

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук

Бакалаврська дипломна робота на тему:

Розробка телеграм-бота для вивчення іноземних мов

Виконав: студент 4 курсу, групи 3КН-206
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)

 Колесник Я. С.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

 Крилик Л. В.
(прізвище та ініціали)

« 07 » червня 2024 р.

Рецензент: к.т.н., доцент каф. АІТ

 Богач І. В.
(прізвище та ініціали)

« 07 » червня 2024 р.

Допущено до захисту

Зав. кафедри Яровий А.А. 

« 07 » червня 2024 р.

Вінниця ВНТУ - 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 122 Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри КН
проф., д.т.н. Яровий А.А.
« 07 » 03. 2024 р.

**ЗАВДАННЯ
НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Колеснику Ярославу Сергійовичу

1. Тема роботи: Розробка телеграм-бота для вивчення іноземних мов.
керівник роботи: Крилик Людмила Вікторівна, к.т.н., доц.
затверджені наказом вищого навчального закладу « 11 » 03. 2024 року № 80
2. Термін подання студентом роботи 27.05.2024 р.
3. Вихідні дані до роботи :
Мова програмування – об'єктно-орієнтована;
Месенджер;
Кількість питань до користувача не менше 2;
Кількість категорій мов для вивчення не менше 2 од.;
Інтерфейс користувача має бути інтуїтивно зрозумілий.
4. Зміст текстової частини (перелік питань, які потрібно розробити):
вступ; обґрунтування доцільності розробки телеграм-бота для вивчення іноземних мов; проектування телеграм-бота для вивчення іноземних мов; програмна реалізація телеграм-бота для вивчення іноземних мов; висновки; список використаних джерел; додатки.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень):
схема алгоритму роботи телеграм-бота для вивчення іноземних мов; загальна структурна схема функціонування системи; структура принципу роботи телеграм-бота для вивчення іноземних мов; загальна UML-діаграма класів; UML-діаграма комунікації системи чат-бота; ER-модель бази даних телеграм-бота для вивчення іноземних мов; ER-діаграма бази даних телеграм-бота для вивчення іноземних мов; приклад роботи програми

6. Консультанти розділів роботи

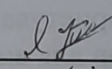
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Крилик Л. В., к.т.н., доц. каф. КН		

7. Дата видачі завдання 07.03.2024 р.

КАЛЕНДАРНИЙ ПЛАН

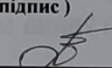
№ з/п	Назва та зміст етапу	Термін виконання		При тк
		початок	закінчення	
1	Вступ. Обґрунтування доцільності розробки телеграм-бота для вивчення іноземних мов	06.05.2024	09.05.2024	
2	Проектування телеграм-бота для вивчення іноземних мов	10.05.2024	14.05.2024	
3	Програмна реалізація телеграм-бота для вивчення іноземних мов	15.05.2024	21.05.2024	
4	Висновки та аналіз отриманих результатів	22.05.2024	24.05.2024	
5	Оформлення додатків	27.05.2024	29.05.2024	
6	Розробка інструкції користувача	30.05.2024	03.06.2024	
7	Оформлення матеріалів до захисту БДР	04.06.2024	06.06.2024	

Студент


(підпис)

Колесник Я.
(прізвище та ініціал)

Керівник роботи


(підпис)

Крилик Л.
(прізвище та ініціал)

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 83 сторінок формату А4, які включають 21 рисунок і 3 таблиці. У списку використаних джерел зазначено 22 найменування.

Основною метою цієї роботи є розширення функціональних можливостей телеграм-бота для вивчення іноземних мов.

У загальній частині роботи на основі аналізу існуючих технічних рішень, методів та технологій доведена доцільність розробки телеграм-бота для вивчення іноземних мов.

Розроблено алгоритм функціонування телеграм-бота для вивчення іноземних мов. Для кращого розуміння роботи програми були створені UML-діаграма класів та діаграма компонентів. У результаті розробки було визначено структуру принципу роботи телеграм-бота для вивчення іноземних мов

Розроблений програваний продукт протестовано, отримані результати підтверджують його ефективність та функціональність.

Розробка продукту здійснювалася в інтегрованому середовищі PyCharm з використанням мови програмування Python.

Ключові слова: Telegram messenger, навчання, Telegram-бот, іноземні мови, допомога людям.

ABSTRACT

Bachelor's thesis consists of 83 pages of A4 format, which include 21 figures and 3 tables. There are 22 names in the list of used sources.

The main goal of this work is to expand the functionality of the Telegram bot for learning foreign languages.

In the general part of the work, based on the analysis of existing technical solutions, methods and technologies, the expediency of developing a Telegram bot for learning foreign languages is proven.

An algorithm for the operation of a Telegram bot for learning foreign languages has been developed. A UML class diagram and a component diagram were created for a better understanding of the program's operation. As a result of the development, the structure of the principle of operation of the telegram bot for learning foreign languages was determined

The developed lost product has been tested, the results obtained confirm its effectiveness and functionality.

Product development was carried out in the PyCharm integrated environment using the Python programming language.

Keywords: Telegram messenger, training, Telegram bot, foreign languages, helping people.

ЗМІСТ

ВСТУП.....	4
1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ТЕЛЕГРАМ-БОТА ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ	6
1.1 Платформи, які підтримують боти.....	7
1.2 Огляд систем-аналогів.....	9
1.3 Формулювання вимог та постановка задачі.....	13
1.4 Висновок	15
2 ПРОЕКТУВАННЯ ТЕЛЕГРАМ-БОТА ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ	16
2.1 Розробка структури телеграм-бота для вивчення іноземних мов	16
2.2 Розробка UML-діаграми класів та комунікацій.....	18
2.3 Розробка схеми алгоритму роботи телеграм-бота для вивчення іноземних мов	20
2.4 Висновок	24
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТЕЛЕГРАМ-БОТА ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ	26
3.1 Обґрунтування вибору мови та бібліотек програмування	26
3.2 Обґрунтування вибору середовища розробки	30
3.3 Обґрунтування вибору бази даних для вивчення іноземних мов.....	34
3.4 Тестування роботи програми та аналіз результатів	38
3.5 Висновок	46
ВИСНОВКИ.....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	49
ДОДАТКИ.....	51
ДОДАТОК А (обов’язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень.....	52
ДОДАТОК Б (обов’язковий) Лістинг програми	53

ДОДАТОК В (обов'язковий) Графічна частина	68
ДОДАТОК Г (довідниковий) Інструкція користувача	81

ВСТУП

Актуальність теми.

Тема «Розробка телеграм-бота для вивчення іноземних мов» є надзвичайно актуальною у сучасному світі. З розвитком глобалізації та збільшенням мобільності, попит на знання іноземних мов стрімко зростає і люди шукають зручні та ефективні методи навчання. Телеграм-боти, які використовують різні сервіси можуть надавати користувачам персоналізовану та інтерактивну допомогу під час вивчення іноземних мов.

Актуальність теми обґрунтовується такими аспектами:

-Легкий доступ до навчальних матеріалів: телеграм-боти можуть забезпечити користувачів швидким і зручним доступом до різних навчальних матеріалів.

-Індивідуалізований навчальний досвід: чат-бот може адаптуватися до потреб користувача, пропонуючи індивідуальні поради та рекомендації, враховуючи рівень знань, темп навчання та особисті вподобання.

-Інтерактивне навчання: телеграм-боти можуть використовувати інтерактивні елементи, такі як тести, ігри та чати з носіями мови, для створення залучаючого та ефективного навчального досвіду.

-Навчання у будь-який час і в будь-якому місці: користувачі можуть вивчати іноземні мови у зручний для них час і в будь-якому місці, це робить процес навчання більш гнучким та доступним.

-Покращення методик викладання: використання телеграм-ботів може допомогти викладачам та навчальним закладам удосконалити свої методики викладання, надаючи додаткові інструменти для дистанційного та самостійного навчання.

Отже, розробка телеграм-бота для вивчення іноземних мов задовольнятиме потреби сучасних учнів, надаючи зручну, персоналізовану та інтерактивну підтримку під час навчання та є доцільною натеper.

Метою дослідження є розширення функціональних можливостей телеграм-бота для вивчення іноземних мов.

Об'єктом дослідження є процес вивчення іноземних мов.

Предметом дослідження є програмні засоби для реалізації телеграм-бота для вивчення іноземних мов.

Задачі дослідження:

1. Обґрунтувати доцільність розробки телеграм-бота для вивчення іноземних мов;
2. Спроекувати телеграм-бота для вивчення іноземних мов;
3. Програмно реалізувати телеграм-бота для вивчення іноземних мов;
4. Виконати тестування програми та провести аналіз отриманих результатів.
5. Розробити інструкцію користувача.

Апробація роботи.

Результати досліджень було апробовано на LIII Всеукраїнській науково-технічній конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ) м. Вінниці у 2024 р. [1].

Публікації: за основними результатами досліджень опубліковано тези доповіді на науково-технічній конференції [1].

1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ТЕЛЕГРАМ-БОТА ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ

Telegram-бот, що розробляється, орієнтований на користувачів з мобільними пристроями та доступом до мобільного інтернету. Основною функцією чат-бота для вивчення іноземних мов є надання вивчення нових слів, повторення цих слів, проходження тестів та вивчення декілької мов.

Telegram-боти – це спеціальні облікові записи в Telegram, які автоматично обробляють і відправляють повідомлення. Вони створюються програмістами та функціонують на сервері, виконуючи різноманітні завдання, такі як навчання, розваги, пошук інформації, нагадування та інтеграція з Інтернетом речей.

Чат-боти поділяються на два типи: публічні та приватні. Публічні чат-боти доступні для всіх користувачів Telegram, тоді як приватні боти мають обмежений доступ, доступний лише певній групі людей. У цьому випадку розроблюваний чат-бот буде публічним.

Для початку роботи з чат-ботом користувачу потрібно мати акаунт у Telegram. Взаємодія з чат-ботом здійснюється через звичайну комунікацію в месенджері, але із програмою, а не реальною людиною. Чат-боти не мають статусів «онлайн» або «був недавно в мережі», натомість відображаються як «бот» у користувачів. Чат-боти не можуть самотійно починати діалоги, тому користувачам необхідно самим ініціювати спілкування з ними.

Користувачі можуть знайти бота за його ім'ям або перейшовши за посиланням типу `telegram.me/<ім'я бота>`. Робота з ботами є простою, оскільки вони надають зручний інтерфейс для взаємодії за допомогою спеціальних кнопок з командами.

Чат-боти мають численні переваги порівняно зі звичайними мобільними додатками [2]:

- Кросплатформенність: боти працюють на різних платформах, таких як iOS, Android, Windows та інші. Це дозволяє користувачам взаємодіяти з ботом незалежно від їхнього пристрою.

- Безкоштовність: використання ботів у Telegram є безкоштовним. Користувачі не сплачують за встановлення або використання бота.

-Швидка реакція: боти здатні оперативно обробляти запити користувачів і відповідати на них протягом 1÷3 секунд, що дозволяє отримувати інформацію дуже швидко.

- Відсутність реєстрації: користувачам не потрібно створювати окремий акаунт або проходити реєстрацію для використання бота. Вони можуть почати взаємодію без зайвих формальностей.

- Відсутність реклами: чат-боти в Telegram не містять рекламних матеріалів, тому користувачі можуть користуватися чат-ботом без перерв на рекламу.

-Синхронізація між пристроями: історія спілкування з чат-ботом синхронізується між різними пристроями користувача, що дозволяє продовжити взаємодію на іншому пристрої без втрати даних.

-Відсутність інсталяції: користувачам не потрібно встановлювати додаткові програми або оновлювати чат-бота. Вони можуть працювати з чат-ботом безпосередньо в Telegram.

Проте, потрібно враховувати, що функціональність бота може бути обмежена можливостями певних платформ або технічними обмеженнями Telegram [3].

1.1 Платформи, які підтримують чат-боти

На сьогодні існує багато платформ для створення інтелектуальних чат-ботів. Щоб обрати найкращу, важливо розглянути ключові аспекти кожної з них. Найпопулярнішими платформами для цього є Telegram, Discord та Facebook Messenger.

Telegram відомий своєю відкритістю та безпекою, це робить його гарним вибором для розробки чат-ботів. Він підтримує різноманітні типи повідомлень, включаючи текстові, голосові та відео. Чат-боти Telegram можуть

використовувати відкриті додатки та мають доступ до докладної документації API.

Discord [4], хоча і призначений в основному для геймерів, широко використовується і для створення чат-ботів. Він підтримує взаємодію через текстові, голосові та відео канали, а також надає доступ до свого API для створення чат-ботів. Discord фінансується за рахунок платних підписок та Nitro, що може вплинути на бізнес-модель.

Для більш наочного порівняння результати аналізу цих платформ представлені у таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика платформ для розробки чат-бота для вивчення іноземних мов

Характеристика	Telegram	Discord	Facebook Messenger
Відкритість і безпека	Відкрита, високий рівень безпеки	Відкрита, висока продуктивність	Частина Facebook, високий рівень безпеки
Підтримка типів повідомлень	Текстові, голосові, відео	Текстові, голосові, відео	Текстові, відео-дзвінки, групові чати
Доступ до API	Так, детальна документація API	Так, доступ до API	Так, доступ до API
Цільова аудиторія	Широке використання, особливо технічні спільноти	Геймери, професіонали, спільноти	Найбільша кількість активних користувачів
Бізнес-модель	Комерційна	Платні підписки (Nitro), спонсорська підтримка	Комерційна
Інтеграції	Відкриті додатки, широкий спектр інтеграцій	Широкий спектр інтеграцій з іншими платформами	Широкий спектр інтеграцій з іншими платформами
Використання для комерційних цілей	Так, висока потенційна аудиторія	Так, особливо для геймерського співтовариства та спільнот	Так, особливо у комерційних цілях

Продовження таблиці 1.1

Характеристика	Telegram	Discord	Facebook Messenger
Особливості	Велика аудиторія, різноманітність типів повідомлень	Велика кількість інтеграцій, створення серверів для спільнот	Найбільша кількість активних користувачів, відеодзвінки, групові чати

Discord є популярною платформою для спілкування геймерів, а також широко використовується для професійного спілкування та розваг. Вона підтримує текстові, голосові та відео-повідомлення, а також функцію створення серверів для спільнот та каналів. Discord відомий своєю високою продуктивністю і має велику кількість інтеграцій з іншими платформами.

Facebook Messenger [5], з іншого боку, є частиною соціальної мережі Facebook і має найбільшу кількість активних користувачів серед усіх месенджерів. Він підтримує різні типи повідомлень, а також відео-дзвінки і групові чати. Чат-боти на платформі Facebook Messenger мають значний потенціал для взаємодії з користувачами, особливо у комерційних цілях.

На основі аналізу даних з таблиці 1.1 можна зробити висновок, що найкращими платформами для розробки чат-бота для вивчення іноземних мов є Telegram [6, 7] і Facebook Messenger. Обидві платформи є комерційними, але Telegram може мати перевагу у випадку, коли важлива велика аудиторія і високий рівень безпеки.

1.2 Огляд систем-аналогів

Існує багато додатків із подібними функціями, але у форматі безкоштовного Telegram-бота розроблюваний телеграм-бот для вивчення іноземних мов відрізнятиметься від аналогів. Аналіз проводитиметься з метою виявлення їхніх особливостей, переваг та недоліків, а також визначення

потенційних можливостей для подальшого впровадження і розвитку розроблюваного телеграм-бот для вивчення іноземних мов.

Для того, щоб провести порівняння розглянемо такі додатки та чат-боти: «Languagecenterbot», «AI English Tutor by Skyeng» [8], «Duoligo» та «Babbel».

«Languagecenterbot» має доволі багато функціоналу, також він простий та зрозумілий у використанні [9].

Основні функціональні можливості цього чат-бота такі:

- вивчення правил вітання з прикладами в текстовому форматі;
- приклади фраз для різних життєвих ситуацій;
- вивчення правил англійської: часів, ввідних слів;
- додаткові функціонали.

Основні недоліки:

- немає бази даних для користувачів;
- все статичне;
- немає перекладача;
- немає покрокового вивчення слів та їх закріплення.

Розглянемо чат-бот «AI English Tutor by Skyeng», який також є популярним серед користувачів. Цей чат-бот створений для комплексного вивчення іноземних мов з інтерактивними функціями.

Основними функціональними можливості цього чат-бота є:

- індивідуальні уроки з адаптацією до рівня користувача;
- інтерактивні вправи для закріплення знань;
- можливість обирати теми для вивчення (подорожі, бізнес, повсякденне життя тощо);
- функція перекладача з можливістю прослуховування вимови слів;
- тестування знань та відстеження прогресу;
- мотиваційні нагадування та підказки.

Основні недоліки:

- відсутність функції збереження персональних даних користувачів;
- деякі функції доступні лише у платній версії;
- обмежений вибір мов для вивчення;
- інтерфейс може здатися складним для новачків.

«Duolingo» є одним з найвідоміших додатків для вивчення мов (рис. 1.1) [10].

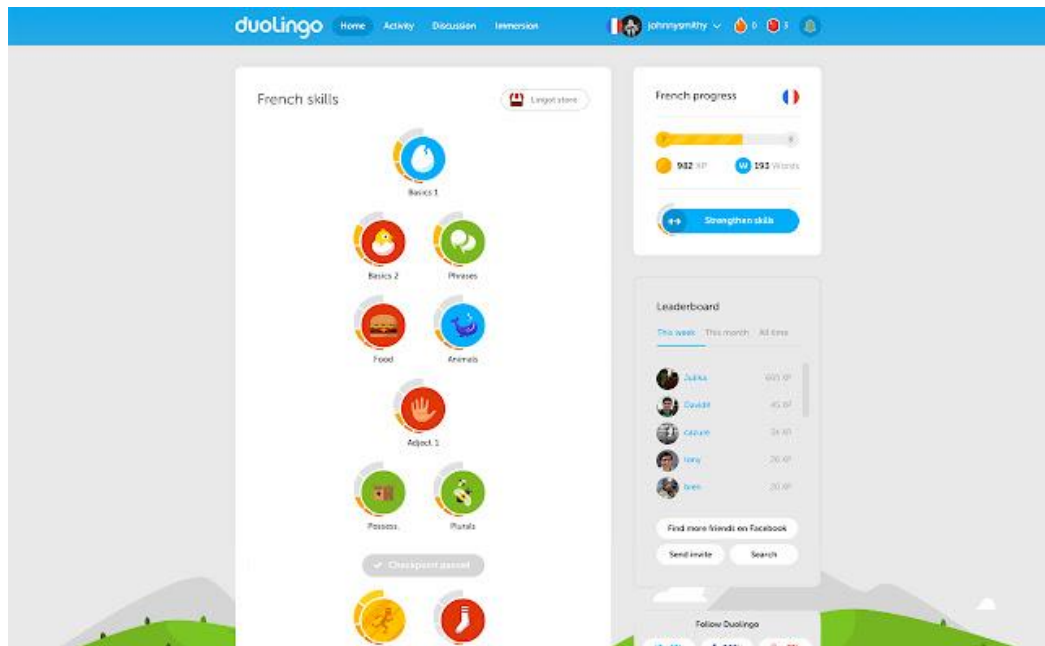


Рисунок 1.1 – Приклад роботи чат-боту «Duolingo»

Основні переваги Duolingo включають:

- Інтерактивне навчання: користувачі можуть виконувати різні вправи, які включають вибір правильних варіантів, написання відповідей та аудіозавдання.
- Гейміфікація процесу навчання: Duolingo використовує систему балів, рівнів та досягнень, що робить навчання більш цікавим і мотивуючим.
- Різноманітність мов: Duolingo пропонує широкий вибір мов для вивчення, включаючи іспанську, французьку, німецьку, японську та багато інших.
- Безкоштовний доступ: основні функції додатку доступні безкоштовно, що робить його доступним для широкої аудиторії.

Основні недоліки:

- Обмеженість вправ: хоча Duolingo пропонує багато різних типів вправ,

деякі користувачі можуть знайти їх повторюваними або недостатньо глибокими для просунутого рівня.

-Обмежений контекст: вправи часто ізольовані від реальних ситуацій, що може обмежувати практичне застосування знань.

-Відсутність персоналізації: хоча додаток адаптується до рівня користувача, відсутня індивідуалізація навчального плану, що може бути необхідним для певних користувачів.

«Babbel» пропонує більш структурований підхід до вивчення іноземних мов, орієнтуючись на реальні ситуації та комунікативні навички (рис 1.2) [11].

