

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ПОДОЛЯН МИХАЙЛО МИХАЙЛОВИЧ

Допускається до захисту:

завідувач кафедри
інформаційних технологій,
доктор технічних наук, доцент

_____ Т. В. Нескородева

«_____» _____ 2022р.

Розробка телеграм-боту розкладу занять в університеті

Спеціальність 122 «Комп'ютерні науки»

Кваліфікаційна (бакалаврська) робота

Керівник:

Штовба С. Д., професор кафедри
інформаційних технологій

Оцінка: _____ / _____ / _____

(бали за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____

(підпис)

Вінниця – 2022

АНОТАЦІЯ

Подольан М. М. Розробка телеграм-боту розкладу занять в університеті. Спеціальність 122 «Комп'ютерні науки», освітня програма «Сучасні інформаційні технології та програмування». Донецький національний університет імені Василя Стуса, Вінниця, 2022.

У кваліфікаційній (бакалаврській) роботі досліджено взаємодії користувача з існуючими системами обміну миттєвими повідомленнями. Проведено огляд ключових факторів взаємодії та алгоритмів, покладених в основу чат-ботів. Описано принципи, які є основними для створення чат-ботів. Наведено актуальність розробки телеграм бота та застосування його в студентському житті. Розглянуто проблеми проєкту, які вирішує розробка чат-боту. Показано детальний опис практичної частини роботи, результат проведеної роботи, а також обґрунтовано вибір використаних технологій.

Ключові слова: телеграм бот, розклад занять, інформаційні технології, розробка продукту.

61 с., 38 рис., 35 джерело.

ABSTRACT

Podolyan M. M. Development of a telegram-bot for university schedule. Specialty is 122 "Computer Sciences", educational program "Modern Information Technology and Programming". Vasyl' Stus Donetsk National University, Vinnitsa, 2022.

In qualifying (to the bachelor) work the interaction of the user with the existing systems of exchange of messages is investigated. A review of the key factors of interaction and algorithms underlying chatbots was carried out. Described the principles that are basic for the creation of chatbots. The relevance of developing a telegram bot in student life is described. The problems of the project, which the development of the chat-bot is solved. A detailed description of the practical part of the work, the result of the work, as well as the choice of technologies used is justified.

Key words: telegram bot, occupation layout, information technologies, product development.

61 p., 38 pict., 35 source.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1.АНАЛІЗ ОБ’ЄКТУ ДОСЛІДЖЕННЯ	6
1.1 Загальна концепція чат-боту та тенденції	6
1.2 Сучасні месенджери.....	12
1.3 Приклади успішних рішень систем оповіщення про розклад занять.....	18
1.4 Деталізація завдань розробки	22
Висновок до розділу 1	27
РОЗДІЛ 2. Програмна реалізація бота	28
2.1 Обґрунтування інструментарію розробки.....	28
2.2 Вибір засобів розробки чат-боту	35
2.3 Підключення бібліотек	43
Висновок до розділу 2.....	43
РОЗДІЛ 3. Практичне застосування на рівні ФПТ ДонНУ	45
3.1 Інструкції користувача та адміністратора	45
3.2 Приклади застосування та операційний аналіз.....	53
3.3 Розміщення чат-бота на віддаленому сервері Heroku	56
Висновок до розділу 3	57
ВИСНОВКИ.....	58
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	59
ДОДАТКИ.....	62

ВСТУП

Ріст популярності використання чат-ботів як сучасного інструменту комунікацій зумовлена тим, що в даний час чат-боти стали широко використовуватися в багатьох сферах для встановлення комунікацій з користувачами Інтернету. Особливу популярність чат-боти набули, коли були інтегровані в месенджери та соціальні мережі, зокрема, у Telegram та Facebook, де з'явилися цілі вітрини та магазини чат-ботів.

Під чат-ботами в даній роботі розуміються спеціальні програми, що імітують мовленнєву поведінку людини і виступають як віртуальні співрозмовники в інтернет-спілкуванні з одним або декількома співрозмовниками.

Чат-боти можуть використовуватися в розважальних цілях або просто для спілкування, також існують функціональні чат-боти, які можуть повідомити погоду, курс валют та ін., а також чат-боти можуть використовуватися в бізнесі для організації комунікації зі споживачами, навчальними закладами використовуються в комерційних та маркетингових цілях, для привернення уваги здобувачів до закладу за рахунок використання модного способу комунікації.

Актуальність роботи диктує ситуація на ринку інформаційних технологій. З кожним роком його поповнюють нові програмні продукти, веб-сервіси та мобільні додатки, за допомогою яких можна побудувати не тільки комунікації, а й сприяти зміцненню бренду університету, як освітнього центру. Чат-бот дозволить спростити щоденні рутинні завдання студента, такі як отримання інформації про розклад, дзвінки, зміни розкладу та інше. Головною перевагою серед інших методів отримання інформації є те, що всі можливості поєднуються на платформі єдиного месенджера.

Мета створити чат-бот для факультету інформаційних і прикладних технологій, який полегшить студентам пошук інформації, відповідей на питання при мінімальній витраті часу.

Завдання кваліфікаційної (бакалаврської) роботи є розробка чат-бота з розкладом занять в месенджері Telegram.

Виходячи з поставленої мети, були поставлені такі завдання:

- розглянути визначення та поняття чат-ботів та коротку історію їх появи;
- пошук та порівняння наявних аналогів чат-ботів;
- виявлення вимог, якими має володіти програмне забезпечення;
- розробка та впровадження розробленого програмного забезпечення;

Об'єктом дослідження є розробка веб-сайту з використанням Typescript та Javascript.

Предмет дослідження - сучасні системи, представлені чат-ботами, які вважаються найкращими за своїм наповненням, швидкістю та з використанням AI.

Практичне значення: чат-бот буде використовуватись під час навчального періоду для популяризації університету серед абітурієнтів, налагодження підтримки чат-бота та роботи з ним допоможе зменшити велику кількість питань, які виникають у здобувачів освіти до викладачів та адміністрації ДонНУ.

Структура роботи: У вступі наведено актуальність розробки телеграмм бота та застосування його в студентському житті. Перший розділ курсової роботи присвячений проблемам проекту, які вирішує розробка чат-боту. Другий розділ акцентує увагу на алгоритмах, покладених в основу чат-ботів. Описано принципи, які є основними для створення чат-ботів. У третьому розділі було розглянуто детальний опис практичної частини роботи. У висновку описаний результат проведеної роботи, а також обґрунтовано вибір використаних технологій.

РОЗДІЛ 1

Аналіз об'єкту дослідження

1.1 Загальна концепція чат-бота та його тенденції

Така програма інтернет-комунікації як чат-бот заснована на сучасних технологіях, в її основі лежать такі інформаційно-комунікаційні технології як штучний інтелект, нейронні мережі, BigData (великі дані), що дозволяє наближати комунікації чат-ботів до людського віртуального спілкування та вирішувати їх допомогою комерційні та маркетингові завдання.

Таке поняття як чат-бот походить від двох англійських слів: to chat – невимущена розмова в Інтернеті, bot (robot) – робот, з чого випливає, що це роботи, призначені для здійснення комунікацій з користувачами в Інтернеті. Також ці програми називають віртуальними співрозмовниками або програмами-співрозмовниками, всі ці поняття синонімічні.

В даний час в літературі представлені різні визначення чат-ботів, проте їх аналіз показує, що вони мають досить багато спільного, що говорить про те, що є досить чітке розуміння цієї технології серед дослідників. Крім того, для користувача спілкування з ботом виглядає як просте листування з справжньою людиною (рис. 1.1).



Рисунок 1.1 – Бізнес-класифікація чат-бот додатків

Чат-бот — це розумна програма, яка живе у месенджерах та виконує різні функції. В даний час можна виділити різні типи чат-ботів в залежності від

різних критеріїв, якими можуть виступати складність чат-ботів, принцип їх роботи, використовувані ними технології, ціль використання, їх функціонал та ін.

За ступенем складності можна виділити два основних типи чат-ботів:

- Чат-боти, що працюють на основі правил та шаблонів. Даний тип чат-ботів дуже органічний у своєму функціоналі та можливості розпізнавання повідомлень та команд, але при цьому і досить простий у розробці;
- Чат-боти, що працюють на основі машинного навчання та мають штучний інтелект. Даний тип чат-ботів має більший функціонал, а також набагато ефективніше аналізує отримані повідомлення та команди, крім цього, вони постійно навчаються в процесі комунікації з користувачем, що розширює їх можливості, проте вони складніші та дорогі в розробці.

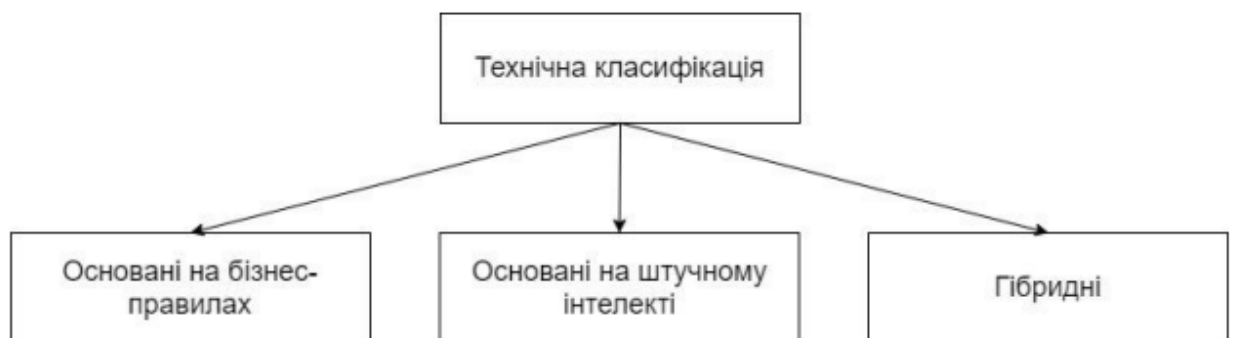


Рисунок 1.2 – Технічна класифікація чат-ботів

Перший тип чат-ботів спілкуються з користувачами, спираючись на заздалегідь написані команди та ключові слова, кожна з таких команд має бути написана розробником окремо з використанням регулярних виразів або інших форм аналізу рядків. Якщо користувач поставив запитання, не використавши жодного ключового слова, робот не може зрозуміти його і, як правило, відповідає повідомленням на кшталт «вибачте, я не зрозумів».

Недоліком даних чат-ботів є їх обмеженість у спілкуванні з користувачами лише тими шаблонами та командами, які задав розробник, що обмежує можливості спілкування з користувачами. Перевагами цього типу є простота їх створення, можна навіть не бути програмістом, а вибрати чат-бота з магазину, наприклад, у Facebook, крім того, їх розробка та підтримка не вимагають великих фінансових вкладень. Вони можуть бути корисні, якщо від них вимагається виконання достатнього вузького функціоналу, наприклад, відповіді на типові питання або виконання нескладних команд, наприклад, пов'язаних з пошуком.

Другий тип чат-ботів спираються на штучний інтелект, що означає, що місце заздалегідь підготовлених відповідей, вони можуть відповідати адекватними повідомленнями на тему, причому мовою, більш наближеною до природного. До того ж, всі слова, сказані чат-боту та ним самим, записуються для подальшої обробки.

Перевагами даного типу чат-ботів є те, що вони функціональніші, можуть дати більше точних відповідей, спілкування з ними більше нагадує спілкування з живою людиною, вони можуть навчатися самостійно, що економить час на розробку та завантаження шаблонів та команд. До їх недоліків відноситься висока вартість і складність розробки та підтримки, а також неможливість повністю контролювати процес навчання.

Як правило, чат-бот веде автоматичне спілкування з користувачем за допомогою тексту чи голосу від імені компанії чи бренду з метою спрощення онлайн-комунікації, щоб надавати користувачеві актуальну інформацію найоперативніше. Таким чином, спілкування користувача з чат-ботом є альтернативою листуванню з живим оператором або дзвінком менеджера компанії (рис. 1.3)

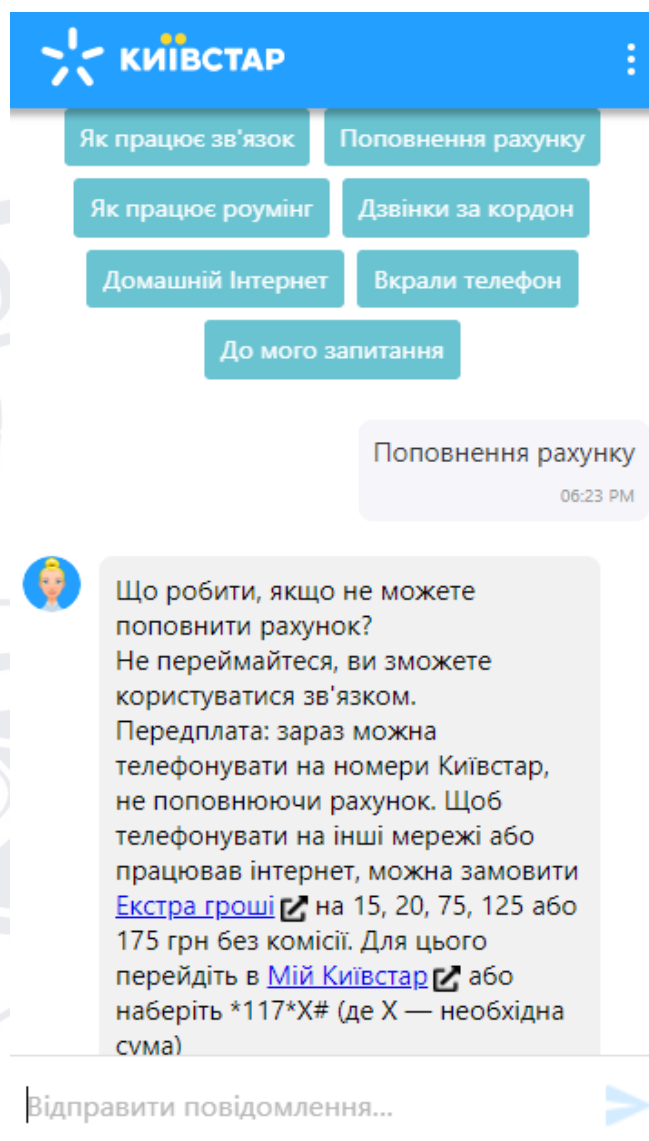


Рисунок 1.3 – Приклад листування з чат-ботом

Чат-бот є програмою, яка працює всередині месенджера, тобто системи обміну миттєвими повідомленнями, і здатна відповідати на запитання, а також самостійно їх ставити. При цьому чат-боти можуть виступати ще й як цифрові асистенти, перебуваючи всередині месенджера, виконуючи команди, даючи рекомендації та здійснюючи якийсь пошук за параметрами, які повідомляє користувач. Чат-бот також може бути

частиною інтерфейсу сайту і бути доступним його користувачеві безпосередньо через віджет (рис. 1.4)

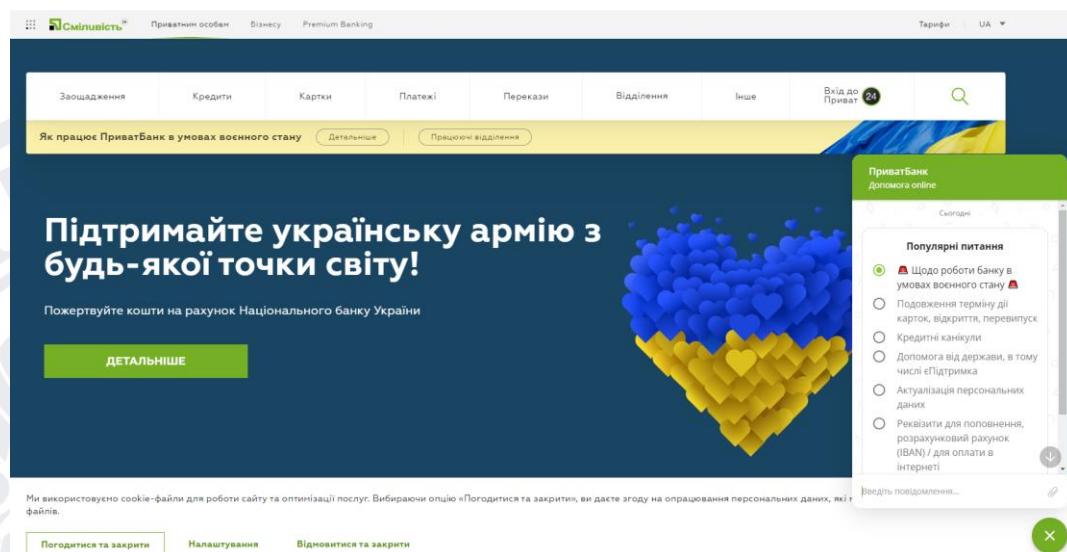


Рисунок 1.4 – Приклад чат-бота в форматі віджету

Таким чином, чат-бот – програма, яка з'ясовує потреби користувачів і потім допомагає задовольнити їх з метою пропозиції рішень при використанні найбільш природної форми спілкування – мовою, зрозумілою користувачеві, тобто при допомоги вербальної або невербальної взаємодії з людиною через онлайн-середовище для вирішення його питань. Чат-бот може автоматизувати завдання, які можуть бути розділені на інші сегменти, такі як підтримка клієнтів, опитування, планування зустрічей та багато іншого. Наприклад, чат-боти успішно допомагають користувачам вирішувати питання, пов'язані з певними брендами, відповідають на запитання, що часто ставляться або надають клієнтам інформацію про продукт або послугу. Крім всього вищезгаданого, боти також можуть допомагати клієнтам записуватися на зустрічі, дізнаватися про доступність продуктів, допомагати в планування поїздок та багато іншого .

