

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ



к.т.н, доц. Михелєв І.Л.

" ____ " _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "бакалавр"

на тему: «Розроблення інформаційної системи для закріплення англійської мови»

Виконав: студент групи 4141


(підпис) *Мозжухін Д.А.*

Керівник роботи:

старший викладач

(посада, науковий ступінь вчене звання)


(підпис) *Павленко А.Ю.*

м. Миколаїв – 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій
Спеціальність 122 "Комп'ютерні науки"
Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

к.т.н, доц. Гайда А.Ю.

" ____ " ____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти "бакалавр"

Студенту Мозжухіну Даніілу Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розроблення інформаційної системи для закріплення англійської мови»

Керівник роботи Павленко Альона Юріївна

Затверджені наказом ректора № 295-уч від "27" березня 2025 року.

2. Термін подання роботи: 20 червня 2025 р.

3. Вихідні дані до проєкту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем.

4. Перелік питань, що належать до розробки (найменування розділів) дослідження та аналіз предметної галузі, розроблення проєкту інформаційної системи для закріплення англійської мови, реалізація проєкту інформаційної системи для закріплення англійської мови.

5. Перелік презентаційних матеріалів вступ, мета та актуальність роботи, схема запропонованої концепції, діаграма варіантів використання, концептуальна модель, фізична модель, діаграма компонентів, результати розроблення, висновок.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Павленко А.Ю. старший викладач	03.02.2025	22.02.2025
2	Павленко А.Ю. старший викладач	22.02.2025	21.03.2025
3	Павленко А.Ю. старший викладач	21.03.2025	25.04.2025
4	Павленко А.Ю. старший викладач	25.04.2025	07.06.2025

7. Дата видачі завдання 03 лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Аналіз предметної галузі	13.02.2025	
2	Постановка задачі	15.02.2025	
3	Розроблення концепції	22.02.2025	
4	Розроблення технічного проекту інформаційної системи	21.03.2025	
5	Реалізація проекту інформаційної системи	25.04.2025	
6	Розроблення питань з охорони праці	07.06.2025	
7	Розроблення графічних матеріалів	12.06.2025	
8	Подання роботи на перевірку на плагіат	10.06.2025	
9	Подання роботи рецензенту	15.06.2025	
10	Подання роботи на захист	20.06.2025	

Студент



(підпис)

Мозжухін Д.А.

Керівник роботи



(підпис)

Павленко А.Ю.

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Розроблення інформаційної системи для закріплення англійської мови».

Актуальність теми зумовлена зростаючою потребою у ефективних цифрових засобах для вивчення англійської мови. У сучасному суспільстві володіння англійською мовою є важливою складовою професійного та особистого розвитку. Особливу роль відіграє граматики, яка часто недостатньо опрацьовується у популярних навчальних застосунках. Тому існує потреба у розробці інструментів, які дозволяють цілеспрямовано тренувати граматичні навички, з урахуванням рівня користувача.

Метою кваліфікаційної роботи є створення інформаційної системи — телеграм-бота, що дозволяє користувачам закріплювати знання з англійської граматики у зручній, інтерактивній формі. Система передбачає використання індивідуальних налаштувань, повторення за темами. Особливу увагу приділено автоматизації перевірки відповідей, збору статистики та формуванню персоналізованих вправ.

Результатом роботи є розроблений телеграм-бот з інтуїтивним інтерфейсом, системою обліку прогресу, підтримкою граматичних тем та можливістю інтеграції платних функцій. Програмний продукт призначений для самостійного використання учнями, студентами та усіма, хто бажає покращити знання англійської мови. Він також може бути корисним викладачам як допоміжний засіб під час навчального процесу.

Дипломний проєкт має обсяг 77 сторінок, складається із 4 розділів, містить 31 рисуноків, 4 таблиць, 24 джерела інформації.

Ключові слова: чат-бот, англійська мова, граматики, навчання, інтерактив, вправи, інформаційна система.

ANNOTATION

Theme of the qualification work: «Development of an information system for improving English language skills».

The relevance of the topic is caused by the growing need for effective digital tools for learning English. In today's society, English language proficiency is an important component of professional and personal growth. Grammar plays a special role, which is often not enough covered in popular learning applications. Therefore, there is a need to create tools that allow for targeted training of grammar skills, taking into account the user's level.

The goal of the qualification work is to create an information system - a telegram bot, that allows users to improve their knowledge of English grammar in a convenient, interactive way. The system provides for the use of individual settings and repetition by topic. Particular attention is paid to automating the processing of answers, collecting statistics, and creating personalized exercises.

The result of the work is a developed telegram bot with an intuitive interface, a progress tracking system, support for grammar topics, and the ability to integrate paid functions. The software product is intended for independent use by pupils, students, and anyone who wants to improve their English language skills. It can also be useful for teachers as an aid in the educational process.

The thesis project is 77 pages long, consists of 4 chapters, 31 figures, 4 tables, and 24 sources of information.

Keywords: chatbot, English, grammar, learning, interactive, exercises, information system.

ЗМІСТ

Вступ.....	8
Розділ 1. Розроблення концепції інформаційної системи для закріплення англійської мови.....	10
1.1 Опис об'єкту дослідження і аналіз предметної області.....	10
1.2 Формування вимог до інформаційної системи.....	22
1.3 Розроблення концепції інформаційної системи.....	23
Розділ 2. Розроблення технічного проєкту інформаційної системи для закріплення англійської мови.....	29
2.1 Загально-системні рішення.....	29
2.2 Рішення по інформаційному забезпеченню.....	30
2.2.1 Розроблення концептуальної моделі даних.....	32
2.2.2 Розроблення логічної моделі даних.....	34
2.2.3 Розроблення фізичної моделі даних.....	35
2.3 Рішення з математичного забезпечення.....	36
2.4 Рішення з програмного забезпечення.....	40
2.5 Рішення з технічного забезпечення.....	44
Розділ 3. Реалізація проєкту інформаційної системи для закріплення англійської мови.....	46
3.1 Розроблення структури програмного забезпечення.....	46
3.2 Результат розроблення програмного забезпечення.....	48
Розділ 4. Охорона праці.....	57
4.1 Загальні питання охорони праці.....	57
4.2 Структура управління питаннями охорони праці на підприємстві.....	57
4.3 Шкідливі та небезпечні фактори, що супроводжують працю програміста.....	59
4.4 Ергономічні вимоги до робочого місця при роботі за комп'ютером.....	60
Висновки.....	62
Список використаної літератури.....	63
Додаток А - Технічне завдання.....	65

Додаток Б - Текст програми.....	68
Додаток В - інструкція користувача.....	72

ВСТУП

Темою дипломної роботи є розроблення інформаційної системи для практики англійської мови у форматі чат-бота.

Метою проєкту є підвищення ефективності вивчення англійської мови завдяки створенню інформаційної системи для закріплення граматичних конструкцій англійської мови шляхом виконання інтерактивних вправ у чат-боті. Такий підхід дозволяє користувачам отримувати миттєвий зворотний зв'язок, доступ до коротких теоретичних пояснень та практичних завдань у будь-який час і в будь-якому місці, використовуючи лише мобільний пристрій із встановленим месенджером Telegram.

У першому розділі проведено аналіз предметної області та сформульовано вимоги до інформаційної системи. Описані основні методи вивчення англійської мови, розглянуті сучасні технологічні рішення для автоматизації навчального процесу, а також проаналізовані існуючі мобільні додатки, що пропонують функціонал для закріплення граматики. На основі цього аналізу визначено концепцію розроблюваної системи, її ключові функції та особливості, що відрізняють її від інших рішень.

У другому розділі розроблено технічний проєкт інформаційної системи. Наведено загально-системні рішення, що забезпечують роботу чат-бота, включаючи інтеграцію з базою даних, механізми обробки користувацьких запитів та логіку формування навчальних завдань. У розділі також представлено модель даних, алгоритми перевірки відповідей користувача та методи генерації граматичних вправ.

У третьому розділі представлена робоча документація, де описані механізми взаємодії користувача з ботом, способи отримання зворотного зв'язку та перевірки виконаних завдань. Наведено приклади сценаріїв роботи чат-бота та знімки екрану з демонстрацією його функціональності.

У четвертому розділі розглянуті питання забезпечення інформаційної безпеки, а також наведені рекомендації щодо використання бота для підвищення ефективності навчального процесу.

Актуальність проєкту полягає в підвищенні доступності та зручності вивчення англійської мови за рахунок інтеграції навчального процесу у месенджер Telegram. На відміну від традиційних мобільних додатків, чат-бот не вимагає встановлення окремого програмного забезпечення та доступний безпосередньо через месенджер, що значно спрощує процес розповсюдження та використання. Крім того, чат-бот не потребує фінансових витрат на розміщення в App Store або Google Play, а для його роботи достатньо лише сервера, що забезпечує гнучкість у розгортанні та підтримці системи. Це дозволяє зробити сервіс більш доступним для широкого кола користувачів та забезпечити безперервний доступ до навчального матеріалу.

РОЗДІЛ 1. РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ЗАКРІПЛЕННЯ АНГЛІЙСЬКОЇ МОВИ

1.1 Опис об'єкту дослідження і аналіз предметної області

26 червня 2023 року до Верховної Ради України внесений проєкт Закону “Про застосування англійської мови в Україні”. Президент України Володимир Зеленський ініціював його розгляд та визначив законопроєкт як невідкладний [1].

Мета проєкту — офіційно закріпити англійську мову як мову міжнародного спілкування в Україні. Це відкриє українцям нові можливості для професійного та особистого розвитку, допоможе конкурувати на світовому ринку, отримувати вищі зарплати та стимулюватиме вивчення англійської мови серед різних груп населення: освітян, держслужбовців, підприємців, громадян і представників місцевого самоврядування. Реалізація цієї ініціативи сприятиме створенню комфортного мовного середовища для всіх, хто живе та працює в Україні [2].

Необхідність підвищення рівня володіння англійською мовою набуває дедалі більшого значення в умовах глобалізації та європейської інтеграції України. Це не лише розширює освітні та кар'єрні перспективи українців, а й сприяє міжнародній співпраці та розвитку інновацій.

Згідно з результатами соціологічного дослідження, проведеного у грудні 2022 - січні 2023 року Київським міжнародним інститутом соціології (КМІС), більшість українців, а саме 68% мають певні знання принаймні однієї іноземної мови, хоча часто ці знання не є глибокими: детальний аналіз показує, що тільки 23% можуть читати, писати і спілкуватися цією мовою на побутовому і навіть на професійному рівнях [3].

Англійська мова є ключовим інструментом для кар'єрного та особистісного розвитку, про що свідчать результати опитувань: 61% респондентів використовують її у повсякденному житті, 19% — у професійній діяльності, а 18% — у навчанні. Поширеними формами її використання серед респондентів є прослуховування пісень (56%), ознайомлення з інструкціями (55%), читання статей (54%) та перегляд відео англійською (52%).

Водночас опитування показують, що на шляху до ефективного вивчення мови існують перешкоди: нестача часу (38%), інші життєві пріоритети (34%), фінансові труднощі (33%) та брак мотивації (31%). Попри це, близько 70% респондентів прагнуть опанувати англійську для професійного розвитку [3].

Формальна освіта залишається одним із найпоширеніших способів вивчення англійської мови: 29% дорослого населення вивчали іноземну мову лише у школі, не вдаючись до практик самоосвіти. У школах, університетах та мовних курсах студенти отримують систематизовані знання з граматики, лексики, аудіювання та письма. Навчання проходить під керівництвом викладача, що дозволяє отримувати зворотний зв'язок та виправляти помилки.

Згідно з інструктивно-методичними рекомендаціями Міністерства освіти і науки України (МОН) щодо викладання навчальних предметів та інтегрованих курсів у закладах загальної середньої освіти у 2024/2025 навчальному році, освітня програма передбачає не менше двох годин на тиждень для вивчення іноземної мови. Проте цього часу може бути недостатньо для формування глибоких мовних навичок, особливо якщо враховувати потребу в регулярній практиці розмовної мови [4].

Хоча формальна освіта забезпечує структуроване навчання, вона має певні обмеження. Навчальний процес часто стандартизований, що може не враховувати індивідуальні особливості учнів, а також вимагає значних часових та фінансових витрат. В умовах обмеженої кількості академічних годин учням і студентам доводиться звертатися до додаткових методів навчання, таких як самоосвіта чи використання цифрових технологій.

Самостійне навчання є ще одним популярним способом опанування англійської мови: 27% дітей вивчають її додатково за межами навчального закладу. Воно може включати читання книг, перегляд відео, прослуховування підкастів, використання підручників і виконання вправ. Цей метод надає велику свободу у виборі матеріалів і темпу навчання, проте вимагає високої самоорганізації та мотивації. Однією з основних проблем є відсутність контролю та можливості отримати виправлення помилок, що може уповільнювати прогрес.

Спілкування з носіями мови є надзвичайно ефективним способом вивчення англійської, оскільки дозволяє не лише вдосконалити мовні навички, а й адаптуватися до реального мовного середовища. Проте не всі мають доступ до такого спілкування, а психологічний бар'єр і страх помилитися можуть ускладнювати процес навчання.