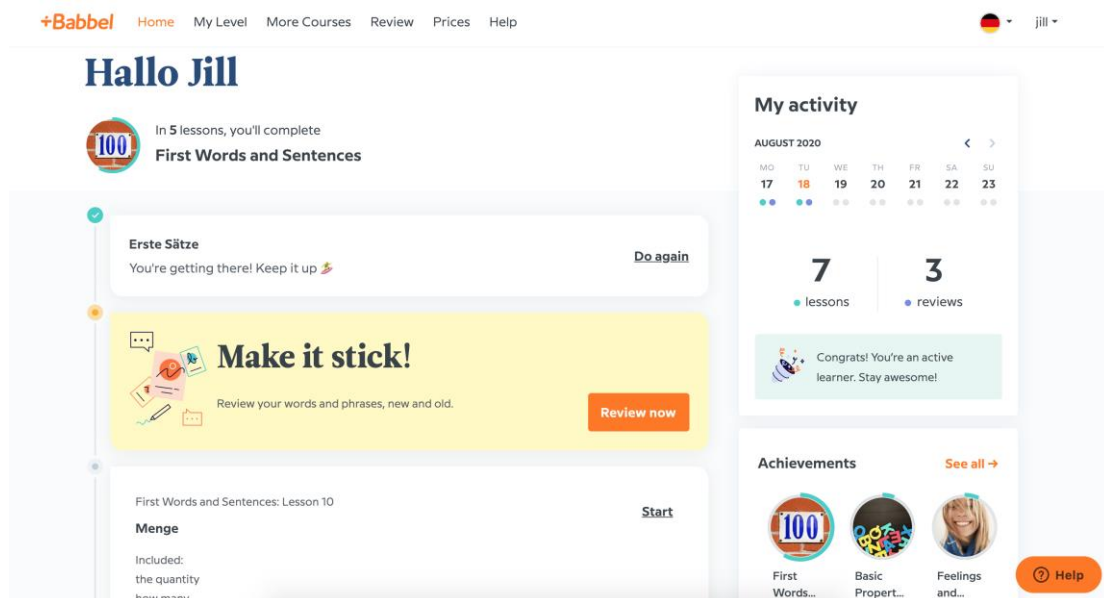


Рисунок 1.2 – Приклад роботи чат-боту «Babbel»

Основні переваги Babbel включають:

-Структуровані курси: Babbel пропонує курси, які розбиті на короткі уроки, орієнтовані на певні теми та ситуації.

-Комунікативний підхід: уроки орієнтовані на реальні діалоги та ситуації, що допомагає користувачам застосовувати знання в реальному житті.

-Персоналізація: Babbel адаптує курси під індивідуальні потреби користувачів, включаючи можливість вибору тем для вивчення.

-Висока якість контенту: уроки створені професійними лінгвістами та

носіями мови, що забезпечує високу якість навчального матеріалу.

Основні недоліки:

-Платний доступ: основні функції Babbel доступні лише за підпискою, що може бути недоступним для деяких користувачів.

-Обмежена кількість мов: хоча Babbel пропонує вивчення кількох популярних мов, вибір не такий широкий, як у деяких інших додатках.

Відсутність гейміфікації: на відміну від Duolingo, Babbel менш орієнтований на гейміфікацію, що може зменшувати мотивацію для деяких користувачів.

На основі аналізу даних та огляду популярних додатків, таких як «Languagecenterbot», «AI English Tutor by Skyeng», «Duolingo» та «Babbel», стає очевидним, що кожен із цих інструментів має свої особливості, переваги та недоліки. Однак, найкращим варіантом для розробки чат-бота для вивчення іноземних мов є Telegram. Це пояснюється тим, що Telegram пропонує високий рівень безпеки, багатофункціональність, кросплатформну сумісність та зручність використання. Telegram також дозволяє інтегрувати бота з різними сервісами, що відкриває широкі можливості для подальшого розвитку та вдосконалення чат-бота.

1.3 Формулювання вимог та постановка задачі

В сучасному світі інтерес до вивчення іноземних мов є актуальним та має практичне значення. Завдяки розвитку технологій та поширенню смартфонів, люди все частіше використовують мобільні додатки для підвищення своєї мовної компетентності, особливо в контексті мандрівок або перебування за кордоном.

Телеграм-бот для вивчення іноземних мов може забезпечити користувачам зручний та персоналізований досвід, допомагаючи засвоювати мову, покращувати граматику, вивчати нові слова та фрази, а також практикувати мовлення. Такий чат-бот може надавати рекомендації, відповідати на запитання, допомагати з вибором навчальних матеріалів та багато іншого.

Для створення телеграм-бота для вивчення іноземних мов необхідно реалізувати такі функції:

1. Розробити чат-бот з покроковим супроводом користувача для Telegram-бота:

- забезпечити інтуїтивно зрозумілий інтерфейс для користувачів;
- реалізувати можливість вибору іноземної мови, яку користувач хоче вивчати.

2. Функція легкого та швидкого знаходження навчальних матеріалів:

- забезпечити доступ до різноманітних відеоуроків, аудіофайлів, текстів для читання тощо;
- надавати рекомендації щодо найкращих матеріалів для конкретного рівня мовлення користувача.

3. Функція практики мовлення та вимови:

- реалізувати можливість запису користувача і надавати зворотній зв'язок щодо правильності вимови та граматичної правильності;
- допомагати коригувати помилки та надавати вказівки для покращення вимови.

4. Функція вивчення нових слів та фраз:

- надавати щоденні завдання для вивчення нових слів та застосування їх у реченнях;
- забезпечити інтерактивні вправи для запам'ятовування слів та фраз.

5. Здійснення тестування телеграм-бота для вивчення іноземних мов:

- провести тестування з метою оцінки функціональності, інтуїтивності і ефективності чат-бота.

Цей чат-бот для вивчення іноземних мов має бути максимально зручним для користувачів і розроблятися з урахуванням користувацького досвіду (UX). Вхідні дані, такі як запити та налаштування, мають легко задаватися користувачем шляхом інтуїтивного та зрозумілого інтерфейсу. Вихідні дані мають бути чітко й зрозуміло представлені, щоб користувач міг легко отримати потрібну інформацію.

Цей чат-бот має допомагати користувачам ефективно вивчати іноземні мови, надаючи доступ до необхідних навчальних матеріалів та практичних завдань у будь-який час і у будь-якому місці.

1.4 Висновок

У цьому розділі було проведено аналіз різних платформ для розробки ботів. Платформа Telegram була обрана як найбільш прийнятна через свою захищеність та доступну документацію. Були розглянуті існуючі застосунки, і виявлено, що вони не повністю задовольняють потреби користувачів і нерідко дратують їх рекламою та нагадуваннями про платні послуги.

В сучасному світі інтерес до вивчення іноземних мов зростає і з цієї причини розробка телеграм-бота для вивчення іноземних мов є дуже актуальною. Різноманітність мобільних платформ і сервісів, призначених для навчання мов, підтверджує значний попит на інноваційні підходи у цій сфері. Зазвичай такі сервіси надають великий вибір уроків, відео, тестів та інших корисних матеріалів, однак наявні значні недоліки

2 ПРОЕКТУВАННЯ ТЕЛЕГРАМ-БОТА ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ

2.1 Розробка структури телеграм-бота для вивчення іноземних мов

Подібно до програм, Telegram-бот складається з набору функціональних компонентів, але зберігати всі ці компоненти у одному файлі незручно і непрактично. Структура Telegram-бота ґрунтується на розбитті на окремі модулі або частини. Кожен модуль відповідає за конкретну функціональність або частину бота. Наприклад, можуть бути окремі модулі для обробки повідомлень, роботи з базою даних, реалізації логіки навчання та інтеграції з зовнішніми сервісами.

Кожен модуль програмується та розробляється незалежно від інших модулів, що дозволяє працювати паралельно та зосередитись на конкретній функціональності без необхідності змінювати весь код чат-бота. Крім того, така структура дозволяє зручно підтримувати та масштабувати чат-бота у майбутньому.

Кожен модуль має свою назву та може мати свою власну структуру файлів, що спрощує розуміння та організацію коду. Наприклад, модуль може містити окремі файли для Python-скриптів, налаштувань конфігурації та ресурсів, пов'язаних з цією частиною бота. Застосування модульної структури дозволяє зберігати код бота організованим, зрозумілим та легко розширюваним.

В Telegram-боті для вивчення іноземних мов будуть такі функції: реєстрація користувача, інтерактивне навчання, відстеження прогресу та інтеграція зі словниками. Кожна функція матиме такі модулі: модуль для графічного оформлення повідомлень, для реалізації функціональності клієнтської частини та для реалізації серверної частини, яка буде взаємодіяти з базою даних. Структура Telegram-бота для вивчення іноземних мов зображена на рисунку 2.1.

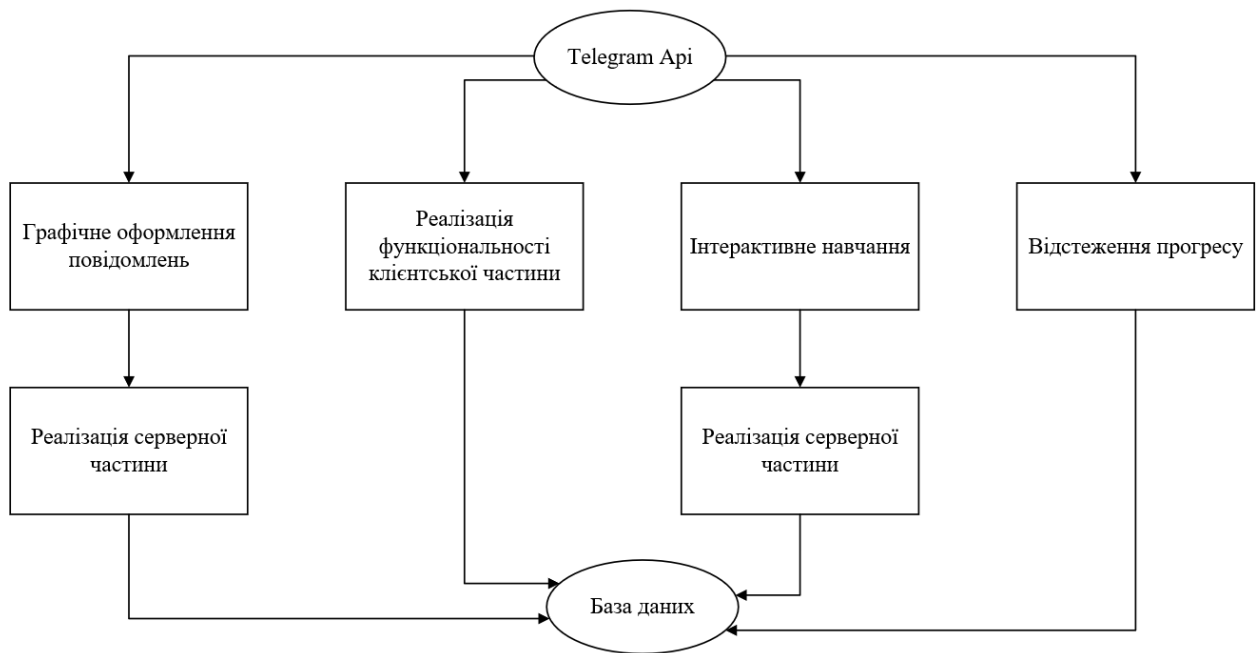


Рисунок 2.1 – Загальна структурна схема функціонування системи

На сторінці інтерактивного навчання користувач зможе проходити уроки, виконувати завдання та отримувати зворотний зв'язок. Серверний модуль відповідає за отримання завдань та відповідей, після чого передає їх до модуля клієнтської частини де вони оброблюються та передаються до інтерфейсу користувача. Також серверний модуль відповідає за збереження результатів користувача та їх відстеження.

На сторінці відстеження прогресу користувачі можуть переглядати свою статистику, прогрес у вивченні мови та отримувати рекомендації. Модуль реалізації функціональності клієнтської частини відповідає за отримання даних з бази даних, після чого передає їх на сервер, який в свою чергу оброблює та надсилає дані до інтерфейсу користувача.

База даних має містити дані для автентифікації користувача, його прогрес, результати виконаних завдань та інші навчальні ресурси. Крім цього, вона має забезпечити швидкий імпорт та автоматичний експорт усієї інформації, необхідної для роботи Telegram-бота для вивчення іноземних мов. Структура та загальний принцип роботи Telegram-бота представлено на рисунку 2.2.

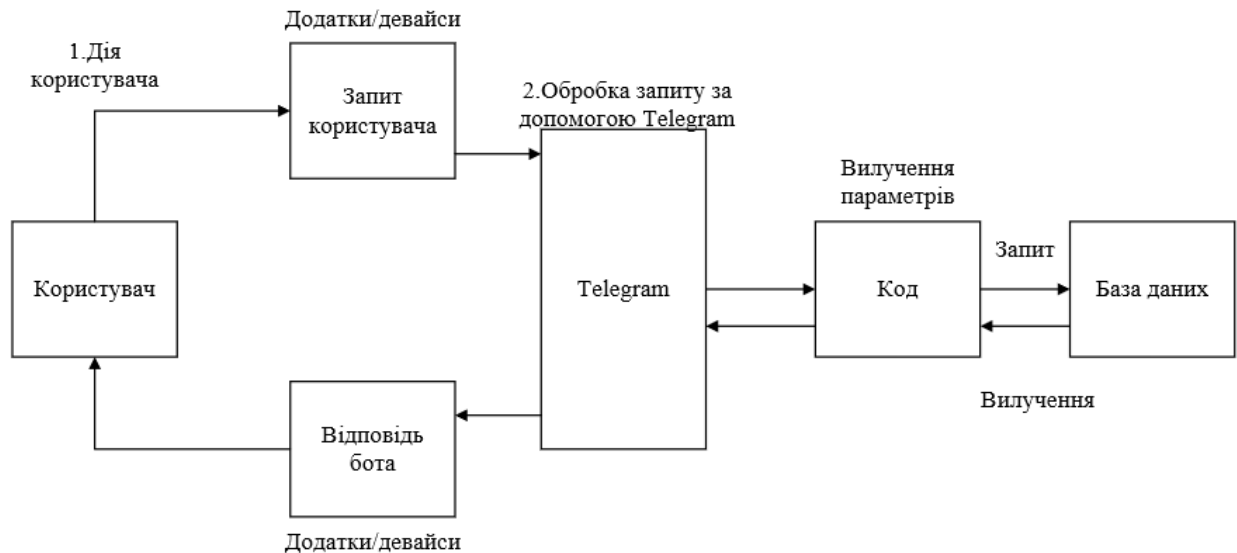


Рисунок 2.2 – Структура принципу роботи телеграм-бота для вивчення іноземних мов

2.2 Розробка UML-діаграми класів та комунікацій

UML-діаграма класів є інструментом графічного моделювання, який використовується для візуалізації структури програмної системи. Вона включає в себе класи, їх атрибути та методи, а також зв'язки між класами, які демонструються за допомогою стрілок (рис. 2.3) [12].

Основні елементи UML-діаграми класів включають:

1. Класи зображуються у прямокутниках, які містять назву класу. Атрибути класу відображаються у вигляді змінних зі знаком «+» або «-» для позначення видимості;
2. Методи класу відображаються у вигляді функцій зі знаком «+» або «-» для позначення видимості;
3. Зв'язки між класами показуються за допомогою стрілок. Різні типи зв'язків включають зв'язки спадкування (inheritance), асоціації (association), агрегації (aggregation) та композиції (composition).

Ці діаграми допомагають розробникам, менеджерам проекту та іншим учасникам команди краще розуміти структуру програми і взаємозв'язки між її частинами.

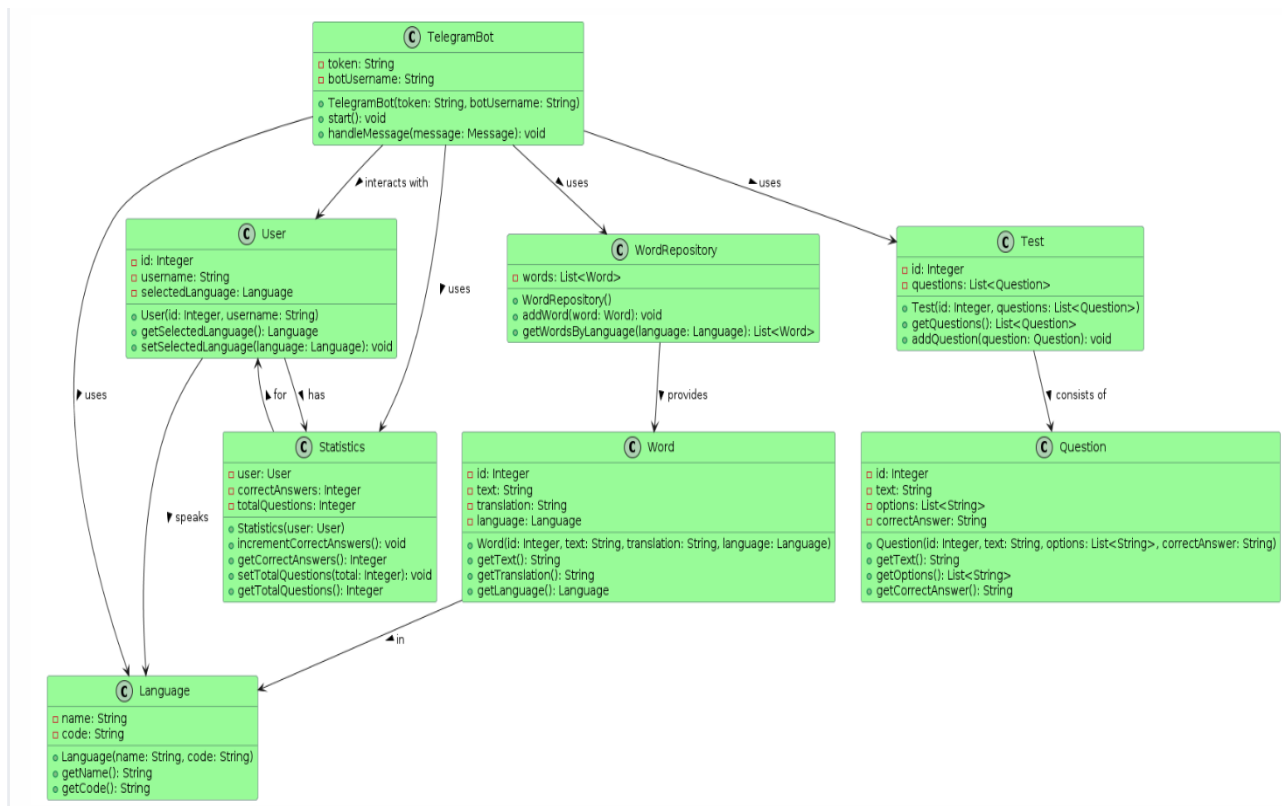


Рисунок 2.3 – Загальна UML-діаграма класів

UML (Unified Modeling Language) – це стандартизована мова моделювання, яка використовується для візуалізації, специфікації, побудови та документування програмних систем. UML-діаграми дозволяють інженерам програмного забезпечення відобразити різні аспекти системи, такі як структура, поведінка, взаємодія та архітектура [13].

У випадку інтелектуального чат-бота для вивчення іноземних мов, UML-діаграма комунікації може показувати взаємодію між різними модулями чат-бота та користувачем. Наприклад, діаграма може відображати взаємодію між модулем голосового розпізнавання, модулем перекладу та модулем навчання.

Модуль голосового розпізнавання може отримувати вимовлені користувачем фрази, передавати їх до модуля перекладу для отримання еквівалентного тексту у вибраній мові, а потім передавати отримані тексти

модуля навчання для аналізу та виправлення помилок. На діаграмі можна показати, як модулі обмінюються даними, щоб забезпечити коректний процес навчання користувача.

Діаграма також може відображати взаємодію модуля інтерфейсу, який відображає рекомендації та інструкції для користувача на основі аналізу результатів навчання та вимовлених фраз. Такий модуль може також отримувати від користувача запити на конкретні справи або теми для вивчення, це може бути передано модулю навчання для подальшого аналізу та формування відповідей.

Таким чином, UML-діаграма комунікації відображає взаємодію різних модулів чат-бота для вивчення іноземних мов, яка дозволяє забезпечити ефективний процес навчання та покращення мовних навичок користувачів (рис. 2.4).



Рисунок 2.4 – UML-діаграма комунікації системи чат-бота для вивчення іноземних мов

2.3 Розробка схеми алгоритму роботи телеграм-бота для вивчення іноземних мов

Алгоритм — це точно визначений набір інструкцій, які виконуються комп'ютером для вирішення певної задачі. Він може бути у вигляді послідовності дій, розгалужень, циклів тощо.