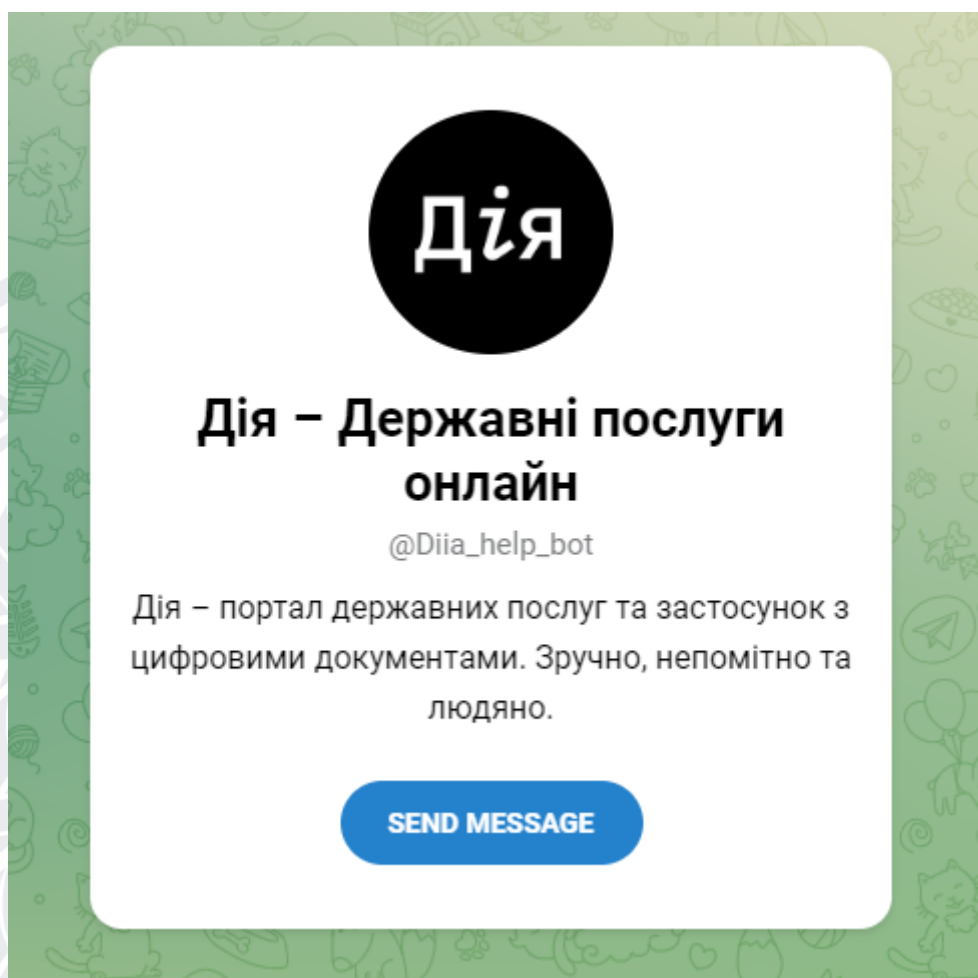


Рисунок 1.5 – Приклад чат-бота державного додатку Дія

Найкорисніші чат-боти розробляються за допомогою технологій машинного навчання і виділяються тим, що їх навчає людина, і вони призначені для використання людьми, таким чином, людина бере активну участь на всіх стадіях розробки такої програми співрозмовника. У той час як зростає поширення та використання чат-ботів, так само продовжує зростати популярність парадигми взаємодії «messaging-as-an-interface», згідно з якою обмін повідомленнями, будучи 7 швидким, прямим, добре візуалізованим і дозволяє вести спілкування відразу з великою кількістю співрозмовників, стає одним з ключових методів комунікації. Зростання популярності цієї парадигми сприяє й те, що більшість підприємств так чи інакше освоює соціальні мережі як канали спілкування з клієнтами. Ще

одна причина - Досить часто користувачам комфортніше написати свої запитання та отримати пряму відповідь, ніж задавати їх під час "живого" дзвінка оператору.

Чат-бот – це програма, з якою може взаємодіяти не лише один користувач в один момент часу, а набагато більше, що дозволяє охоплювати великі аудиторії, при цьому користувачі можуть спілкуватися з чатом різними способами: за допомогою текстових повідомлень, голосу, жестів та натискання на елементи інтерфейсу, якщо він призначений для конкретного бота. При цьому чат-бот доступний користувачам цілодобово, без перерв, на відміну живого оператора.

Протягом декількох років чат-боти, як правило, використовувалися в середовища обслуговування клієнтів, але в даний час спектр ролей використання чат-ботів повсюдно зростає, що зумовлено прагненням до підвищення якості обслуговування та ефективності бізнесу на різних підприємствах. При цьому варто враховувати, що чат-боти також мають різну ступенем інтелекту: від найпростіших чат-ботів, які можуть бути трохи більш ніж інтерфейсним рішенням для відповідей на стандартні питання, до просунутих діалогових AI-чат-ботів, що володіють інтелектуальними можливостями та здатністю розуміти те, що пише їм користувач, а також самонавчати і підлаштовуватися під особливості конкретного користувача, наприклад, під його поведінку чи манеру вести спілкування.

1.2 Сучасні месенджери

Останні роки популярність знову набирають месенджери. Існує безліч додатків для обміну повідомленнями: є очевидні та неминучі як Viber та Facebook Messenger, але крім них, є ще багато інших, призначених для любителів стікерів, зайнятих професіоналів, які дбають про безпеку та навіть геймерів.

Месенджери — це програма для обміну повідомленнями. Миттєвий обмін повідомленнями — це вид онлайн-чату, який пропонує передачу тексту в режимі реального часу через інтернет. З моменту появи смартфонів та подальшого вибухового зростання мобільних програм недорогі або безкоштовні чати та програми для обміну повідомленнями в соціальних мережах зарекомендували себе як дешева альтернатива текстовим повідомленням за допомогою операторів через SMS. Багато програм для обміну повідомленнями пропонують такі функції, як групові чати, обмін графікою, відео та навіть звуковими повідомленнями, а також наклейками та смайликами.

На графіку (рис 1.6) представлено кількість активних користувачів у найпопулярніших месенджерах станом на квітень 2021 року, за даними дослідження сайту statista.com. Також станом на квітень 2022 року у месенджера WhatsApp два мільярди користувачів і він є одним із найпопулярніших мобільних соціальних додатків у світі. У лютому 2014 року соціальна мережа Facebook придбала мобільний додаток за 19 мільярдів доларів США.

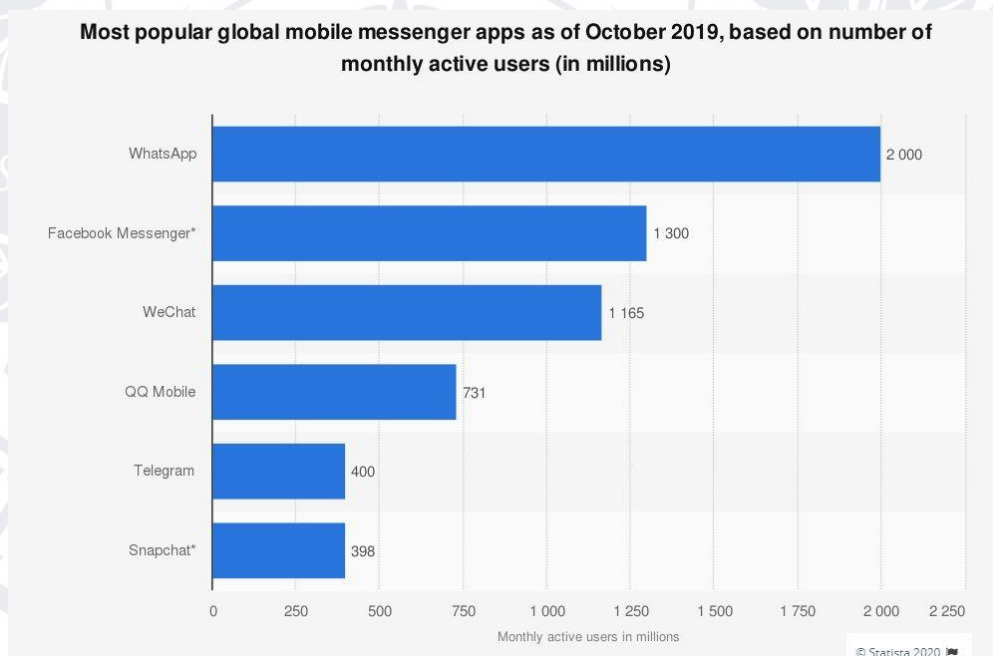


Рисунок 1.6 – Кількість активних користувачів у найпопулярніших месенджерах

За своїм устроєм месенджери поділяються на хмарні та звичайні. Відмінності в тому, що хмарний месенджер зберігає повідомлення на сервері і ви ніколи не втратите листування навіть якщо пристрій, з якого ви її вели буде втрачено. У звичайних месенджерах усі повідомлення зберігаються виключно на пристрої користувача.

Є багато суперечок який підхід правильніший. З одного боку використовуючи хмарні послуги ви йдете на компроміс довіряючи своє листування компанії, яка стоїть за месенджером. З іншого боку, це набагато зручніше. І у вас є багато додаткових можливостей. Наприклад, ви можете завантажити месенджер на комп'ютер, почати листуватись в ньому, а потім піти від комп'ютера і продовжити листування на смартфоні. До того ж для конфіденційних листування існують секретні чати, які не зберігаються на сервері.

У частині базового функціоналу, надсилання коротких повідомлень, усі месенджери приблизно однакові. Відмінності починаються у додатковому функціоналі, безпеці, зручності реалізації тих чи інших функцій. Тому важко відповісти однозначно який месенджер краще, оскільки кожен добрий у чомусь своєму.

Найбільш функціональним на території України можна вважати Telegram (рис. 1.7) з його хмарним зберіганням чатів, наскрізним шифруванням, каналами та ботами. Але зрештою користувачам доводиться вибирати той месенджер, в якому спілкуються більшість ваших контактів. А найчастіше доведеться поставити два, а то й усі три месенджери, щоб бути на зв'язку з

усіма.

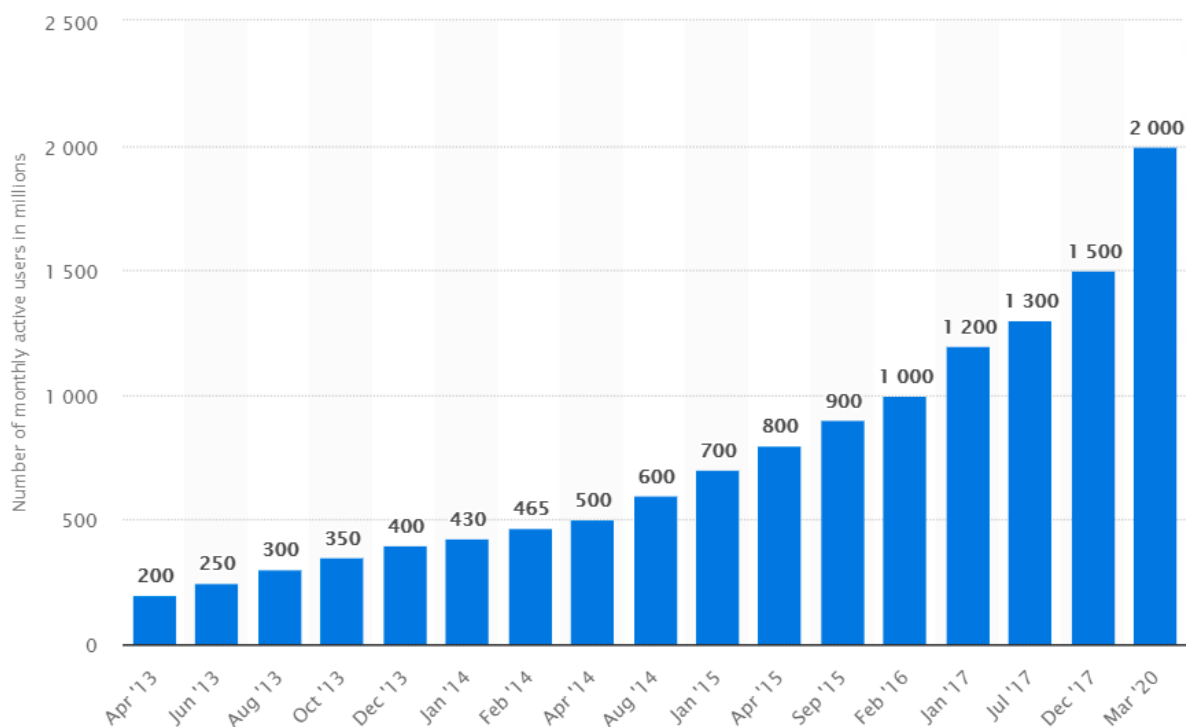


Рисунок 1.7 – Кількість активних користувачів WhatsApp за місяць.

Telegram – 400 млн. користувачів. Telegram схожий на більшість програм для обміну повідомленнями і досить відомий своєю безпекою. Є кілька способів, якими бренди можуть використовувати Telegram, крім надання підтримки клієнтів наодинці. Наприклад, вони можуть створювати чат-ботів для платформи або використовувати функцію каналу Telegram для трансляції повідомлень необмежену кількість передплатників.

В період із березня 2021 по квітень 2022 кількість активних користувачів зросла на 80% (рис. 1.8). За той же період у WhatsApp цей показник зріс тільки на 15%.

У квітні 2022-го аудиторія Telegram (дітища росіянина Павла Дурова) перейшла позначку 400 млн. користувачів по всьому світу (за рік +100 млн.). І ця цифра продовжує зростати. Україна тут не є винятком.

Так, у 2022 році трійку найпопулярніших месенджерів серед українців очолили: Viber, Facebook Messenger та Telegram. І якщо про перші два можемо сміливо сказати, що чули/бачили/користувалися, то останній досі для багатьох залишається темним лісом. І при цьому – абсолютна більшість політичних скандалів останнім часом розпочиналася саме зі «зливою» у телеграм-каналах.

Функціонал Telegram дозволяє заводити і підписуватися на «канали» (такі блоги), обмінюватися файлами, створювати чат-ботів та багато іншого.

«Цей месенджер вважається одним із найбезпечніших для спілкування із списку всіх популярних месенджерів», – пишуть про Telegram у Мережі.

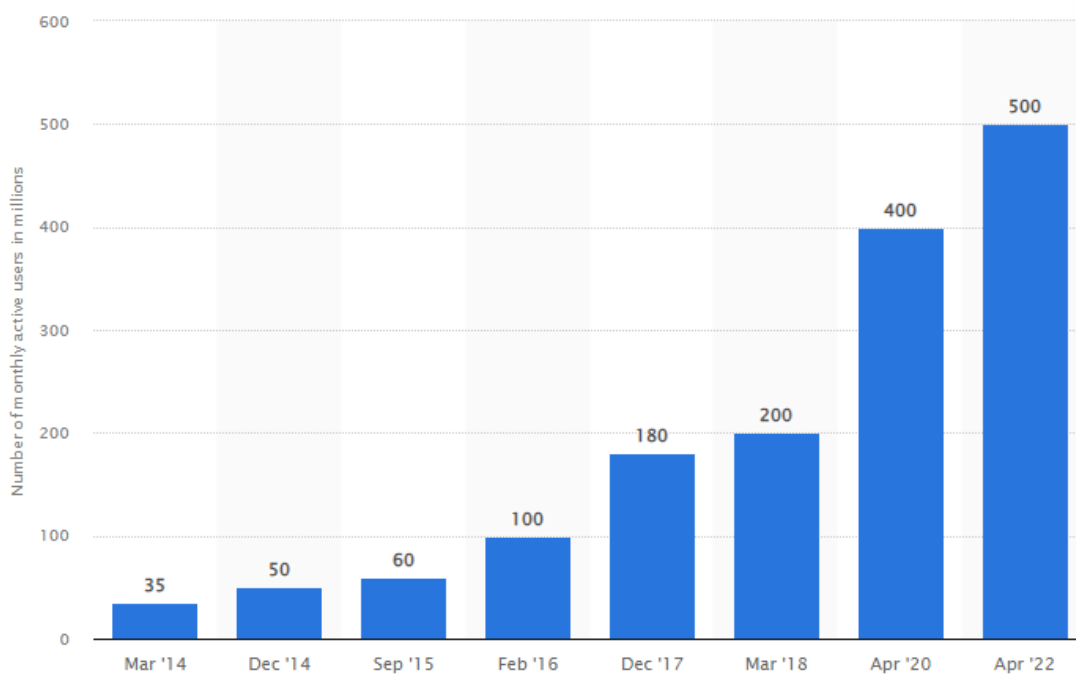


Рисунок 1.8 – Кількість активних користувачів Telegram за місяць, млн

Український сегмент Telegram особливо швидко почав зростати після блокування російських соцмереж Вконтакте та Однокласники – у 2017 році. Проте ті, які обсяги цей продукт займає на вітчизняному ринку, навіть 2022 року

оцінити складно. На це є причина. Наприклад, якщо у Facebook можна написати до центрального офісу чи регіонального представництва та запитати, скільки витрачає на рекламу український бізнес за рік, то відносно Telegram маємо: «На жаль, Telegram таких даних не надає. Тому що прямо на платформі немає вбудованої реклами. Вся екосистема комерційних розміщень у Telegram вона побудована на приватних домовленостях власників каналів (адміністраторів) та рекламодавців», – розповідає Андрій Бродецький.