Дослідження, проведене на прикладі індонезійської сім'ї, що переїхала до Австралії, демонструє, що навіть повне занурення в мовне середовище не гарантує швидкого оволодіння мовою. Батько сім'ї, який мав значний досвід формального навчання англійської мови та використовував її у професійній діяльності, показав найкращі результати: його рівень знання англійської був достатньо високим ще до переїзду (IELTS 6.5 - вище середнього), що дозволило йому покращити навички в англomовному середовищі. У той же час мати, яка не використовувала англійську протягом десяти років, та їхня донька, яка зовсім не володіла мовою, зіткнулися з труднощами у спілкуванні. Навіть після року життя та роботи в англomовному середовищі жоден із членів сім'ї не засвоїв англійську морфологію повністю, а лише батько досяг 80% правильного використання часів [5].

Ці результати узгоджуються з іншими дослідженнями та підтверджують кілька важливих аспектів процесу оволодіння другою мовою. По-перше, чим вищий рівень володіння мовою до моменту занурення в мовне середовище, тим ефективніше воно сприяє подальшому розвитку навичок. По-друге, сам факт перебування в англomовному середовищі не є достатнім для швидкого засвоєння граматичних конструкцій. По-третє, формальне навчання відіграє ключову роль у прискоренні процесу опанування мови дорослими.

Кожен із методів вивчення англійської має свої переваги та недоліки. У багатьох випадках для досягнення найкращого результату доцільно комбінувати різні підходи. У перших двох випадках – формальній освіті та самостійному навчанні – для засвоєння граматичних конструкцій зазвичай використовують спеціалізовані підручники. Вони містять практичні завдання, короткі теоретичні пояснення та відповіді для самоперевірки. Однак відповіді часто розміщені в кінці

підручника, що змушує користувачів витрачати час на їх пошук, перевіряючи правильність своїх рішень.

Незалежно від обраного підходу до вивчення мови – формальної освіти, самостійного навчання чи занурення у мовне середовище – граматики залишається важливою складовою навчального процесу. У формальній освіті її викладають систематично за навчальними програмами, у самостійному навчанні – через підручники та онлайн-курси, а при спілкуванні з носіями мови граматичні конструкції засвоюються природним шляхом, через практику. Однак, попри значну увагу до граматики, існує низка викликів, пов'язаних із її ефективним засвоєнням та застосуванням у реальному мовленні.

Грамматика вважається найскладнішою частиною навчання, і через це практичні навички володіння мовою нерідко залишаються нерозвинутими. Водночас головною метою вивчення іноземної мови є комунікація, тому граматичний матеріал, який необхідно засвоїти, має бути достатнім для ефективного спілкування та відповідати реальним потребам мовлення.

При формуванні граматичної компетенції важливо зосереджуватися не лише на формальному аспекті, що стосується правильного використання граматичних структур, а й на функціональному аспекті – тобто на здатності застосовувати граматику відповідно до комунікативних ситуацій. Знання правил саме по собі не є достатнім: необхідно розуміти, коли і як використовувати ті чи інші граматичні конструкції у спілкуванні.

Оскільки складні граматичні явища потребують значної кількості прикладів та часу на їх опанування, доцільним підходом є теоретично-практичний метод. Він поєднує коротке граматичне пояснення з реальними мовленнєвими прикладами, що демонструють утворення та використання певних конструкцій у конкретному контексті. Таким чином, оптимальний результат досягається завдяки комбінуванню механічного запам'ятовування правил та їх усвідомленого застосування у мовленні. Крім того, ефективність засвоєння граматики підвищується при групуванні подібних конструкцій і проведенні паралелей між ними [6].

Не менш важливим аспектом при вивченні англійської мови є засвоєння нових слів, адже без достатнього лексичного запасу неможливо ефективно використовувати граматичні конструкції для побудови речень. Перехід інформації з короткочасної до довготривалої пам'яті, як правило, відбувається шляхом багаторазового повторення матеріалу, введеного спочатку у короткочасну пам'ять.

Повторення забезпечує дві речі. Перш за все, доки інформація повторюється, доти вона зберігається у короткочасній пам'яті. Але важливішим є інше - перехід цієї інформації у довготривалу пам'ять завдяки повторенню.

З цієї причини традиційними інструментами для вивчення нових слів є словники та картки. В обох випадках користувач бачить англійське слово та його переклад українською, після чого намагається пригадати переклад поточного слова, і такий процес відбувається багато разів. Проте такий підхід найчастіше призводить до механічного запам'ятовування без глибокого розуміння, що знижує ймовірність довготривалого засвоєння інформації.

Для прикладу: запам'ятати набір літер А Р Ш С Р Ю А У Б Ф С Б доволі складно при простому повторенні. Але якщо об'єднати їх у знайомі, значущі одиниці (США, ЮАР, ФБР, СБУ), запам'ятати їх стане значно легше. Це свідчить про важливість групування та осмислення інформації при її засвоєнні.

На думку вчених, більшість інформації ми вводимо у пам'ять шляхом семантичного кодування: запам'ятовуємо загальне значення. А при відтворенні інформації не повторюємо її дослівно, а перефразовуємо, однак при цьому зберігаємо загальну суть [7].

Таким чином, можна зробити висновок, що метод карток не є найбільш ефективним для довготривалого засвоєння нових слів, оскільки не сприяє формуванню міцних асоціативних зв'язків. У зв'язку з цим доцільно впровадити іншу методику навчальних завдань для користувача, яка базуватиметься на таких принципах:

- замість простого перекладу слова — надання контексту його використання;
- стимулювання користувача до складання речень з новим словом, що сприяє активному закріпленню знання;

- відтворення аудіо з правильною вимовою слова, що дозволяє формувати слухову асоціацію.

Застосування такої методики допоможе глибше засвоювати лексику без надмірного фокусування на переклад, розвиваючи водночас контекстне мислення, асоціативне запам'ятовування та навички активного використання мови.

У цьому процесі значну роль відіграють сучасні технології, які роблять навчання більш інтерактивним і гнучким. Цифрові платформи дозволяють отримувати миттєвий зворотний зв'язок, адаптувати матеріали під індивідуальні потреби учня та автоматизувати багато аспектів навчання, що значно підвищує його ефективність. Завдяки цьому учні можуть не лише краще розуміти граматичні правила, а й навчитися застосовувати їх у реальному мовному середовищі.

Розвиток інформаційних технологій значно розширив можливості для вивчення іноземних мов. Цифрові платформи та мобільні додатки стали невід'ємною частиною навчального процесу, забезпечуючи доступ до інтерактивних вправ, тестів і матеріалів.

Однією з ключових переваг цифрових технологій є доступність навчальних матеріалів у будь-який час і з будь-якого місця. Онлайн-курси, мобільні додатки та освітні платформи надають можливість практикувати різні аспекти мови — граматику, лексику, аудіювання, говоріння та письмо.

Одним із найпопулярніших інструментів є мобільні додатки для вивчення мов, такі як Duolingo, Busuu, Memrise та Babbel. Вони пропонують навчання в ігровій формі, завдяки нагородам та змагальному елементу, вони утримують увагу користувача та роблять процес навчання більш захопливим. Частина додатків використовують алгоритми машинного навчання для аналізу результатів користувача та на їх основі пропонують індивідуальні плани занять, орієнтовані на сильні та слабкі сторони учня. Проте більшість з них мають і певні недоліки, серед яких відсутність глибокого опрацювання граматики та мовлення [8].

Ще одним ефективним методом є онлайн-курси та відео-уроки, які можна знайти на платформах Coursera, Udemy, YouTube та інших. Вони пропонують структурований матеріал і дають можливість навчатися у зручний час. Багато

курсів містять інтерактивні завдання, тести та сертифікацію після завершення навчання. Проте такі курси часто не мають зворотнього зв'язку з користувачем та потребують багато часу, на опрацювання матеріалу, що може знижувати ефективність засвоєння матеріалу.

Таким чином, цифрові технології не лише розширюють можливості для вивчення англійської мови, а й сприяють формуванню навичок самостійного навчання, підвищують мотивацію та роблять процес засвоєння матеріалу більш зручним і ефективним.

Проведено аналіз існуючих рішень, спрямованих на використання цифрових технологій для вивчення англійської мови. Метою аналізу є виявлення сильних і слабких сторін сучасних інструментів, таких як мобільні додатки та чат-боти. Розгляд існуючих рішень дозволить визначити їхні ключові функціональні особливості, методи реалізації та підходи до взаємодії з користувачами. Це, своєю чергою, сприятиме формуванню вимог і визначенню напрямів для розроблення власного рішення, що враховуватиме сучасні тенденції та потреби цільової аудиторії.

Duolingo є одним із найпопулярніших мобільних додатків для вивчення мов (рисунок 1.1), зокрема англійської. Його ключовою особливістю є акцент на гейміфікації та залученні користувача через ігрові елементи: система балів, досягнень, нагород та щоденних серій мотивує користувачів регулярно практикуватися: розробники перетворили те, що може бути нудним та потребує неабиякої мотивації, на щось, що викликає залежність.

Основна методика Duolingo полягає у частому повторенні схожих мовних конструкцій, що сприяє запам'ятовуванню простих мовних патернів. Завдяки цьому користувачі набувають базових навичок читання, письма та сприйняття мови на слух. Проте застосунок часто нехтує глибоким поясненням граматичних правил, роблячи акцент на запам'ятовуванні готових фраз без розуміння їхньої структури.

Однією з переваг Duolingo є його доступність та інтерактивність: завдання подаються у короткому та зручному форматі, що робить додаток придатним для навчання у будь-який час. Проте недоліком є обмеженість у розвитку навичок

говоріння та спілкування, оскільки взаємодія з мовою відбувається здебільшого у форматі тестів і вправ

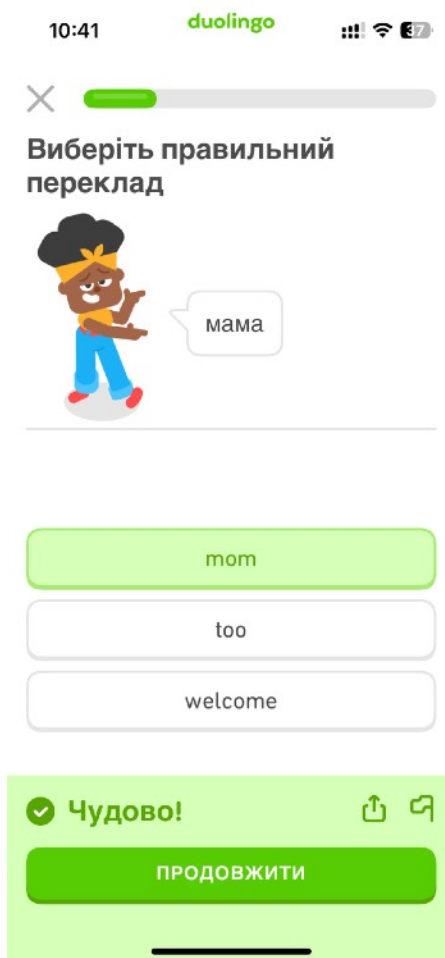


Рисунок 1.1 — Екран застосунку Duolingo

Таким чином, Duolingo ефективно допомагає користувачам розвивати базові мовні навички завдяки ігровим механікам та системі повторень, однак має обмеження у глибокому вивченні граматики та розвитку мовлення.

Додаток Gramaro (рисунок - 1.2) спеціалізується на практиці граматики англійської мови, що робить його корисним інструментом для користувачів, які прагнуть покращити свої граматичні навички. Він надає можливість обирати рівень складності та конкретну граматичну тему для повторення, що дозволяє адаптувати навчання під індивідуальні потреби користувача. Цей функціонал особливо зручний для тих, хто хоче працювати над певними аспектами граматики без перевантаження іншими темами.

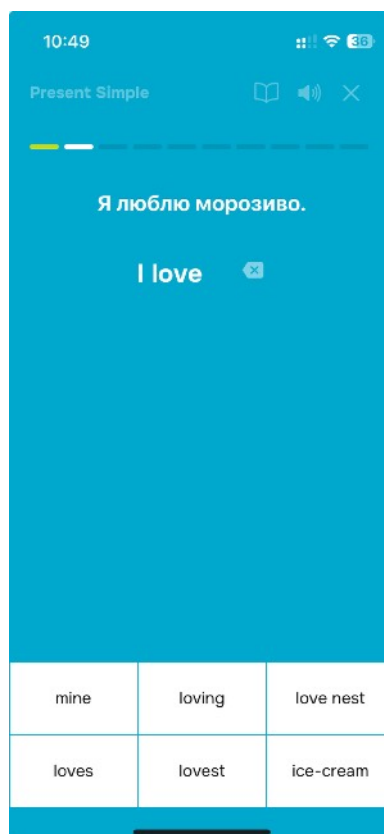


Рисунок 1.2 — Екран застосунку Gramaro

Також додаток має коротку теорію до кожної теми, яка дозволяє користувачам швидко згадати аспекти граматичних правил. Це дуже корисно для користувачів, які хочуть освіжити свою пам'ять або розібратися з незрозумілими моментами.

Практика граматичних конструкцій здійснюється через поетапне будування речень. Це допомагає користувачам освоювати правильне використання граматичних структур, що є ефективним підходом до навчання. Додатково, озвучення побудованих речень дає змогу користувачам слухати правильну вимову та інтонацію, що сприяє покращенню їхніх слухових навичок.

Загалом, Gramaro є корисним інструментом для вивчення граматики, особливо для тих, хто хоче зосередитися на конкретних темах та розвивати навички побудови граматичних конструкцій.

Додаток Drops (рисунок 1.3) орієнтований на вивчення нових слів та малих конструкцій речень, застосовуючи дуже інтерактивний підхід до навчання. Користувачам необхідно перетягувати слова до зображень, які асоціюються з відповідними словами. Це дозволяє створити візуальні асоціації, що допомагає вивчати слова більш ефективно. Завдяки постійному повторенню таких завдань, асоціації між словами та їх значенням значно зміцнюються. Це є основною особливістю додатку, адже створення таких візуальних і когнітивних зв'язків може бути дуже ефективним методом для вивчення лексики.