Симетричний алгоритм – це алгоритм, який використовує один і той самий ключ для шифрування і дешифрування інформації. Це означає, що той хто шифрує повідомлення і той хто його отримує, мають мати один ключ.

Послідовний алгоритм – це алгоритм, який виконується в строго визначеному порядку, одна за одною послідовною дією. Кожен крок виконується лише після завершення попереднього, що робить його простим у розумінні та відлагодженні.

Для розробки чат-бота для вивчення іноземних мов найбільш прийнятними є послідовні алгоритми. Такий підхід дозволяє крок за кроком виконувати важливі функції, такі як розпізнавання мови, генерація відповідей, переклад текстів тощо. Послідовний алгоритм забезпечує систематичність і надійність роботи чат-бота, що особливо важливо для такого виду програм.

З іншого боку, симетричні алгоритми зазвичай застосовуються для захисту конфіденційності інформації, і хоча вони не підходять для розробки чат-ботів для вивчення мов, вони можуть бути використані для шифрування даних, зберігаючи приватність користувачів.

Отже, для розробки чат-бота для вивчення іноземних мов доцільно використовувати послідовні алгоритми, які забезпечують послідовність виконання завдань та ефективність роботи чат-бота. Крім того, симетричні алгоритми використовуються для інших цілей, наприклад, для забезпечення безпеки даних у транспортних каналах або для зберігання конфіденційної інформації (рис. 2.5).

І випадок

Алгоритм у випадку натиску на кнопку «Слова, які вивчили» складається з таких кроків:

- Крок 1. Користувач натискає кнопку «start»;
- Крок 2. Яка кнопка натиснута;
- Крок 3. Користувач натискає кнопку «Англійська»;
- Крок 5. Внесення даних в БД;
- Крок 6. Виведення вихідних даних;

- Крок 7. Яка кнопка натиснута;
- Крок 8. Користувач натискає кнопку «Вивчення слів»;
- Крок 9. Виведення вихідних даних;
- Крок 10. Користувач натискає кнопку «Слова, які вивчили»;
- Крок 11. Виведення вихідних даних.

II випадок

Алгоритм у випадку натиску на кнопку «Вивчення слів» складається з таких кроків:

- Крок 1. Користувач натискає кнопку «start»;
- Крок 2. Яка кнопка натиснута;
- Крок 3. Користувач натискає кнопку «Англійська»;
- Крок 5. Внесення даних в БД»;
- Крок 6. Виведення вихідних даних;
- Крок 7. Яка кнопка натиснута;
- Крок 8. Користувач натискає кнопку «Вивчення слів»;
- Крок 9. Виведення вихідних даних;
- Крок 12. Користувач натискає кнопку «Вивчення нових слів»;
- Крок 13. Виведення вихідних даних;
- Крок 14. Користувач натискає кнопку «Запам'ятав»;
- Крок 15. Виведення вихідних даних.

III випадок

Алгоритм у випадку натиску на кнопку «Вивчення слів» складається з таких кроків:

- Крок 1. Користувач натискає кнопку «start»;
- Крок 2. Яка кнопка натиснута;
- Крок 3. Користувач натискає кнопку «Англійська»;
- Крок 5. Внесення даних в БД;
- Крок 6. Виведення вихідних даних;
- Крок 7. Яка кнопка натиснута;

Крок 16. Користувач натискає кнопку «Тести»;

Крок 17. Виведення вихідних даних;

Крок 18. Користувач натискає кнопку «Вибір правильного варіанту».

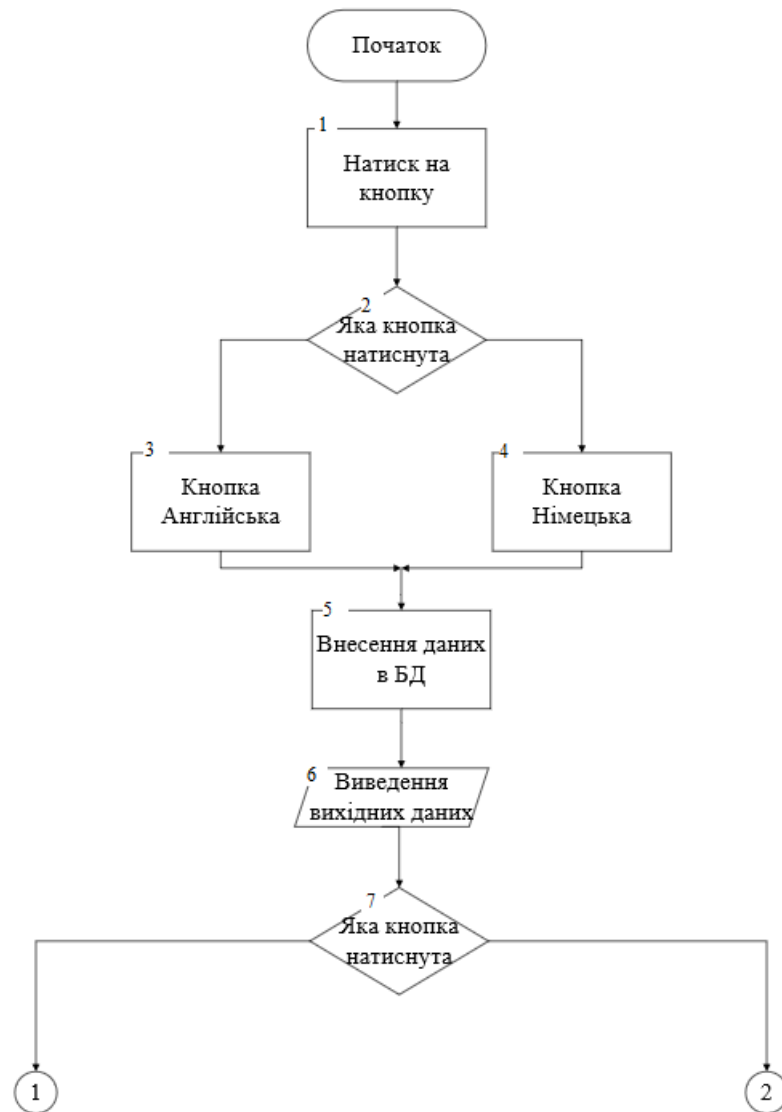


Рисунок 2.5 – Схема алгоритму роботи телеграм-бота для вивчення іноземних мов

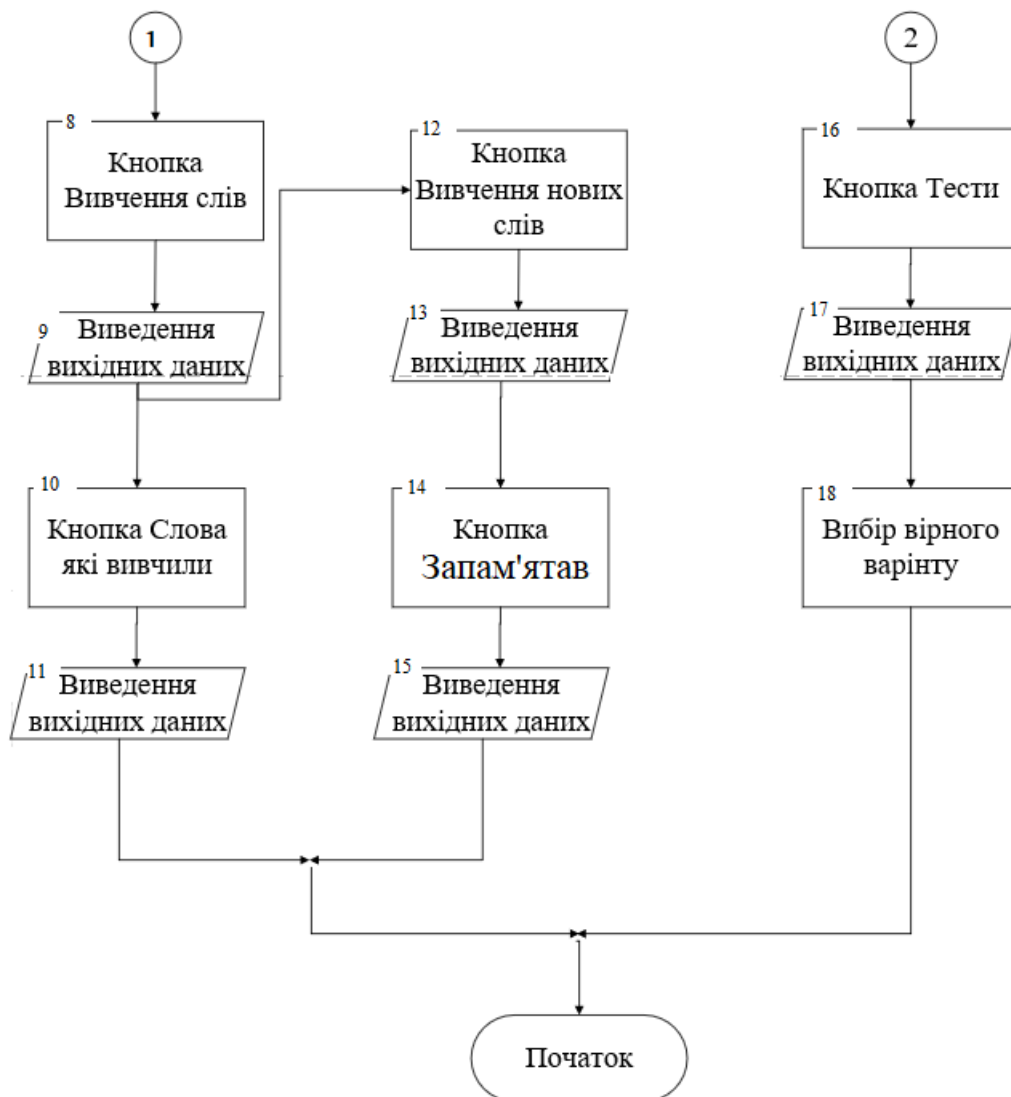


Рисунок 2.5, аркуш 2

2.4 Висновок

Загальна структура Telegram-бота для вивчення іноземних мов визначається потребою в модульній організації, що дозволяє зберігати код організованим, зрозумілим та легко розширюваним. Кожен модуль відповідає за конкретну функціональність, таку як обробка повідомлень, робота з базою даних, інтерактивне навчання, інтеграція зі словниками тощо.

Модульна структура дозволяє розробляти і підтримувати різні частини чат-бота незалежно одна від одної, спрощує розуміння коду та полегшує масштабування проекту в майбутньому. Кожен модуль має свою назву, власну структуру файлів, налаштування конфігурації та ресурси, що покращує організацію та управління проектом.

Для кращого розуміння і взаємодії між компонентами системи використовується UML-діаграма класів та комунікацій. Ці діаграми дозволяють візуалізувати структуру програмної системи, показуючи класи, їх атрибути, методи та зв'язки між ними. Це допомагає розробникам краще розуміти систему та взаємозв'язки між її частинами.

Розробка схеми алгоритму роботи чат-бота для вивчення іноземних мов показує важливі кроки, які виконує чат-бот у відповідь на дії користувачів, наприклад, вивчення нових слів чи відстеження прогресу у вивченні іноземної мови. Використання послідовних алгоритмів забезпечує надійність та систематичність роботи чат-бота, що є важливим у розробці таких програм.

Отже, зазначена структура, використання UML-діаграм та послідовні алгоритми дозволяють ефективно розробляти, впроваджувати та підтримувати Telegram-бота для вивчення іноземних мов, забезпечуючи користувачам зручний інтерфейс і надійність системи.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТЕЛЕГРАМ-БОТА ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ

3.1 Обґрунтування вибору мови та бібліотек програмування

При розробці програмного забезпечення, вибір мови програмування є одним з найважливіших рішень, яке визначає ефективність, продуктивність та можливості майбутнього продукту. Це особливо актуально у випадку розробки чат-ботів, які мають бути не лише функціональними, але й здатними швидко обробляти великі обсяги даних та надавати користувачам релевантні відповіді у реальному часі. Для створення Телеграм-ботів найбільш популярними мовами програмування є Python, C# та Node.js. Розглянемо їхні характеристики, переваги та недоліки, а також обґрунтуємо вибір мови для створення чат-бота для вивчення іноземних мов.

Мова програмування має відповідати кільком ключовим критеріям: легкість у навчанні та використанні, наявність потужних бібліотек для обробки природної мови, можливості для інтеграції з різними API та сервісами, а також здатність обробляти великі обсяги даних з високою швидкістю та ефективністю. Враховуючи ці критерії, проаналізуємо три основні мови програмування для розробки чат-ботів: Python, C# та Node.js [14].

Python [15]

Переваги:

- Простота та легкість у навчанні та використанні: Python відомий своєю простотою синтаксису та легкістю в освоєнні, що робить його дуже популярним серед початківців.
- Багато бібліотек та фреймворків для створення чат-ботів: великий вибір бібліотек, таких як Python-telegram-bot для Telegram, ChatterBot для створення розмовних ботів, Rasa для розробки AI чат-ботів.

- Можливості для обробки даних та машинного навчання: Python є однією з провідних мов для обробки даних та реалізації алгоритмів машинного навчання.

Недоліки:

- Відносно повільне виконання: у порівнянні з іншими мовами, такими як C++ або Java, Python може бути менш ефективним з точки зору швидкодії.
- Не підходить для високонавантажених систем: за рахунок високого рівня абстракції та динамічної типізації, Python може мати обмежену продуктивність у великих та високонавантажених системах.

C#[16]

Переваги:

- Висока продуктивність та швидкість виконання: C# має високу продуктивність та швидкість, оскільки компілюється в машинний код.
- Підтримка різних парадигм програмування: C# підтримує об'єктно-орієнтоване програмування, аспектно-орієнтоване програмування, а також функціональне програмування.
- Глибока інтеграція з екосистемою .NET: C# є основною мовою для платформи .NET, що включає в себе велику кількість бібліотек та інструментів для розробки.

Недоліки:

- Складність у навчанні для новачків: C# може бути складнішою для освоєння, особливо для тих, хто тільки починає з програмування.
- Менша кількість бібліотек для створення чат-ботів: у порівнянні з Python, C# має меншу кількість бібліотек, спеціалізованих на створення чат-ботів.

Node.js

Переваги:

- Висока продуктивність завдяки асинхронній природі JavaScript: Node.js використовує однопотокову, але асинхронну модель виконання, що робить його ефективним для високонавантажених систем.

- Велика кількість бібліотек для створення чат-ботів: у Node.js є багато фреймворків та бібліотек для створення чат-ботів, таких як Botpress, Microsoft Bot Framework.

Недоліки:

- Складність у налагодженні та відстеженні помилок: за рахунок асинхронної природи, в Node.js може бути складніше відстежувати послідовність подій та відлагоджувати програми.

- Менша стабільність та зрілість у порівнянні з іншими мовами: у порівнянні з C# або Java, Node.js може мати меншу стабільність, оскільки він молодша технологія.

На основі аналізу, було обрано мову програмування Python. Її простота, велика кількість доступних бібліотек та потужні можливості для обробки даних роблять її ідеальним вибором для створення чат-бота для вивчення іноземних мов. Python забезпечує легкість у розробці та швидке освоєння для новачків, а також пропонує широкий спектр інструментів для інтеграції з машинним навчанням та обробкою природної мови, що є ключовими вимогами для створення інтелектуального бота.

Розробка чат-бота для вивчення іноземних мов вимагає використання надійних та функціональних бібліотек, які забезпечують легкість у створенні, налаштуванні та підтримці програмного модуля [7]. У цьому розділі розглянемо основні бібліотеки для Python, які найчастіше використовуються для розробки чат-ботів, та проаналізуємо їхні характеристики, переваги та недоліки.

Python є одним з найпопулярніших мов програмування завдяки своїй простоті, читабельності та великій кількості доступних бібліотек, що робить його ідеальним вибором для розробки чат-ботів. В таблиці 3.1 подана порівняльна характеристика платформ для розробки чат-бота для вивчення іноземних мов.

Таблиця 3.1 – Порівняльна характеристика платформ для розробки чат-бота для вивчення іноземних мов

Назва бібліотеки	Опис	Переваги	Недоліки
PYTHON-TELEGRAM-BOT	Популярна бібліотека для створення Telegram-ботів на Python, яка надає зручний інтерфейс для роботи з Telegram Bot API.	Повна підтримка всіх функцій Telegram Bot API. Докладна документація та велика спільнота користувачів. Легкість у використанні завдяки високорівневому інтерфейсу.	Орієнтована лише на Telegram, що обмежує використання в інших месенджерах. Залежність від оновлень Telegram Bot API.
ChatterBot	Бібліотека для створення чат-ботів на основі машинного навчання. Вона дозволяє створювати боти, які можуть навчатися з діалогів.	Підтримка різних алгоритмів машинного навчання. Можливість навчання на власних даних. Легкість у налаштуванні та розширенні функціоналу.	Потребує значних обчислювальних ресурсів для навчання. Може бути складною для новачків у машинному навчанні
Rasa	Платформа з відкритим вихідним кодом для створення чат-ботів, яка включає в себе інструменти для обробки природної мови та управління діалогом.	Потужні інструменти для обробки природної мови (NLP). Можливість створення складних діалогових сценаріїв. Інтеграція з різними месенджерами та сервісами.	Висока складність налаштування та розгортання. Потребує знань у сфері NLP та машинного навчання.

Продовження таблиці 3.1

Назва бібліотеки	Опис	Переваги	Недоліки
Botpress	Платформа для створення чат-ботів з відкритим вихідним кодом, яка надає веб-інтерфейс для дизайну діалогів.	Зручний ВЕБ-інтерфейс для налаштування ботів. Інтеграція з різними месенджерами та ВЕБ-сайтами. Підтримка складних діалогів та інтерактивних елементів.	Вимагає окремого сервера для розгортання. Може бути складною для новачків.

На основі аналізу даних таблиці 3.1, стає очевидним, що для розробки чат-бота для вивчення іноземних мов оптимальним вибором є бібліотека Python-Telegram-bot. Це пояснюється її простотою у використанні, активною спільнотою та повною підтримкою всіх функцій Telegram Bot API, що робить її ідеальним інструментом для створення ефективного навчального чат-бота.

3.2 Обґрунтування вибору середовища розробки

В сучасному світі програмування вибір інструменту для редагування коду або розробки програм має величезне значення для продуктивності та задоволення від процесу розробки. Серед найпопулярніших редакторів інтегрованого середовища розробки (IDE) можна виділити Visual Studio Code, PyCharm та Sublime Text. Кожен з них має свої переваги та недоліки, які варто врахувати при виборі. Розглянемо їх детально.

Visual Studio Code (VS Code) [17]

Основні переваги:

- Інтеграція з Git: VS Code має вбудовану підтримку Git, це дозволяє легко взаємодіяти з репозиторіями.
- Розширюваність: одна з найбільших переваг VS Code – це можливість розширення за допомогою розширень. Існує велика кількість розширень для

різноманітних завдань, таких як підтримка різних мов програмування, відлагодження, робота з Docker тощо.

- **Спільнота та підтримка:** VS Code має велику та активну спільноту користувачів. Ви зможете знайти відповіді на більшість своїх питань через документацію, форуми або соціальні мережі.
- **Легкість налаштування:** VS Code легко налаштовується під потреби користувача. Ви можете змінювати теми, налаштовувати клавіатурні скорочення, вибирати редактори коду та багато іншого.
- **Мовна підтримка:** підтримка багатьох мов програмування, включаючи Python, JavaScript, TypeScript, Java, C++, PHP та інші.
- **Інтеграція з середовищами виконання:** VS Code підтримує підключення до серверів і віртуальних машин для розробки у різних середовищах.

Основні недоліки:

- **Високі вимоги до ресурсів:** у порівнянні з легшими текстовими редакторами, такими як Sublime Text, VS Code використовує більше ресурсів комп'ютера.
- **Швидкодія:** хоча VS Code є досить швидким для багатьох завдань, деякі користувачі відзначають, що швидкодія може бути меншою, ніж у більш спеціалізованих редакторів або IDE (рис. 3.1).