Telegram має програму під будь-яку платформу: iOS, Android, Windows, Mac OS, Linux і навіть Chrome OS. А для решти є веб-версія, в яку можна потрапити з будь-якого сучасного браузера. Також підтримується розумний годинник від Apple, від Samsung і навіть страждальці на базі VR OS. Більш того, користувач можете користуватися всіма цими платформами на необмеженій кількості пристроїв і всі вони будуть між синхронізуватися. До того ж, логінуватися на новому пристрої дуже просто. Вам не потрібно сканувати жодні QR-коди, а код підтвердження надходить у сам Telegram. Тобто, користувач може його подивитися з будь-якого пристрою, на якому вже логінілися раніше.

Якщо користувачу не вистачає функціональності офіційних клієнтів Telegram, він завжди можете скористатися кастомними клієнтами від сторонніх розробників. По суті всі вони створені на основі вихідного коду Telegram, але кожен з них має свої унікальні особливості та можливості. Наприклад, Unigram має версію xbox One. Bettergram існує для всіх десктопних платформ і дозволяє закріпити до 50 чатів замість 5. Під мобільні версії можна виділити Getogram – там дуже багата можливість кастомізації, є вбудований перекладач і можна додати більше трьох аккаунтів. Найважливіше, що такі клієнти безпечні. Їхні розробники не мають доступу до кодів шифрування та використовують спеціальні протоколи Telegram, які гарантують безпеку.

Безпека – це ще одна причина, через яку, на мій погляд, Telegram найкращий месенджер. Для доступу до свого облікового запису можна налаштувати двофакторну авторизацію, а це означає, що зламати його буде набагато

складніше навіть, якщо у зловмисника є доступ до всіх смсків, які приходять до вас на телефон. Усі дані, що надсилаються через Telegram і всі дані, які вже зберігаються на серверах месенджера, шифруються. Telegram, за заявою засновників, нікому не передає ключі шифрування, зокрема державним органам безпеки.

Крім цього, є можливість спілкуватися в секретних чатах. Трафік у них не потрапляє на сервер Telegram, а проходить безпосередньо між учасниками такого чату. А ще в секретних чатах можна надіслати повідомлення із таймером самознищення. Для цього натисніть на іконку годинника, а таймер запуститься, відповідно, коли одержувач прочитає повідомлення. Відмінно для обміну оголеними тілами з коханими). Просто чудово.

Якщо сталося якесь лихо та історію листування треба стерти, то це можна зробити. Причому стерти можна як свої повідомлення, а й повідомлення співрозмовника. Щойно його пристрій підключитися до інтернету – вони зникнуть, наче їх там і не було.

Отже, питання розробки та наукового супроводження ДонНУ на платформі Telegram, на поточний час, є актуальним.

1.3 Аналіз конкурентних систем оповіщення про розклад занять

Чат-бот для факультету інформаційних і прикладних технологій буде виконано українською мовою, тому під час аналізу наявних рішень, було обрано найпопулярніші та найбільш затребувані боти українських компаній та продуктів. Серед найпопулярніших функцій, які можуть задовольнити боти, є зв'язок та взаємодія з іншими мобільними додатками та сервісами.

Прикладом такого боту є RailwayBot (рис. 1.9), який допомагає моніторити, шукати та купувати квитки на потяг. За допомогою боту можна поставити моніторинг на певний потяг, дату, часу та тип місця. Як тільки квиток з'явиться в наявності - бот надішле сповіщення. Алгоритм дії схожий на купівлю

квитків на uz.booking.gov.ua. Бот реалізовано на основі кнопок з введенням міст відправлення та прибуття.

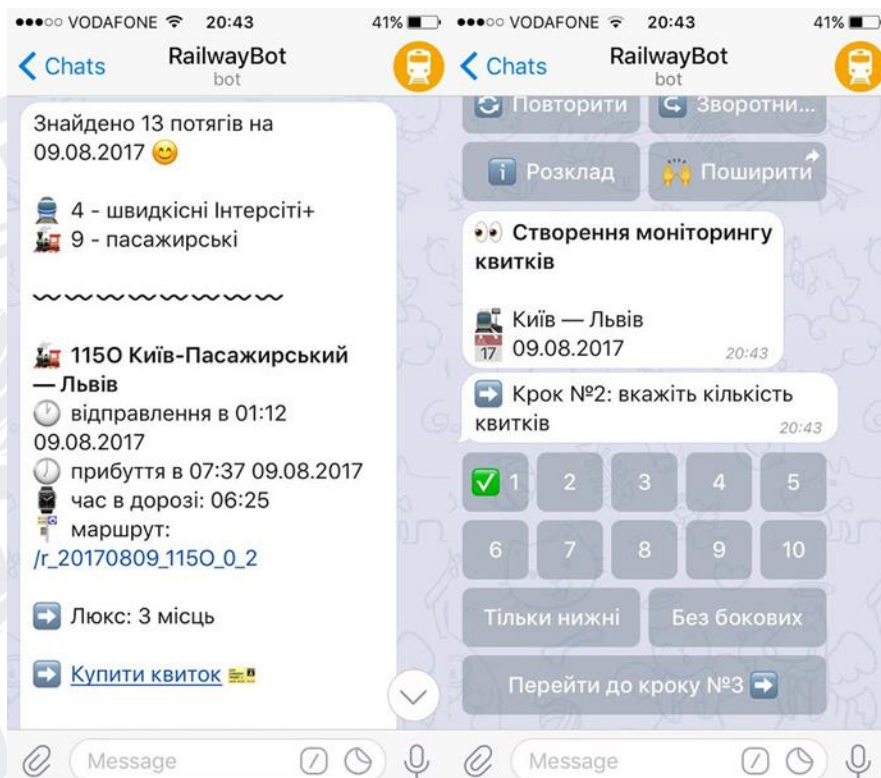


Рисунок 1.9 – RailwayBot

PrivatBankBot (рис. 1.10) - бот від ПриватБанку, який спрощує систему платежів. Це легкий спосіб відправити кошти, не відкриваючи додаток. Для надсилання коштів іншому користувачу, у нього теж має бути активований бот.

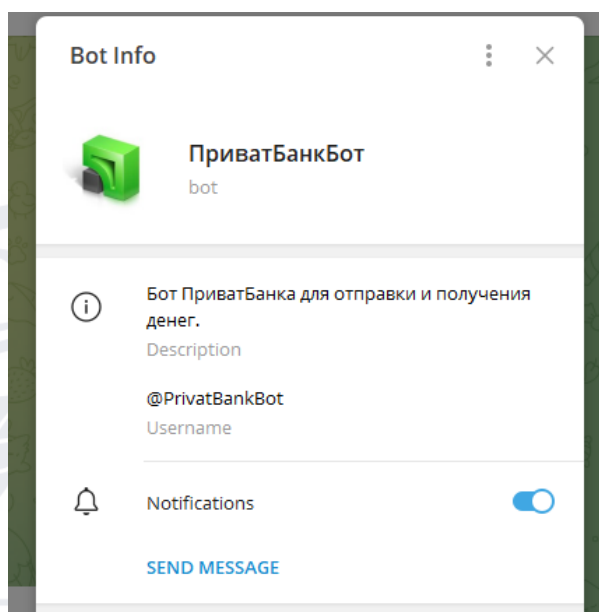


Рисунок 1.10 – PrivatBankBot

NovaPoshtaBot (рис. 1.11) - бот для моніторингу відправлень, їх статусу, пошуку найближчого відділення. “Індекс якості повітря в Києві” - надає інформацію по двох обраних районах. Наприклад, Позняки - дім, Поділ - робота. Окрім інформації про забруднення повітря доступні ще рекомендації чи потрібен додатковий захист. “Коронавірус в Україні” - бот, який надає актуально статистику щодо кількості заражених, тих, хто одужав, та тих, хто загинув від коронавірусу в Україні та Світі.

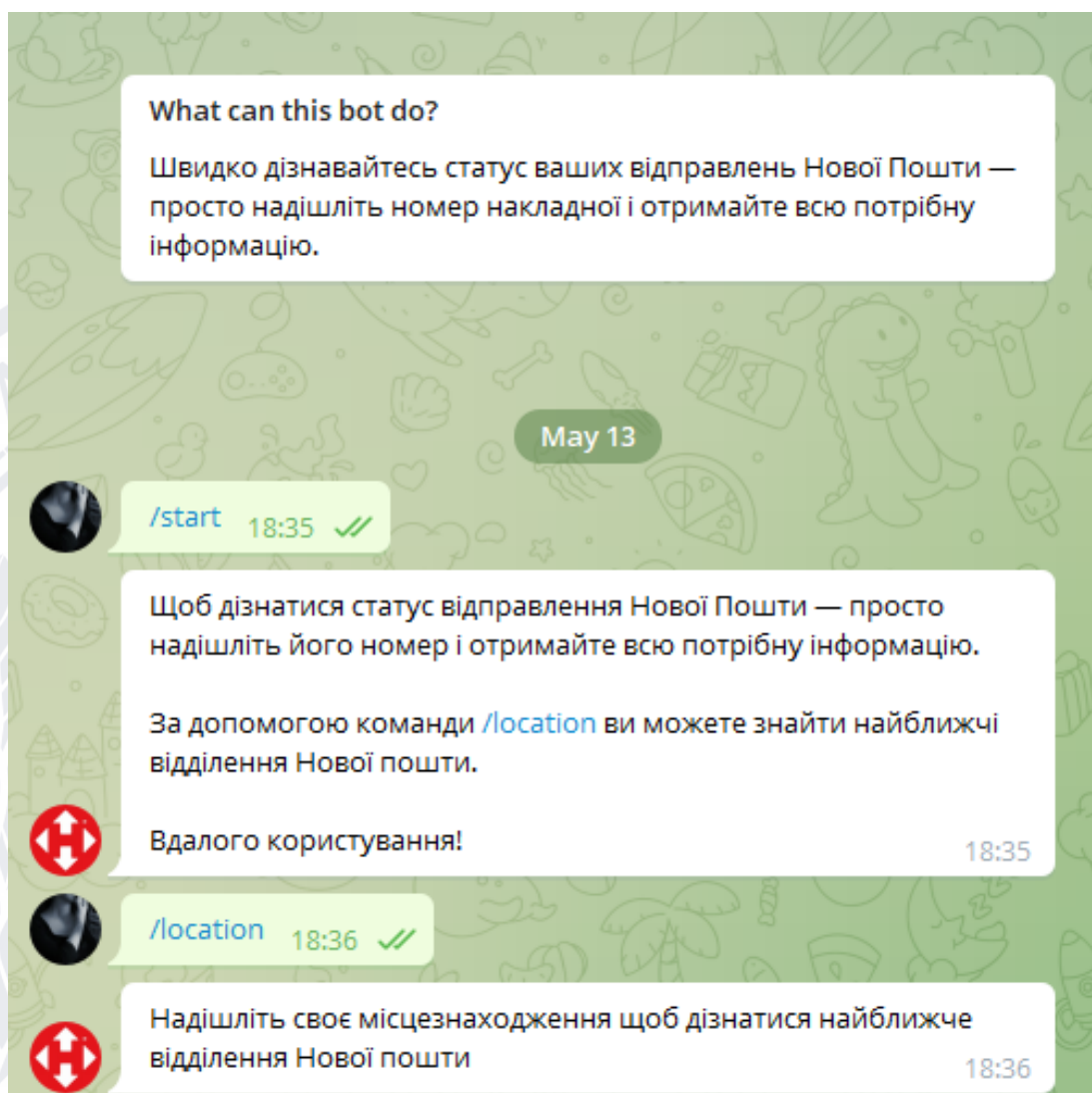


Рисунок 1.11 – NovaPoshtaBot

Більшість ресторанів та кафе мають власні боти для замовлення та доставки у популярних месенджерах. Проаналізувавши готові рішення, було взято до уваги, що швидкість боту є ключовою вимогою. Якщо час очікування перевищує декілька секунд, бот втрачає свою перевагу над пошуком у Google чи відкриттям додатку. AI боти більше нагадують живе спілкування, тому мають перевагу над скриптовими ботами. Водночас, гібридні боти є більш зручними у користуванні. Надання готових варіантів спрощує комунікацію, вимагає менше дій від клієнта.

1.4 Деталізація завдань розробки

Головною метою кваліфікаційної (дипломної) роботи було визначено створення чат-боту з розкладом занять для оповіщення студентів з використанням месенджера Telegram. Перед початком проектування і розробки телеграм бота потрібно визначити основні вимоги та описати їх. В попередньому розділі було проаналізовано предметну область та розглянуту існуючі аналоги, щоб переконатися в доцільності розробки.

Сучасні методи навчання та доставки інформація до кінцевого споживача постійно вимагають досконалостей у будь-яких сферах життя, починаючи від продуктових магазинів і закінчуючи великими державними підприємствами. Те, що було актуально 5 років тому, вже не користується популярністю, або втрачає аудиторію (рис. 1.12). Донесення актуальної інформації до студентів у стислі терміни, а саме зміна в розкладі, великий захід (добровільне здавання крові або день народження університету) та інше, вимагає негайного оповіщення. Єдиний спосіб доставки інформації до студента не завжди зручний, а часто забирає більше часу, ніж справді могло б. Тому одним із головних завдань у цій галузі є проблема актуалізація подачі інформації під сучасні платформи зв'язку в суспільстві.

Чим більше студентів вступають і навчаються, тим важливіше охопити всю аудиторію разом у короткий інтервал часу. Студенти, через свій вік, досить активно вивчають і використовують у своєму повсякденному житті всі новинки, зокрема месенджер Telegram. Як правило, не всі постійно моніторять сайти свого університету та додаткові офіційні джерела, необхідна централізація всіх необхідних та значимих даних. Отже, метою даної є створення проекту для інформаційної підтримки студентів на основі месенджера Telegram. Розроблена система має надавати зручний сервіс для студентів, які навчаються в університеті.

Виходячи з поставлених вище цілей, потрібна розробка Telegram-бота, який:

- а) зможе знати про будь-які зміни розкладу та про формат зберігання цих даних;
- б) буде зберігати отримані дані в спеціально створеній для цього базі даних, яка буде оптимізована для зберігання таких даних;
- в) буде певним чином зберігати дані кожного користувача та розклад занять кожної підгрупи;
- г) дозволить відстежувати поточний час та підлаштовувати сповіщення для користувачів бота;
- д) дозволить отримувати інформацію про розклад дзвінків;

А тепер деталізуємо кожен із наведених пунктів. Користувачу чат-бота потрібно буде знати про розклад занять на певний день(сьогодні/завтра) та на певну неділю.

Так як розклад дзвінків – це параметр, який змінюється дуже рідко, ми зможемо зберігати інформацію про дзвінки в базі даних для того, щоб не витрачати зайвий час для нового і нового пошуку. Так само ми зможемо зробити з інформацією про розклад занять, але кожен раз будемо підтягувати цю інформацію з сайту університету для актуалізації даних.

Telegram – чат – це можливість обміну миттєвими повідомленнями між учасниками конференції – бесіди. Telegram – канал є одностороннім способом отримання інформації, а саме надсилання її від імені адміністратора. Telegram – бот також є учасником конференції, що складається тільки з безпосередньо учасника та бота, який видає інформацію залежно від запитів користувача. Telegram – боти створюються з метою отримання інформації шляхом вибору певного шляху з дерева ланцюжків питання-відповідь. У представленому проекті об'єктом дослідження є Telegram – бот, тому що на даний момент є одним із

найпопулярніших способів отримання будь-якої інформації, починаючи від текстової та закінчуючи відеооб'єктами.

