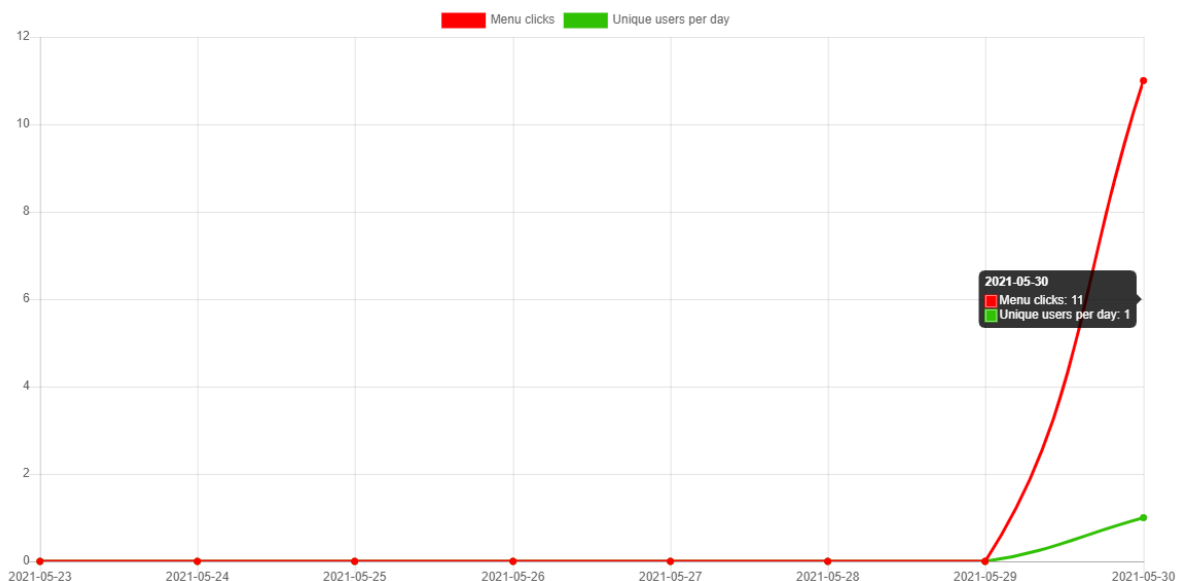


Демонстрація останньої кнопки, де користувачу виводиться повідомлення про потрібність певної інформації від користувача і після введення даної інформації відбувається такий же запит у зовнішню систему і відображається результат із зовнішньої системи:



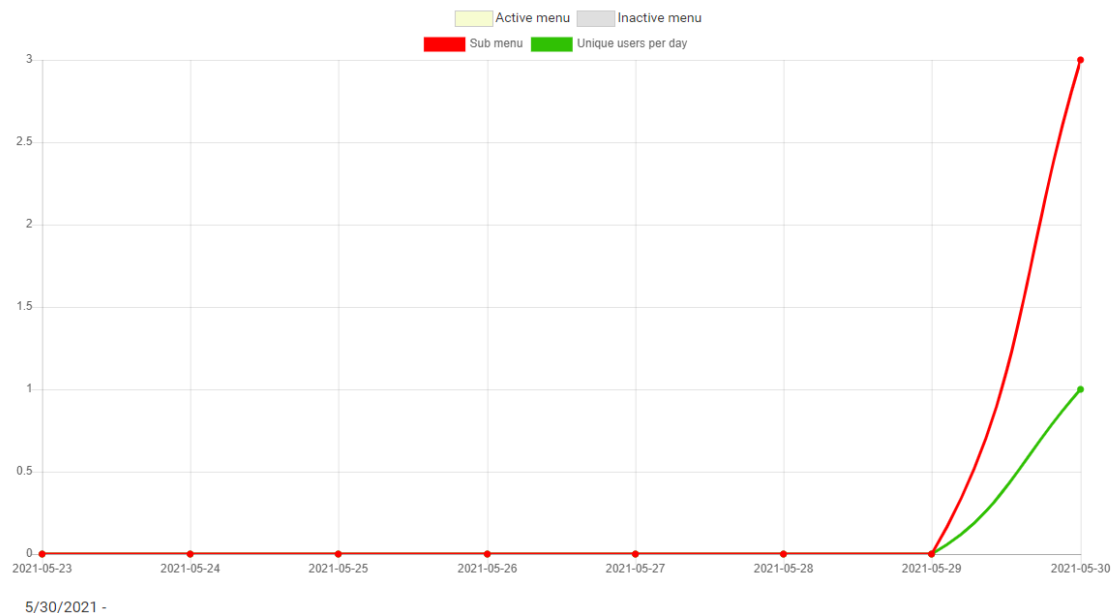
7.6 Демонстрація функціоналу статистики в універсальному модулі підключення зовнішніх систем до месенджерів в WeChat