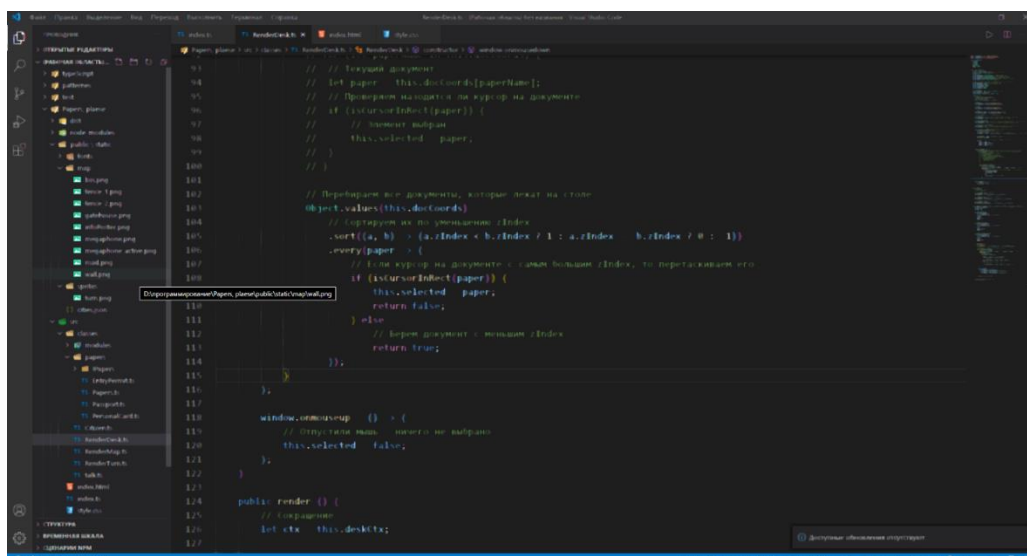


Рисунок 3.1 – Інтерфейс Visual Studio Code (VS Code)

PyCharm [18]

Основні переваги:

- Спеціалізація на Python: PyCharm розроблений спеціально для роботи з Python і має великий набір інструментів, призначених для полегшення розробки на цій мові.
- Інтегроване відлагодження: підтримка відлагодження з можливістю встановлення точок зупинки, інспекції змінних тощо.
- Кодові інспектори: інструменти для аналізу коду, виявлення помилок та покращення якості коду.
- Підтримка віртуальних середовищ: підтримка створення та управління віртуальними середовищами (virtual environments) для Python.
- Широкий набір інструментів: інтеграція з Django, Flask, NumPy, SciPy тощо.
- Підтримка багаторівневих проектів: підтримка розробки багаторівневих проектів з розширеними можливостями організації коду.

Основні недоліки:

- Високі вимоги до ресурсів: PyCharm потребує значно більше пам'яті та процесорного часу, ніж легші текстові редактори.
- Вартість: повна версія PyCharm є платною, хоча є і безкоштовна версія для спрощеної роботи (рис. 3.2).

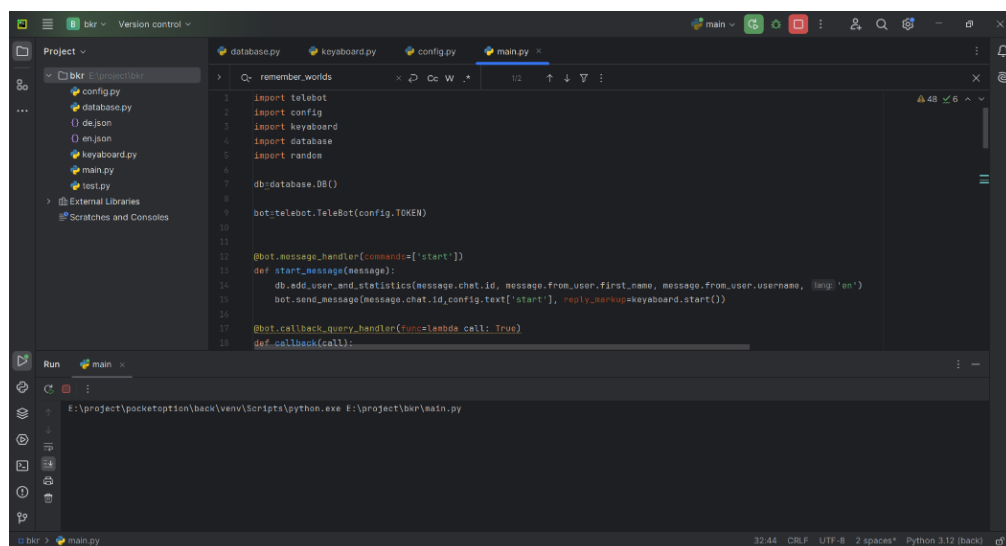


Рисунок 3.2 – Інтерфейс PyCharm

Sublime Text [19]

Основні переваги:

- Швидкодія: одна з головних переваг Sublime Text – це його швидкодія. Він легкий і швидкий в порівнянні з іншими редакторами.
- Розширюваність: Sublime Text підтримує розширення (плагіни), які значно покращують його функціонал. За допомогою плагінів можна розширити підтримку мов програмування, функції редактора та інше.
- Керування проектами: підтримка роботи з багаторівневими проектами та можливість налаштування різних налаштувань проекту.
- Підтримка мов програмування: підтримує багато мов програмування, хоча вона менш високорівнева порівняно з IDE, такими як PyCharm або VS Code.

Основні недоліки:

- Відсутність спеціалізованих інструментів для розробки: у порівнянні з PyCharm, Sublime Text не має таких спеціалізованих інструментів для розробки на Python або інших мовах програмування.
- Активність розробки: розробка Sublime Text не така активна, як у VS Code, що може вплинути на швидкість впровадження нових функцій та виправлення помилок (рис. 3.3).

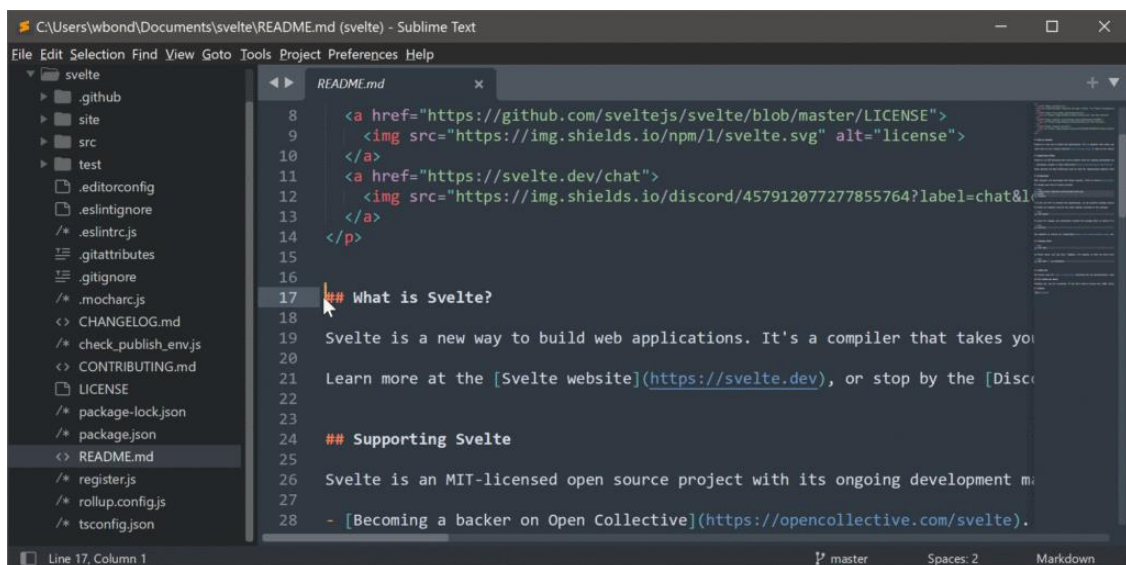


Рисунок 3.3 – Інтерфейс Sublime Text

На основі аналізу можна зробити висновок, що кожен з редакторів має свої сильні та слабкі сторони, які залежать від конкретних потреб розробника.

Visual Studio Code – ідеальний варіант для загального використання, має великий вибір розширень і активну спільноту користувачів.

PyCharm – найкращий вибір для розробки на Python, пропонує широкий набір спеціалізованих інструментів, хоча вимагає більше ресурсів комп'ютера.

Sublime Text – надзвичайно швидкий та легкий, з високим рівнем розширюваності, але може бути менш прийнятним для серйозної розробки на певних мовах програмування, таких як Python.

На основі проведеного аналізу було прийнято рішення вибрати PyCharm для розробки на Python. Це рішення обумовлене такими факторами: PyCharm має вищі вимоги до ресурсів і вартість у порівнянні з іншими редакторами. Однак його можливості та інструменти для розробки на Python переважають ці недоліки, це робить його найкращим вибором для розробки телеграм-бота для вивчення іноземних мов.

3.3 Обґрунтування вибору бази даних для вивчення іноземних мов

Для розробки інтелектуального чат-бота для вивчення іноземних мов важливо обрати відповідну базу даних, яка буде ефективною, надійною і прийнятною для специфічних вимог проекту. Розглянемо три популярні бази даних: MongoDB, MySQL [20] та PostgreSQL [21], щоб зробити обґрунтований вибір на користь MySQL.

MongoDB є NoSQL базою даних, яка використовує документоорієнтований підхід до зберігання даних [22].

Переваги:

-Гнучкість структури даних: можливість зберігання даних в JSON-подібних документах, що дозволяє легше змінювати схему даних.

-Горизонтальне масштабування: легко масштабувати дані за рахунок шардінгу.

Недоліки:

-Відсутність складних транзакцій: підтримка транзакцій на рівні одного документа, що може бути недоліком для складних операцій.

-Використання пам'яті: може використовувати більше пам'яті порівняно з реляційними базами даних через зберігання документів у вигляді BSON.

PostgreSQL є потужною реляційною базою даних з великими можливостями.

Переваги:

-Розширені функції: підтримка складних запитів, транзакцій, реплікації та масштабування.

-Підтримка JSON: можливість роботи з JSON-типами даних, що дозволяє використовувати гібридний підхід до зберігання даних.

-Надійність та консистентність: високий рівень надійності даних та забезпечення цілісності.

Недоліки:

-Складність налаштувань: висока гнучкість налаштувань може бути складною для початківців.

-Швидкодія: може бути повільнішою для деяких операцій порівняно з іншими СУБД через забезпечення консистентності.

MySQL є однією з найпопулярніших реляційних баз даних [20].

Переваги:

-Широка підтримка: підтримується багатьма хостинг-провайдерами та має велике ком'юніті.

-Простота використання: легка в налаштуванні та управлінні, підходить для різних рівнів досвіду користувачів.

-Висока швидкодія: оптимізована для швидкого читання та запису даних.

-Підтримка транзакцій: підтримка ACID-транзакцій забезпечує надійність та консистентність даних.

Недоліки:

-Менш гнучка схема: зміни у схемі можуть вимагати значних зусиль порівняно з NoSQL рішеннями.

-Обмеження масштабування: вертикальне масштабування (збільшення потужності одного сервера) є основним методом масштабування, що може бути обмеженням для дуже великих проєктів.

Зважаючи на вимоги проєкту, який полягає у розробці телеграм-бота для вивчення іноземних мов, вибір на користь MySQL є виправданим з таких причин:

1. Простота використання та налаштування: MySQL підходить для проєктів різного рівня складності і не потребує глибоких знань для початкового налаштування.

2. Підтримка транзакцій та ACID властивостей: це забезпечує цілісність та надійність даних, що важливо для освітнього контенту.

3. Висока швидкодія: оптимізована для швидкого доступу до даних, що покращить продуктивність чат-бота.

4. Широка підтримка та ком'юніті: легко знайти ресурси та допомогу у разі виникнення проблем.

Таким чином, MySQL є оптимальним вибором для розробки телеграм-бота для вивчення іноземних мов, забезпечуючи баланс між продуктивністю, надійністю та легкістю використання.

Оскільки буде використана СУБД MySql, потрібно розробити ER-модель (рис. 3.4) та ER-діаграму бази даних телеграм-бота для вивчення іноземних мов (рис. 3.5).

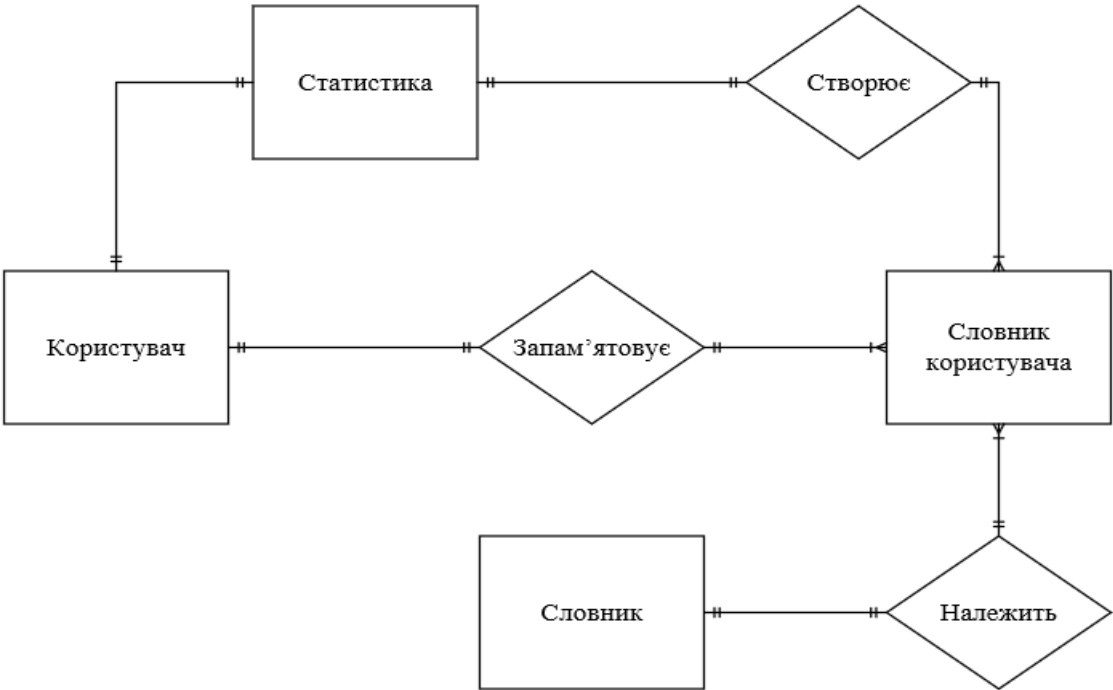


Рисунок 3.4 – ER-модель бази даних телеграм-бота
для вивчення іноземних мов

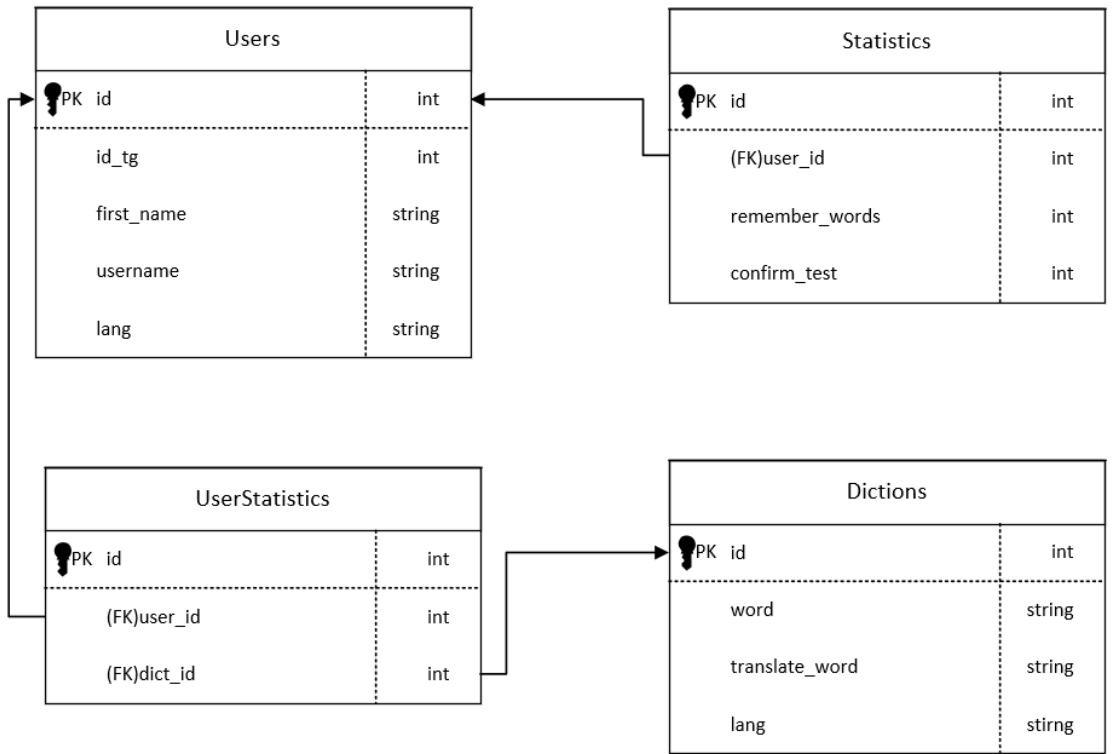


Рисунок 3.5 – ER-діаграма бази даних телеграм-бота
для вивчення іноземних мов

3.4 Тестування роботи програми та аналіз результатів

Останніми роками технології розробки програмного забезпечення стали ключовими для різних галузей комп'ютерних наук, допомагаючи долати складнощі, характерні для сучасних програмних систем. Тестування, як процес своєчасного виявлення помилок та дефектів, порівнює фактичний стан і поведінку продукту зі специфікацією. Важливо розрізняти тестування програмного забезпечення та забезпечення його якості, яке охоплює всі аспекти бізнес-процесу, а не тільки тестування.

Було проведено детальне тестування розробленого телеграм-бота для вивчення іноземних мов (понад 100 тестових запусків). Після відкриття силки на чат-бота для вивчення іноземних мов з'явиться привітальне вікно, де знаходиться кнопка /start (рис. 3.6).

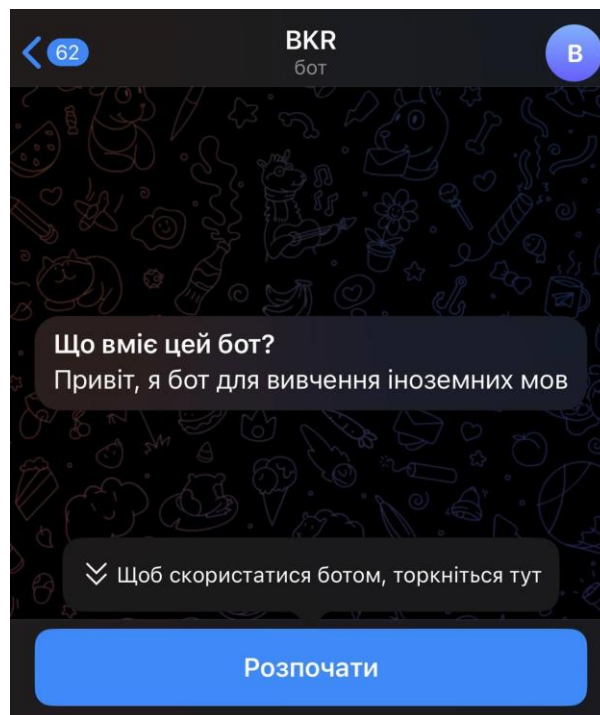


Рисунок 3.6 – Привітальне вікно телеграм-бота для вивчення іноземних мов

Після натискання кнопки /start чат-бот пропонує користувачеві вибрати мову для вивчення (рис. 3.7). Це є першим кроком у налаштуванні роботи з чат-

ботом, оскільки мова, яку вибере користувач, визначатиме інтерфейс і комунікацію з чат-ботом надалі. Вибір мови важливий для забезпечення комфортного використання і максимальної зручності під час навчання.

Чат-бот надає такі варіанти мов: «Англійську» та «Німецьку», серед яких користувач може вибрати ту, яка йому найбільше потрібна або, яку він хоче вивчати. Крім того, у налаштуваннях чат-бота передбачена можливість зміни вибраної мови в будь-який момент (рис. 3.8). Це дозволяє користувачам легко адаптувати інтерфейс до своїх потреб або перейти до вивчення іншої мови без необхідності створювати новий акаунт.

Такий підхід забезпечує гнучкість та зручність використання чат-бота, дозволяючи кожному користувачеві налаштувати його відповідно до своїх індивідуальних уподобань та потреб. Завдяки цьому чат-бот стає корисним інструментом для навчання, який може адаптуватися до різних ситуацій та вимог користувачів.