Для розробки будь-якого нового продукту та виведення його на ринок необхідно подолати безліч етапів, починаючи з розробки ідеї та закінчуючи підведенням результатів. Однак специфіка розробки може сильно відрізнятися в залежності від виду нового продукту, так етапи розробки нового промислового продукту можуть значно відрізнятися від стадій розвитку цифрового продукту.

У зв'язку з використанням великої кількості вхідних даних для проекту було прийнято рішення розподілу на етапи. Перший етап має на увазі визначення структури web - порталу, а саме шлях отримання інформації користувачем: лінійний тип руху від головної сторінки і далі, мережна структура, при якій користувачі можуть переміщатися від сторінки до сторінки без строго порядку, або ж ієрархічна структура, при якій на головній сторінці міститься коротка інформація про підрозділи. Другим етапом є вибір моделі побудови сайту, а саме вибір між: модель водоспаду, спіральна модель або MSF (Microsoft Solutions Framework). Заключним етапом визначаються технології розробки web-порталу, а саме технічна частина back-end і користувальницька froent-end.

Етапи аналізу ускладнені тим, що на даний момент недостатньо вивчені моменти взаємодії web - сайтів і Telegram - ботів, здатності студентів отримувати інформацію використовуючи даний месенджер, а також надмірність джерел інформації, що належать ДонНУ.

Верхній тиждень					
День тижня	Час навчання	Б18 д/122М			
		Дисципліна	Тип за洋яття	Аудиторія/ Платформа	Викладач
Понеділок	IV	Прикладні інформаційні технології	Лек (А+Б)	204-Кристал / MS Teams	Зелінська О.В., доцент
	V	Прикладні інформаційні технології	Лаб(А)	306-Кристал / MS Teams	Гончар В.М., асист.
Вівторок	II	Комп'ютерна графіка	Лек (А+Б)	314-Кристал / MS Teams	Ткаченко В.С., доцент
	IV	Прикладні інформаційні технології	Лаб(Б)	129-Кристал / MS Teams	Гончар В.М., асист.
Четвер	II	Дизайн інтерфейсів	Лек(В)	127-Кристал / MS Teams	Кожем'яко А.В., доцент
	III	Теорія програмування	Лек	205-Кристал / MS Teams	Потапова Н.А., доцент
	IV	Аналіз даних	Лек	307-Кристал / MS Teams	Штовба С.Д., професор
П'ятниця	II	Аналіз даних	Лаб(В)	306-Кристал / MS Teams	Штовба С.Д., професор
	III	Аналіз даних	Лаб(А)	306-Кристал / MS Teams	Штовба С.Д., професор
		Теорія програмування	Лаб(Б)	202-Кристал / MS Teams	Потапова Н.А., доцент
		Комп'ютерно-математичне моделювання в СІТ	Лек(В)	303-А-Кристал / MS Teams	Нескорородева Т.В., доцент
	IV	Аналіз даних	Лаб(Б)	306-Кристал / MS Teams	Штовба С.Д., професор
		Теорія програмування	Лаб(А)	202-Кристал / MS Teams	Потапова Н.А., доцент

Рисунок 1.12 – Приклад формату розкладу на сайту університету

Основні елементи інтерфейсу чат-ботів є універсальними, і властиві кожній програмі для обміну повідомленнями. Як зазначено у документації Телеграм API, спілкуватися з серверами Телеграм-бот може двома способами:

1. `getUpdates - pull`: бот постійно звертається до сервера Телеграми перевіряє чи є нові повідомлення;
2. `setWebhook - push`: у міру надходження нових повідомлень сервер Телеграм відправляє їх роботу.

Різницю цих двох способів демонструє рисунок 1.13

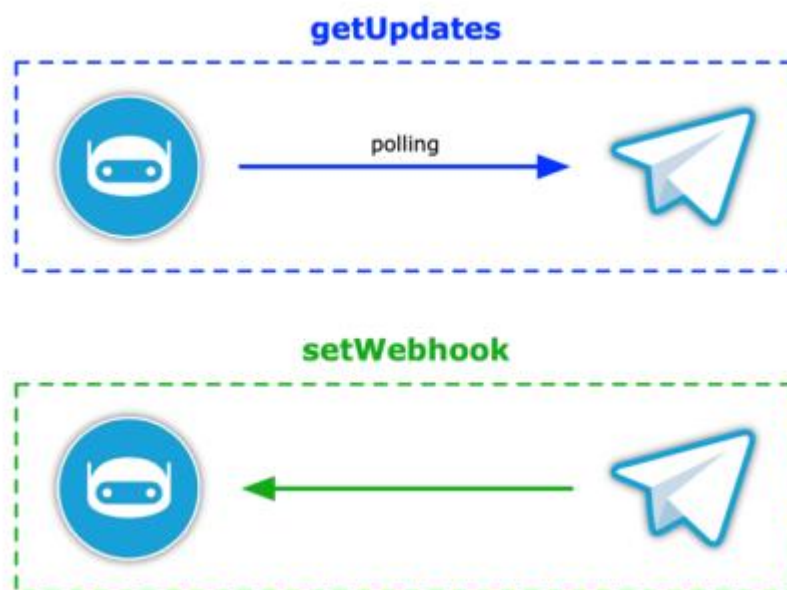


Рисунок 1.13 – Способи взаємодії з ботом

Зрозуміло, що другий спосіб (setWebhook) раціональніший для всіх учасників процесу. Однак у ньому є неявна складність: хтось повинен приймати повідомлення від телеграм на стороні бота, тобто потрібний веб-сервер або його еквівалент. Також необхідний SSL сертифікат, щоб зв'язок проходив захищеним каналом HTTPS. Це необхідна і головна частина взаємодії по вебхуку.

До складу програмного забезпечення входять веб-сервер та Телеграм-бот.

Основні ідеї, які були реалізовані в додатку, представлені далі:

1. Обробка повідомлень користувачів у режимі реального часу;
2. Миттєве уявлення можливих варіантів відповіді користувачеві;
3. Обробка рішення користувача в режимі реального часу та пересилання відповіді;

Веб-сервер, у свою чергу, складається з серверного програмного забезпечення телеграм бота. Тобто зв'язок з базою даних, виконання команд який запитує користувач.

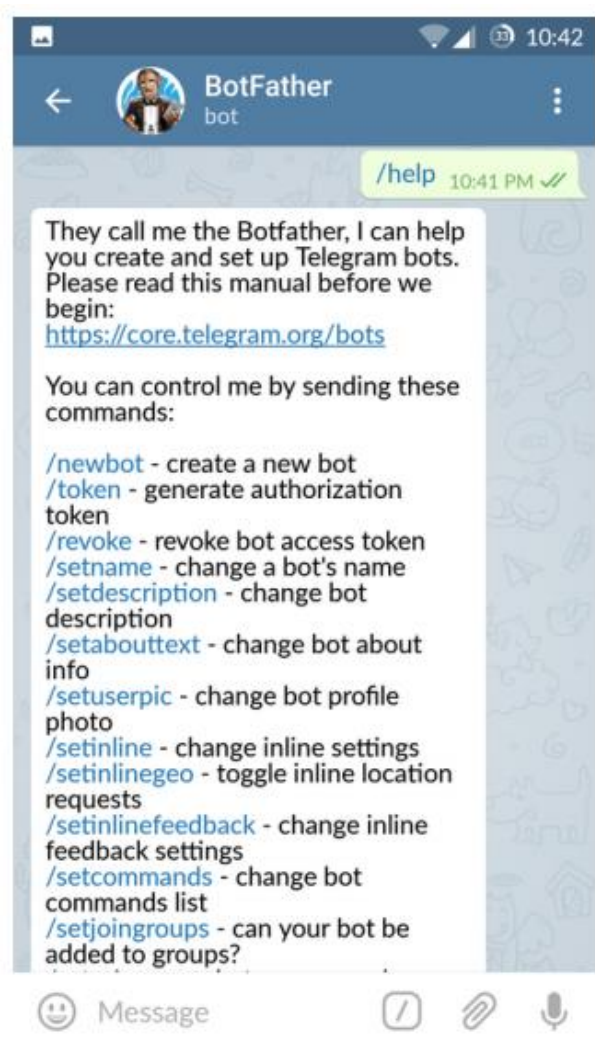


Рисунок 1.14 – Результат обробки команди «/help» ботом BotFather

Висновок до розділу 1

Перший розділ дипломної роботи присвячений проблемам проєкту, які вирішує розробка чат-боту. Детально розглянуто наявні види ботів, їх поширення та використання. Вивчено приклади успішних рішень та шляхи їх реалізації.

Також перший розділ акцентує увагу на алгоритмах, покладених в основу чат-ботів. Описано принципи, які є основними для створення чат-ботів. Проведено аналіз для вибору алгоритму за яким буде створюватись продукт. Розглянуто який результат буде досягнуто за допомогою обраних принципів.

РОЗДІЛ 2

Програмна реалізація бота

2.1 Обґрунтування інструментарію розробки

Для розробки програми необхідно спершу визначити певні засоби та інструменти які використовуватимуться.

До таких можна віднести:

- а) Мова програмування;
- б) Середовище розробки;
- в) Використані інструменти розробки;
- г) Система контролю версій;

У цьому розділі буде проаналізовано можливості вибору таких засобів та наведено приклад розробки бота за їх допомогою.

BotFather – найпростіший спосіб для реєстрації, налаштування та керування іншими telegram-ботами. Робота з ним проста і не потребує специфічних навичок. За допомогою BotFather можна зареєструвати необмежену кількість нових роботів. Єдиною умовою для реєстрації нового бота є його унікальний username.

BotFather стане відмінним рішенням для тих, хто не розуміється на програмуванні і не хоче довіряти створення та управління своїм ботом

стороннім людям або стороннім організаціям.

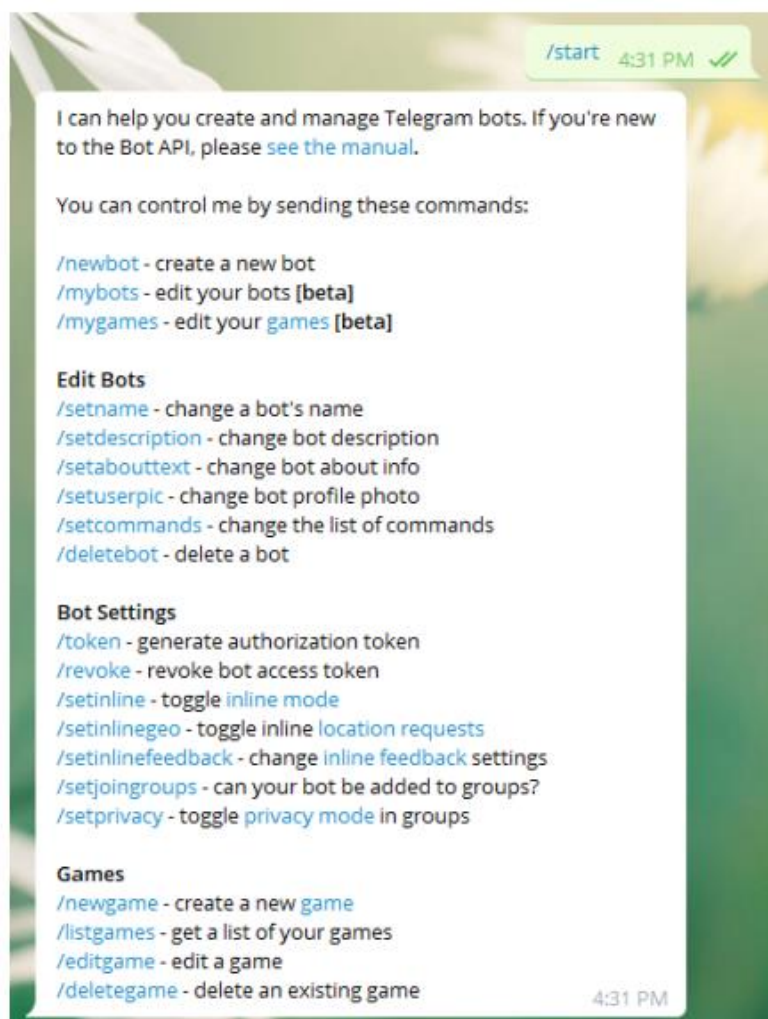


Рисунок 2.1 – Початок роботи із ботом BotFather та доступні команди

Взаємодія з BotFather здійснюється за допомогою простих команд. Наприклад, для того, щоб зареєструвати нового бота, достатньо відправити в чат команду /newbot і слідувати простим інструкціям:

1. Придумати ім'я бота, яке відображатиметься у чатах та контактах. Надалі його можна буде змінити. Тут все залежить тільки від вашої фантазії та вимог;
2. Придумати username – це складніше: ім'я має бути унікальним і закінчуватися на «bot». Допускаються літери латинського алфавіту,

цифри та символ підкреслення (приклад – MySuper_wow_bot). Загальна кількість символів не менше 5 та не більше 32;

3. Якщо все гаразд, то ми отримаємо повідомлення з токеном. Токен необхідний для роботи з Bot API за допомогою http-протоколу. Не можна передавати його іншим та бажано не втрачати. Хорошим рішенням буде скопіювати його: зберегти в текстовий файл і покласти в надійне, завжди доступне місце, наприклад, хмарне сховище.

Після реєстрації можна приступати до облагородження бота: встановити аватар, додати опис та інше.

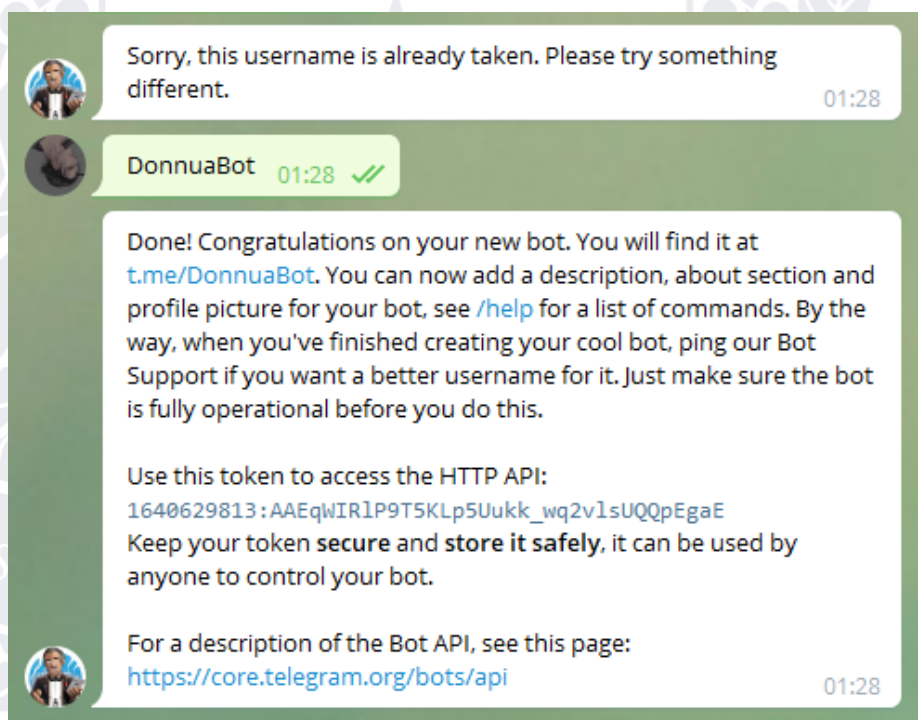


Рисунок 2.2 – Процес отримання токена за допомогою BotFather

Наприклад, якщо раптом вам перестала подобатися ім'я робота, то його можна змінити командою /setname.

Встановити аватарку можна командою /setuserpic, а змінити чи додати короткий опис - командою /setdescription. Зміна інформації у профілі виконується командою /setabouttext. На цьому кастомізацію зовнішнього

вигляду робота можна вважати закінченою, і можна зайнятися його функціональними можливостями.

Щоб встановити список команд, введіть `/setcommands` у вікні чату.