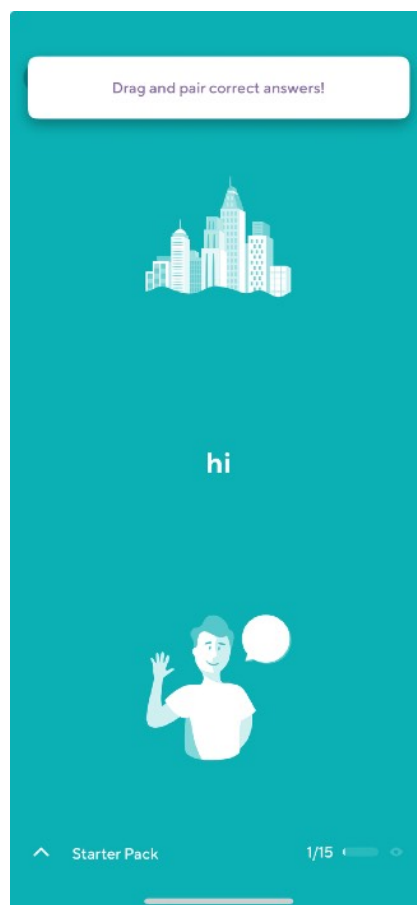


Рисунок 1.3 — Екран застосунку Drops

Однак варто зазначити, що додаток Drops не вчить граматиці, що може бути недоліком для тих, хто хоче комплексно вивчати мову. Відсутність граматичних вправ означає, що цей додаток більше підходить для початкових етапів вивчення мови, зокрема для збільшення словникового запасу, а не для глибокого розуміння структури мови.

Lingq — це додаток для вивчення мов (рисунок 1.4), який поєднує читання, слухання та інтерпретацію контексту для полегшення вивчення лексики. Основною особливістю Lingq є використання контекстуальних матеріалів, таких як статті, аудіо та відео, що дозволяє користувачам занурюватися у мову, яку вони вивчають. Це створює можливість для вивчення слів та фраз без перекладу на рідну мову, що сприяє розвитку інтуїтивного розуміння.

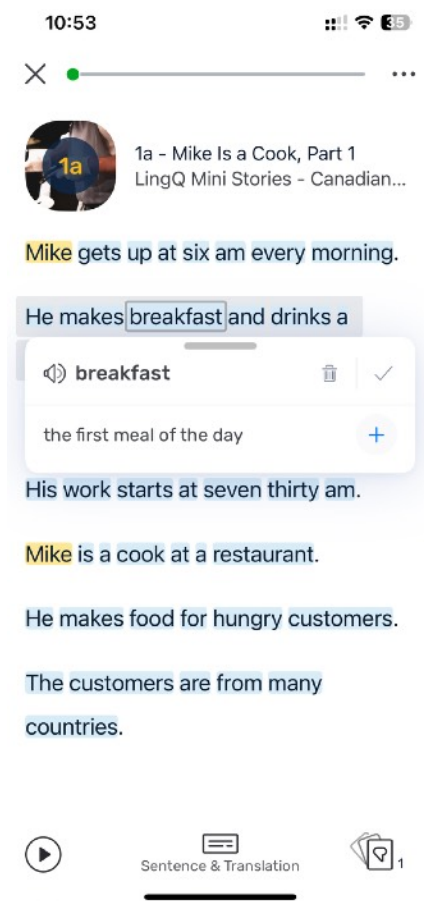


Рисунок 1.4 — Екран застосунку LingQ

Один з головних принципів додатку — це «сплутування» або «непряме вивчення» мови. Користувачі мають можливість розглянути незнайомі слова в контексті, що дозволяє їм краще запам'ятовувати їх. Кожне слово, яке зустрічається вперше, можна позначити як нове, і додаток допоможе користувачам відслідковувати їхній прогрес, надаючи можливість для повторення.

Додаток також має функцію автоперекладу, що дозволяє користувачам отримувати миттєву допомогу, коли вони натрапляють на складні слова або фрази. Користувачі можуть прослуховувати правильну вимову, що є великим плюсом для тих, хто хоче поліпшити свої слухові навички.

Загалом, Lingq — це потужний інструмент для тих, хто хоче вивчати мову через контекст, допомагаючи будувати навички через читання та слухання, без необхідності вивчати граматику чи лексичні одиниці ізольовано.

Проаналізувавши перелічені додатки, можна зробити висновок, що Gramago використовує підхід, який відповідає концепції помічника у засвоєнні граматики. Додаток автоматизує виконання практичних завдань, пропонуючи можливі відповіді та забезпечуючи миттєву перевірку. Також він містить короткі теоретичні матеріали для кожної теми, що дозволяє користувачам швидко згадати основні граматичні правила.

Однак Gramago має кілька недоліків. Додаток пропонує завдання, в яких потрібно будувати речення з нуля, маючи лише їх переклад українською мовою. Такий підхід може відволікати користувача від опрацювання конкретної граматичної теми, оскільки він змушений зосереджуватися на формулюванні речення, а не на засвоєнні правил. Крім того, безкоштовна версія містить обмежену кількість тем, а повноцінний доступ коштує 412 грн на місяць, що є доволі високою ціною [8].

Також варто зазначити, що методика вивчення слів, реалізована в мобільному додатку Drops, має багато спільного з раніше описаним підходом. У цьому додатку застосовується механіка перетягування слів до відповідних зображень, які асоціюються з цими словами. Таким чином, забезпечується одночасне залучення моторної активності та формування асоціативних зв'язків між лексемою та візуальним образом. Це сприяє більш глибокому засвоєнню нової лексики завдяки залученню декількох каналів сприйняття — зорового, моторного та семантичного.

З огляду на це, доцільно розробити інформаційну систему, яка автоматизуватиме виконання практичних завдань і надаватиме миттєві відповіді. Основна увага має бути зосереджена на практиці вже вивчених граматичних тем

та їх правильному використанні, без зайвого акценту на побудову речень. Водночас система повинна пропонувати доступнішу ціну підписки.

З огляду на це, наступним етапом є визначення ключових вимог до інформаційної системи, які забезпечать її ефективність та зручність для користувачів.

1.2 Формування вимог до інформаційної системи

Функціональні вимоги:

- надавання користувачу практичних завдань;
- миттєва перевірка відповідей користувача;
- можливість вибору теми для практики;
- скасування відправки відповіді на перевірку;
- отримання статистики виконаних вправ.

Програмні вимоги:

- мова програмування: Java;
- фреймворк: Spring Boot;
- база даних: PostgreSQL;
- кеш сховище: Redis;
- операційна система: Linux Arch (Не включає в себе користувацький інтерфейс, завдяки чому споживає мало ресурсів).

Вимоги до апаратного забезпечення серверу:

- оперативна пам'ять: DDR4 8GB (рекомендується 16 GB для більшої надійності при високих навантаженнях);
- процесор: AMD Ryzen 7 - 8 ядер (для покращення продуктивності в багатозадачному режимі);
- сховище: Samsung EVO 970 256GB (для об'ємної бази даних).

Вимоги до якості:

- інтерфейс Telegram-бота має бути інтуїтивно зрозумілим та зручним для користувачів;
- сумісність з усіма основними платформами, де можна використовувати Telegram (мобільні телефони, планшети, комп'ютери з доступом до Telegram);
- бот повинен бути швидким і реагувати на запити користувачів без значних затримок. Важливо, щоб час відгуку був мінімальним;
- бот має стабільно працювати протягом тривалого часу без збоїв, а також мати систему аварійного відновлення для відновлення роботи в разі непередбачених помилок чи збоїв у системі;
- масштабованість системи є важливим фактором для забезпечення її ефективної роботи в умовах зростання кількості користувачів. Бот повинен мати можливість працювати з великими обсягами даних і забезпечувати високий рівень продуктивності навіть при значному навантаженні;
- бот має захищати особисті дані користувачів, зокрема інформацію про підписки та прогрес у навчанні.

1.3 Розроблення концепції інформаційної системи

У цьому розділі представлено концептуальну модель інформаційної системи, що реалізує функціональність Telegram-бота для практики англійської граматики. Запропонована модель концепції, яка представляє взаємодію користувача з системою, основними бізнес-процесами та зовнішніми сервісами, з якими працює система зображена на рисунку 1.5 [9].

Взаємодія користувача з інформаційною системою відбувається через месенджер Telegram, де вона реалізована у вигляді чат-бота. Бот працює за заздалегідь запрограмованими сценаріями, реагуючи на команди користувача та забезпечуючи природну комунікацію у форматі листування в чаті. Повідомлення, передаються через сервери Telegram, а з них до інформаційної системи.

Інформаційна система обробляє запити користувача, зокрема генерує завдання, надає коротку теорію, збирає статистику виконання та дозволяє скасовувати відповіді. Також система містить механізм оплати, що забезпечує доступ до тарифних планів через інтеграцію з банківськими сервісами.

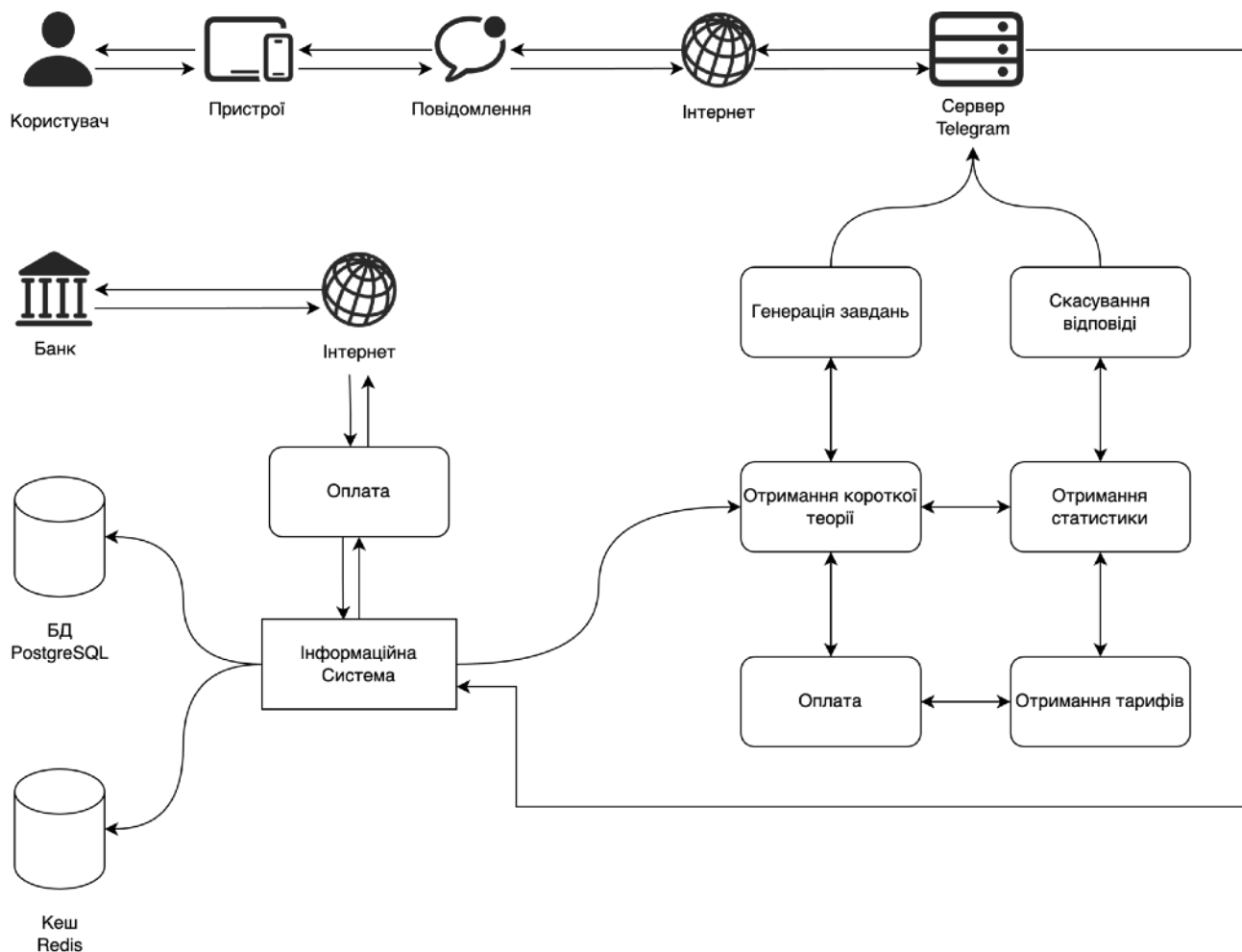


Рисунок 1.5 — Запропонована модель концепції

Для збереження даних використовується база даних PostgreSQL, а для прискорення обробки запитів застосовується кешування в Redis. Така архітектура дозволяє забезпечити швидку та надійну роботу Telegram-бота, підвищуючи його продуктивність і зручність використання.

Інформаційна система для закріплення англійської мови призначена для автоматизації виконання та перевірки вправ, спрямованих на засвоєння граматичних конструкцій англійської мови. Вона дозволяє користувачам

ефективно тренуватися завдяки миттєвій перевірці відповідей та можливості швидкого доступу до коротких теоретичних пояснень.