Main menu clicks



Unique users for period 23/05/2021 - 30/05/2021: 1

Sub-Menu clicks



first button

- ☐ link to google
- ☒ simple text content

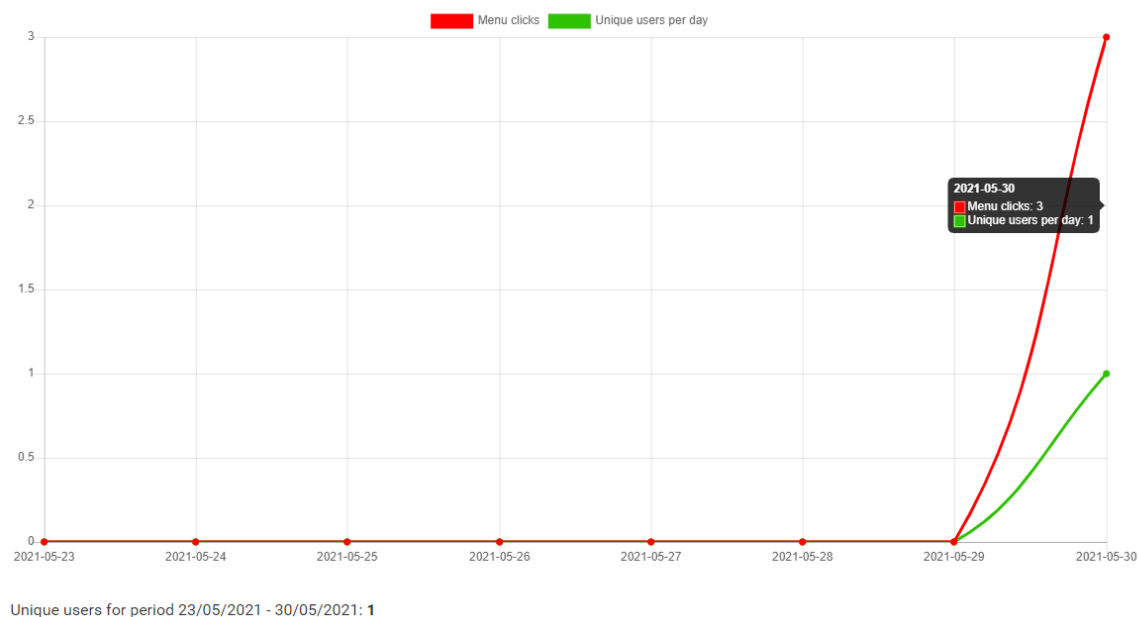
second

- ☐ second menu
- ☐ package info
- ☐ reach menu item

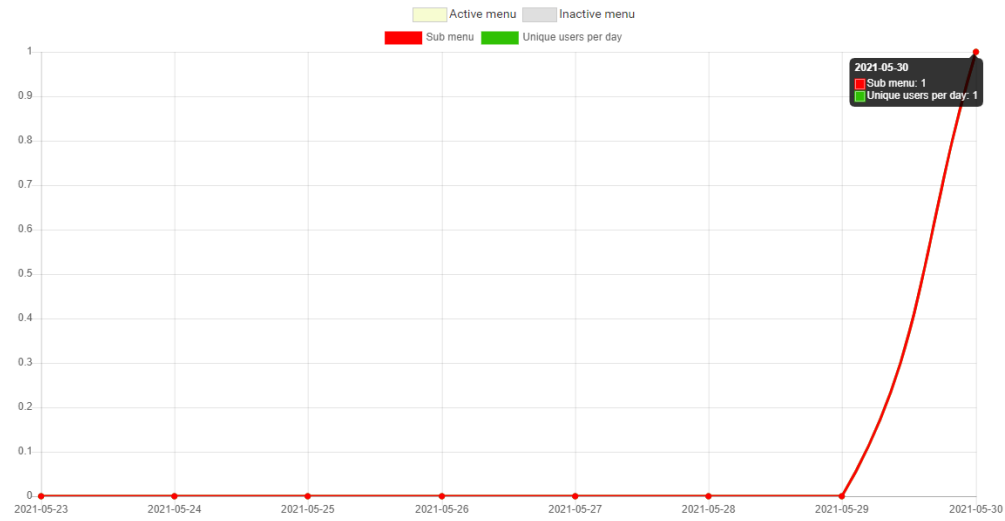
Статистику по конжному з підменю можна переглядати окремо.

7.7 Демонстрація функціоналу статистики в універсальному модулі підключення зовнішніх систем до месенджерів в Telegram

Main menu clicks



Sub-Menu clicks



5/30/2021 -

first button	second
<input type="radio"/> link to google	<input checked="" type="radio"/> second menu
<input type="radio"/> simple text content	<input type="radio"/> package info
	<input type="radio"/> reach menu item

ВИСНОВКИ

Виконуючи дану роботу, було реалізовано систему, яка надає можливість легко інтегруватись з месенджерами WeChat та Telegram. Даний універсальний модуль підключення зовнішніх систем до месенджерів надає можливість створювати меню каналу, обробляти різні типи подій, які створюються користувачем та збирати статистику кліків по меню каналу, створити двосторонню комунікацію між зовнішньою системою та користувачем в месенджері.

При роботі з існуючою системою, яка була розроблена в рамках курсової роботи, було впроваджено архітектурні зміни, які надають можливість легко додавати нові месенджери в універсальний модуль для інтеграції. Для того, щоб додати новий месенджер, потрібно створити всього лиш два додаткові класи, контролер яких би відкривав ендпоінт для викликання його по API серверами месенджера, який був доданий. Також клас, який включає в собі бізнес логіку, яка належить до месенджера. Він може перевикористовувати функціонал, спільний для усіх месенджерів, як от: збирання статистики, відправлення повідомлення зовнішній системі та отримання відповіді від неї та інше.