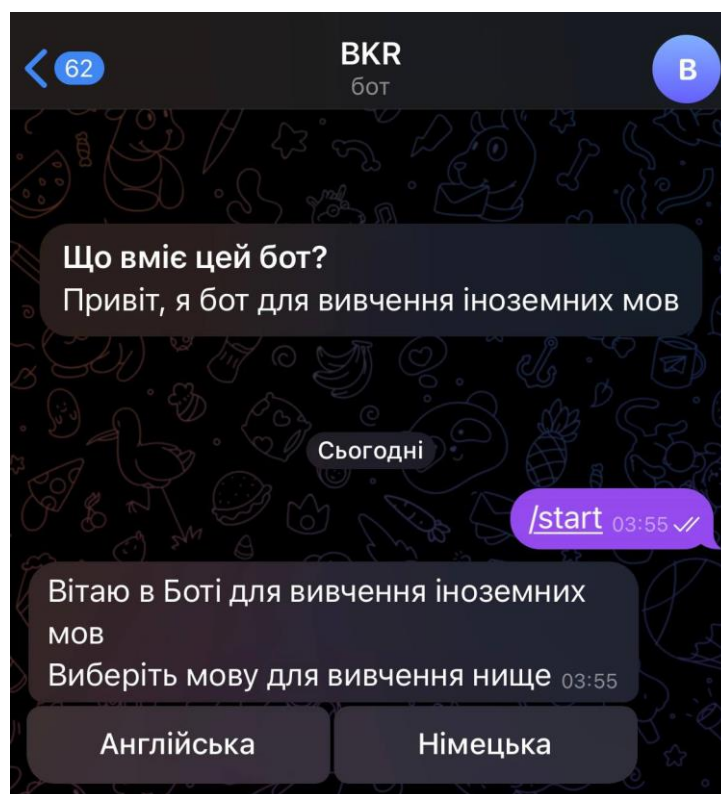


Рисунок 3.7 – Вибір мов для вивчення іноземних мов

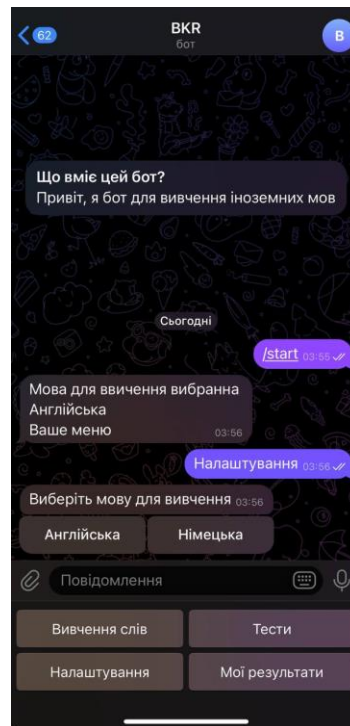


Рисунок 3.8 – Вибір мов для вивчення іноземних мов за допомогою кнопки «Налаштування»

Після вибору мови для вивчення з'являються кнопки: "Вивчення слів", "Тести", "Налаштування" та "Мої результати" (рис. 3.9).

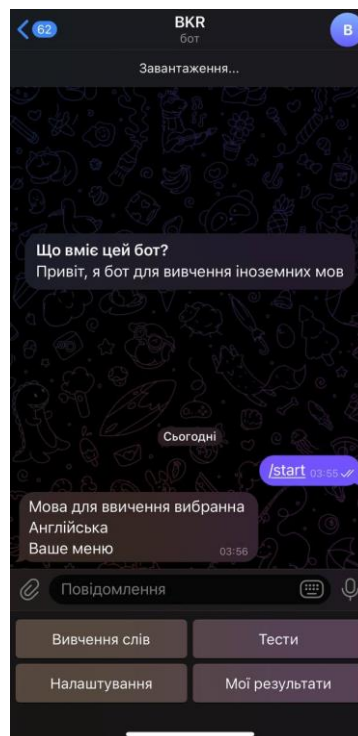


Рисунок 3.9 – Після вибору мови

При натисканні на кнопку "Вивчення слів" з'являються інлайн-кнопки "Вивчені слова" та "Вивчення нових слів" (рис 3.10).

При натисканні інлайн-кнопки "Вивчені слова" з'являються останні 20 слів, які користувач вже вивчив, це допомагає йому їх повторювати та закріплювати у пам'яті (рис 3.11).

Натисканні інлайн-кнопки "Вивчення нових слів" дозволяє користувачеві ознайомитися з новими словами, які він ще не вивчав за допомогою чат-бота (рис. 3.12).

Після натискання кнопки "Запам'ятав слово" нове слово автоматично додається до списку вивчених слів, що спрощує процес навчання та забезпечує систематизацію вивчених матеріалів. Таким чином, користувач має змогу легко повторювати вже відомі слова та поступово розширювати свій словниковий запас. Також є кнопка "Закрити" в двох випадках, щоб перейти до вибору.

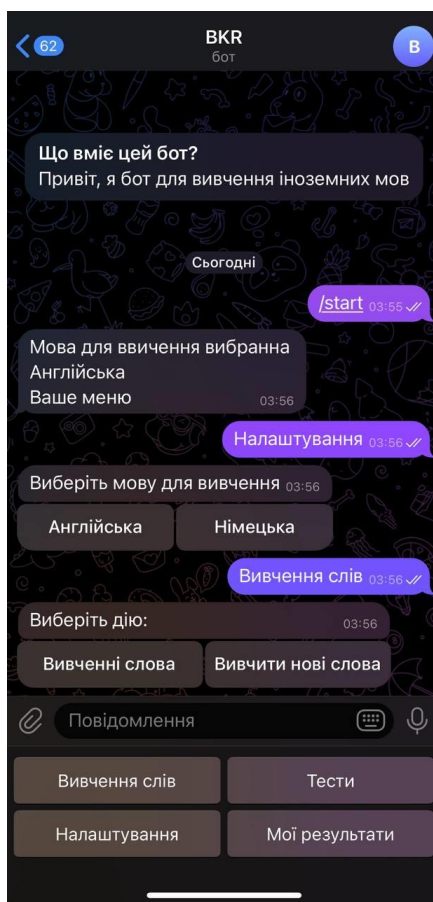


Рисунок 3.10 – Натиск на кнопку " Вивчення слів"

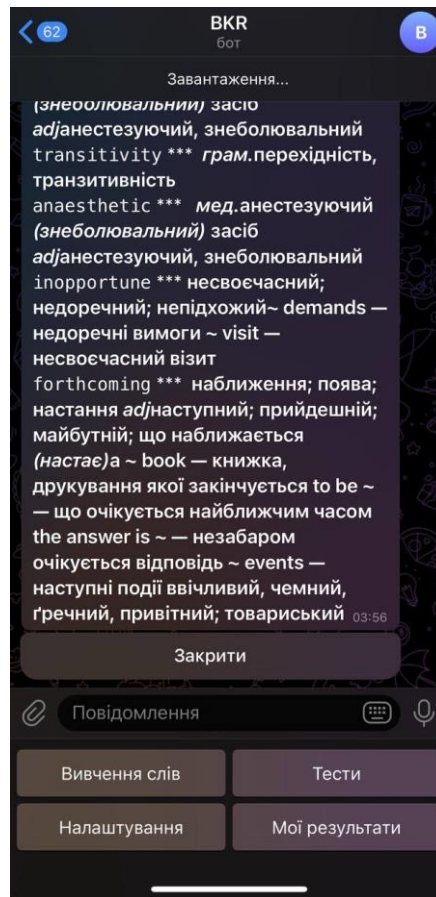


Рисунок 3.11 – Натиск на кнопку " Вивчені слова"

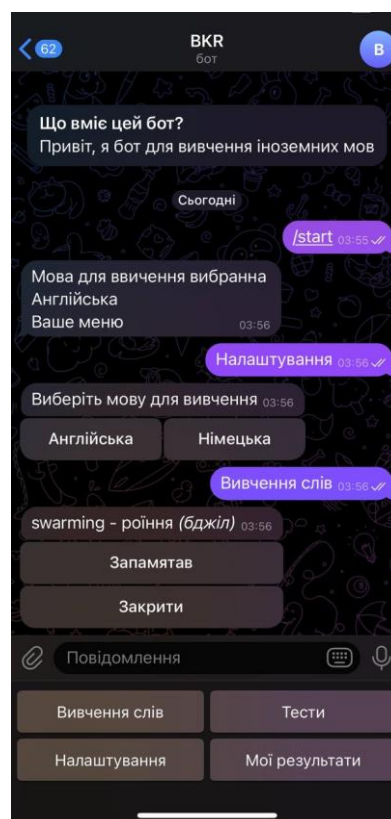


Рисунок 3.12 – Натиск на кнопку " Вивчення нових слів"

Кнопка "Тести" генерує випадковий тест зі слів, які користувач вже вивчив. Користувачу пропонується слово на іноземній мові та чотири варіанти перекладу, серед яких потрібно вибрати правильний (рис. 3.13).

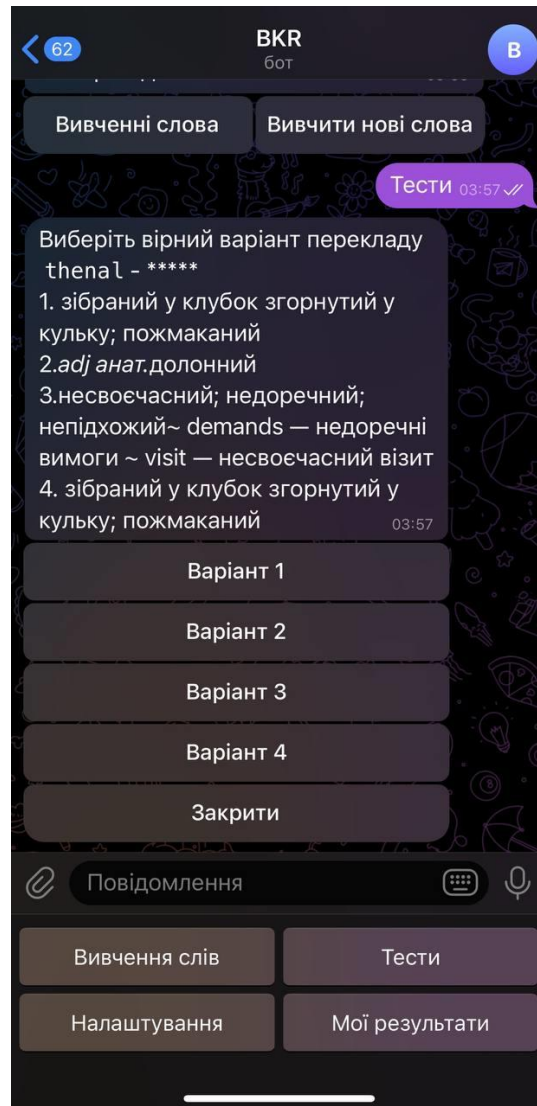


Рисунок 3.13 – Натиск на кнопку "Тест"

Такий формат тестування допомагає закріпити знання та перевірити рівень засвоєння нових слів. Крім того, це сприяє більш ефективному запам'ятовуванню та активному використанню вивченого словникового запасу. Результати тестів зберігаються, що дозволяє користувачеві відстежувати свій прогрес та виявляти слабкі місця для подальшого опрацювання.

Крім того, користувач може переглянути свою статистику, натиснувши на кнопку "Мої результати". Тут відображається кількість вивчених слів та кількість успішно пройдених тестів (рис. 3.14).

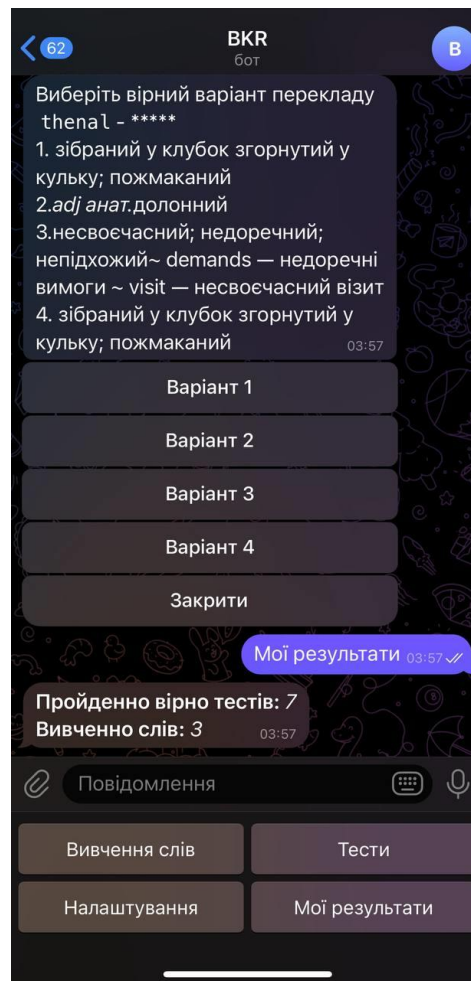


Рисунок 3.14 – Натиск на кнопку "Мої результати"

Ця інформація допомагає користувачеві відслідковувати свій прогрес у вивченні іноземної мови, мотивує до подальшого навчання та дозволяє визначити, які аспекти потребують додаткової уваги. Завдяки цьому користувач має змогу ефективніше планувати свій навчальний процес та досягати кращих результатів.

Розроблений інтелектуальний чат-бот для вивчення іноземних мов відрізняється від конкурентів такими ключовими особливостями:

- Гнучкість вибору мови: телеграм-бот надає можливість користувачам вибирати іноземну мову для вивчення (англійська чи німецька) під час першого

входу і змінювати її в будь-який момент без необхідності створення нового акаунту. Це робить чат-бота більш гнучким і зручним у використанні.

- Система вивчення слів: користувачі мають можливість вивчати нові слова, а також переглядати вивчені слова. Додатково вони можуть вказувати, що вже запам'ятали певне слово, це сприяє систематизації вивченого матеріалу та підвищує ефективність навчання.

- Тести: чат-бот генерує випадкові тести зі словникових одиниць, які користувач вже вивчив, що дозволяє перевірити рівень засвоєння матеріалу. Такий підхід допомагає активно використовувати вивчені слова і підвищує їхню запам'ятовуваність.

- Статистика і результати: користувачі можуть переглядати свої результати та статистику, що включає кількість вивчених слів та кількість успішно пройдених тестів. Це дозволяє відстежувати прогрес у навчанні і мотивує до подальшого вдосконалення.

- Зручність і доступність: чат-бот має інтуїтивно зрозумілий інтерфейс, який сприяє легко переміщуватися між різними функціями і налаштуваннями. Користувачам легко орієнтуватися і використовувати всі доступні можливості.

Отже, завдяки цим особливостям розроблений чат-бот стає потужним інструментом для навчання, який відповідає сучасним вимогам ефективного і зручного вивчення іноземних мов, забезпечуючи користувачам більш гнучкі та розширені можливості порівняно з іншими програмами і чат-ботами в цій сфері.

Розроблений телеграм-бот для вивчення іноземних мов успішно пройшов тестування та відповідає всім заданим вимогам.

В таблиці 3.2 подано результати тестування розробленого програмного продукту та аналогів.

В результаті розробки було реалізовано функціонал, який був відсутній в системах-аналогах, а саме:

- український інтерфейс;
- тести за вивченими словами;
- словник вивчених слів.

Таблиця 3.2 – Результати тестування розробленого програмного продукту та аналогів

	Розроблений модуль	Languagecenterbot	AI English Tutor by Skyeng	Duolingo	Babbel
Український інтерфейс	+	–	–	+	–
Тести за вивченими словами	+	+	–	–	+
Словник вивчених слів	+	+	+	+	–
Вивчення слів	+	+	+	+	+
Безкоштовний	+	+	+	–	–
Вивчення більше однієї мови	+	–	–	+	+

Отже, проаналізувавши та протестувавши розроблений програмний продукт, можна дійти висновку, що поставлених цілей було досягнуто.

З результатів таблиці 3.2 видно, що розроблений телеграм-бот для вивчення іноземних мов має покращений функціонал у порівнянні з системами-аналогами щонайменше на 3 можливості.

3.5 Висновок

На основі проведеного аналізу та тестування розробленого чат-бота для вивчення іноземних мов можна стверджувати, що поставлені цілі були досягнуті. Чат-бот є безкоштовним для користувачів і абсолютно без реклами, це робить його доступним і зручним для використання. Він підтримує кросплатформеність, це означає, що його можна використовувати на різних пристроях та операційних системах.

На основі аналізу найпопулярніших текстових редакторів: Visual Studio, Sublime Text, Pycharm для розробки телеграм-бота для вивчення іноземних мов обрано Pycharm. Такий вибір обумовлений тим, що PyCharm має великі можливості та інструменти для розробки телеграм-бота на мові Python.

Розроблено ER-модель бази даних телеграм-бота для вивчення іноземних мов, яка сприятиме кращому розумінню взаємозв'язків між сутностями бази даних і спростить проектуванню бази даних.

Проведено тестування телеграм-бота для вивчення іноземних мов та проведено порівняння його з програмами-аналогами. Встановлено, що розробка має розширений функціонал порівняно з аналогами щонайменше на 3 можливості, а саме: український інтерфейс; тести за вивченими словами; словник вивчених слів.

ВИСНОВКИ

Всі задачі, поставлені для реалізації телеграм-бота для вивчення іноземних мов були виконані в повному обсязі, а саме: обґрунтовано доцільність розробки телеграм-бота для вивчення іноземних; спроектовано телеграм-бот для вивчення іноземних; програмно реалізовано телеграм-бот для вивчення іноземних; виконано тестування програми та проведено аналіз отриманих результатів; розроблено інструкцію користувача.

Для розробки телеграм-бота для вивчення іноземних була використана мова програмування Python, яка забезпечує гнучкість і широкий вибір бібліотек для реалізації різноманітних функціоналів. Базою даних була обрана MySQL, для якої були сформульовані необхідні таблиці для зберігання даних користувачів, їхнього прогресу та словникового запасу. Платформа Telegram була обрана як оптимальний варіант для реалізації, оскільки вона відзначається зручністю використання, широким функціоналом, а також підтримкою чат-ботів, що дозволило інтегрувати інтелектуальні алгоритми в зручний для користувачів інтерфейс. Телеграм-бот для вивчення іноземних мов включає такі основні функції: вивчення нових словникових одиниць; проходження тестів для перевірки знань; можливість вибору мови для вивчення.

Розроблений телеграм-бот для вивчення іноземних мов успішно пройшов тестування та відповідає всім заданим вимогам. В порівнянні з системами-аналогами надає користувачам щонайменше 3 додаткові можливості, що суттєво підвищує задоволення у використанні користувачами і робить його корисним інструментом для вивчення іноземних мов.

Мета роботи була досягнута шляхом розширення функціональних можливостей телеграм-бота для вивчення іноземних мов. Порівняно з аналогами, розроблений програмний продукт містить український інтерфейс; тести за вивченими словами, а також словник вивчених слів.

Отже, розроблений телеграм-бот для вивчення іноземних мов відповідає заданим вимогам.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Колесник Я. С., Крилик Л. В. Аналіз передумов розробки телеграм-бота для вивчення іноземних мов. *LIII Всеукраїнська науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації*. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2024/paper/view/20766/17238> (дата звернення: 06.05.2024)
2. Що таке боти в сервісі Telegram та як ними користуватися – TechToday. URL: <https://techtoday.in.ua/news/shho-take-boti-v-servisi-telegram-ta-yak-nimi-koristuvatisya-64222.html#> (дата звернення: 06.05.2024).
3. Reach of mobile messenger apps among Android users in Ukraine in June 2020, by platform. URL: <https://www.statista.com/statistics/1188579/most-popular-messengers-in-ukraine/> (дата звернення: 06.05.2024).
4. Discord. URL: <https://discord.com> (дата звернення: 06.05.2024).
5. Facebook. URL: <https://uk-ua.facebook.com/> (дата звернення: 02.06.2024).
6. Telegram. URL: <https://web.telegram.org> (дата звернення: 06.06.2024).
7. Програмний модуль. URL: https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D1%8C%D0%BD%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F (дата звернення: 06.05.2024).
8. AI English Tutor by Skyeng. URL: https://t.me/english_tester_sky_bot (дата звернення: 06.05.2024).
9. Language center. URL: <https://t.me/Languagecenterbot> (дата звернення: 10.06.2024).
10. Duolingo. URL: <https://uk.duolingo.com/> (дата звернення: 06.05.2024).
11. Babbel : <https://ua.babbel.com/> (дата звернення: 06.05.2024).
12. Діаграма класів. URL : <https://uk.wikipedia.org/wiki/%D0%94%D1%96%D0%B0%D0%B3%D1%80%D0>

%B0%D0%BC%D0%B0_%D0%BA%D0%BB%D0%B0%D1%81%D1%96%D0%B2 (дата звернення: 06.05.2024).

13. Діаграма компонентів. URL: https://uk.wikipedia.org/wiki/%D0%94%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0_%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D1%96%D0%B2 (дата звернення: 06.05.2024).