Однією з ключових переваг такої реалізації є мінімальні вимоги до інфраструктури. Для запуску системи достатньо лише сервера, що дозволяє уникнути додаткових витрат на розповсюдження застосунку. Відсутність необхідності розміщення в App Store чи Google Play усуває фінансові витрати на публікацію, що суттєво знижує поріг входу для розробника та дозволяє зробити сервіс доступним без зайвих обмежень.

Чат-бот має суттєві переваги перед веб-застосунками та традиційними мобільними додатками. Його не потрібно встановлювати окремо, адже достатньо мати встановлений месенджер, що спрощує доступ для користувачів. Взаємодія відбувається в знайомому середовищі, що зменшує поріг входу та сприяє швидшому освоєнню функціоналу. Завдяки push-сповіщенням користувач отримує нагадування без необхідності запускати окремий додаток, що підвищує регулярність використання. Оновлення та вдосконалення сервісу відбуваються безпосередньо на сервері, без потреби примушувати користувачів встановлювати нові версії. Крім того, Telegram надає розширені можливості інтеграції з іншими сервісами, що дозволяє у майбутньому доповнювати функціонал без суттєвих змін у самій архітектурі системи.

Чат-бот для практики англійської граматики повинен мати кілька основних функцій, які дозволяють користувачам автоматизувати процес навчання та робити його більш ефективним.

По-перше, бот повинен надавати користувачам практичні завдання, що складаються з кількох етапів. Основна мета такої структури – поступове опрацювання граматичних правил без механічного вибору готових відповідей.

Кожне завдання передбачає заповнення пропусків у реченнях відповідно до правил обраної граматичної теми. Однак замість того, щоб одразу отримати кілька варіантів відповіді на вибір, користувач виконує завдання поетапно. Спочатку він заповнює перший пропуск, після чого переходить до наступного. Такий підхід дозволяє краще осмислити власні відповіді, уникнути звичайного вгадування та

сформувати усвідомлене розуміння граматичних конструкцій та контекст їх використання.

Поетапне виконання також дає змогу користувачеві вчасно помітити помилку ще до перевірки відповіді системою. Якщо він розуміє, що зробив неправильний вибір на попередньому етапі, він може його виправити, що сприяє більш глибокому аналізу граматичних закономірностей. Завдання можуть мати різний рівень складності, змінюючи кількість пропусків, складність речень та варіативність граматичних конструкцій.

Не менш важливою функцією є миттєвий зворотний зв'язок. Після того, як користувач надасть свою відповідь, бот повинен негайно повідомити, чи правильна вона, та вивести правильний варіант. Це дозволить користувачам швидко коригувати свої помилки і покращувати свої знання. Важливо, щоб бот також пояснював помилки, зазначаючи, чому певна відповідь є правильною чи неправильною.

Крім того, користувач повинен мати можливість вибирати одну або кілька тем для практики. Ця функція буде корисна, наприклад, коли користувач тільки-но вивчив певну тему з репетитором і хоче закріпити матеріал шляхом додаткових вправ. Бот також повинен бути здатен надавати коротке пояснення теорії щодо поточної теми, щоб користувач міг освіжити свої знання в разі забуття.

Особливо корисною є можливість скасування відповіді на попередньому етапі перед її перевіркою. Це дає змогу користувачеві виправити помилку, якщо він зрозумів її ще до того, як отримав зворотний зв'язок від бота. Така функція підвищує точність виконання завдань і забезпечує додаткову гнучкість у процесі навчання.

Інформаційна система повинна підтримувати механізм підписки, який дозволяє користувачам отримувати доступ до бота на обмежений термін за певну плату. Підписка може бути місячною або річною, що дасть користувачам гнучкість у виборі варіанту доступу.

Для залучення нових користувачів передбачена реферальна система, яка дозволяє отримувати знижки або бонуси за запрошення нових учасників. Це

стимулюватиме користувачів активно ділитися ботом і залучати більше людей, що збільшить кількість користувачів та потенційно підвищить прибуток.

Крім того, бот повинен мати можливість зберігати прогрес користувача, фіксуючи виконані тести та досягнуті результати. Це дозволить користувачеві відслідковувати свій розвиток у навчанні, побачити свої слабкі місця та працювати над ними. Прогрес також може включати рівні досягнень або бали, що додатково мотивуватиме користувача продовжувати навчання.

Можливість придбання підписки та перегляду власної статистики буде реалізована у форматі міні-застосунку — веб-сторінки, інтегрованої з Telegram. Завдяки цій інтеграції користувач сприйматиме систему як єдине ціле. Використання міні-застосунку для відображення статистики забезпечує зручніший та наочніший формат подачі даних, що покращує взаємодію з користувачем і підвищує комфорт використання сервісу.

Програмні вимоги:

- мова програмування: Java;
- фреймворк: Spring Boot;
- база даних: PostgreSQL;
- кеш сховище: Redis;
- операційна система: MacOS Sequoia 15.3.1.

Вимоги до апаратного забезпечення:

- оперативна пам'ять: DDR4 16GB;
- процесор: Apple M1 - 8 ядер;
- сховище: APPLE SSD AP0256Q 256GB.

Висновок за розділом 1

У світлі проєкту ініціативи президента України та законопроєкту щодо офіційного закріплення англійської мови як мови міжнародного спілкування в Україні, актуальність вивчення англійської набуває нової ваги. Глобалізаційні

процеси, європейська інтеграція та потреба в конкурентоспроможності українців на міжнародному ринку підсилюють необхідність розвитку мовної компетентності, особливо граматики та лексики, як основи ефективної комунікації.

Аналіз існуючих підходів до вивчення англійської мови — засвідчує, що кожен метод має як переваги, так і обмеження. Формальна освіта забезпечує структурованість і методичну підтримку, проте часто не враховує індивідуальні потреби й вимагає додаткових зусиль поза межами навчального закладу. Самонавчання дає свободу вибору матеріалів і темпу, однак потребує високої мотивації та самоконтролю.

Грамматика залишається складним, але необхідним елементом процесу навчання. Ефективне засвоєння граматичних конструкцій можливе лише за умови поєднання теоретичних знань із практичним застосуванням у реальних комунікативних ситуаціях. Фокус має зміщуватися з механічного запам'ятовування правил на їхнє осмислене використання у мовленні. Водночас засвоєння лексики потребує осмислення, повторення та групування інформації, що значно підвищує ймовірність її переходу до довготривалої пам'яті.

Тому на основі розробленої концепції інформаційної системи розробляється технічний проєкт, який дозволить досягнути поставленої мети.

РОЗДІЛ 2. РОЗРОБЛЕННЯ ТЕХНІЧНОГО ПРОЄКТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ЗАКРІПЛЕННЯ АНГЛІЙСЬКОЇ МОВИ

2.1 Загально-системні рішення

Інтеграція з готовою еквайринг-системою є важливою складовою для забезпечення зручності та безпеки платіжних операцій в інформаційній системі. В Україні є кілька популярних сервісів для приймання та обробки онлайн-платежів, таких як LiqPay (ПриватБанк), MonoPay (Монобанк), NovaPay (Нова Пошта), Fondy, Portmone та інші. Для реалізації оплати підписки на сайті інформаційної системи було обрано еквайринг-систему LiqPay від ПриватБанку.

Цей вибір зумовлений рядом переваг. По-перше, LiqPay має добре розвинену документацію та надійно інтегрується з популярними платіжними системами, такими як Apple Pay, Google Pay, PrivatPay, а також підтримує стандартну оплату картками (Visa, MasterCard). По-друге, LiqPay надає можливість оформлення підписок, що дає змогу автоматизувати процес щомісячних списань без потреби створення окремого модуля для цієї функції.

Крім того, використання зовнішньої еквайринг-системи дозволяє значно знизити ризики, пов'язані зі зберіганням платіжних даних користувачів, оскільки всі платіжні транзакції обробляються на сторонньому сервері, що забезпечує більшу безпеку та відповідність вимогам захисту персональних даних. Це також дозволяє зосередити увагу на розвитку основних функцій системи, не витрачаючи ресурси на розроблення та підтримку платіжних механізмів.

Згідно з вимогами AppStore (магазин додатків для пристроїв на iOS) та Google Play (магазин додатків для пристроїв на Android), всі платежі всередині застосунків, що розповсюджуються на цих платформах, повинні оброблятися через їх вбудовані платіжні системи. Тому використання обраної раніше еквайринг-системи LiqPay для здійснення оплат всередині додатка є неприпустимим відповідно до політики цих платформ [10, 11].

У зв'язку з цим для чат-боту буде використана система Telegram Stars, яка дозволяє користувачам придбати цифрові бали через вбудовані платіжні системи Apple та Google. Ці бали можна витратити на цифрові продукти, пропоновані чат-ботами, такі як електронні книги або онлайн-курси. Telegram Stars відповідає вимогам політики Apple та Google щодо продажу цифрових товарів і надає можливість здійснювати внутрішні покупки через мобільні додатки.

Проте, важливим недоліком використання Telegram Stars є комісія у розмірі 47%, що є досить високим відсотком. Для того, щоб зменшити витрати користувачів, буде також надана можливість оформити підписку на офіційному сайті бота, де оплата здійснюватиметься через систему LiqPay за більш вигідною вартістю.

Цей варіант також можна використовувати як акцію або спеціальну пропозицію для користувачів, які хочуть отримати підписку за більш вигідними умовами. Таке рішення може стати привабливим для клієнтів, адже дає їм можливість вибору між двома способами оплати — стандартним через Telegram Stars з більшою вартістю або більш економічним через офіційний сайт за допомогою LiqPay. Цей підхід не тільки знизить витрати користувачів, але й створить додаткові стимули для оформлення підписки через сайт бота.

2.2 Рішення по інформаційному забезпеченню

Для збереження даних користувача та завдань потрібна база даних. Для управління БД планується використання СУБД PostgreSQL.

PostgreSQL - це потужна об'єктно-реляційна система баз даних з відкритим вихідним кодом, яка вже понад 35 років активно розвивається і завоювала міцну репутацію завдяки надійності, функціональності та продуктивності.

PostgreSQL постачається з багатьма функціями, які допомагають розробникам створювати додатки, адміністраторам захищати цілісність даних і створювати відмовостійкі середовища, а також допомагають вам керувати даними,

незалежно від того, наскільки великий чи малий обсяг даних. Крім того, що PostgreSQL є безкоштовною і з відкритим кодом, вона дуже легко розширюється. Наприклад, є можливість визначати власні типи даних, створювати власні функції і навіть писати код на різних мовах програмування без перекомпіляції бази даних [12].

Також важливо передбачити збереження проміжних станів під час виконання завдань, щоб користувач мав можливість скасувати останню дію. Для реалізації цієї функціональності необхідне надшвидке сховище, яке дозволяє оперативно зберігати та отримувати тимчасові дані. Застосування кеш-менеджера дозволяє не лише забезпечити високу швидкість обробки, а й зменшити навантаження на основну базу даних.

Одним із ефективних рішень для таких задач є використання in-memory сховищ — систем зберігання, що працюють в оперативній пам'яті, на відміну від традиційних SQL-СУБД, які зберігають дані на жорстких чи твердотільних дисках. Завдяки цьому забезпечується миттєвий доступ до даних, що критично для взаємодії з користувачем у режимі реального часу.

Redis є однією з найпопулярніших NoSQL-баз даних, яка підтримує in-memory зберігання та відома своєю високою продуктивністю й широкими можливостями використання в якості кешу.

Проте, оскільки Redis зберігає дані в оперативній пам'яті, існує ризик їх втрати у разі аварійного завершення роботи системи або її вимкнення. Цей аспект необхідно враховувати під час проєктування системи та реалізації стратегій резервного копіювання, щоб мінімізувати можливі втрати даних у надзвичайних ситуаціях [13].

Наступним етапом є створення концептуальної моделі бази даних, яка визначає структуру та взаємозв'язки даних, що зберігаються в системі, та забезпечує логічне і зручне їх використання.

2.2.1 Розроблення концептуальної моделі даних

Концептуальна модель – це абстрактне представлення даних, яке описує їхню структуру та взаємозв'язки. Її створення ґрунтується на аналізі предметної області. Вона гарантує не лише впорядковане зберігання інформації, але й зручне та логічне її використання.

На початковому етапі проектування концептуальної моделі бази даних необхідно виділити основні сутності, які будуть зберігатися в системі (таблиця 2.1).

Таблиця 2.1 — Атрибути сутностей

Сутність	Атрибути
Користувач	Ім'я користувача
	Баланс бонусів
Завдання	Текст завдання
	Переклад завдання
	Тема (FK)
	Коротка теорія
Тема	Назва
	Коротка теорія
Відповідь	Завдання (FK)
	Час відповіді
	Користувач (FK)
	Прийнято відповідь чи ні
	Витрачений час
Слова	Три форми слова
	Переклад
План підписки	Назва
	Вартість
	Тривалість
	Знижка

Продовження таблиці 2.1.

Сутність	Атрибути
Оплата	План підписки (FK)
	Користувач (FK)
	Вартість
	Використані бонуси
	Час оплати
Підписка	Оплата (FK)
	Користувач (FK)
	План підписки (FK)
	Дата активації
	Дата закінчення

Після виділення сутностей та встановлення зв'язків між ними створено діаграму ER яка зображена на рисунку 2.1, яка візуально представить концептуальну модель БД [14].