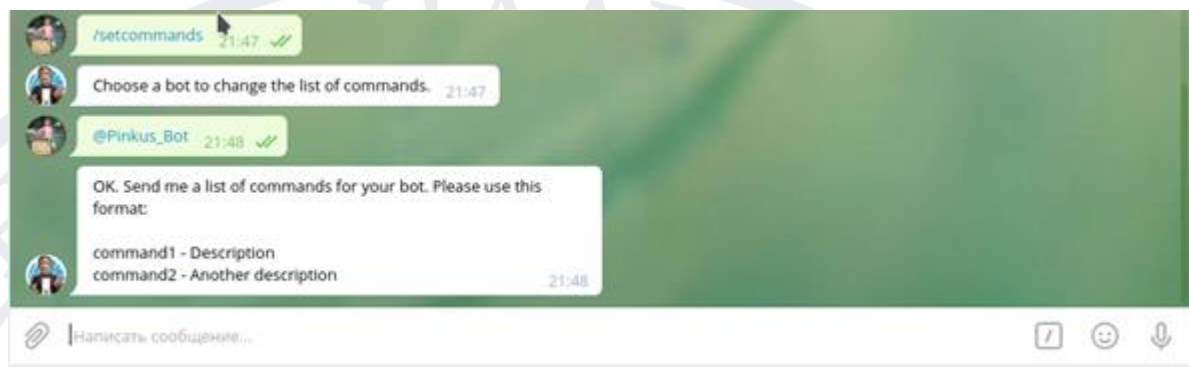


Рисунок 2.3 Встановлення команд

Команди вводяться без сліша, у форматі "command1 - опис команди"

Ось приклади написання основних команд та їх розшифровка:

`/newbot` – зареєструвати нового робота

`/mybots` – редагувати своїх роботів [бета]

Редагування

`/setname` - Змінити ім'я бота;

`/setdescription` - Змінити опис бота;

`/setabouttext` - змінити інформацію про бот;

`/setuserpic` - Змінити фотографію профілю бота;

`/setcommands` – змінити список команд;

`/deletebot` – видалити робота.

Налаштування бота

`/token` - генерувати токен авторизації;

`/revoke` - відкликати токен доступу до робота;

`/setinline` – увімкнути вбудований режим (дозволяє звертатися безпосередньо до бота з будь-якого каналу, групи або чату, написавши його ім'я у полі надсилання повідомлень) (<https://core.telegram.org/bots/inline>);

`/setinlinegeo` – перемикає запити розташування під час використання бота у вбудованому режимі. (<https://core.telegram.org/bots/inline#location-based-results>);

`/setinlinefeedback` – змінити налаштування зворотного зв'язку (збір статистики команд, що найчастіше відправляються) (<https://core.telegram.org/bots/inline#collecting-feedback>);

`/setjoingroups` – визначає можливість додавання вашого робота в групи;

`/setprivacy` – переключити режим конфіденційності у групах (<https://core.telegram.org/bots#privacy-mode>).

Ігри

`/mygames` – редагуйте свої ігри (<https://core.telegram.org/bots/games>) [бета];

`/newgame` – створити нову гру (<https://core.telegram.org/bots/games>);

`/listgames` - Отримати список ваших ігор;

`/editgame` – редагувати гру;

`/deletegame` - Видалити існуючу гру.

Таким чином можна додати будь-які необхідні команди та завершити налаштування.

Мільйони вибирають Telegram за його швидкість. Щоб залишатися конкурентоспроможним у цьому середовищі, ваш бот також має бути швидким. Щоб допомогти розробникам підтримувати свої роботи у формі, @BotFather надсилає сповіщення про статус, якщо бачить, що щось не так.

Telegram перевірятиме кількість відповідей та коефіцієнт конверсії запитів/відповідей для популярних ботів (300 запитів на хвилину: це значення може змінитися в майбутньому). Якщо ми отримаємо аномально низькі свідчення, ви отримаєте повідомлення від @BotFather.

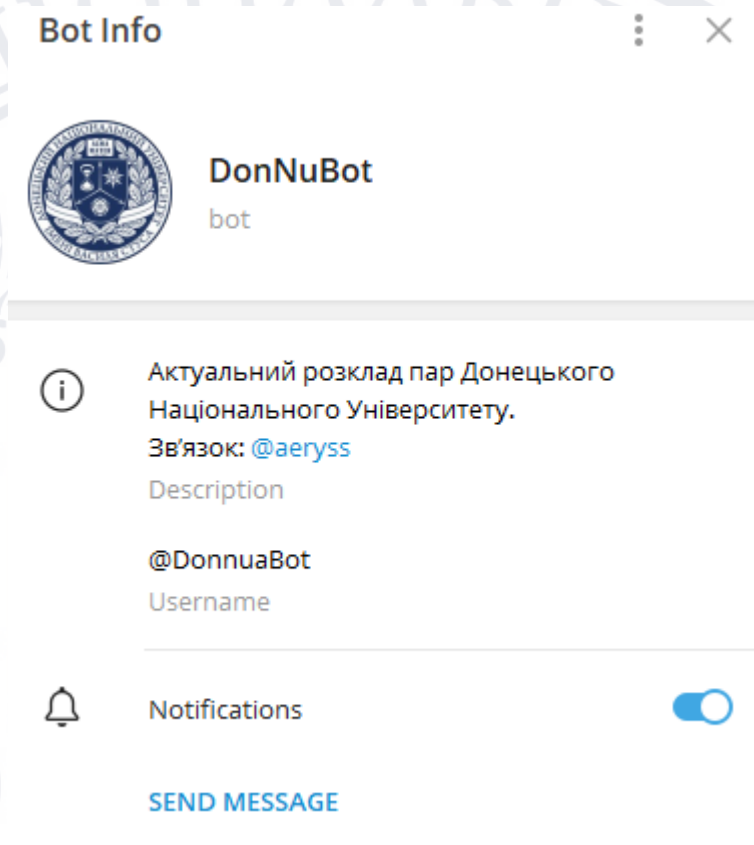


Рисунок 2.3.1 – Вікно інформації про бота

Перед тим як обирати інструменти розробки, необхідно ознайомитися із прикладним програмним інтерфейсом (англ. Application Programming Interface, API), який надає месенджер для розробників. На даний момент є два основні інструменти API, за допомогою яких можна використовувати сервіси Telegram — Telegram Bot API і Telegram API. Перший служить для розробки чат-ботів, другий дозволяє робити повністю кастомні телеграм-клієнти. Розробникам також доступна відкрита бібліотека TDLib (Telegram Database Library), за допомогою якої можна створювати свою версію месенджера з унікальними опціями (наприклад, Telegram X, побудований саме на TDLib). Telegram Bot API

є надбудовою над Telegram API, тому користуватися Bot API можна без знань про механізм використовуваного протоколу MTProto.

Для його роботи задіяний проміжний сервер з інтерфейсом HTTPS, який шифрує трафік і забезпечує зв'язок з Telegram API. Bot API дозволяє легко створювати програми, які використовують інтерфейс Telegram для виконання коду на локальному сервері. Користувачі можуть взаємодіяти з ботами, надсилаючи їм повідомлення, команди та вбудовані запити.

Принцип роботи будь-якого робота полягає в тому, що він перманентно надсилає запити на сервер і регулярно отримує оновлення. Отримувати їх можна двома способами. По-перше, можна використовувати вебхуки, коли сервер здійснює зворотний дзвінок на вказаний URL. По-друге, можна просто «закидати» запитами Telegram, отримуючи постійні відповіді.

Зверніть увагу - отримувати повідомлення про нові повідомлення в роботі та інших подіях ви можете всього один раз. Тому, якщо дані чату видаються вам дуже важливими, доведеться самостійно зберігати список чатів та історію старих повідомлень. Якщо ви випадково зітре/втратите цю інформацію, то ви її ніяк не відновите.

На відміну від Bot API, де отримувати оновлення можна лише один раз, у Telegram API це обмеження можна обійти, якщо використовувати кілька клієнтів. У такому випадку робот отримуватиме всі оновлення на кожному із запущених клієнтів. Також у Bot API немає можливості розсилки повідомлень усім користувачам одночасно.

2.2 Вибір засобів розробки чат-боту

Однією з унікальних переваг JavaScript є його поширеність.

Цю мову можна зустріти буквально всюди. Він підтримується на всіх операційних системах, у всіх видах браузерів, і настільних комп'ютерах, і мобільних пристроях. Дуже важливо ще й те, що JavaScript-програми працюють без встановлення їх на комп'ютери користувачів. Насправді, вже складно згадати не такі вже й давні часи, коли компанії, розгортаючи клієнт-серверні програми у своїх внутрішніх мережах, витрачали тижні, борючись із проблемами налаштування цих додатків та з несподіваними помилками. Зіткнувшись з подібними кошмарами, ви можете оцінити привабливість JavaScript.

JavaScript, завдяки його поширеності, не обов'язково бути найкращою мовою програмування. Йому лише треба бути досить гарною мовою.

Схожі механізми працюють і тоді, коли йдеться про вивчення програмування. Спочатку проблема поширеності мови не здається особливо важливою для того, чия мета полягає в тому, щоб вивчити перевірені часом практичні прийоми розробки програм. Але безліч тих самих проблем актуальні, як і для професійного програмування, у навчанні програмування. Якщо ви хочете, щоб тим, що ви створили, міг би скористатися будь-хто, то JavaScript - це єдина альтернатива.

Припустимо, ви розробляєте чудову програму, яка, ставлячи користувачеві питання, визначає його особистісний профіль. Налаштувати середовище розробки досить легко. Але що станеться, коли ви захочете надіслати свою програму товаришу? Чи знадобиться йому встановлювати середовище часу виконання або відтворювати те саме середовище розробки, що й у вас? Чи потрібно буде «підкрутити» налаштування безпеки його комп'ютера, перш ніж він зможе завантажити та встановити вашу програму? Може, ваш товариш використовує операційну систему, яка просто не підтримує вашу програму, або мобільну платформу, на яку, якщо не вдаватися до деталей, можна

встановлювати лише професійні програми, що розповсюджуються через iTunes або Google Play? Якщо ж ви, для розробки програми, використовуєте JavaScript, а для її розповсюдження - простий веб-сайт (наприклад, GitHub дозволяє безкоштовно створювати подібні сайти), всі ці проблеми просто зникнуть.

Наразі ми обговорювали сильні сторони JavaScript. Але, звичайно, говорячи про JavaScript як мову програмування для початківців, не можна не згадати і про його недоліки. (рис. 2.4).

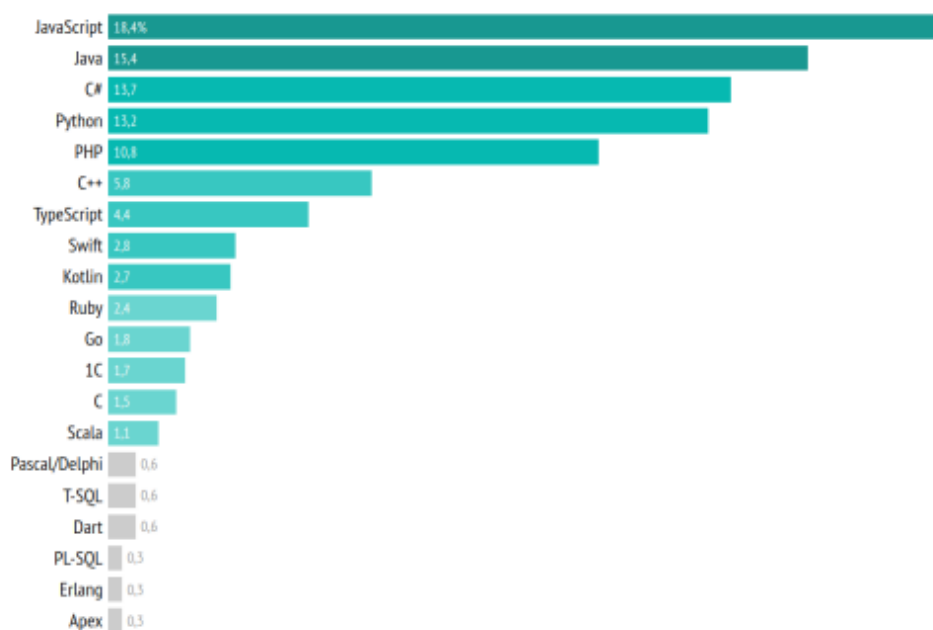


Рисунок 2.4 – Рейтинг мов програмування у комерційних проектах, %

Найважливішою концепцією, яку програмісти-початківці освоюють дуже рано, є ідея змінних, контейнерів, які зберігають інформацію під час роботи програми. Проблема JavaScript полягає в тому, що ця мова надто вільно та неакуратно поводить себе зі змінними. Він дозволяє робити те, що не виглядає правильним і ігнорує очевидні нестиковки. Його недбалість здатна перетворювати незначні помилки на катастрофи, що порушують роботу програм.

Ось приклад коду, який приречений на провал:

```
var myNumber = 100;  
myNumber = myNumbr + 1;
```

Рисунок 2.5 – Приклад коду

У другому рядку, насправді, використовуються дві змінні: змінна `myNumber`, яка була оголошена в першому рядку, і якась фіктивна змінна `myNumbr`. Друга змінна - це результат помилки, але JavaScript повідомить нам про це тільки під час виконання програми, видавши повідомлення про помилку, яке виглядає як `ReferenceError: myNumbr is not defined`. Те саме відбувається і в тому випадку, якщо при записі імені змінної переплутали великі та маленькі літери (наприклад, змінна з `myNumber` випадково перетворилася на `MyNumber`).

Проблема, але вже іншого характеру, може виникнути й у тому випадку, якщо ви, вважаючи, що якась змінна зберігає число, додаєте до неї інше число, а ця змінна насправді містила рядок або щось інше. JavaScript у такому разі не повідомить про помилку, але результат обчислень буде зовсім не таким, на який можна очікувати від складання двох чисел. Кожен JavaScript-розробник може згадати неприємну історію, коли плутанина з типами викликала дивні помилки в програмах.

Досить складно вчитися програмувати, і, водночас, турбуватися про те, як мова програмування, що вивчається, може вам нашкодити. Але, на щастя, проблеми безпечної роботи з типами даних можна легко вирішити за допомогою відповідних інструментів розробки.

Далі розглянемо рейтинг популярності за особистими уподобаннями, але для комерційних проектів, що представлено малюнку 2.6.

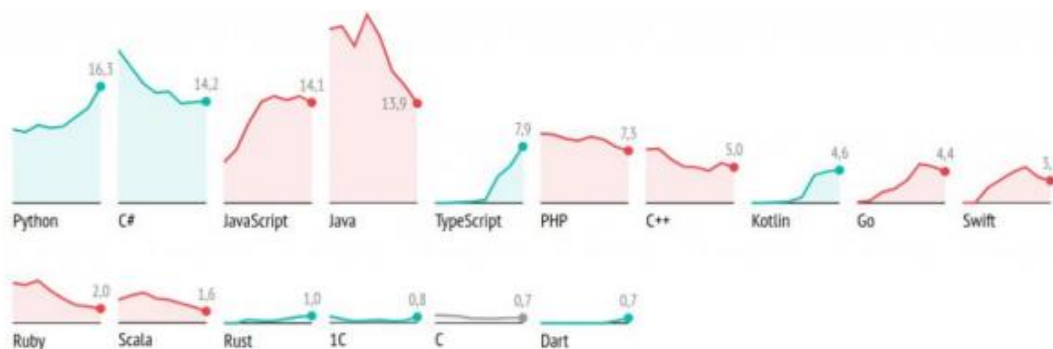


Рисунок 2.6 – Рейтинг популярності мов за особистими уподобаннями, %

Найвідоміше застосування JavaScript - це, звичайно, веб-розробка. Найчастіше до JS вдаються, коли потрібно додати динамічності на сайт або зробити браузерну програму, але за бажання на ньому можна написати і бекенд.

JavaScript відрізняється:

- прототипним стилем програмування (замість класів та успадкування - прототипи та клонування);
- слабкою динамічною типізацією;
- функціями як об'єктами першого класу (тобто їх можна створювати безпосередньо під час виконання програм).

Отже, у яких областях знайшов себе JavaScript.

Підказки, рухомі картинки, падаючі сніжинки та інші анімашки - за всім цим можна йти до JS. JS-код вбудований у веб-сторінку, і коли користувач відкриває її, скрипт виконується у браузері.

Без JavaScript майже неможливо створити повноцінний сайт. Додайте до JS знання HTML, CSS, і ви отримаєте непоганий набір навичок початківця фронтенд-розробника. Для повноти картини варто освоїти пару популярних фреймворків: наприклад, React, Svelte, Vue.js

Так, на JavaScript можна спокійно написати веб-сервер. Але якщо в браузері у JS практично немає конкурентів, то у світі бекенда йому доводиться змагатися з іншими мовами: PHP, Python та іншими. Головна перевага JavaScript у цій битві

JavaScript дістався і до настільних програм. Так, GitHub об'єднав згаданий вище Node.js, двигун рендерингу Chromium і розробив фреймворк Electron, на якому можна писати кросплатформні десктопні проекти. Серед прикладів – GitHub Desktop, Visual Studio Code, Skype, WordPress Desktop. Electron має аналог — NW.js, який в основному використовується для створення настільних версій сайтів та ігор.— безліч корисних бібліотек та розширень мови можна легко використовувати у своїх проектах завдяки гранично уніфікованим механізмом імпорту та програмним інтерфейсів;

TypeScript (TS, TScript або тайпскрипт) — це мова програмування для веб-розробки, що базується на JavaScript. Робить код зрозумілішим і надійнішим, додає статичну типізацію (змінні прив'язані до конкретних типів даних), а також може бути скомпільований у JavaScript. TypeScript використовують фронтенд- та бекенд-розробники.