14. NodeJS. URL: <https://nodejs.org/en> (дата звернення: 06.05.2024).

15. Python. URL: <https://www.python.org/> (дата звернення: 06.05.2024).

16. C Sharp. URL: https://uk.wikipedia.org/wiki/C_Sharp (дата звернення: 06.05.2024).

17. Microsoft | Visual Studio. URL: <https://visualstudio.microsoft.com/en/> (дата звернення: 06.05.2024).

18. JetBrains PyCharm. URL: <https://www.jetbrains.com/pycharm/> (дата звернення: 02.06.2024).

19. Sublime Text. URL: <https://www.sublimetext.com/> (дата звернення: 06.05.2024).

20. MySQL. URL: <https://www.mysql.com/> (дата звернення: 06.05.2024).

21. PostgreSQL. URL: <https://www.postgresql.org/> (дата звернення: 06.05.2024).

22. MongoDB. URL: <https://www.mongodb.com/> (дата звернення: 06.05.2024).

ДОДАТКИ

ДОДАТОК А (обов'язковий)**Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень**Назва роботи: Розробка телеграм-бота для вивчення іноземних мовТип роботи: бакалаврська дипломна робота
(БДР, МКР)Підрозділ кафедра комп'ютерних наук, ФІІТА
(кафедра, факультет)**Показники звіту подібності Unicheck**Оригінальність 91,3% Схожість 8,7%

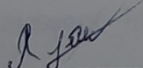
Аналіз звіту подібності (відмітити потрібне):

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку  Озеранський В.С.

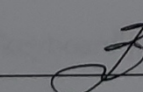
Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи



Колесник Я.С.

Керівник роботи



Крилик Л.В.

ДОДАТОК Б (обов'язковий)**Лістинг програми**

```
main.py
import telebot
import config
import keyboard
import database
import random

db=database.DB()

bot=telebot.TeleBot(config.TOKEN)

@bot.message_handler(commands=['start'])
def start_message(message):
    db.add_user_and_statistics(message.chat.id, message.from_user.first_name,
message.from_user.username, 'en')
    bot.send_message(message.chat.id,config.text['start'],
reply_markup=keyboard.start())

@bot.callback_query_handler(func=lambda call: True)
def callback(call):
    if call.data == 'en' or call.data == 'de':
        db.update_user_lang(call.message.chat.id, call.data)
        bot.delete_message(call.message.chat.id, message_id=call.message.message_id)
        bot.send_message(call.message.chat.id,
text=config.text['menu'].format(config.text['keyboard']['start'][call.data]),
reply_markup=keyboard.menu())
    elif call.data == 'learning_new_worlds':
```

```

lang = db.get_user_by_id_tg(call.message.chat.id)[4]
new_word = db.get_random_word_by_lang(lang)
bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=f'{new_word[0]} - {new_word[1]}',
reply_markup=keyaboard.remember_btn(new_word[2]), parse_mode='HTML')

elif call.data == 'close_learning':
    bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=config.text['learn_new_world'],
reply_markup=keyaboard.learn_new_world())
elif call.data == 'remember_worlds':
    text = config.text['your_dictoin']
    for words in db.get_last_50_user_diction_by_lang(call.message.chat.id):
        text+=config.text['remember_worlds'].format(words[2], words[3])

    bot.edit_message_text(chat_id=call.message.chat.id,
message_id=call.message.message_id, text=text,
reply_markup=keyaboard.return_to_choise_words(), parse_mode='HTML')
elif call.data == 'win_test':
    bot.answer_callback_query(callback_query_id=call.id, text=config.text['win_test'])
    db.add_confirm_test(call.message.chat.id)
    bot.delete_message(chat_id=call.message.chat.id,
message_id=call.message.message_id)
    dictoin = db.get_random_4_from_last_50(call.message.chat.id)
    if len(dictoin) < 4:
        bot.send_message(call.message.chat.id, text=config.text['error_test'])
    else:
        rand_diction = random.sample(dictoin, k=4)
        trust = random.choice(rand_diction)
        index = rand_diction.index(trust)

```

```

bot.send_message(chat_id=call.message.chat.id,
                  text=config.text['test'].format(trust[2], rand_diction[0][3],
                                                    rand_diction[1][3], rand_diction[2][3], rand_diction[3][3]),
                  reply_markup=keyboard.test(index),
                  parse_mode='HTML')

```

```

elif call.data == 'lose_test':
    bot.answer_callback_query(callback_query_id=call.id,
                              text=config.text['lose_test'])

```

```

elif call.data == 'menu':
    bot.delete_message(chat_id=call.message.chat.id,
                       message_id=call.message.message_id)
    lang = db.get_user_by_id_tg(call.message.chat.id)[4]
    bot.send_message(chat_id=call.message.chat.id,
                     text=config.text['menu'].format(config.text['keyboard']['start'][lang]),
                     reply_markup=keyboard.menu())

```

```

elif 'remember' in call.data:
    id_dictoin = call.data.split(':')[1]
    user = db.get_user_by_id_tg(call.message.chat.id)
    db.add_user_diction(user[0], id_dictoin)
    db.add_remember_word(user[0])
    new_word = db.get_random_word_by_lang(user[4])
    bot.edit_message_text(chat_id=call.message.chat.id,
                          message_id=call.message.message_id, text=f'{new_word[0]} - {new_word[1]}',
                          reply_markup=keyboard.remember_btn(new_word[2]), parse_mode='HTML')

```

```

@bot.message_handler(content_types=['text'])
def text(message):
    me_text = config.text['keyboard']
    if message.text == me_text['menu']['learn_new_world']:
        bot.send_message(message.chat.id, text=config.text['learn_new_world'],
reply_markup=keyaboard.learn_new_world())
    elif message.text == me_text['menu']['settings']:
        bot.send_message(message.chat.id, text=config.text['settings'],
reply_markup=keyaboard.start())
    elif message.text == me_text['menu']['tests']:
        dictoin = db.get_random_4_from_last_50(message.chat.id)
        if len(dictoin) < 4:
            bot.send_message(message.chat.id, text=config.text['eror_test'])
        else:
            rand_diction = random.sample(dictoin, k=4)
            trust = random.choice(rand_diction)
            index = rand_diction.index(trust)
            bot.send_message(chat_id=message.chat.id,
                            text=config.text['test'].format(trust[2], rand_diction[0][3],
rand_diction[1][3], rand_diction[2][3], rand_diction[3][3]),
reply_markup=keyaboard.test(index),
                            parse_mode='HTML')

    elif message.text == me_text['menu']['my_result']:
        statistics = db.get_statistics(message.chat.id)
        bot.send_message(message.chat.id,
text=config.text['my_result'].format(statistics[3], statistics[2]), parse_mode='HTML')

```

```
if __name__ == '__main__':
```

```
    db.create_table()
```

```
    bot.polling(non_stop=True)
```

```
keyboard.py
```

```
from telebot.types import InlineKeyboardMarkup, InlineKeyboardButton,
```

```
KeyboardButton, ReplyKeyboardMarkup
```

```
import config
```

```
text = config.text['keyboard']
```

```
def start():
```

```
    inline = InlineKeyboardMarkup()
```

```
    inline.add(InlineKeyboardButton(text=text['start']['en'], callback_data='en'),
```

```
    InlineKeyboardButton(text=text['start']['de'], callback_data='de'))
```

```
    return inline
```

```
def menu():
```

```
    button = ReplyKeyboardMarkup(row_width=2, resize_keyboard=True)
```

```
    my_text = text['menu']
```

```
    button.add(KeyboardButton(text=my_text['learn_new_world']),
```

```
    KeyboardButton(text=my_text['tests']))
```

```
    button.add(KeyboardButton(text=my_text['settings']),
```

```
    KeyboardButton(text=my_text['my_result']))
```

```
    return button
```

```

def learn_new_world():
    inline = InlineKeyboardMarkup()

    inline.add(InlineKeyboardButton(text=text['learn_new_world']['remember_worlds'],
    callback_data='remember_worlds'),

    InlineKeyboardButton(text=text['learn_new_world']['learning_new_worlds'],
    callback_data='learning_new_worlds'))

    return inline

def remember_btn(id_diction):
    inline = InlineKeyboardMarkup()
    inline.add(InlineKeyboardButton(text=text['remember'],
    callback_data=f'remember:{id_diction}'))
    inline.add(InlineKeyboardButton(text=text['close'],
    callback_data='close_learning'))

    return inline

def return_to_choose_words():
    inline = InlineKeyboardMarkup()
    inline.add(InlineKeyboardButton(text=text['close'],
    callback_data='close_learning'))

    return inline

def test(win_index):
    inline = InlineKeyboardMarkup(row_width=2)
    for n in range(4):

```

```

        inline.add(InlineKeyboardButton(text=f'Бапиаҥт {n+1}',
callback_data='win_test' if n == win_index else 'lose_test'))

```

```

        inline.add(InlineKeyboardButton(text=text['close'], callback_data='menu'))

```

```

    return inline

```

database.py

```

import mysql.connector

```

```

from config import db_host, db_user, db_password

```

```

import random

```

```

class DB():

```

```

    def __init__(self):

```

```

        self.connection = None

```

```

        self.connection = mysql.connector.connect(host=db_host, user=db_user,
password=db_password, database='bkr')

```

```

    def query(self, sql, args):

```

```

        cursor = self.connection.cursor()

```

```

        cursor.execute(sql, args)

```

```

        return cursor

```

```

    def create_table(self):

```

```

        cursor = self.query("""CREATE TABLE IF NOT EXISTS Diction (
            id MEDIUMINT NOT NULL AUTO_INCREMENT,
            world TEXT,
            translate_world TEXT,
            lang TEXT,

```

```

        PRIMARY KEY (id)
    );""", None)
self.connection.commit()
cursor.close()
cursor = self.query("""
    CREATE TABLE IF NOT EXISTS Users (
        id INT AUTO_INCREMENT PRIMARY KEY,
        id_tg VARCHAR(255) NOT NULL,
        first_name VARCHAR(255),
        username VARCHAR(255),
        lang VARCHAR(10)
    );
""", None)
self.connection.commit()
cursor.close()
cursor = self.query("""
    CREATE TABLE IF NOT EXISTS Statistics (
        id INT AUTO_INCREMENT PRIMARY KEY,
        id_user INT,
        remember_words INT DEFAULT 0,
        confirm_test INT DEFAULT 0,
        FOREIGN KEY (id_user) REFERENCES Users(id)
    );
""", None)
self.connection.commit()
cursor.close()
cursor = self.query("""
    CREATE TABLE IF NOT EXISTS UserDiction (
        id INT AUTO_INCREMENT PRIMARY KEY,
        user_id INT,

```

```

        dict_id MEDIUMINT,
        FOREIGN KEY (user_id) REFERENCES Users(id),
        FOREIGN KEY (dict_id) REFERENCES Diction(id)
    );
    """
    self.connection.commit()
    cursor.close()

```

```

def add_new_world(self, world, translate_world, lang):
    cursor = self.query("""INSERT INTO Diction (world, translate_world, lang)
                        VALUES (%s, %s, %s);""", (world, translate_world, lang))
    self.connection.commit()
    cursor.close()

```

```

def get_user_by_id_tg(self, id_tg):
    cursor = self.query("SELECT * FROM Users WHERE id_tg = %s", (id_tg,))
    result = cursor.fetchone()
    cursor.close()

```

```

    return result

```

```

def add_user_and_statistics(self, id_tg, first_name, username, lang,
remember_words=0, confirm_test=0):
    result = self.get_user_by_id_tg(id_tg)
    if result is None:
        cursor = self.query("""
            INSERT INTO Users (id_tg, first_name, username, lang)
            VALUES (%s, %s, %s, %s)
            """, (id_tg, first_name, username, lang))
        user_id = cursor.lastrowid

```

```

self.connection.commit()
cursor.close()
self.query("""
    INSERT INTO Statistics (id_user, remember_words, confirm_test)
    VALUES (%s, %s, %s)
    ON DUPLICATE KEY UPDATE
        remember_words = VALUES(remember_words),
        confirm_test = VALUES(confirm_test);
""", (user_id, remember_words, confirm_test))

```

```

self.connection.commit()
cursor.close()

```

```

def update_user_lang(self, id_tg, new_lang):

```

```

    self.query("""
        UPDATE Users
        SET lang = %s
        WHERE id_tg = %s;
""", (new_lang, id_tg))
    self.connection.commit()

```

```

def get_random_word_by_lang(self, lang):

```

```

    cursor = self.query("""
        SELECT world, translate_world, id
        FROM Diction
        WHERE lang = %s
        ORDER BY RAND()
        LIMIT 1;
""", (lang,))

```

```

result = cursor.fetchone()
cursor.close()

return result if result else None

def get_last_50_user_diction_by_lang(self, id_tg):
    user = self.get_user_by_id_tg(id_tg)
    cursor = self.query("""
        SELECT ud.id, u.id_tg, d.world, d.translate_world, d.lang
        FROM UserDiction ud
        JOIN Users u ON ud.user_id = %s
        JOIN Diction d ON ud.dict_id = d.id
        WHERE d.lang = %s
        ORDER BY RAND()
        LIMIT 20;
    """, (user[0], user[4],))
    result = cursor.fetchall()
    cursor.close()

    return result

def get_random_4_from_last_50(self, id_tg):
    lang_records = self.get_last_50_user_diction_by_lang(id_tg)
    random.shuffle(lang_records)
    return lang_records[:4]

def add_user_diction(self, user_id, dict_id):
    cursor = self.query("""
        INSERT INTO UserDiction (user_id, dict_id)
    """)

```

```

        VALUES (%s, %s);
        """ , (user_id, dict_id))
    cursor.close()

    self.connection.commit()

def get_last_50_user_diction(self):
    cursor = self.query("""
        SELECT *
        FROM UserDiction
        ORDER BY id DESC
        LIMIT 50;
        """, None)

    result = cursor.fetchall()
    return result

def get_statistics(self, id_tg):
    id_user = self.get_user_by_id_tg(id_tg)[0]
    cursor = self.query("""
        SELECT * FROM Statistics WHERE id_user = %s
        """, (id_user,))

    statistics = cursor.fetchone()

    cursor.close()

    return statistics

def add_confirm_test(self, id_tg):

```

```

id_user = self.get_user_by_id_tg(id_tg)[0]
cursor = self.query("""
    UPDATE Statistics SET confirm_test = confirm_test + 1 WHERE id_user =
%s
    """, (id_user,))
cursor.close()
self.connection.commit()

```

```

def add_remember_word(self, id_user):
    cursor = self.query("""
        UPDATE Statistics SET remember_words = remember_words + 1 WHERE
id_user = %s
        """, (id_user,))
    cursor.close()
    self.connection.commit()

```

```

def __del__(self):
    if self.connection != None:
        self.connection.close()

```

config.py

```
TOKEN = '7407067523:AAGl2_4-m4R1XTyy_OTyoJMpVua8iU-0ks4'
```

```
db_host = 'localhost'
```

```
db_user = 'root'
```

```
db_password = 'root'
```

```
db_port = 3306
```

```
text = {
```

```
    'keyboard' : {
```

```

'start': {
    'en': 'Англійська',
    'de': 'Німецька'
},
'menu': {
    'learn_new_world': 'Вивчення слів',
    'tests': 'Тести',
    'settings': 'Налаштування',
    'my_result': 'Мої результати'
},
'learn_new_world': {
    'remember_worlds': 'Вивчення слова',
    'learning_new_worlds': 'Вивчити нові слова'
},
'remember': 'Запам'ятав',
'close': 'Закрити'
},
'start': 'Вітаю в Боті для вивчення іноземних мов\n'
    'Виберіть мову для вивчення нижче',
'menu': 'Мова для вивчення вибрана {0}\n'
    'Ваше меню',
'learn_new_world': 'Виберіть дію:',
'settings': 'Виберіть мову для вивчення',
'remember_worlds': '<code>{0}</code> *** <b>{1}</b>\n',
'your_dictoin': 'Ваш словник\n',
'error_test': 'Ви ще не вивчили достатньо слів для проходження тестів',
'test': 'Виберіть вірний варіант перекладу\n <code>{0}</code> - *****\n'
    '1.{1}\n'
    '2.{2}\n'
    '3.{3}\n'

```

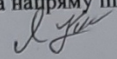
```
'4.{4}',  
'win_test': 'Вірний варіант',  
'lose_test': 'Не вірний варіант',  
'my_result': '<b>Пройденно вірно тестів: </b><i>{0}</i>\n'  
            '<b>Вивчено слів: </b><i>{1}</i>'  
}
```

ДОДАТОК В (обов'язковий)

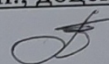
ГРАФІЧНА ЧАСТИНА

«РОЗРОБКА ТЕЛЕГРАМ-БОТА ДЛЯ ВИВЧЕННЯ
ІНОЗЕМНИХ МОВ»

Виконав: студент 4 курсу, групи ЗКН-206
спеціальності 122 – Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)

 Колесник Я. С.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

 Крилик Л. В.
(прізвище та ініціали)

« 07 » червня 2024 р.