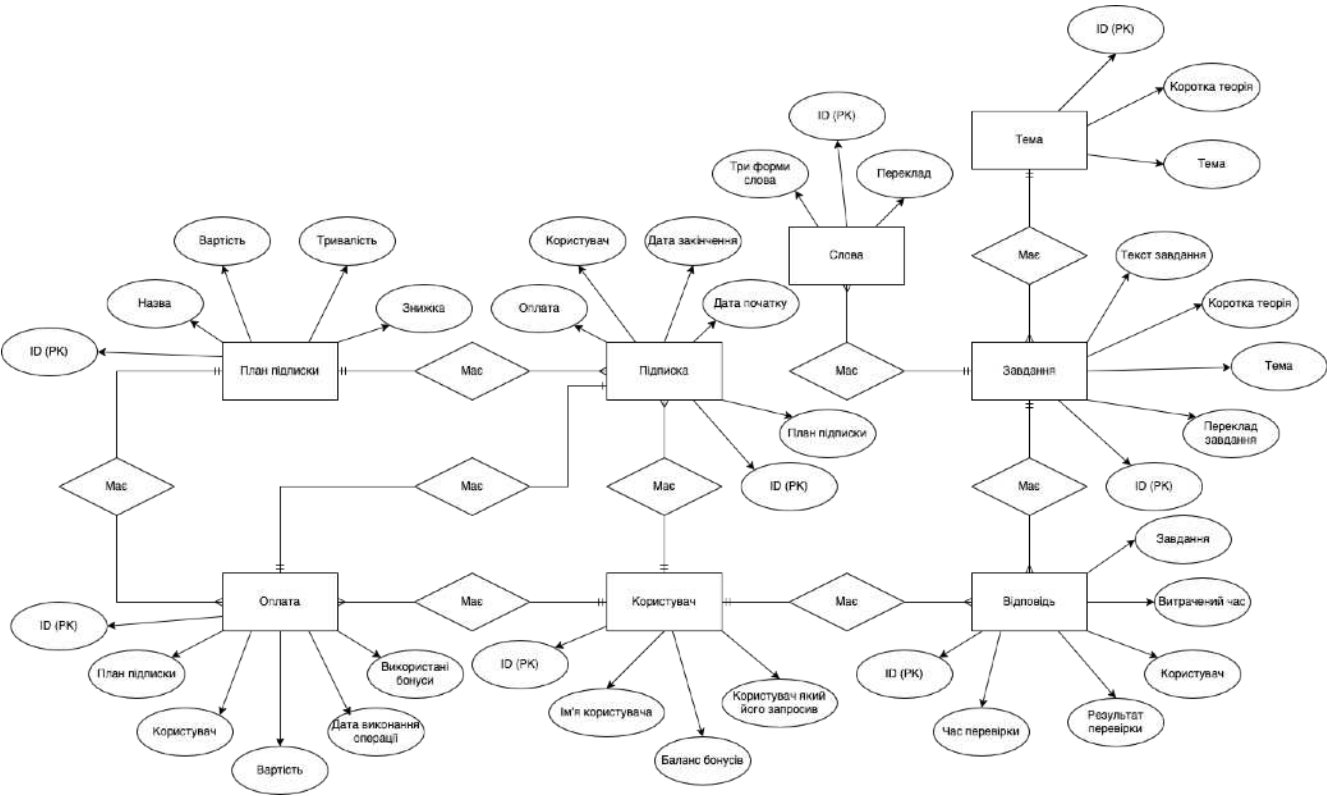


Рисунок 2.1 — Концептуальна модель БД

Візуалізація концептуальної моделі у вигляді ER-діаграми дозволяє наочно побачити структуру предметної області, виділені сутності та взаємозв'язки між ними. Така діаграма є важливим етапом у проєктуванні бази даних, оскільки вона дає змогу ще на ранньому етапі виявити можливі логічні помилки, дублювання інформації або надлишкові зв'язки. Концептуальна модель не залежить від конкретної СУБД і слугує базою для побудови логічної моделі, яка вже деталізує технічні аспекти зберігання інформації.

2.2.2 Розроблення логічної моделі даних

Логічна модель даних є більш детальним описом структури бази даних, ніж концептуальна модель. Вона ґрунтується на концептуальній моделі, але додає більше інформації про типи даних, атрибути, обмеження та ключі.

Розроблену логічну модель даних можна побачити на рисунку 2.2.

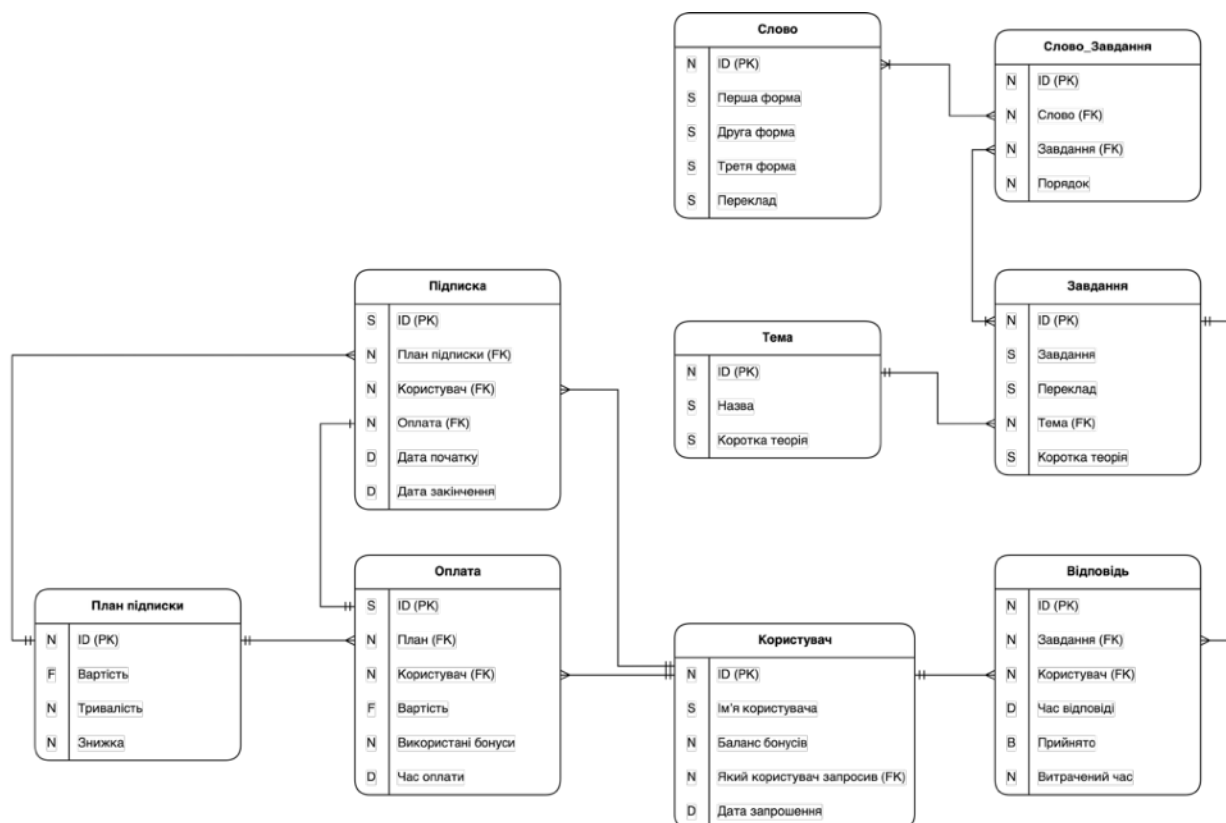


Рисунок 2.2 — Логічна модель даних

У результаті створена фізична модель є основою для генерації SQL-скриптів, які використовуються для створення таблиць у базі даних на сервері. Завдяки чіткому структуруванню даних та правильному визначенню зв'язків між об'єктами, модель забезпечує ефективне зберігання інформації, необхідної для функціонування ІС, зокрема даних про користувачів, граматичні завдання, результати виконаних вправ та інші сутності системи.

2.3 Рішення з математичного забезпечення

Для кращого розуміння логіки взаємодії користувача з системою під час виконання граматичного завдання, на рисунку 2.4 представлено блок-схему, що відображає основні етапи цього процесу. Схема ілюструє послідовність дій, починаючи з ініціації завдання та завершуючи його виконанням, збереження проміжного стану та можливість скасування останньої дії. У даному випадку «Вихід» — означає не завершення роботи програми, а завершення взаємодії користувача із програмою.

На блок-схемі зображено процес виконання граматичного завдання, який поділяється на кілька послідовних кроків. Поточний стан виконання завдання зберігається на сервері інформаційної системи, а саме — у кеш-сховищі Redis, що забезпечує швидкий доступ до даних завдяки збереженню їх в оперативній пам'яті.

Після отримання першої частини завдання користувачеві надається набір варіантів відповіді, серед яких лише один є правильним. Як тільки користувач обирає варіант, що, на його думку, є правильним, відповідь надсилається до інформаційної системи. Система оновлює поточний стан виконання завдання відповідно до вибраної відповіді.

До моменту остаточної перевірки користувач має можливість повертатися до попередніх кроків і змінювати свої відповіді. Якщо користувач доходить до

останнього кроку завдання, система пропонує перевірити результат. У разі згоди користувача бот повертає результат перевірки.

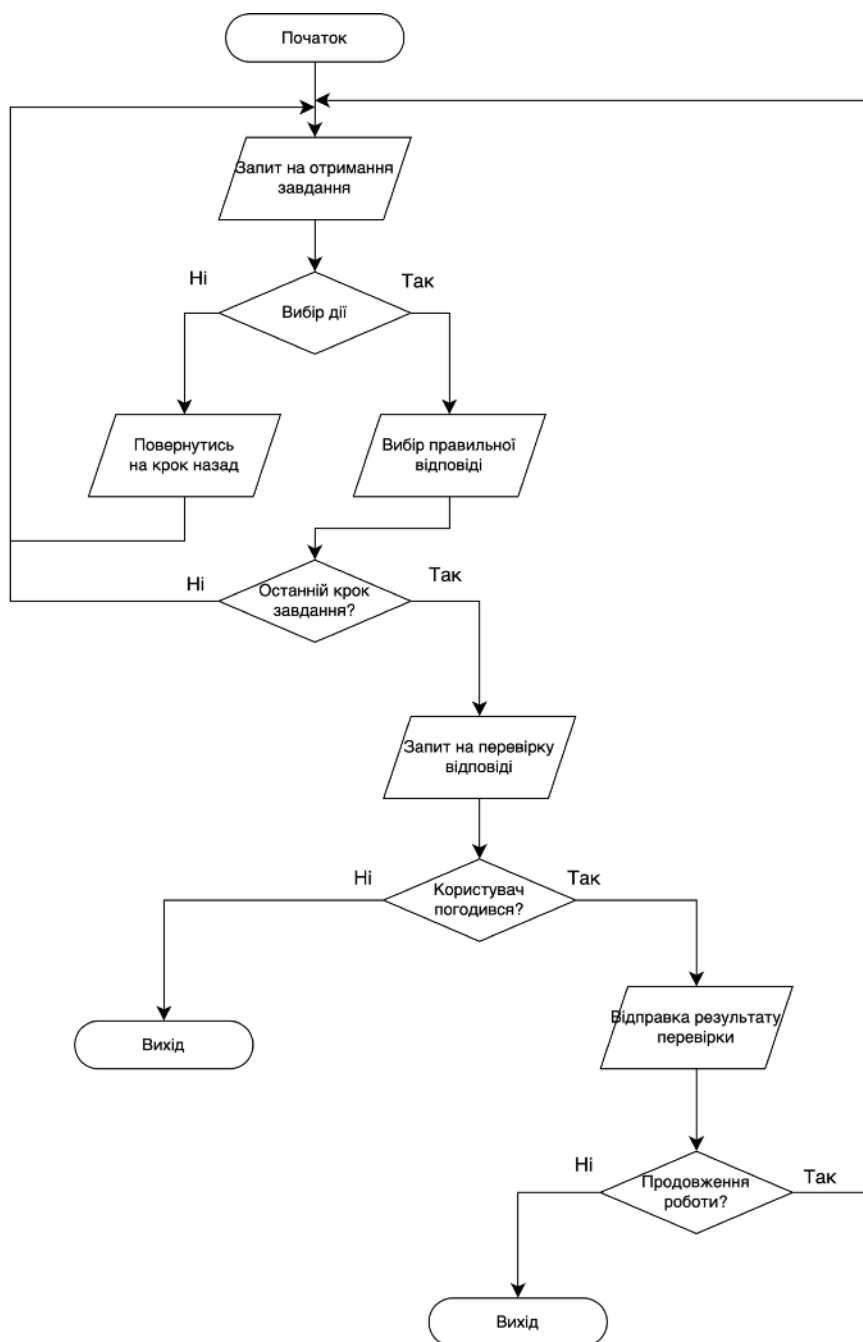


Рисунок 2.4 — Діаграма користувацького досвіду виконання граматичного завдання

Можливість повернення до попередніх кроків реалізовано завдяки збереженню проміжних станів завдання за принципом, подібним до роботи системи контролю версій Git. У Git кожна зміна фіксується у вигляді окремого

знімка (коміту), що дозволяє у будь-який момент повернутися до попередньої версії проєкту. Аналогічно, у даній системі кожна дія користувача створює новий стан, що зберігається окремо, забезпечуючи гнучкість та зручність взаємодії з завданням [15].