Отже, подальший розвиток даної системи виглядає наступним чином: додавання підтримки нових месенджерів у систему.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. A Solid Guide to SOLID Principles [Електронний ресурс] – Режим доступу до ресурсу: <https://www.baeldung.com/solid-principles>
2. Respond.io | The #1 Business Messaging Platform for all channels [Електронний ресурс] – Режим доступу до ресурсу: <https://respond.io/>
3. Sunshine Conversations Docs | Overview [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.smooch.io/guide/>
4. Weixin public doc [Електронний ресурс] – Режим доступу до ресурсу: https://developers.weixin.qq.com/doc/offiaccount/en/Basic_Information/Get_access_token.html
5. Weixin public doc (Creating Custom Menus) [Електронний ресурс] – Режим доступу до ресурсу: https://developers.weixin.qq.com/doc/offiaccount/en/Custom_Menus/Creating_Custom-Defined_Menu.html
6. Weixin public doc (Receiving Common Messages) [Електронний ресурс] – Режим доступу до ресурсу: https://developers.weixin.qq.com/doc/offiaccount/en/Message_Management/Receiving_standard_messages.html
7. Bots: An introduction for developers) [Електронний ресурс] – Режим доступу до ресурсу: <https://core.telegram.org/bots>
8. How Much Time Do People Spend on Social Media in 2019? [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@JBBC/how-much-time-do-people-spend-on-social-media-in-2019-infographic-cc02c63bede8>
9. Базалицький В. М. Розробка системи інтеграції систем управління бізнес-процесами з месенджерами – Київ, 2020.