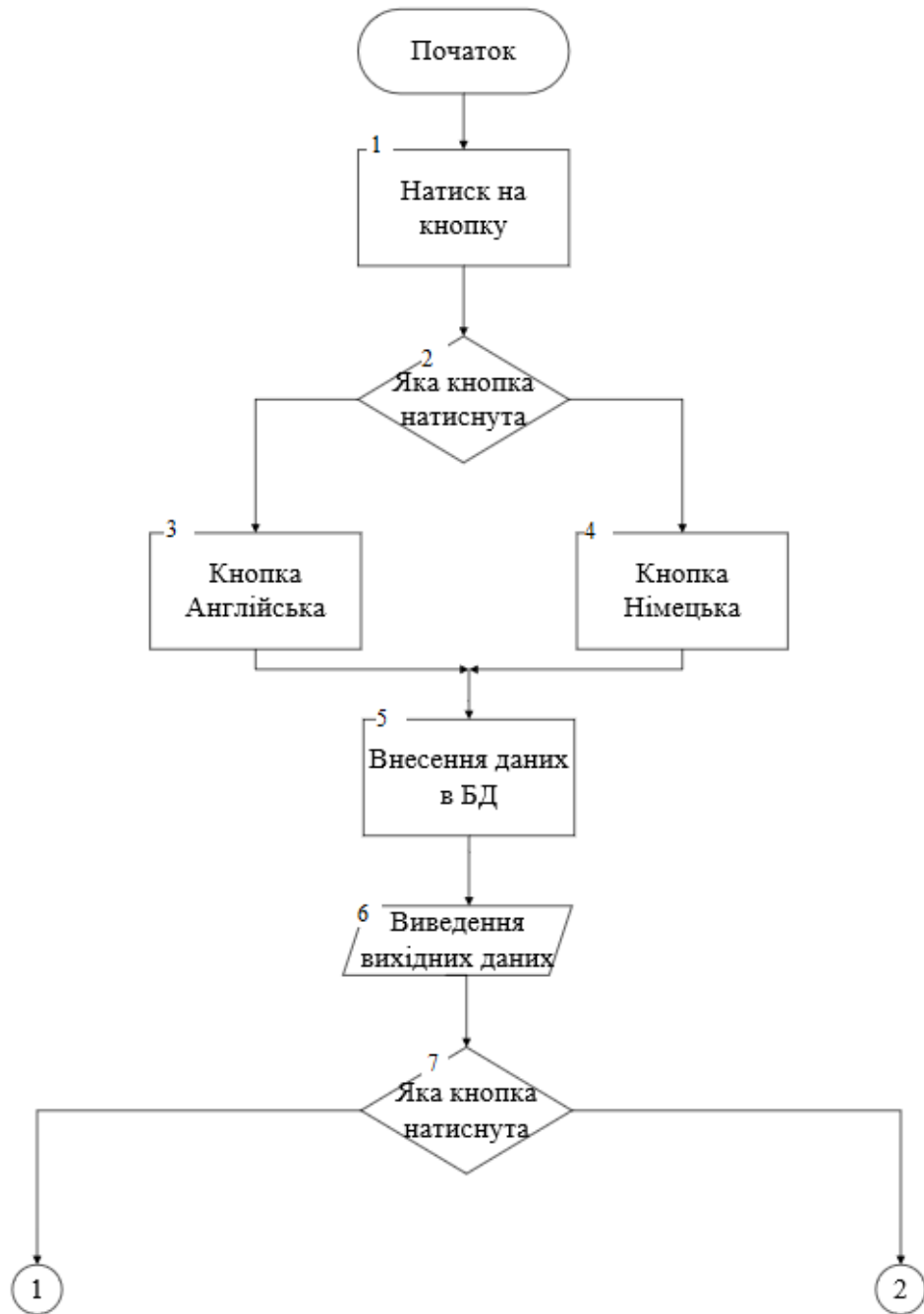


Рисунок В.1 – Схема алгоритму роботи телеграм-бота для вивчення іноземних мов

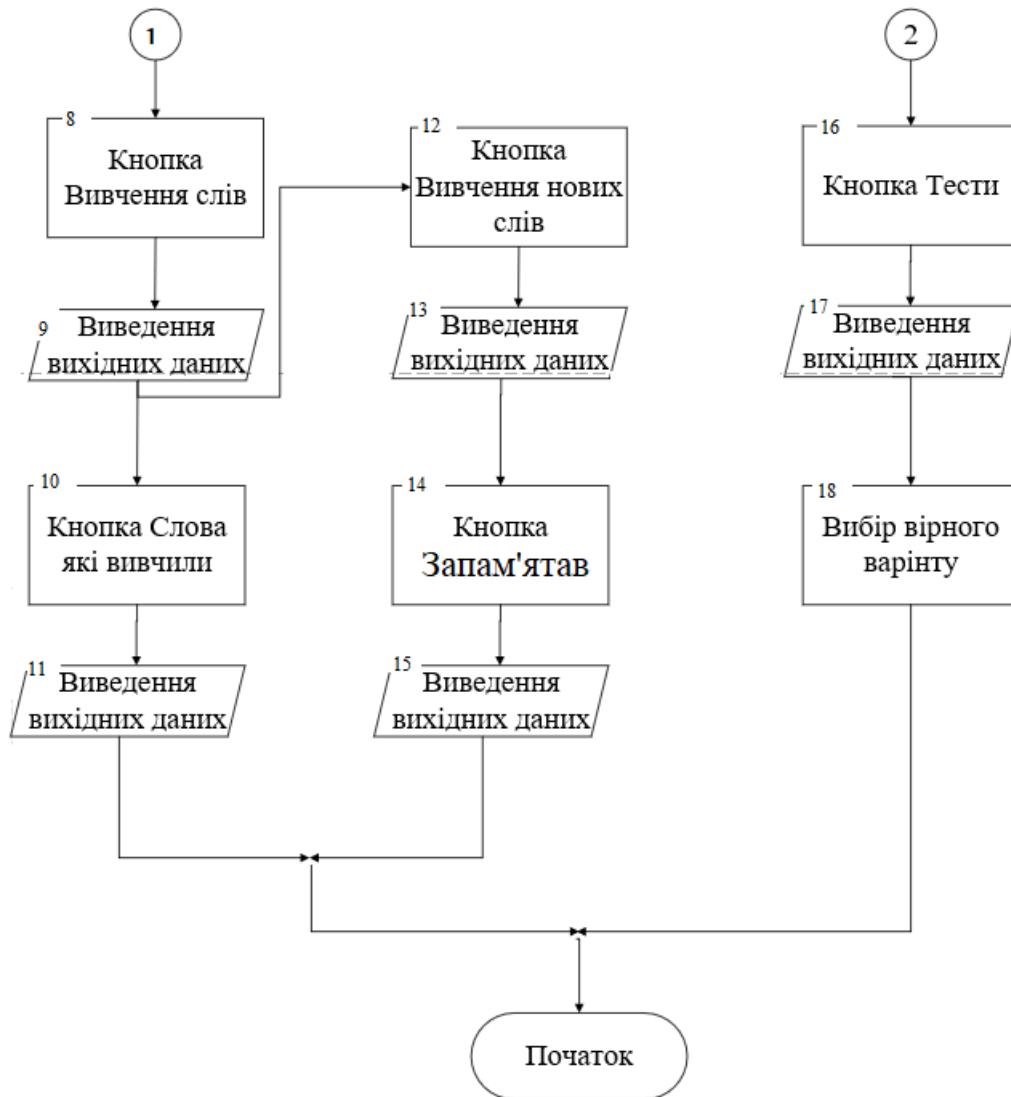


Рисунок В.1, аркуш 2

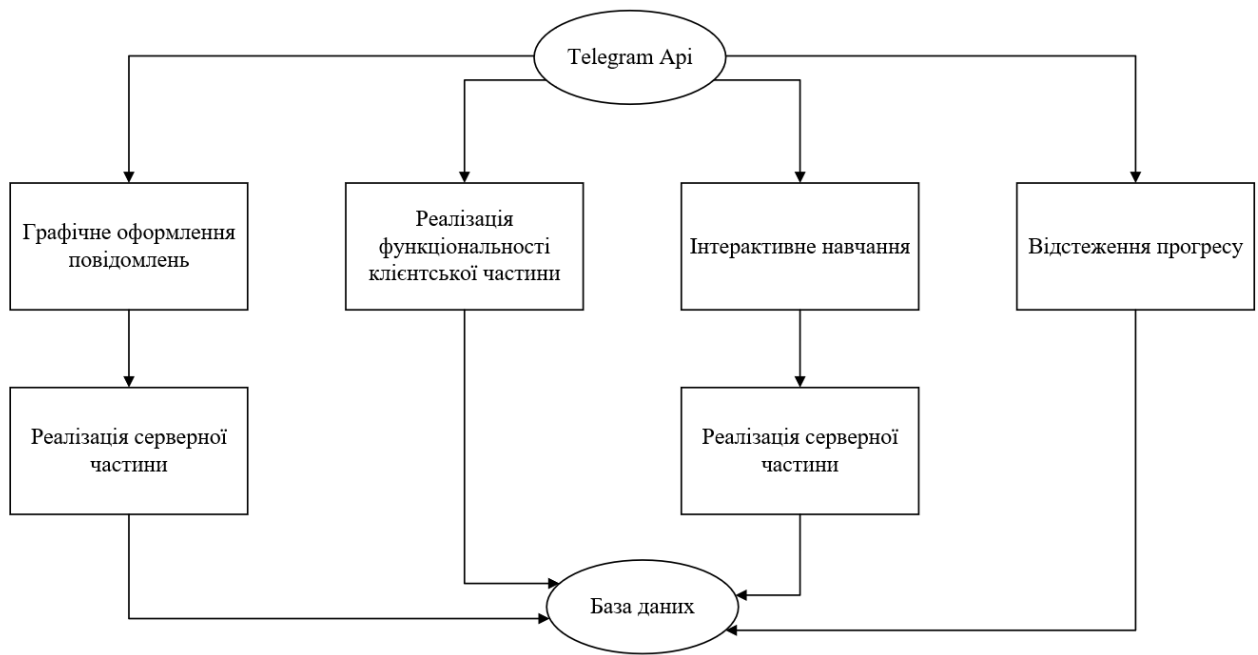


Рисунок В.2 – Загальна структурна схема функціонування системи

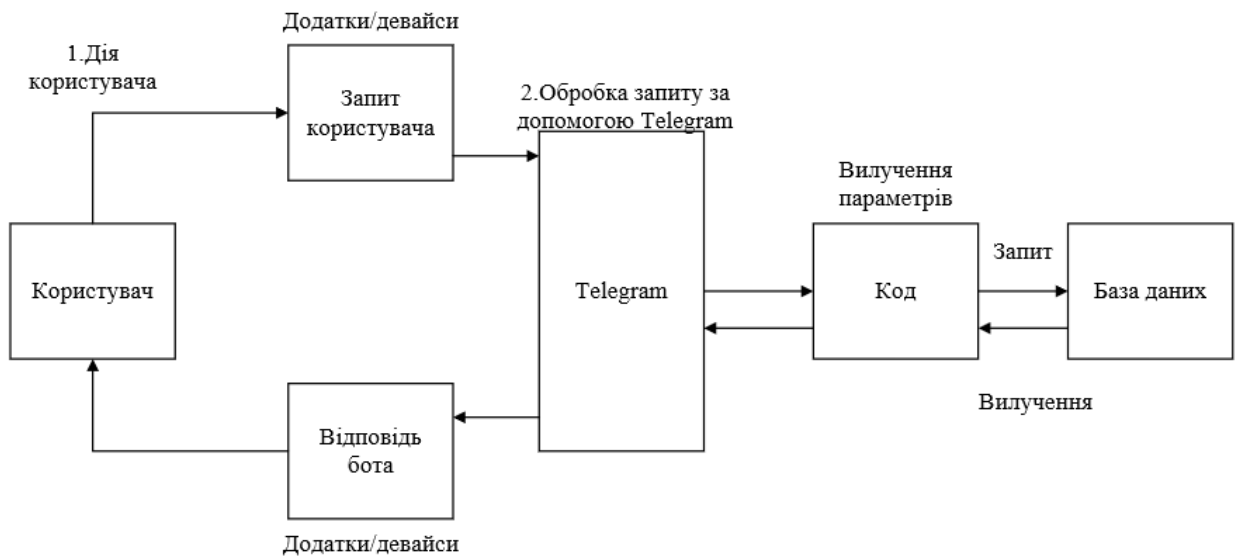


Рисунок В.3 – Структура принципу роботи телеграм-бота для вивчення іноземних мов

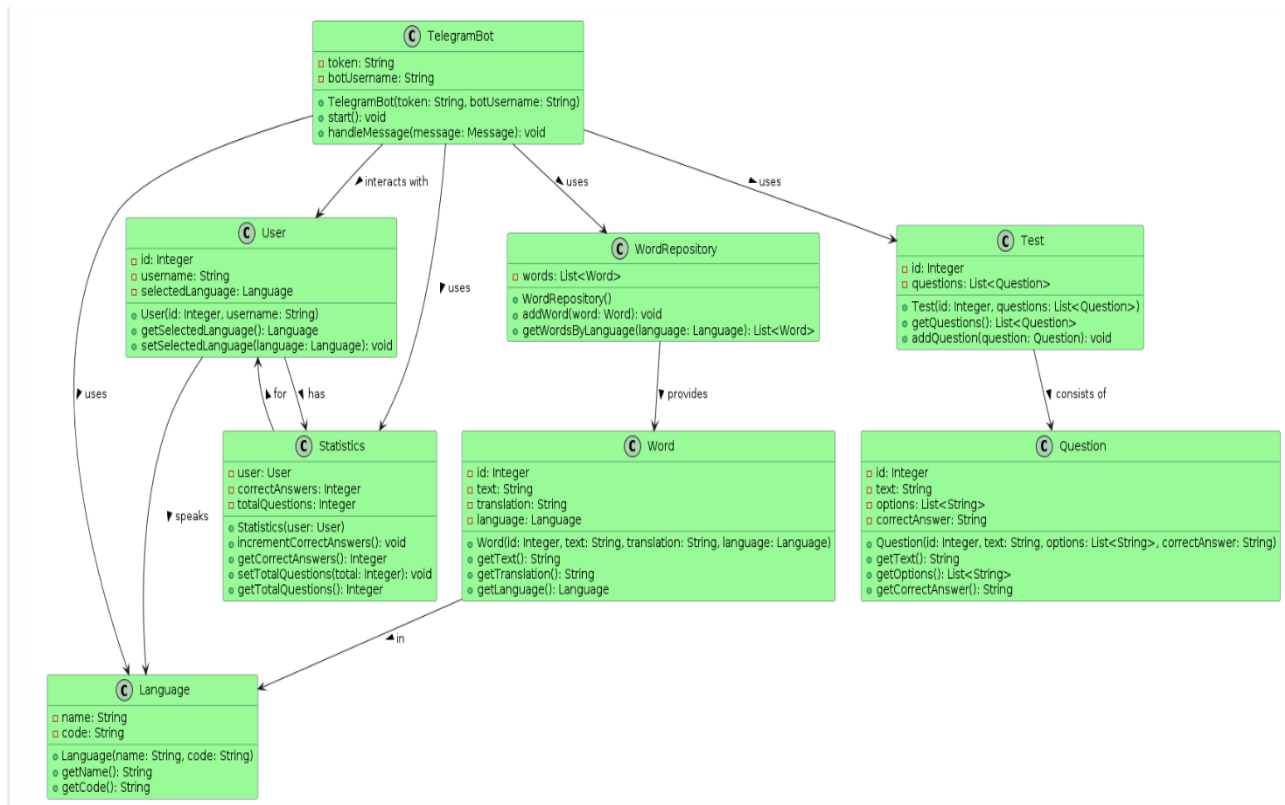


Рисунок В.4 – Загальна UML-діаграма класів

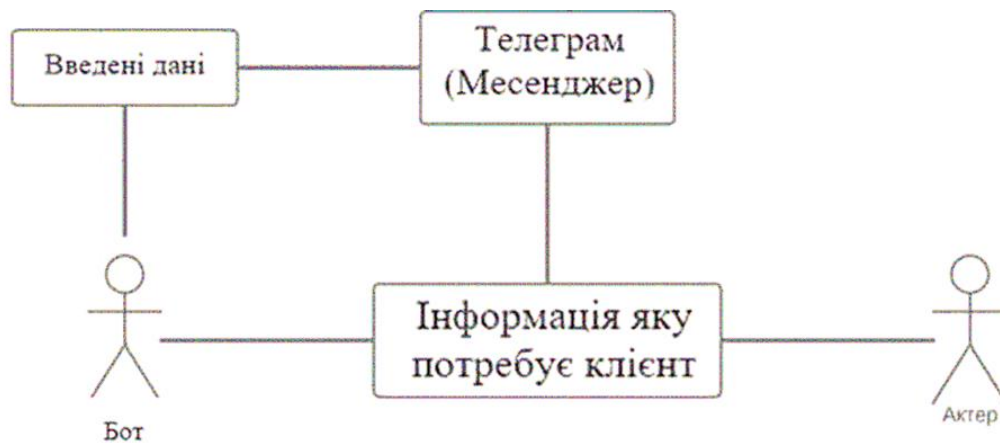


Рисунок В.5 – UML-діаграма комунікації системи чат-бота

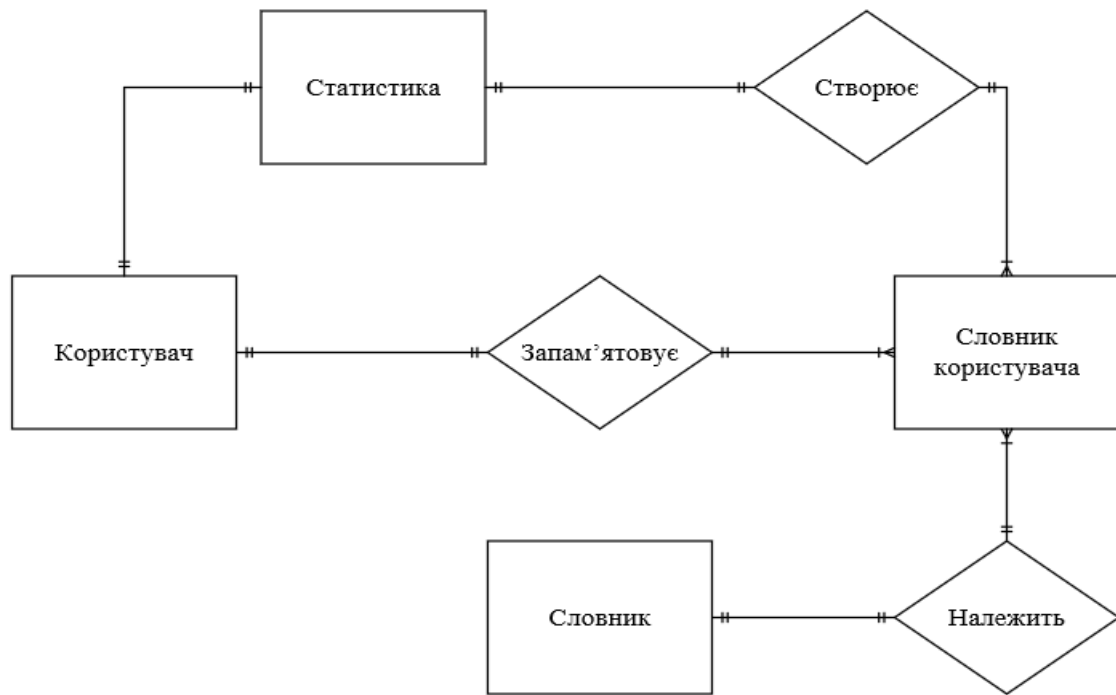


Рисунок В.6 – ER-модель бази даних телеграм-бота
для вивчення іноземних мов

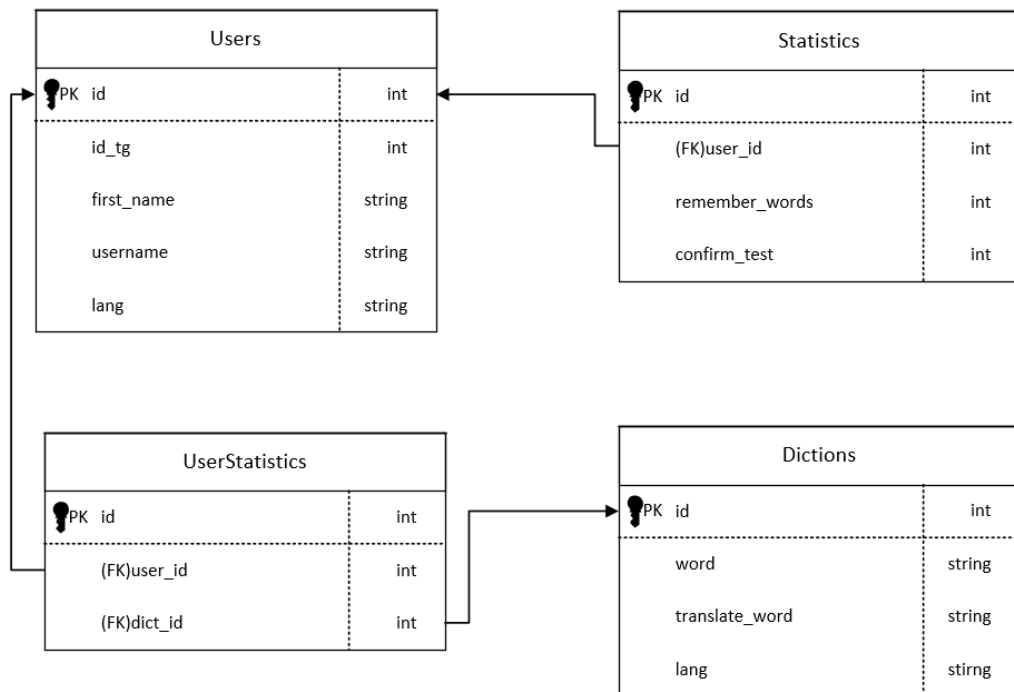


Рисунок В.7 – ER-діаграма бази даних телеграм-бота
для вивчення іноземних мов

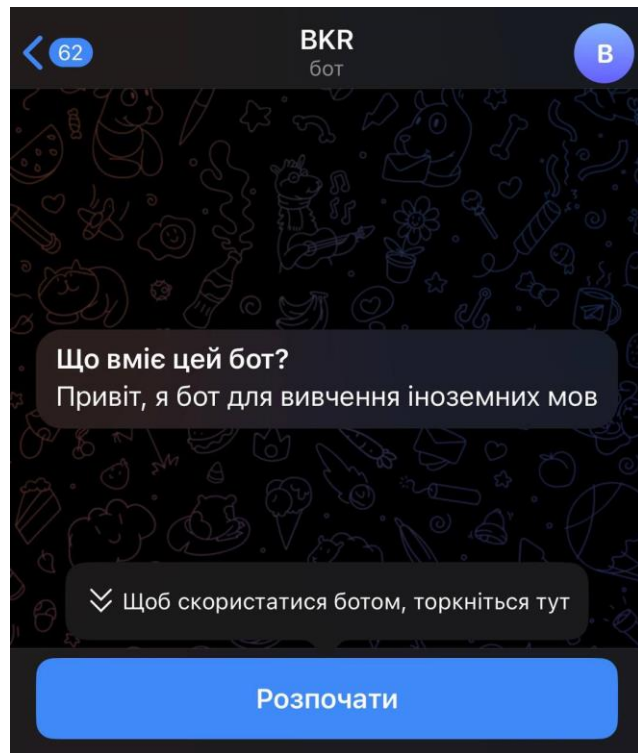


Рисунок В.8 – Привітальне вікно інтелектуального чат-бота для вивчення іноземних мов