Для оцінки правильності відповіді користувача система використовує алгоритм, який порівнює її з правильною відповіддю та визначає ступінь збігу. Такий підхід дозволяє не лише перевірити точність, але й бути стійким до дрібних помилок, таких як зайві або пропущені символи. На початковому етапі обидві відповіді проходять попередню обробку, яка передбачає очищення від зайвих символів та приведення до уніфікованого вигляду. Після цього ініціалізуються змінні, необхідні для обліку кількості збігів між символами та для можливого коригування індексів у випадках зміщення символів. Основна частина алгоритму полягає у поетапному порівнянні кожного символу з обох відповідей. Якщо символи збігаються — фіксується збіг. Якщо ж символи не збігаються, перевіряється, чи є поточний символ у правильній відповіді пробілом. Якщо так - алгоритм пропускає символ у відповіді користувача, інакше - пропускається символ у правильній відповіді. Така логіка дозволяє частково згладжувати помилки, пов'язані з пропусками чи додатковими символами, не знижуючи при цьому загальну точність. Після завершення порівняння виконується розрахунок відсотка збігів відносно максимальної довжини однієї з відповідей. Отримане значення і є оцінкою точності відповіді користувача. На рисунку 2.5 подано блок-схему, яка відображає логіку цього процесу.

Також розроблено алгоритм який виконує підготовку речення для граматичного завдання, в якому потрібно приховати окремі слова. Його основна задача — замінити вибрані слова у реченні на спеціальні пропуски, щоб користувач міг самостійно їх заповнити. Для цього речення спочатку розбивається на окремі слова, далі слова, які потрібно приховати, обробляються: з них прибираються зайві символи, після чого вони замінюються на позначку пропуску. У деяких випадках до пропуску може додаватись підказка, яка потрібна аби спростити виконання завдання для користувача. Після обробки всі частини збираються назад у єдине речення. Наприкінці алгоритм також усуває дублікати

пропусків, які можуть з'явитися поруч, це потрібно для того щоб користувач не бачив кількість кроків у завданні.

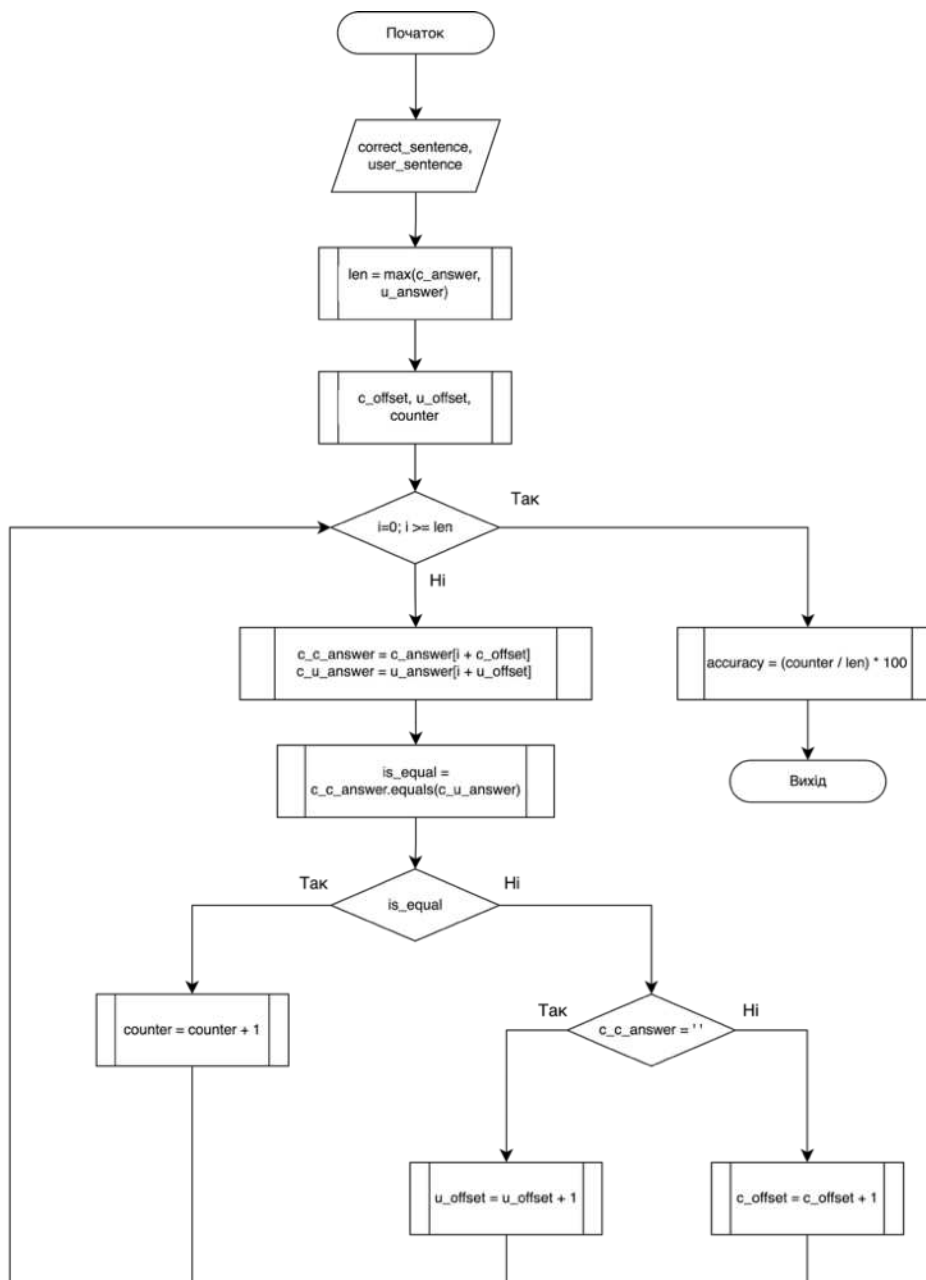


Рисунок 2.5 — Алгоритм порівняння відповідей

2.4 Рішення з програмного забезпечення

Під час проєктування системи першочергову увагу було зосереджено на забезпеченні масштабованості, гнучкості та підтримуваності архітектури. Основне завдання полягало у створенні таких механізмів, які дозволяють розширювати функціональність без значного переписування існуючого коду, а також гарантувати стійку поведінку системи при зростанні складності чи кількості користувачів.

Для формалізації логіки роботи з інтерфейсом було побудовано діаграми переходів станів, що ілюструють ключові сценарії взаємодії із системою. Зокрема, на рисунку 2.6 зображено процес виконання граматичного завдання.

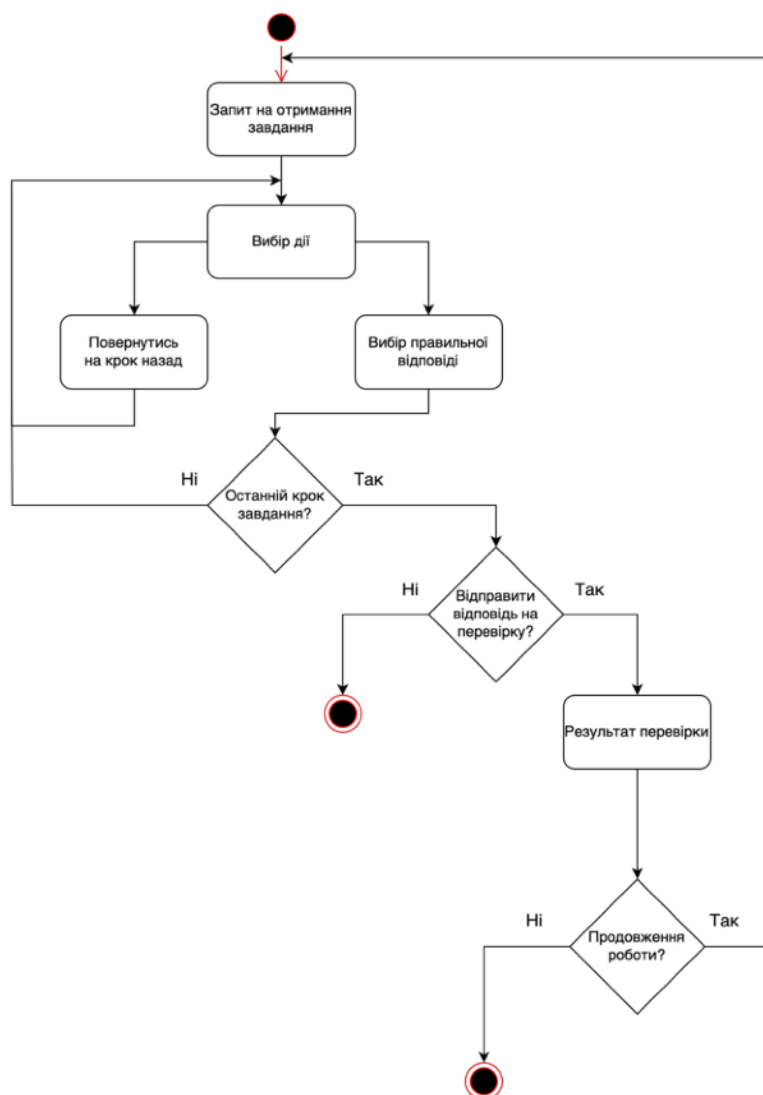


Рисунок 2.6 — Діаграма переходу станів

Він передбачає як лінійне проходження етапів, так і можливість повернення на попередні кроки, що ускладнює внутрішню логіку і вимагає збереження проміжних станів.

Розроблення такої поведінки накладає вимоги до вибору технологій та архітектурних підходів: система повинна мати ефективне сховище станів, що дозволяє реалізувати механізми "відміни", повторного виконання тощо.

Крім того, у структурі додатку мають застосовуватися рішення, які спрощують реалізацію патернів проєктування, зокрема патерна *Команда*, який дозволяє інкапсулювати дії користувача та полегшує контроль за їх виконанням, скасуванням або повторенням.

Таким чином, основну увагу при моделюванні програмного забезпечення варто приділяти забезпеченню чистої архітектури, відокремленню бізнес-логіки від представлення, а також підготовці до використання інструментів і фреймворків, які спрощують реалізацію складної поведінки системи при мінімальних витратах на її підтримку.

Враховуючи ці вимоги, бізнес-логіка інформаційної системи буде реалізована на мові програмування Java. Ця мова програмування з'явилась ще у 1995, і дуже широко застосовується у фінансовій сфері, адже саме в такі застосунки потребують великої надійності та використовуються в умовах великого навантаження, також до 2017 року, саме мова програмування Java була основною, для розроблення мобільних додатків для операційної системи Android [16].

Для розроблення серверних застосунків на мові програмування Java здебільшого використовується сімейство фреймворків Spring. Spring Core - ядро цього сімейства, дозволяє спростити розроблення великих систем, завдяки використанню патерну Dependency Injection (DI), що дозволяє досягти слабкого зв'язування між компонентами системи та полегшує їхнє тестування і масштабування. Однією з головних переваг Spring є модульність – розробник може використовувати лише ті модулі, які потрібні у конкретному проєкті, наприклад Spring Boot, Spring Data, Spring Security тощо.

Spring Boot, значно спрощує конфігурацію та запуск застосунку. Завдяки Spring Boot можна швидко створити production-ready застосунок з мінімальними

зусиллями, оскільки фреймворк надає набір автоматичних налаштувань (auto-configuration) та вбудований сервер (наприклад, Tomcat), що дозволяє запускати застосунок як звичайну Java-програму без необхідності окремого розгортання на сервері [17].

Ще однією перевагою є інтеграція з базами даних за допомогою Spring Data JPA, що спрощує роботу з реляційними базами даних, автоматично реалізуючи основні CRUD-операції. Завдяки цьому, можна зосередитися на логіці застосунку, не витрачаючи час на рутинну реалізацію з'єднання з базою даних.

Також не менш важливою є підтримка аспектно-орієнтованого програмування (AOP), яка дозволяє легко реалізовувати наприклад логування, без зміни основної бізнес-логіки.

Spring має велику спільноту, добре задокументований та постійно оновлюється, що робить його надійним вибором для створення сучасних, масштабованих і підтримуваних веб-застосунків.

Redis — це високопродуктивна in-memory база даних, яка дозволяє зберігати та обробляти дані напряму в оперативній пам'яті, що забезпечує надзвичайно швидкий доступ до них.

Однією з головних причин вибору Redis є його швидкодія — читання та запис відбуваються в мілісекундах, що особливо корисно для кешування, зберігання сесій користувачів, тимчасових токенів або обмеження кількості запитів (rate limiting). У порівнянні з традиційними реляційними базами даних, Redis забезпечує значно меншу затримку та вищу пропускну здатність при обробці великої кількості однотипних запитів.

Ще однією перевагою Redis є простота інтеграції з Spring Framework. За допомогою Spring Data Redis можна легко налаштувати з'єднання з Redis-сервером, серіалізацію об'єктів та реалізацію операцій зберігання/читання.

Задля спрощення розроблення, тестування та розгортання системи варто використовувати на Docker. Docker дозволяє ізолювати компоненти системи у вигляді незалежних контейнерів, що містять увесь необхідний для роботи програмний стек: код, залежності, конфігурації, середовище виконання.

У рамках проєкту Docker використовується для запуску Redis-сервера без потреби ручного встановлення та налаштування. Замість складного конфігурування Redis, достатньо скористатися офіційним Docker-образом, який містить усі необхідні налаштування для швидкого старту. Це дозволяє мінімізувати час, необхідний для запуску допоміжних сервісів, і зосередитися на розробці бізнес-логіки [18].

Однією з ключових переваг Docker є портативність середовища. Застосунок, запущений у контейнері на локальній машині, буде працювати ідентично на сервері чи в хмарному середовищі, що знижує ризик проблем, пов'язаних із “несумісністю середовищ”.