TypeScript став однією з необхідних навичок для сучасного веб-розробника. У 2019 він увійшов до ТОП-10 найбільш використовуваних мов GitHub, його підтримку повністю додали в Create react app, і можна знайти масу інших свідчень зростання його популярності. У той же час такі мови, як Java і C, продовжують здавати позиції.

Коли ми говоримо про переваги TypeScript, на думку зазвичай спадає наступний список:

- TypeScript підтримує статичну типізацію
- TypeScript робить код простіше для читання та розуміння
- TypeScript допомагає уникнути безлічі болючих багів, які зазвичай роблять розробники, завдяки перевірці типів у коді
- TypeScript заохочує розробників слідувати найкращим ООП практикам
- Як наслідок перерахованого вище — TypeScript заощаджує час розробників

Цікаво, що всі пункти цього списку беруться на віру без критичного погляду. Сьогодні я пропоную Вам розглянути ці пункти уважніше і з'ясувати, чи вони настільки корисні для нас.

Оскільки головним завданням TypeScript є зменшення кількості багів, давайте визначимося, чому так важливо? Відповідь проста: чим менше багів — тим менше дорогого часу розробників і тестувальників ми витратимо, отже — отримаємо продукт, що стоїть, за менші гроші і він почне приносити дохід раніше.

Пам'ятаючи про це, давайте з'ясуємо, як TypeScript може допомогти нам покращити нашу продуктивність та ефективність.

Дані JS зберігаються в змінних і мають будь-який тип: рядок, число, логічне значення. JavaScript несувора, динамічна типізація. Це означає, що одній змінній можна надати спочатку числове значення, а потім, наприклад, текстове.

TypeScript додає до мови строгу типізацію. Кожній змінній при створенні надається певний тип (type) - стандартний або створений самим розробником. Створити тип можна в межах можливостей мови: наприклад, число від 1 до 31 для запису дня на місяць або масив двох елементів для запису координат.

Що таке «змінній надається тип»? Це означає, що протягом усього скрипту вона зможе набувати лише значень у межах цього типу. Змінна, задана як число, буде лише числом. Якщо спробувати надати їй рядкове значення, програма видасть помилку.

В основі розробки серверу чат-боту було покладено використання Telegram Bot API, що дозволяє легко і швидко створювати інтегровані чатботи в месенджері Telegram. Це API можна використовувати на основі HTTP запитів на сервер телеграму. Чат-бот в свою чергу створюється за допомогою іншого боту

від месенджера, який називається BotFather, що і є батьківським класом в системі телеграму для інших ботів.

За допомогою запитів можна легко отримувати інформацію щодо оновлення боту (отримання нових повідомлень, змінення повідомлень, відправлення різних даних мультимедіа тощо).

На основі використання Telegram Bot API було побудовано асинхронний фреймворк aiogram, що був використаний під час розробки чат-боту. Фреймворк aiogram було обрано за його простоту у використанні та за те, що він повністю асинхронний. Побудований за допомогою Typescript стандартної бібліотеки asyncio та асинхронного вебфреймворку aiohttp.

Найголовнішою особливістю є те, що він може витримувати дуже велике навантаження при використанні невеликих машинних ресурсів (фізичних або віртуальних), має вбудований скінчений автомат станів для проектування більш складних чат-ботів, підтримує особливість Telegram API вебхук (можливість робити запити у відповідь на оновлення).

Для написання коду нам потрібний текстовий редактор – це незалежна комп'ютерна програма або компонент програмного комплексу, призначений для створення та зміни текстових даних та текстових файлів [21]. Редактор відіграє важливу роль у кожній галузі роботи розробника. Код написаний, налагоджений та виконаний з використанням текстового редактора.

WebStorm - це інтегроване середовище для розробки на JavaScript та пов'язаних з ним технологіях. Як і інші IDE JetBrains, WebStorm дозволяє автоматизувати рутинну роботу та легко справлятися зі складними завданнями, роблячи розробку більш цікавою.

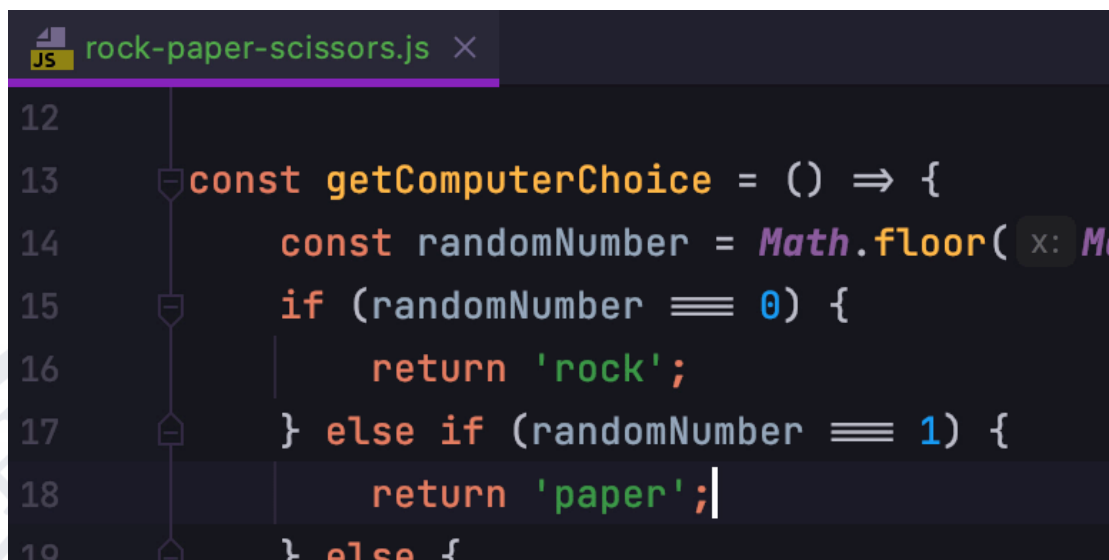


Рисунок 2.7 – Інтерфейс Webstorm

"Голий" WebStorm (без плагінів) не представляє можливостей, описаних нижче, однак, за замовчуванням має більше 100 встановлених доповнень, які забезпечують підтримку та зручну розробку, використовуючи різні JS-фреймворки, бібліотеки, різні нововведення CSS/HTML і т.д. . Це лише мала частина того, з чим WebStorm здатний працювати із коробки:

Модифікація файлів .css, .html, .js з одночасним переглядом результатів (англ. Live Edit, у деяких джерелах ця функціональність називається «редагування файлів на льоту» або «в реальному часі» або «без перезавантаження сторінки»)

- Підтримка HTML5
- Підтримка JSDoc[4]
- Підтримка Node.js
- Підтримка React, JSX
- Можливості Zen Coding та Emmet
- Налаштування коду на JavaScript

Віддалене розгортання за протоколами FTP, SFTP, на монтованих мережних дисках тощо з можливістю автоматичної синхронізації

Інтеграція із системами керування версіями: Subversion, Git, GitHub, Perforce, Mercurial, CVS підтримуються з коробки з можливістю побудови списку змін та відкладених змін.

2.3 Підключення бібліотек

Для того, щоб зберігати наші дані з розкладом занять та іншим, була обрана база даних SQLite.

SQLite – це компактна інтегрована реляційна база даних. Вихідний код бібліотеки було передано на громадське надбання. Це чисто реляційна база даних. Типи даних, що підтримуються:

- NULL: значення NULL.
- INTEGER: ціле число зі знаком, що зберігається в 1, 2, 3, 4, 6 або 8 байтів.
- REAL: число з плаваючою комою, що зберігається в 8-байтовому формат IEEE.
- TEXT: текстовий рядок, закодований UTF-8, UTF-16BE або UTF-16LE.
- BLOB: тип даних, що зберігаються в тій самій формі, в якій вони були отримані. Переваги:
 - Файлова: вся база даних зберігається в одному файлі, що полегшує переміщення.
 - Стандартизована: SQLite використовує SQL.

– Відмінно підходить для розробки та навіть тестування: на етапі розробки більшості потрібно масштабоване рішення. SQLite з його багатим набором функцій може надати більше чим достатню функціональність, а також бути достатньо простим для роботи з одним файлом та пов'язаною бібліотекою.

Недоліки:

– Відсутність управління користувача: просунуті БД надають користувачам можливість керувати зв'язками в 33 таблицях відповідно до привілеїв, але у SQLite такої функції ні.

– Неможливість додаткового налаштування: SQLite не можна зробити більш продуктивною. У цьому підрозділі була представлена база даних, необхідна для роботи чат-бота, було розглянуто її поняття, переваги та недоліки.

Висновок до розділу 2

У цьому розділі було розглянуто ті поняття чат-бота, які ми будемо використовувати в розробці, показано функції, наведена класифікація всіх існуючих на даний момент чат-ботів. Далі було обґрунтовано вибір програмного забезпечення для реалізації чат-бота, за результатами якого були обрані такі засоби: мова програмування JavaScript та TypeScript, текстовий редактор Webstorm, месенджер Telegram та база даних SQL Lite.

РОЗДІЛ 3

Практичне застосування на рівні ФІПТ ДонНУ імені Василя Стуса

3.1 Інструкції користувача та адміністратора

Для більш детального ознайомлення з результатом практичної частини роботи, розглянемо декілька сценаріїв взаємодії користувача з чат-ботом.

В описі чат-боту поданий короткий опис його можливостей. Для зображення боту використано дизайн, розроблений для факультету інформаційних технологій ДонНУ. Знайти бот у пошуці в месенджері Telegram можна ввівши його назву: «DONNUABOT» (рис. 3.1).

Для початку роботи з ботом, користувач повинен натиснути на «Send message».

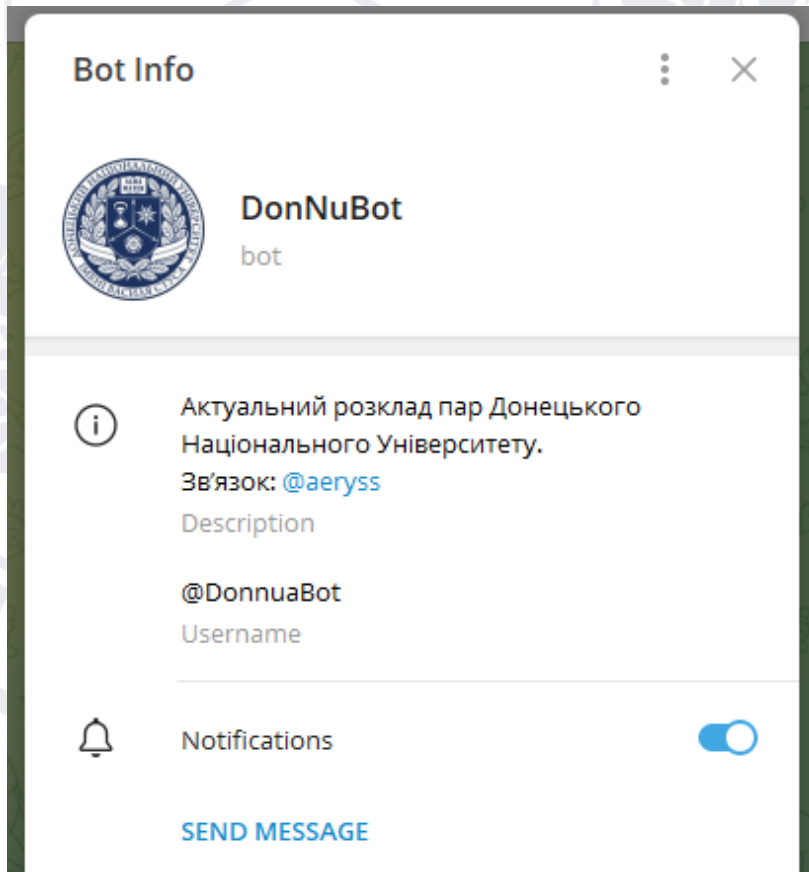


Рисунок 3.1 - Опис чат-бота

Відповідь говорить про те, що бот підлаштовується під комунікацію з людиною, надаючи відчуття живого спілкування. Перша відповідь (див. перше повідомлення від боту на рис.1.10) є загальною і несе в собі інформацію про те, що бот досі на стадії навчання та розробки. Друга відповідь – це власне відповідь на питання користувача. Відразу надається можливість оцінити наскільки відповідь була корисною. Третє повідомлення – це пояснення до оцінки і можливість зв'язатись із експертом.

Можливість отримати зворотний зв'язок від людини, а не лише від боту, є дуже важливою функцією. Оскільки, не усі питання можуть бути легко розв'язані за допомогою пошуку або навчання.

Після приєднання до чату користувачу доступна тільки одна команда /start. Відправивши цю команду, у відповідь отримуємо подальшу інструкцію для використання боту (рис.3.2).

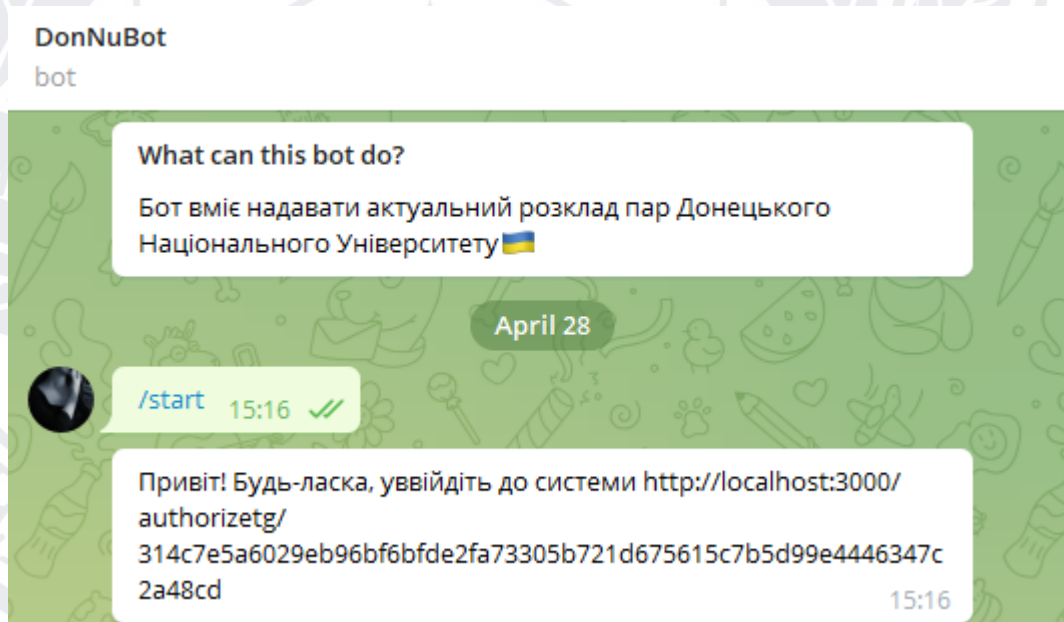


Рисунок 3.2 – Перша взаємодія з чат-ботом

Отримуємо посилання для авторизації студента для подальшого використання боту. Для цього переходимо за посиланням та авторизуємось за допомогою корпоративного аккаунту ДонНУ (рис. 3.3)

Авторизуйтеся використовуючи корпоративний аккаунт ДонНУ

Увійти

Рисунок 3.3 – Авторизація

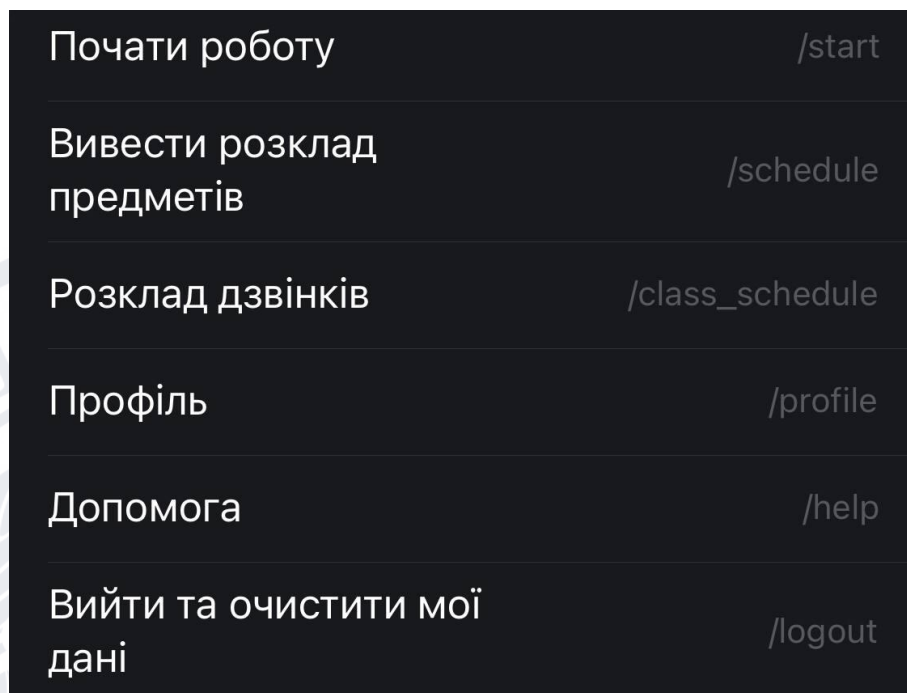
Після вдалого авторизування користувач має змогу бачити свій ПІБ та надпис «Успішна авторизація» (рис. 3.4). Далі можемо переходити назад в Телеграм та продовжувати роботу з ботом.