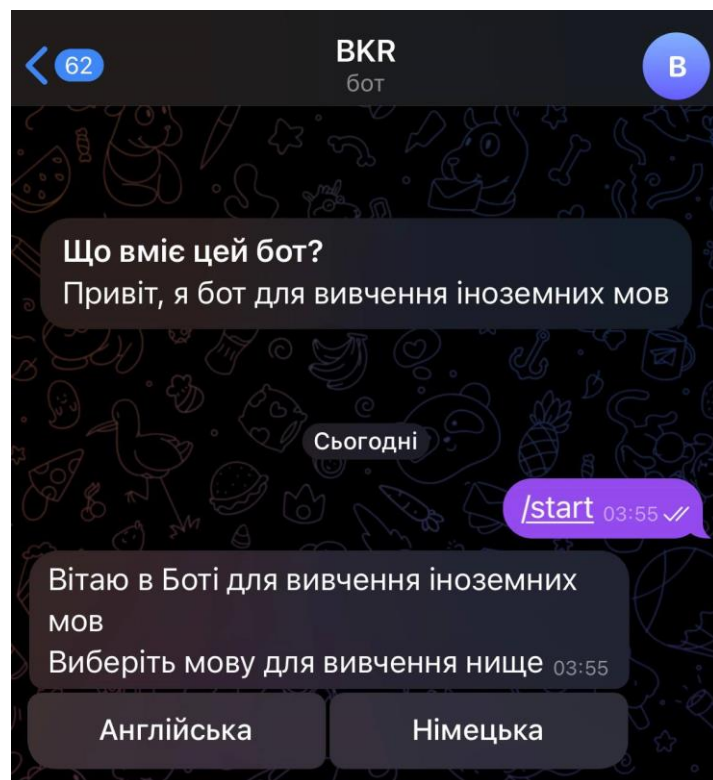


Рисунок В.9 – Вибір мов для вивчення іноземних мов

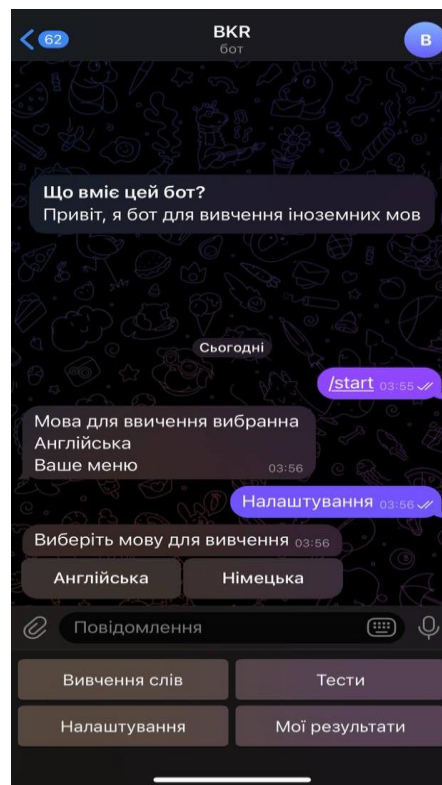


Рисунок В.10 – Вибір мов для вивчення іноземних мов за допомогою кнопки Налаштування

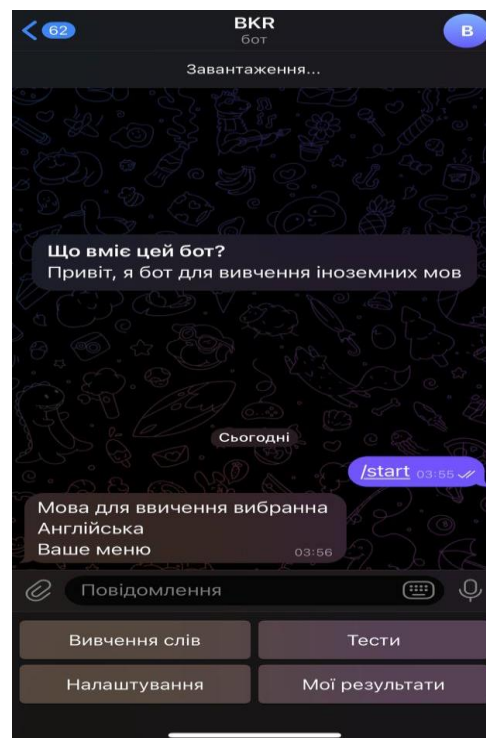


Рисунок В.11 – Після вибору мови

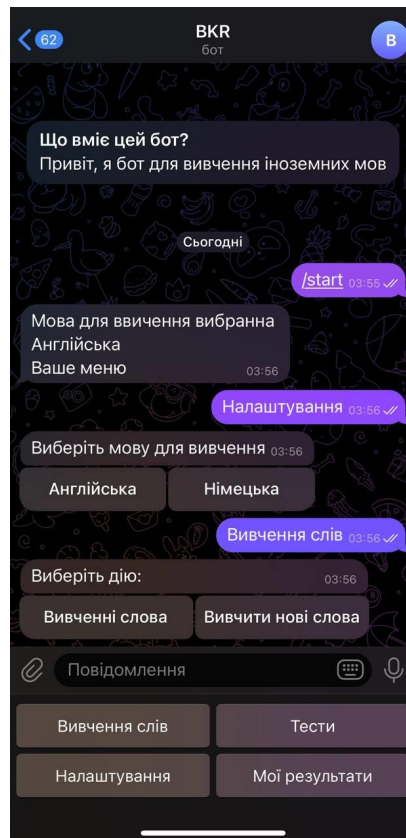


Рисунок В.12 – Натиск на кнопку " Вивчення слів"

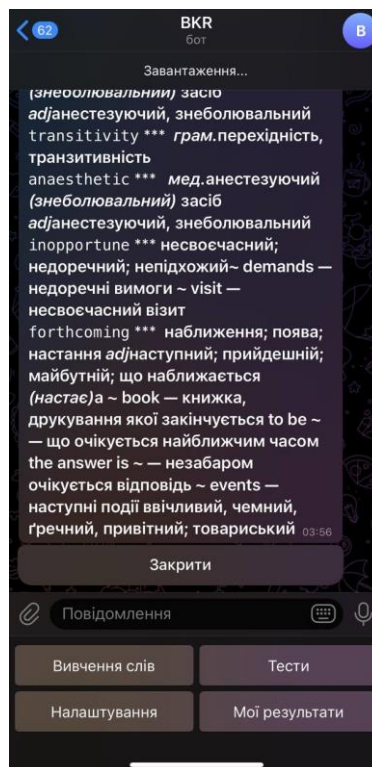


Рисунок В.13 – Натиск на кнопку " Вивчені слова"

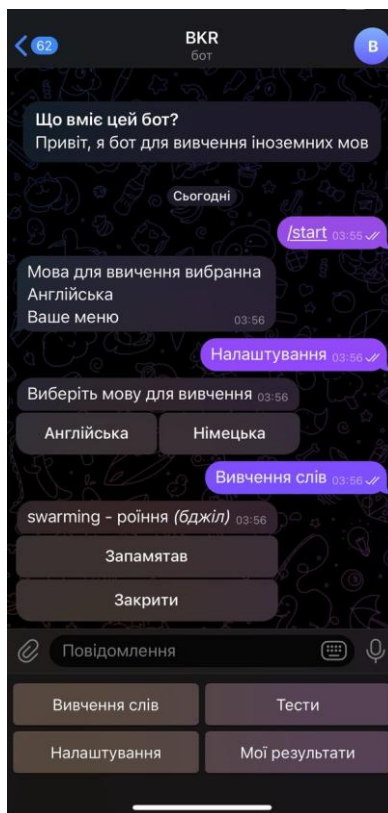


Рисунок В.14 – Натиск на кнопку " Вивчення нових слів"

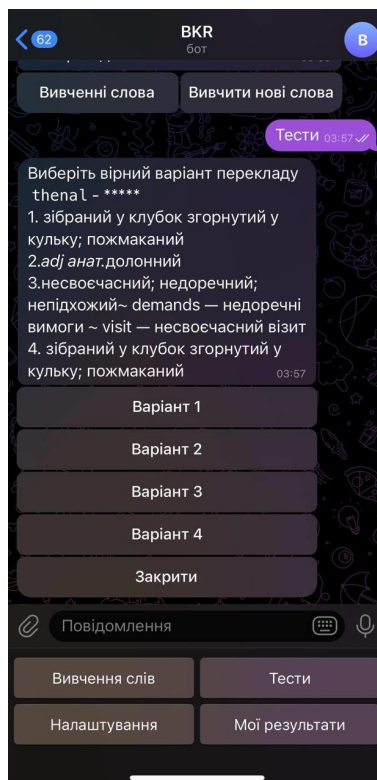


Рисунок В.15 – Натиск на кнопку "Тест"

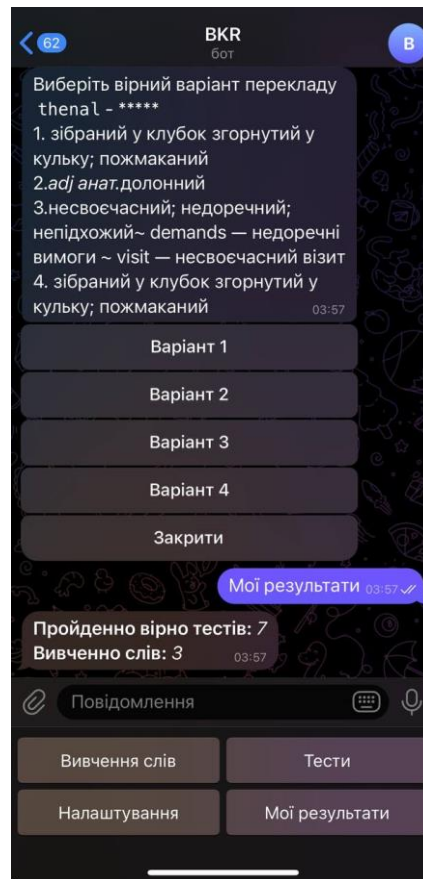


Рисунок В.16 – Натиск на кнопку " Мої результати "

ДОДАТОК Г (довідниковий)

Інструкція користувача

Після відкриття силки на чат-бот для вивчення іноземних мов з'явиться привітальне вікно, де знаходиться кнопка /start (рис. Г.1).

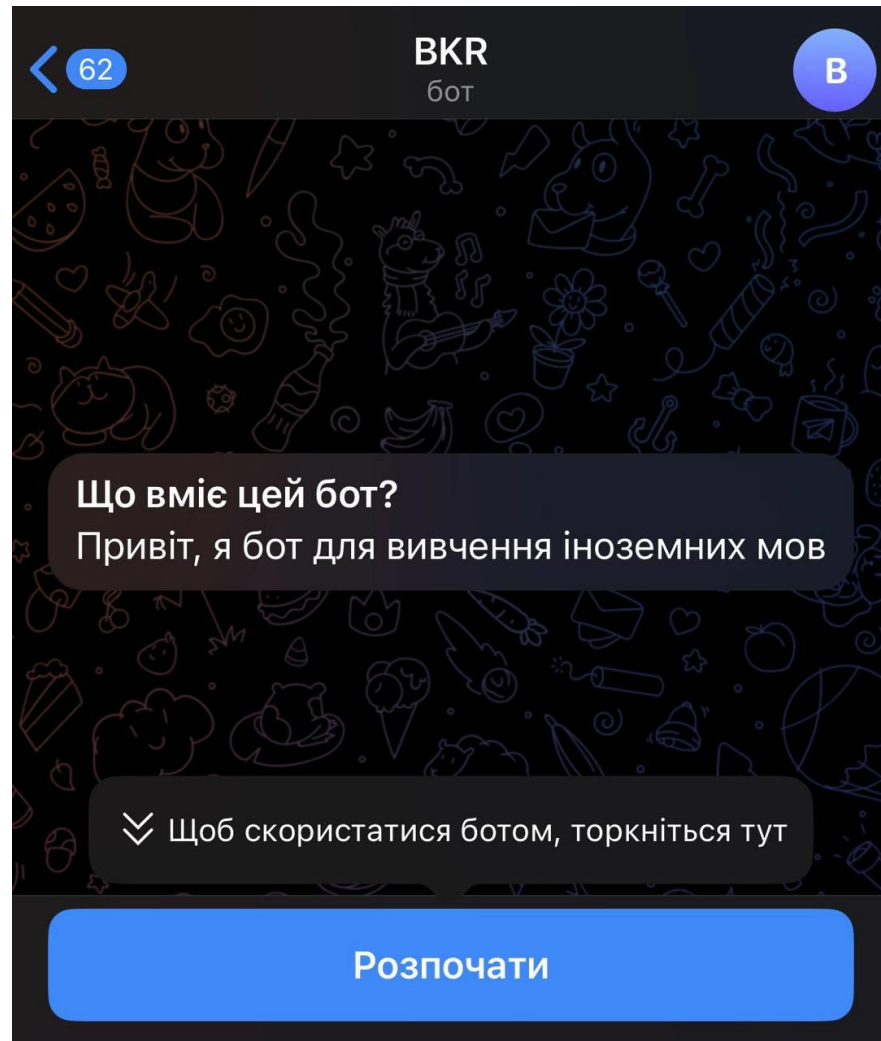


Рисунок Г.1 – Привітальне вікно інтелектуального чат-бота для вивчення іноземних мов

Натиснувши кнопку /start з'являться дві кнопки, потрібно вибрати мову яку бажає користувач вивчити (рис. Г.2).

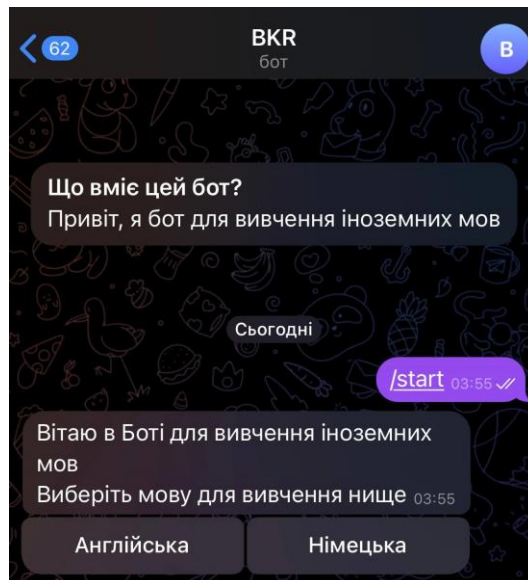


Рисунок Г.2 – Вибір мов для вивчення іноземних мов

Після вибору іноземної мови для вивчення з'являються кнопки: "Вивчення слів", "Тести", "Налаштування" та "Мої результати" (рис. Г.3).

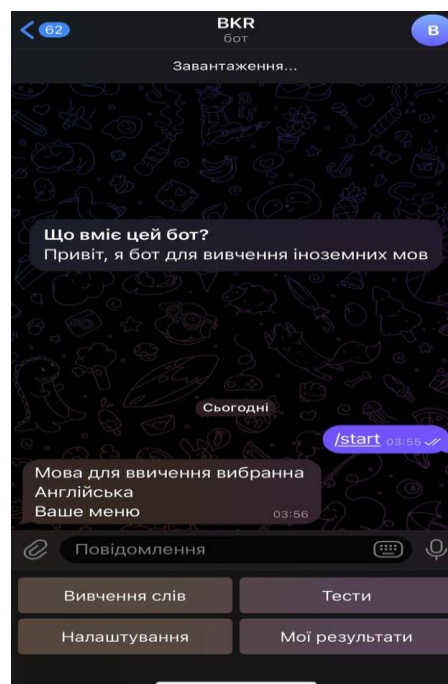


Рисунок Г.3 – Вибір мови

При натисканні на кнопку "Вивчення слів" з'являються інлайн-кнопки "Вивчені слова" та "Вивчення нових слів" (рис. Г.4).

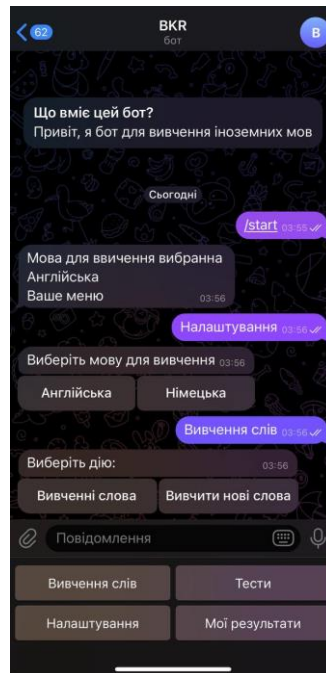


Рисунок Г.4 – Натиск на кнопку " Вивчення слів"

При натисканні інлайн-кнопки "Вивчені слова" з'являються останні 20 слів, які користувач вже вивчив, це допомагає йому їх повторювати та закріплювати у пам'яті (рис. Г.5).

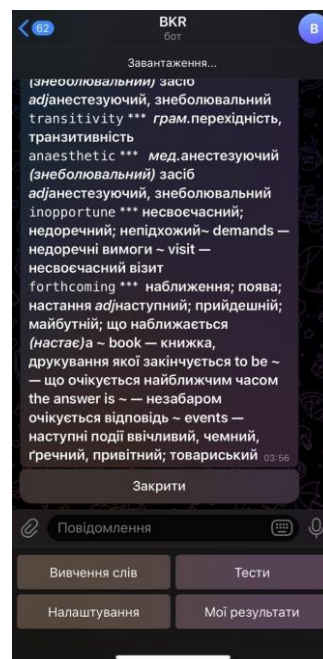


Рисунок Г.5 – Натиск на кнопку " Вивчені слова"

Натискання інлайн-кнопки "Вивчення нових слів" дозволяє користувачеві ознайомитися з новими словами, які він ще не вивчав за допомогою чат-бота (рис. Г.6).

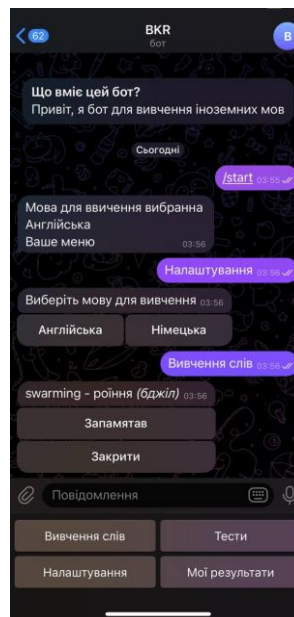


Рисунок Г.6 – Натиск на кнопку " Вивчення нових слів"

Кнопка «Закрити» в обох випадках повертає до див.(рис Г.4). Кнопка "Тести" генерує випадковий тест зі слів, які користувач вже вивчив. Користувачу пропонується слово на іноземній мові та чотири варіанти перекладу, серед яких потрібно вибрати правильний варіант (рис. Г.7).

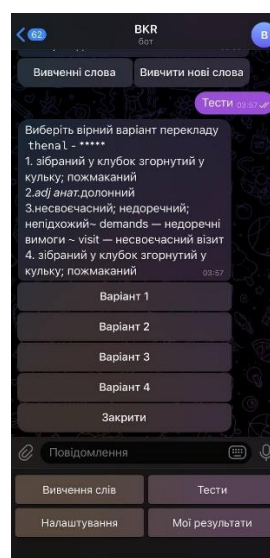


Рисунок Г.7 – Натиск на кнопку "Тест"

Крім того, у налаштуваннях чат-бота передбачена можливість зміни вибраної мови в будь-який момент (рис. Г.8).

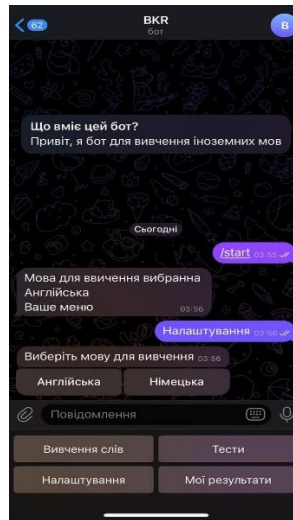


Рисунок Г.8 – Вибір мов для вивчення іноземних мов за допомогою кнопки «Налаштування»

Крім того, користувач може переглянути свою статистику, натиснувши на кнопку "Мої результати". Тут відображається кількість вивчених слів та кількість успішно пройдених тестів (рис. Г.9).

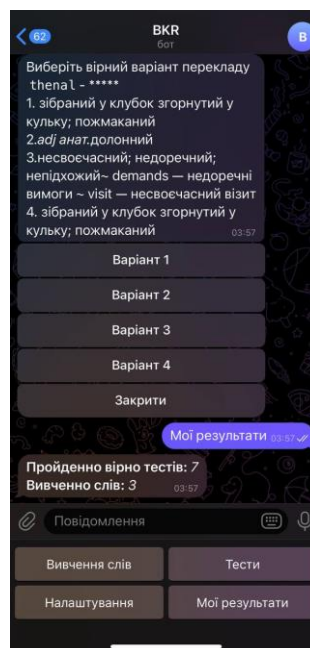


Рисунок Г.9 – Натиск на кнопку "Мої результати"