Для розроблення веб-інтерфейсу інформаційної системи було використано фреймворк Svelte. На відміну від більшості сучасних JavaScript-фреймворків, таких як React чи Vue, які виконують більшу частину роботи у браузері під час виконання, Svelte здійснює компіляцію компонентів під час збірки. Це означає, що готовий код не містить об'ємного фреймворку і працює швидше, споживаючи менше ресурсів клієнта.

Однією з ключових переваг Svelte є простота синтаксису, що дозволяє створювати інтерактивні користувацькі інтерфейси з мінімальним обсягом коду. Це суттєво прискорює розроблення та зменшує кількість помилок. Svelte також має вбудовану реактивність, яка реалізується без використання складних станів або зовнішніх бібліотек [19].

Крім того, результуючий веб-застосунок, зібраний Svelte, має менший розмір у порівнянні з аналогами, що позитивно впливає на швидкість завантаження сторінки та загальну продуктивність системи.

На основі обраних технологій, стає можливим сформулювати загальне уявлення про структуру розгортання застосунку. На рисунку 2.7 представлено діаграму розгортання, яка ілюструє взаємодію між основними компонентами системи. Надсилаючи повідомлення користувачі взаємодіють із Telegram-сервером, який у свою чергу, надсилає HTTP-запити до сервера застосунку за допомогою механізму webhook.

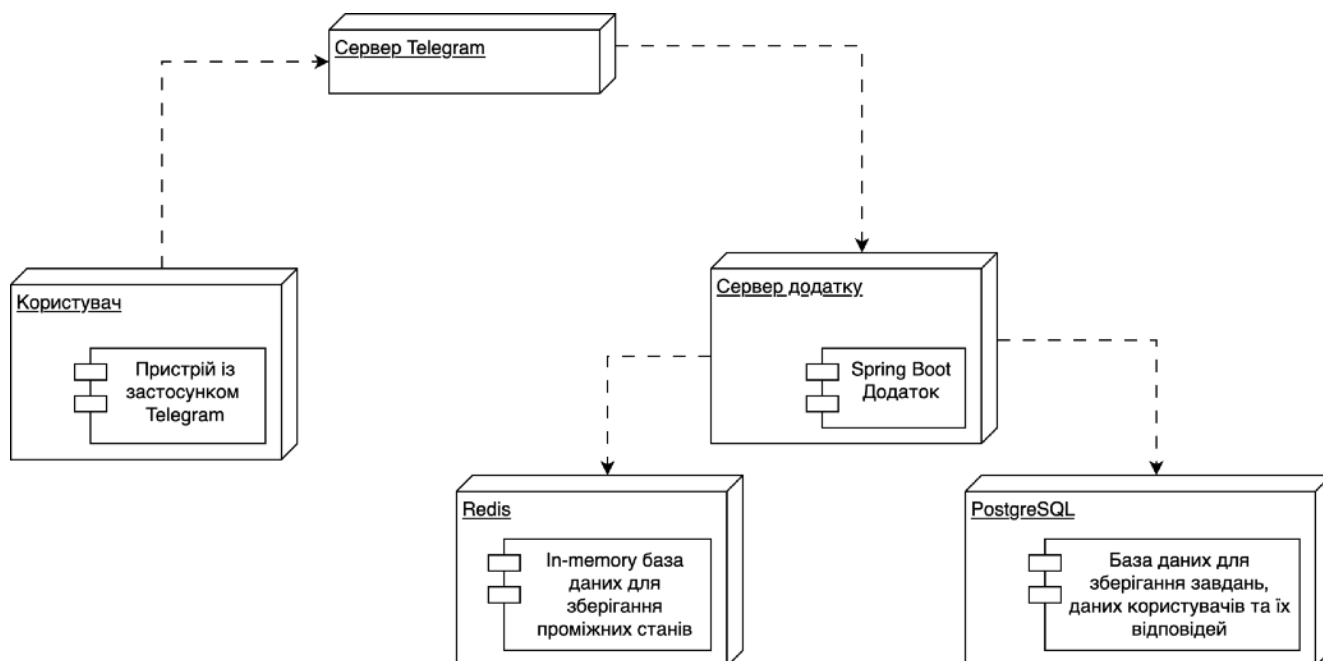


Рисунок 2.7 — Діаграма розгортання системи

Принцип роботи цього механізму полягає в тому, що Telegram автоматично надсилає запити на заздалегідь задану публічну URL-адресу сервера бота щоразу, коли користувач надсилає нове повідомлення. Такий підхід є більш ефективним порівняно з періодичним опитуванням (long polling), оскільки забезпечує миттєву реакцію бота та зменшує навантаження на сервер.

2.5 Рішення з технічного забезпечення

Для забезпечення коректного функціонування інформаційної системи, реалізованої у вигляді Telegram-бота для закріплення англійської граматики, необхідно визначити технічні вимоги як до серверного середовища, де знаходиться інформаційна система, так і до клієнтських пристроїв користувачів, що взаємодіють із системою через додаток Telegram або браузер.

Серверна частина реалізується з використанням мови програмування Java (версія 22), збереженням даних у базі PostgreSQL та кеш сховищем Redis. Для забезпечення стабільної роботи достатньо розмістити сервер у хмарному середовищі або на фізичному сервері з операційною системою Linux, зокрема

Ubuntu 20.04 LTS або новішою. Мінімальні системні характеристики для запуску Java-дodatка, що використовує Spring Boot і PostgreSQL, включають одноядерний процесор з частотою від 2.0 ГГц, щонайменше 512 МБ оперативної пам'яті та близько 2 ГБ вільного місця на накопичувачі. Однак для забезпечення комфортної роботи з урахуванням навантаження, логування та можливої масштабованості, рекомендованим є використання принаймні 8-ядерного процесора, 8 ГБ оперативної пам'яті та 15–20 ГБ дискового простору. Система повинна мати постійне підключення до мережі Інтернет з відкритим доступом до Telegram API, а також підтримувати протокол HTTPS для безпечного з'єднання з клієнтською стороною при використанні webhook-методу. Безкоштовні TLS/SSL сертифікати, наприклад Let's Encrypt, дозволяють реалізувати цей захист без додаткових витрат.

З боку користувача взаємодія з ботом здійснюється через Telegram — багатоплатформовий месенджер, який доступний для різних операційних систем, включаючи Android, iOS, Windows, macOS, Linux, а також у вигляді вебзастосунку. Вимоги до апаратного забезпечення мінімальні, однак залежать від платформи. Для пристроїв на базі Android підтримуються версії починаючи з Android 6.0. Telegram для iOS працює на версіях від iOS 12.0 і сумісний з пристроями починаючи з iPhone 5s, згідно з даними офіційного магазину додатків App Store. На десктопах Telegram підтримується на операційних системах Windows, починаючи з Windows 7, а також на macOS версій від 10.13 і новіших. Вебверсії Telegram WebK та WebZ сумісні з усіма сучасними браузерами, такими як Google Chrome, Mozilla Firefox, Safari та Microsoft Edge, що дає змогу користувачам працювати з ботом без встановлення додатка [20, 21, 22].

У цьому розділі було розглянуто ключові аспекти технічної реалізації інформаційної системи для закріплення знань з англійської мови. Проведено обґрунтований вибір еквайринг-системи для забезпечення прийому платежів. Було встановлено, що інтеграція з LiqPay є оптимальним рішенням для оплати підписок на вебсайті системи завдяки широкій функціональності, підтримці підписок та високому рівню безпеки. Для внутрішніх покупок у мобільному додатку, відповідно до вимог Apple і Google, передбачено використання Telegram Stars, незважаючи на значну комісію.

Висновок за розділом 2

У рамках інформаційного забезпечення обґрунтовано використання СУБД PostgreSQL як основної бази даних системи, що поєднує надійність, розширюваність та відповідність вимогам до складних сучасних проєктів. Для обробки проміжних станів завдань та забезпечення швидкої взаємодії з користувачем впроваджено in-memory сховище Redis, яке дозволяє зберігати тимчасові дані з мінімальною затримкою.

З математичної точки зору було розроблено алгоритми, що забезпечують логіку виконання граматичних завдань, збереження проміжних станів та оцінювання відповідей користувача з урахуванням можливих незначних помилок. Такі алгоритми підвищують гнучкість взаємодії з системою та її адаптивність до потреб користувача.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЄКТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ЗАКРІПЛЕННЯ АНГЛІЙСЬКОЇ МОВИ

3.1 Розроблення структури програмного забезпечення

Перед безпосереднім описом реалізації окремих модулів та використаних технологій варто розглянути загальну структуру програмного коду. Це дозволяє краще зрозуміти, як організовано компоненти системи, які між ними існують зв'язки та як вони взаємодіють між собою. З цією метою нижче представлено діаграму класів (рисунок 3.1, 3.2), що відображає основні концептуальні рішення, прийняті під час розроблення.

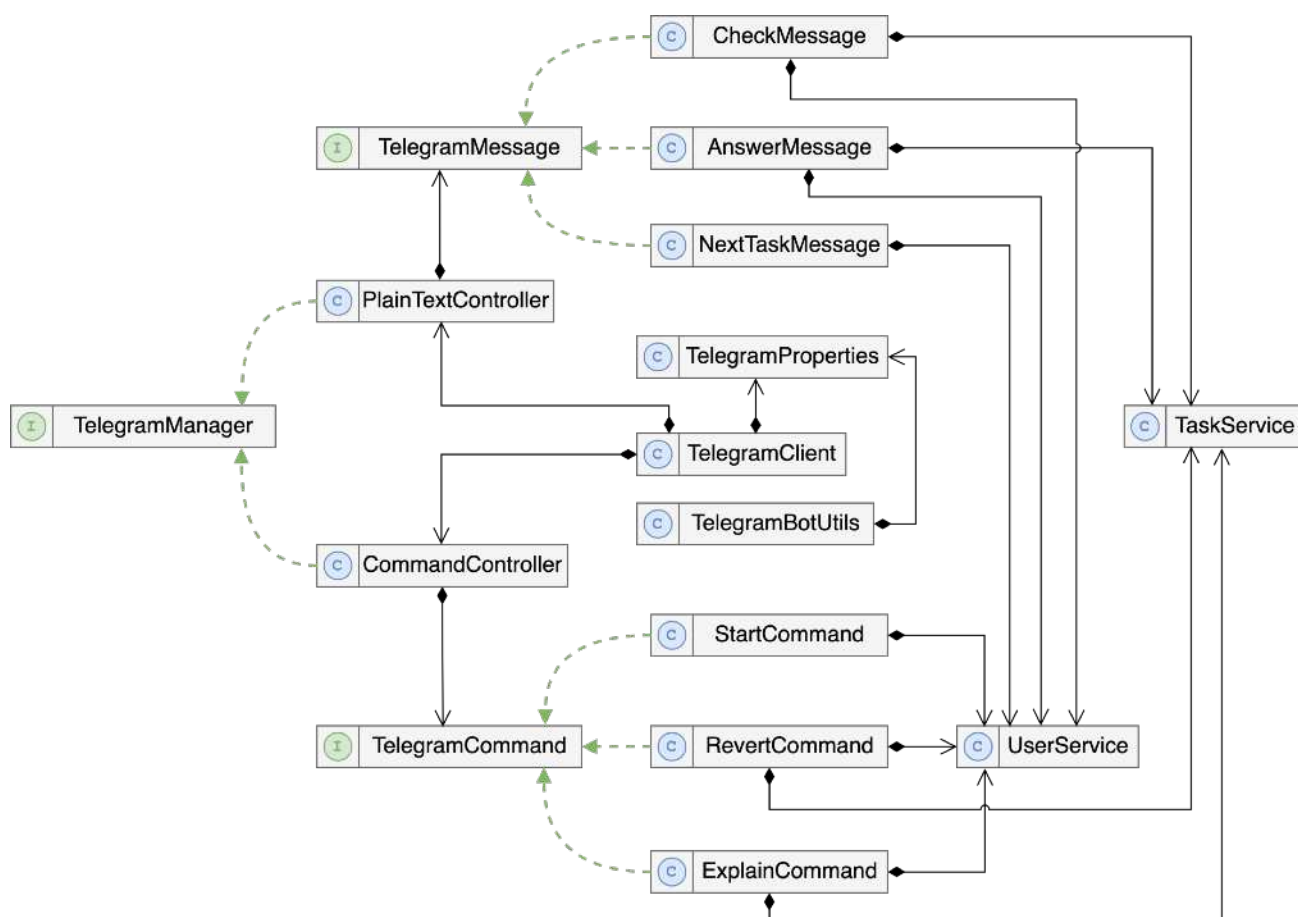


Рисунок 3.1 — Діаграма класів системи 1

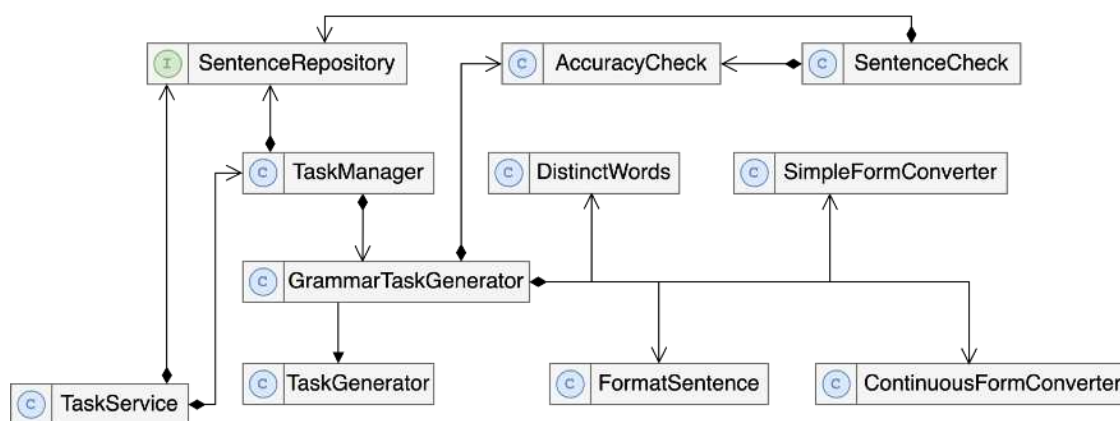


Рисунок 3.2 — Діаграма класів системи 2

На діаграмі класів зображено ключові програмні компоненти системи, що реалізують модульну архітектуру з чітким розмежуванням відповідальностей. Це дозволяє досягти гнучкості, масштабованості та полегшує підтримку застосунку в майбутньому. Архітектура побудована відповідно до принципів об'єктно-орієнтованого проєктування та орієнтована на повторне використання і розширюваність компонентів.