Успішно авторизовано користувача
Подольан Михайло Михайлович. [Перейти до бота](#)

Вийти

Рисунок 3.4 – Успішна авторизація

Користуватись ботом досить просто, тому що були створені всі потрібні кнопки для переходу до наступного меню (рис. 3.5) без введення тексту користувачем вручну.



Почати роботу	/start
Вивести розклад предметів	/schedule
Розклад дзвінків	/class_schedule
Профіль	/profile
Допомога	/help
Вийти та очистити мої дані	/logout

Рисунок 3.5 – Головне меню

Розклад на сьогодні або завтра доступний двостороннім способом. Розглянемо перший варіант - через поступовий ввід даних: необхідно вибрати факультет, курс та групу. Велику роль грає те, що атрибут «текст» об'єкта «повідомлення» - це текстове значення вибору пункту меню.

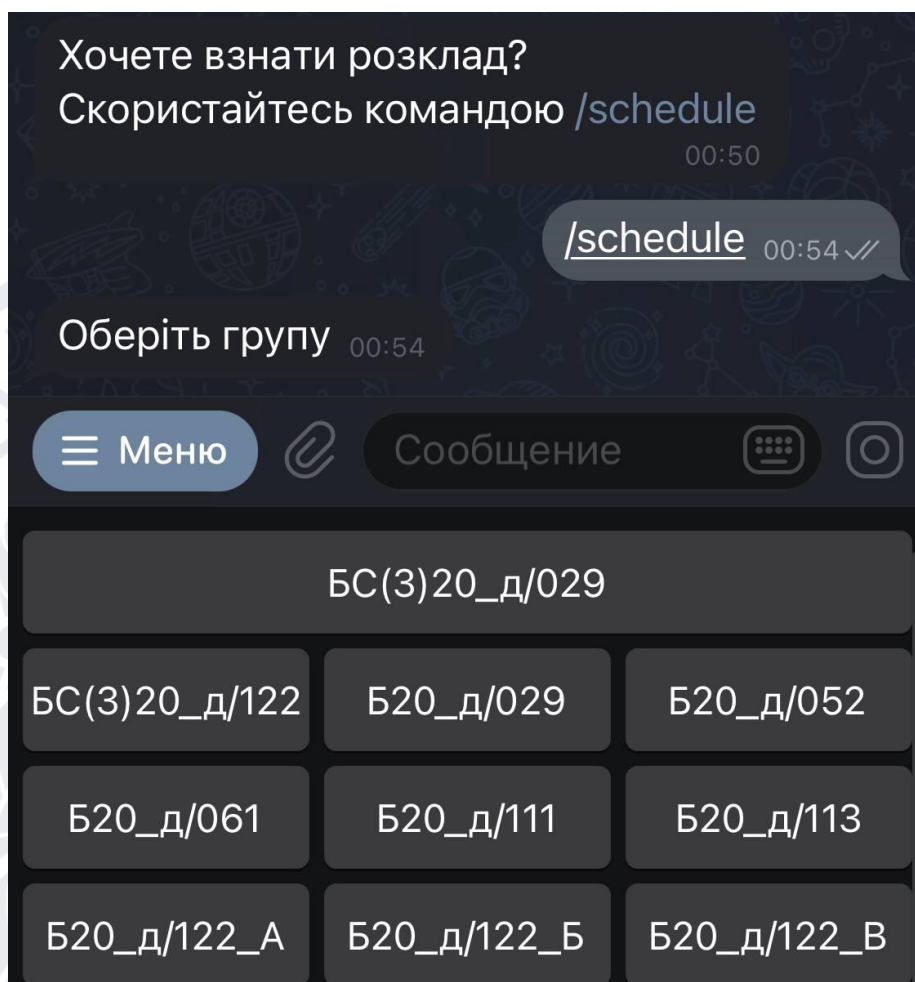


Рисунок 3.6 Вибір своєї групи

Вибір курсу та групи, скрипт якого кілька разів відрізняється від попередніх, тому що відображення списку групи на екрані відбувається після запрошення до таблиць із повним виписуванням, де виходить весь список груп за заданим користувачем факультету та курсу.

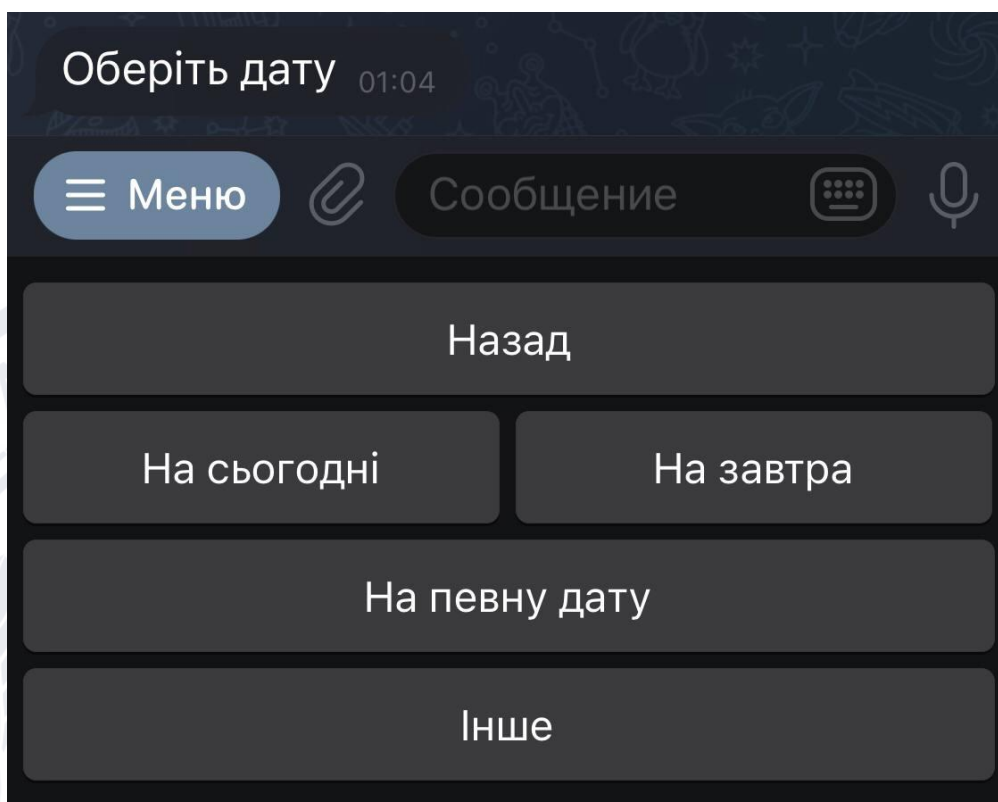


Рисунок 3.7 Вибір дня з розкладом

Кнопка «На сьогодні» або «На завтра», код якої містить актуальний набір даних (сьогодні, завтра, індекс сьогоднішнього дня), перевірка вводу від користувача та самостійне відображення розкладу (рис. 3.8).

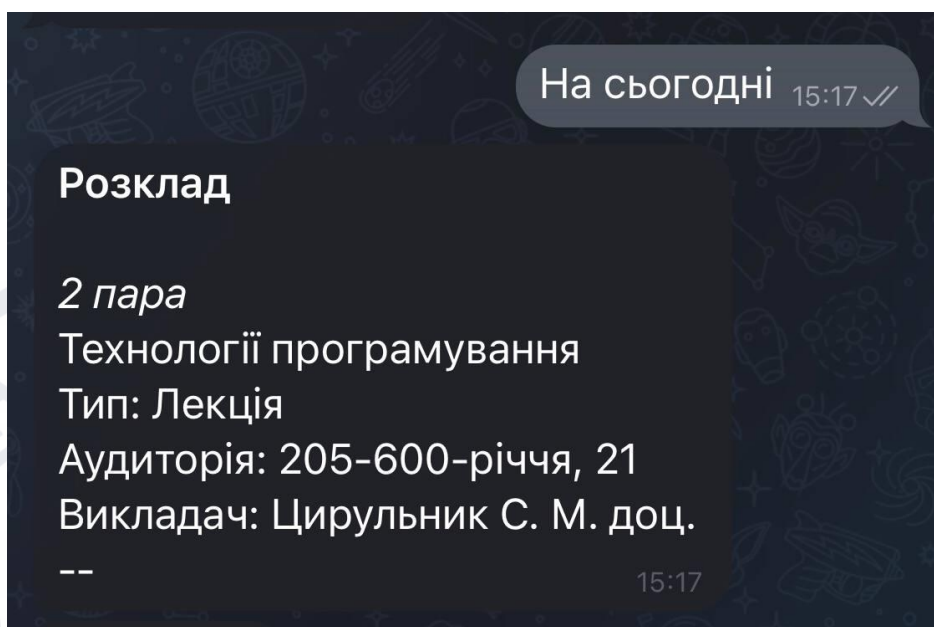


Рисунок 3.8 Отримання розкладу на сьогодні

Кнопка «Отримати розклад» - користувач присилає базовий набір даних.

Кнопки «Розклад дзвінків» (рис.3.9), «Профіль» (рис.3.10), та «Вивести розклад» являються інформативними, тобто з їх використанням можна отримати необхідну для користувача інформацію.

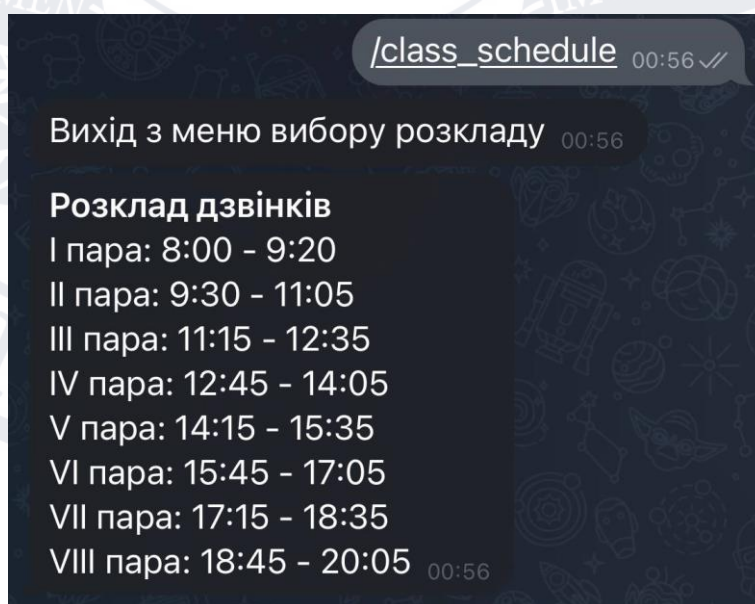


Рисунок 3.9 Приклад отримання розкладу дзвінків



Рисунок 3.10 Приклад отримання інформації про профіль

Функція оформлення доступу до розкладу груп на два дні з першим меню, заснованим на одному вступі своєї групи в таблицю, якщо необхідно вибрати бажану групу та підтвердити підписку, яка доступна в самому останньому меню.

Також користувачу доступні ще дві функції:

- 1) Підписка та відписка від розкладу на певну групу (рис. 3.11) та вибір часу отримання розкладу (рис. 3.12)
- 2) Отримання допомоги адміністратора в разі якихось помилок в боті

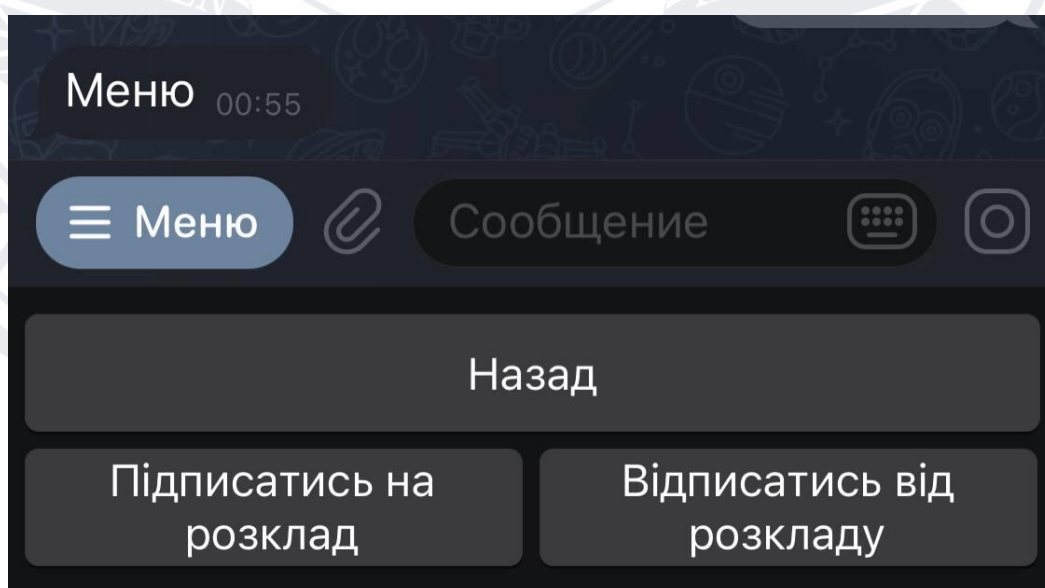


Рисунок 3.11 Підписка на групу

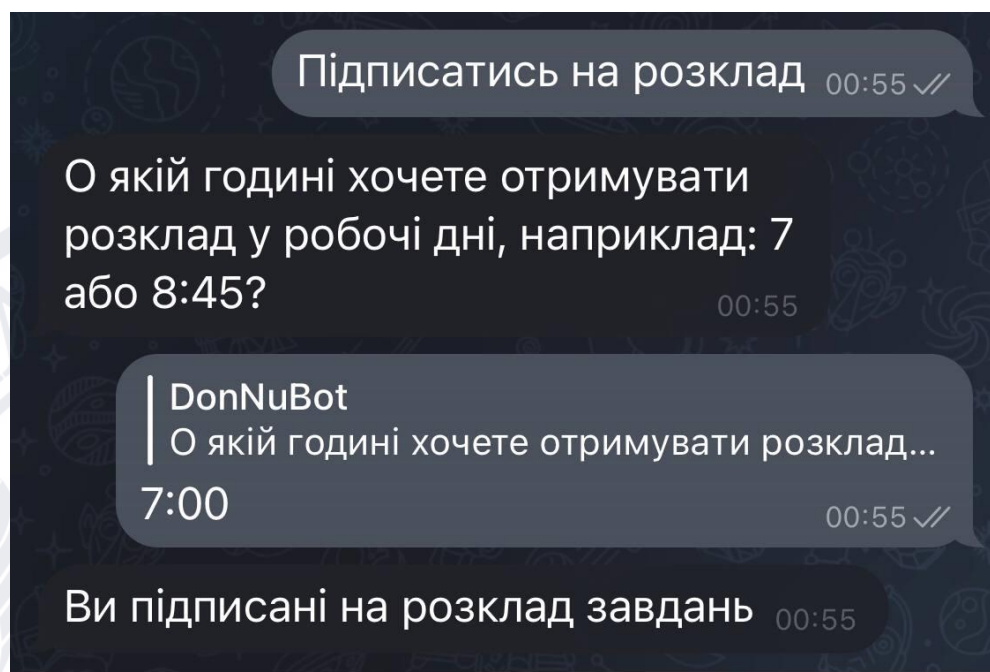


Рисунок 3.12 Вибір часу отримання розкладу

3.2 Приклади застосування та операційний аналіз

Чат-бот не може отримати старі повідомлення з чату. Бот не може отримати список усіх користувачів. Все, що може отримувати бот - це інформація про оновлення. У цьому полягає головна складність розробки роботів.

Ви можете отримувати інформацію про нові повідомлення в роботі та інших подіях, але тільки один раз. Вам доведеться самим зберігати список чатів, старих повідомлень (якщо це навіщось потрібно) і так далі. Якщо ви випадково зітріть/втратите цю інформацію, ви її більше ніяк не отримаєте.

У Telegram API бот може трохи більше: він може отримувати повідомлення по id, отримувати список учасників групи та інше. ж) оцінювання коректності відповідей.

Звичайно, на запити до сервера є ліміти. У Bots FAQ на сайті Telegram названі наступні:

- Не більше одного повідомлення на секунду в один чат,
- Не більше 30 повідомлень за секунду взагалі,
- Не більше 20 повідомлень за хвилину в одну групу.

Ці ліміти не строгі, а зразкові. Ліміти можуть бути збільшені для великих роботів через підтримку.

Нижче в Bots FAQ сказано, що Bot API не дозволяє розсилати повідомлення всім користувачам одночасно і що в майбутньому, можливо, вони щось для цього зроблять. І це написано вже кілька років.

Вони радять розтягнути розсилку на тривалий час (8-12 годин) і зауважують, що API не дозволить надсилати повідомлення більш ніж 30 користувачам в секунду.

Восени 2020 року з'явилася можливість передавати роботів іншій людині. Це можна зробити в налаштуваннях робота в BotFather. Для цього на вашому обліковому записі повинна бути включена двофакторна авторизація - не менше ніж за 7 днів до передачі. Передати бота можна лише користувачеві, який щось писав.

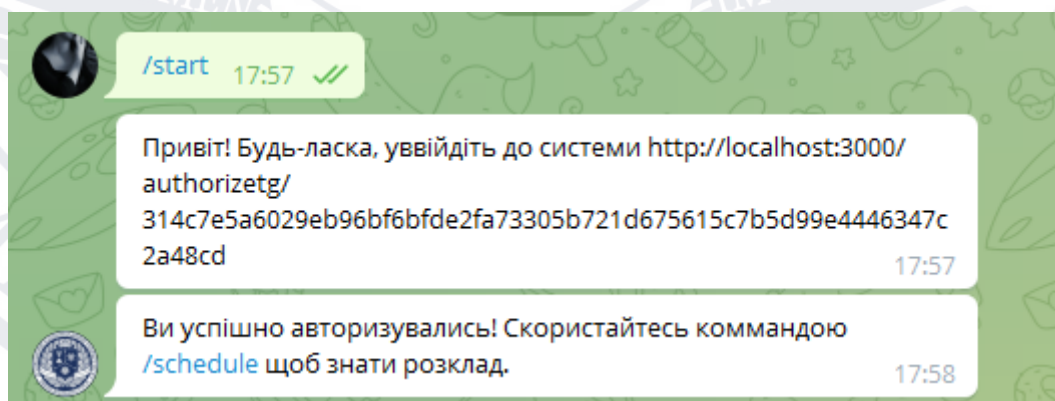


Рисунок 3.13 Команда `/start`

Для того, щоб завантажити розклад чи оновити файл з розкладом для бота потрібно перейти в відповідну папку /assets та замінити чи відредагувати файл (рис. 3.14) init.csv.

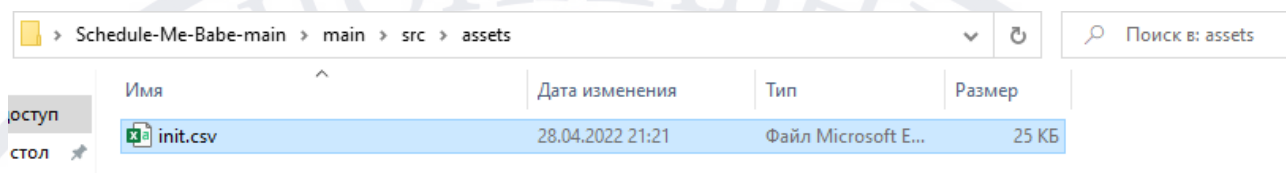


Рисунок 3.14 Файл init.csv

Після внесення змін адміністратором – за допомогою слухача файлів в нашій директорії файл з розкладом автоматично підгружається. При цьому наш чат-бот з сервером не падає і продовжує свою роботу без помилок.

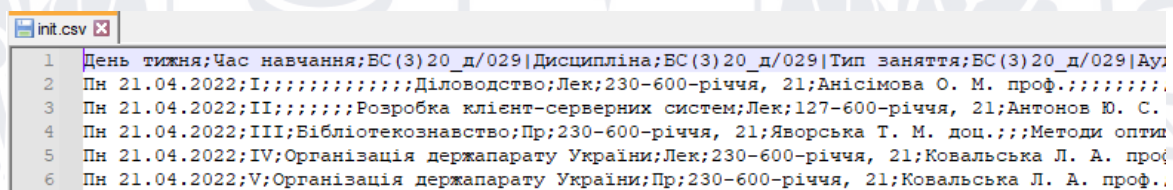


Рисунок 3.15 Формат файлу init.csv

Також важливим є те, щоб всі наступні файли з розкладом відповідали таким критеріям:

1. Назва файлу повинна залишатись без змін, а саме «init»
2. Формат файлу з розкладом бот сприймає лише в форматі csv
3. Кодировка файлу бажано повинна бути в форматі utf-8

Для перевірки кодировки використовуємо сервіс convertcsv.com (рис. 3.16) в якому прикріплюємо файл та змінюємо його кодировку

What can this tool do? ▾

See also [CSV to HTML Table](#) and [HTML Table to CSV](#)

Step 1: Select your input

Option 1 - Choose a CSV/Excel file Файл не выбран Encoding

Option 2 - Enter an URL

Option 3 - paste into Grid below

Step 2: Choose input options (optional) ▾

Save Your result: .csv or .xlsx EOL: ☒ Include Header

	A
1	

Рисунок 3.16 Зміна кодування файлу

3.3 Розміщення чат-бота на хмарному сервері Heroku

Для безперебійного отримання доступу до нашого чат-бота після завершення його розробки необхідно розмістити на віддаленому сервері. Тому в якості хмарної платформи був обраний сервіс Heroku.

Розміщення на хмарній PaaS-платформі Heroku схоже на роботу системи розподіленого контролю версій відомого як Git. Існує три способи розгортання:

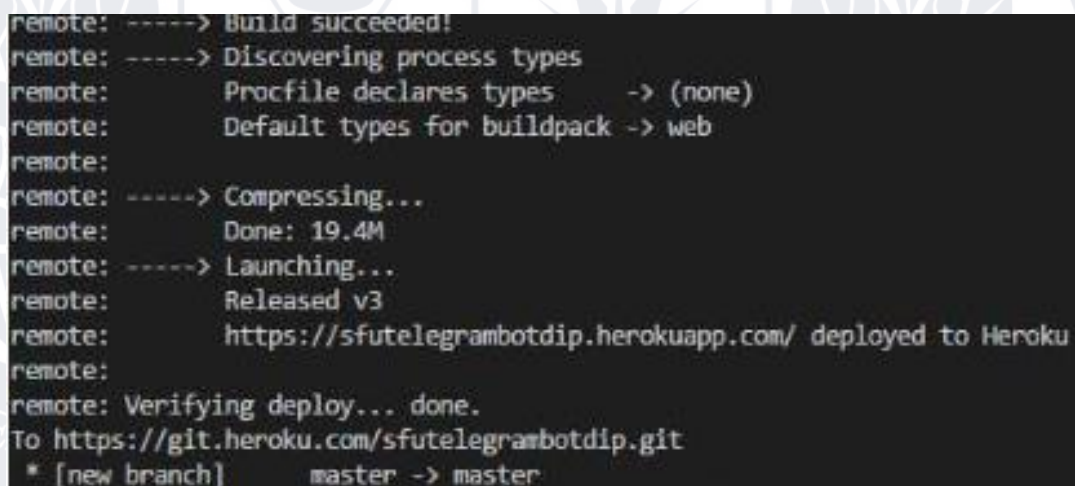
- Heroku Git та Container Registry використовуючи Heroku CLI;
- підключення облікового запису GitHub з автоматичним розгортанням;
- підключення облікового запису Dropbox з автоматичним розгортанням.

У кваліфікаційній (бакалаврській) роботі було обрано спосіб розгортання на хмарному сервісі Heroku CLI. Для розгортання чат-бота були виконані наступні дії:

- реєстрація на хмарній SaaS-платформі Heroku;
- завантажено та встановлено Heroku CLI;
- в терміналі операційної системи або вбудованому в IDE виконано вхід до облікового запису Heroku команду «heroku login»;

- клонований репозиторій з віддаленого Git-сервера Heroku на локальну машину за допомогою команди `heroku git:clone -a <APP_NAME>;`
- зафіксовані зміни в коді за допомогою команд `git add .` та `«git commit -am <COMMIT_NAME>»;`
- всі зафіксовані зміни були відправлені на віддалений сервер хмарних обчислень Heroku командою `«git push heroku master»`.

Після завершення всіх зазначених вище команд у відповідній послідовності автоматично розпочинається передача наших даних на віддалені сервери Heroku. Якщо розгортання пройшло успішно, то на терміналі ми зможемо побачити довідкову інформацію про стан роботи програми та режим доступу(рис. 3.3).



```

remote: -----> Build succeeded!
remote: -----> Discovering process types
remote:      Procfile declares types      -> (none)
remote:      Default types for buildpack -> web
remote:
remote: -----> Compressing...
remote:      Done: 19.4M
remote: -----> Launching...
remote:      Released v3
remote:      https://sfutelegrambotdip.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/sfutelegrambotdip.git
 * [new branch]      master -> master

```

Рисунок 3.17 – Результат розгортання чат-бота

Висновок до розділу 3

Третій розділ є детальним описом практичної частини роботи. У розділі детально описано як було вибрано середовище розробки, мову програмування. Міститься детальний опис використаних технологій з перевагами та недоліками на основі схожих рішень. Розглянуто та описано процес розробки з кодом. Додатково представлено скріншоти інтерфейсу та інструкція щодо користування чат-ботом.

ВИСНОВКИ

Результатом кваліфікаційної (дипломної) роботи став чат-бот месенджеру Telegram для отримання розкладу студентами, основною ціллю якого є легкий та швидкий доступ до інформації про факультет інформаційних технологій ДонНУ та розклад занять.

В рамках кваліфікаційної (дипломної) роботи було виконано поставлені завдання. По-перше, було вивчено месенджери. Було проведено порівняння та аналіз переваг та недоліків, внаслідок чого був обраний месенджер Telegram як найзручніший і найдоступніший у плані документації Telegram Bot API.

По-друге, були вивчені наявні аналоги чат-бота на платформі Telegram, а також виявлено їх переваги, недоліки та цікаві рішення.

У четвертому розділі дипломної роботи детально описано та обґрунтовано вибір використаних технологій. Описано процес розробки фронтенду та бекенду програми, розглянуто роботу з базою даних. Додатково подано алгоритм роботи чат-боту та структуру програми. Проведено тестування чат-боту.

В перспективі, бот допоможе мінімізувати багато рутинних процесів, які зараз існують під час виборчої кампанії та на початку навчального року. Бот стане швидким способом дізнатись інформацію без залучення студентів, викладачів чи адміністрації ДонНУ.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. [Електронний ресурс] Five Different Types of Chatbot <https://medium.com/voiceui/five-different-types-of-chatbot17bb255b23b4>
2. [Електронний ресурс] Natural Language Processing with Python <http://www.nltk.org/book/>
3. [Електронний ресурс] A brief history of Chatbots <https://chatbotslife.com/a-brief-history-of-chatbots-d5a8689cf52f>
4. [Електронний ресурс] What are Chatbots and Why are They Becoming so Popular? <https://www.uctoday.com/contact-centre/what-are-chatbotsand-why-are-they-becoming-so-popular/>
5. [Електронний ресурс] Chatbot Report 2019: Global Trends and Analysis <https://chatbotsmagazine.com/chatbot-report-2019-globaltrends-and-analysis-a487afec05b>
6. [Електронний ресурс] What Are The Inner Workings of a Chatbot? <https://chatbotsmagazine.com/what-is-the-working-of-a-chatbote99e6996f51c>
7. [Електронний ресурс] Building a Simple Chatbot from Scratch in Python (using NLTK) <https://medium.com/analytics-vidhya/building-a-simple-chatbot-in-python-using-nltk-7c8c8215ac6e>
8. [Електронний ресурс] Watson Assistant <https://www.ibm.com/cloud/watson-assistant/>
9. [Електронний ресурс] Dialogflow Case Studies <https://dialogflow.com/case-studies/>
10. [Електронний ресурс] Dialogflow vs Lex vs Watson vs Wit vs Azure Bot | Which Chatbot Service Platform To Use? <https://www.kommunicate.io/blog/dialogflow-vs-lex-vs-watson-vs-witvs-azure-bot/>

11.[Електронний ресурс] Chatbots: Are they Really Useful?
https://jlcl.org/content/2allissues/20Heft12007/Bayan_AbuShawar_and_Eric_Atwell.pdf

12.[Електронний ресурс] Deep Learning Chatbots: Everything You Need to Know
<https://hackernoon.com/deep-learning-chatbot-everything-youneed-to-know-r11jm30bc>

13.[Електронний ресурс] Recipes for apps you can talk to <https://wit.ai/docs/recipes>

14.[Електронний ресурс] What is a Container?
<https://www.docker.com/resources/what-container>

15.[Текстовий ресурс] Кундік К.В. Розробка веб-сервісу інтернет оголошень / Кундік К.В., Гречко А.В. – К.: Києво-Могилян. акад., 2019. – 19 с.

16.[Електронний ресурс] Welcome to python.org <https://www.python.org/>

17.[Електронний ресурс] API Telegram
https://api.telegram.org/bot/METHOD_NAME

18.[Електронний ресурс] Telegram Bot API - <https://core.telegram.org/bots/api>

19.[Електронний ресурс] Welcome to aiogram's documentation!-
<https://aiogram.readthedocs.io/>

20.[Електронний ресурс] Python and fast HTTP clients
<https://julien.danjou.info/python-and-fast-http-clients/>

21.[Електронний ресурс] Welcome to Flask – Flask Documentation
<https://flask.palletsprojects.com/en/1.1.x/>

22.[Електронний ресурс] PyTorch <https://pytorch.org/>

23.[Електронний ресурс] Welcome to pdfminer.six's documentation!
<https://pdfminersix.readthedocs.io/en/latest/>

24.[Електронний ресурс] Gunicorn - WSGI server
<https://docs.gunicorn.org/en/stable/> 48

- 25.[Электронный ресурс] PostgreSQL: The World's Most Advanced Open Source Relational Database <https://www.postgresql.org/>
- 26.[Электронный ресурс] Get Started with Elasticsearch <https://www.elastic.co/>
- 27.[Электронный ресурс] Ingest Attachment Processor Plugin <https://www.elastic.co/guide/en/elasticsearch/plugins/master/ingestattachment.html>
- 28.[Электронный ресурс] Telegram API: наглядный разбор с примерами <https://highload.today/telegram-api/>
- 29.[Электронный ресурс] Всё, о чём должен знать разработчик Телеграм-ботов <https://habr.com/ru/post/543676/>
- 30.[Электронный ресурс] Чем хорош (и чем плох) Typescript: опыт UI-разработчиков <https://blog.skillfactory.ru/glossary/typescript/>
- 31.[Электронный ресурс] Отличия TypeScript от JavaScript <https://blog.skillfactory.ru/glossary/typescript/>
- 32.[Электронный ресурс] Всё, о чём должен знать разработчик Телеграм-ботов <https://habr.com/ru/post/543676/>
- 33.[Электронный ресурс] Javascript плюсы и минусы https://xn--90abhccf7b.xn--plai/raznoe/javascript-plyusy-i-minusy-javascript-luchshij-yazyk-programmirovaniya-dlya-nachinayushhix-tak-eto-ili-net.html#_JavaScript
- 33.[Электронный ресурс] ТОП МЕССЕНДЖЕРОВ В УКРАИНЕ И МИРЕ 2020 <https://marketer.ua/top-messengers-in-ukraine-and-the-world/>
- 34.[Электронный ресурс] 20 причин почему Telegram — лучший мессенджер <https://apps4.life/20-prichin-pochemu-telegram-luchshij-messendzher/>
- 35.[Электронный ресурс] Инструкция по работе с BotFather ботом <https://botcreators.ru/blog/botfather-instrukciya/>
- 35.[Электронный ресурс] Telegram API: наглядный разбор с примерами <https://highload.today/telegram-api/>

Декларація щодо унікальності текстів роботи
та невикористання матеріалів інших авторів без посилань

ПІБ

Прізвище, ім'я, по батькові

Факультет інформаційних і прикладних технологій

Факультет

122 Комп'ютерні науки

Шифр і назва спеціальності

Сучасні інформаційні технології та програмування

Освітня програма

ДЕКЛАРАЦІЯ

Усвідомлюючи свою відповідальність за надання неправдивої інформації, стверджую, що подана кваліфікаційна (бакалаврська) робота на тему: «РОЗРОБКА ТЕЛЕГРАМ-БОТУ РОЗКЛАДУ ЗАНЯТЬ В УНІВЕРСИТЕТІ» є написаною мною особисто.

Одночасно заявляю, що ця робота:

- не передавалась іншим особам і подається до захисту вперше;
- не порушує авторських та суміжних прав, закріплених статтями 21-25 Закону України «Про авторське право та суміжні права»;
- не отримувались іншими особами, а також дані та інформація не отримувались у недозволений спосіб.

Я усвідомлюю, що у разі порушення цього порядку моя кваліфікаційна (бакалаврська) робота буде відхилена без права її захисту, або під час захисту за неї буде поставлена оцінка «незадовільно».

дата

підпис