Взаємодія з Telegram Bot API абстрагована за допомогою класів, які реалізують шаблон фасаду (`TelegramManager`, `TelegramClient`). Вони інкапсулюють логіку обробки запитів і відповідають за маршрутизацію повідомлень між клієнтом та системою. Механізм обробки команд реалізовано через патерн «Команда»: абстрактний інтерфейс `TelegramCommand` дозволяє незалежно розширювати функціональність, додаючи нові класи-команди (`StartCommand`, `ExplainCommand`, `RevertCommand`) без модифікації наявного коду, що відповідає принципу відкритості/закритості.

Для відповіді на запити користувача використано ієрархію повідомлень, побудовану на основі патерну спадкування. Базовий інтерфейс `TelegramMessage` дозволяє уніфікувати формат відповіді, тоді як конкретні реалізації (`CheckMessage`, `AnswerMessage`, `NextTaskMessage`) відповідають за відправлення відповідей різного типу в залежності від контексту взаємодії.

Генерація навчальних завдань реалізована з дотриманням принципу інверсії залежностей: сервіс `TaskService` координує роботу генераторів та компонентів перевірки відповідей (`AccuracyCheck`, `SentenceCheck`). Це забезпечує високу

тестованість логіки та можливість гнучкого конфігурування. Компоненти для форматування речень реалізують вузькоспеціалізовану функціональність і дотримуються принципу єдиної відповідальності.

3.2 Результат розроблення програмного забезпечення

В телеграм-ботах взаємодія користувача із ботом здебільшого побудоване на повідомленнях, які можна розглядати як команди, на які ІС повинна певним чином реагувати. Наприклад, кожен користувач коли вперше розпочинає роботу з будь-яким телеграм-ботом бачить екран з однією кнопкою “Розпочати” (Рисунок 3.3), натиснувши на яку автоматично до бота буде надіслана команда «/start», саме тому для цієї команди слід додати логіку автоматичного створення аккаунту користувача та надсилати повідомлення із кнопкою для початку практики граматики (Рисунок 3.4).

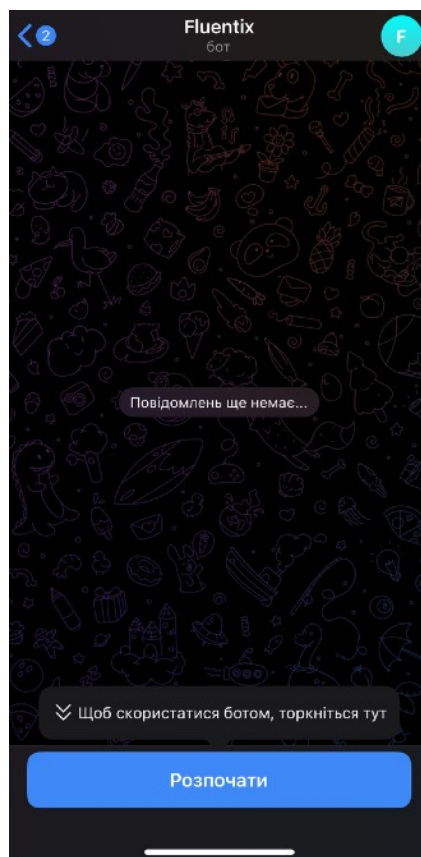


Рисунок 3.3 — Початковий екран телеграм-боту

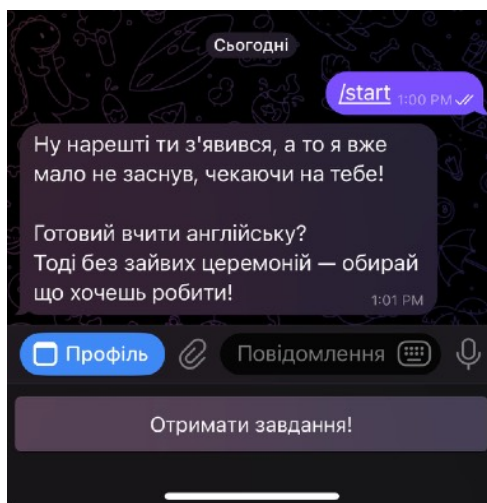


Рисунок 3.4 — Початок роботи із IC

Для такого випадку добре підійде патерн програмування «Команда» (Command Pattern), оскільки він дозволяє інкапсулювати запит як окремий об'єкт, що містить усю необхідну інформацію для його обробки. У контексті Telegram-бота кожне вхідне повідомлення або команда («/start», «/help», або власні текстові запити користувача) може розглядатися як окремий запит до системи, який повинен бути оброблений специфічним чином. Реалізація патерну “Команда” забезпечує чисту архітектуру. Кожна команда обробляється окремим класом, що реалізує спільний інтерфейс або базовий абстрактний клас. Це дозволяє ізолювати логіку обробки окремих команд, спростити модульне тестування, а також розширювати функціонал без модифікації вже існуючого коду, замість цього просто додається нова реалізація команди.

Особливо зручно впроваджувати патерн “Команда” в проєктах, що використовують Spring, адже його механізм ін'єкції залежностей (Dependency Injection) дозволяє автоматизувати процес реєстрації обробників команд. Достатньо визначити спільний інтерфейс, який реалізують усі класи-команди, після чого Spring автоматично знайде всі відповідні бін-класи та додасть їх у потрібну структуру, наприклад, колекцію типу `Map<String, Command>`, де ключем є команда, а значенням її реалізація. Це значно спрощує розроблення, адже для додавання нової команди достатньо створити новий клас-бін, який реалізує інтерфейс, і Spring сам подбає про його реєстрацію та доступність у системі.

Такий підхід не лише усуває необхідність вручну підтримувати диспетчер команд, роблячи архітектуру ІС більш гнучкою й масштабованою.

На цьому етапі взаємодії користувач має змогу натиснути одну з доступних кнопок: «Отримати завдання!» або «Профіль» (синя кнопка в лівому нижньому куті на рисунку 3.4). Якщо користувач обирає кнопку «Отримати завдання!», бот сприймає це як нову команду та формує для користувача наступне завдання. Якщо користувач раніше вже обрав теми для практики, завдання підбирається з-поміж них; у протилежному випадку бот самостійно визначає відповідні теми на основі попередніх відповідей користувача та історії виконаних завдань.

Після вибору теми, бот надсилає завдання у вигляді повідомлення, яке містить текст із пропущеними словами. До цього повідомлення додаються кнопки з варіантами відповідей (рисунок 3.5), з яких користувач повинен обрати правильну. Кожна така відповідь подана у вигляді окремої кнопки.

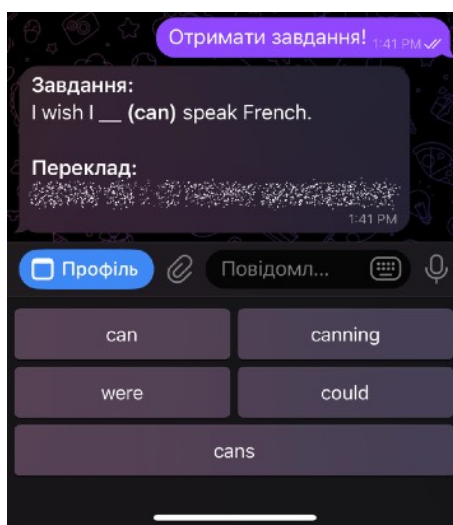


Рисунок 3.5 — Отримане завдання

Після натискання на варіант відповіді бот обробляє вибір та надсилає наступний крок у межах цього ж завдання (рисунок 3.6), у випадку якщо це не останній крок завдання, інакше він надсилає текст заповнений обраними користувачем відповідями, та пропонує перевірити відповідь, тим самим забезпечуючи покрокову, інтерактивну взаємодію.

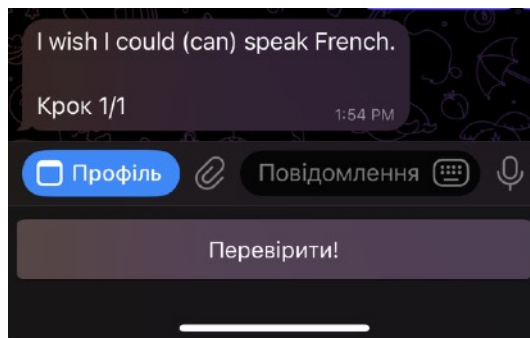


Рисунок 3.6 — Наступний крок завдання

Після заповнення всіх пропусків у завданні, на думку користувача, правильним чином, він має змогу надіслати відповідь на перевірку, натиснувши кнопку «Перевірити». У відповідь система надає повідомлення з результатом перевірки (чи прийнято відповідь), правильною відповіддю та показником точності (Рисунок 3.7).

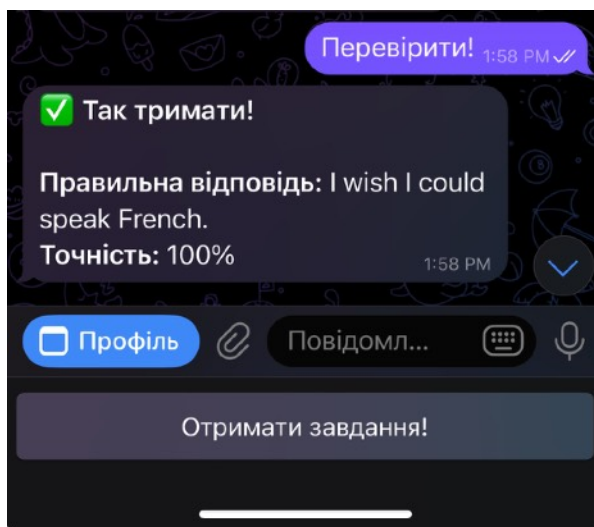


Рисунок 3.7 — Результат перевірки відповіді

У разі, якщо користувач змінив рішення та бажає скоригувати свою відповідь, він може скористатися командою «/revert», яка повертає його до попереднього кроку (Рисунок 3.8).

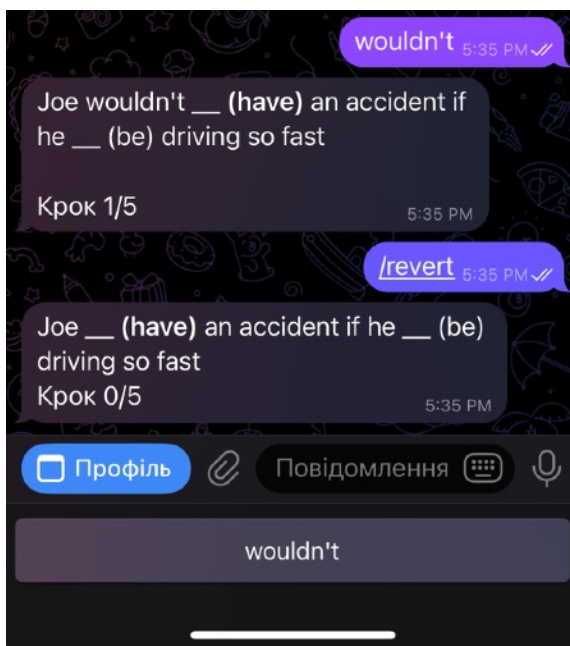


Рисунок 3.8 — Скасування відповіді

Якщо користувач потребує нагадування щодо теми завдання, він може отримати стислий теоретичний матеріал, надіславши команду «/explain».

Натиснувши синю кнопку «Профіль», користувач переходить на вебсторінку (рисунок 3.9), де може обрати теми для практики, переглянути власну статистику за попередні дні, зокрема кількість виконаних завдань, відсоток правильно виконаних завдань, час, витрачений на їх виконання, а також інформацію щодо підписки.

Користувач має змогу обрати одну або декілька тем для поточної практики. Після вибору тем, йому послідовно пропонуватимуться завдання відповідно до кожної з обраних тем. У разі натискання кнопки «Очистити обране», вибрані теми буде видалено зі списку, і надалі користувачеві пропонуватимуться рекомендовані теми.

У блоці «Ваша активність» користувач може переглянути інтенсивність своїх занять у попередні дні: чим яскравіший колір, тим більше часу було приділено виконанню завдань у відповідний день.

У блоці «Сьогодні» відображається зведена статистика за поточну добу: кількість виконаних завдань, точність відповідей (у відсотках) та загальний час, витрачений на виконання завдань.

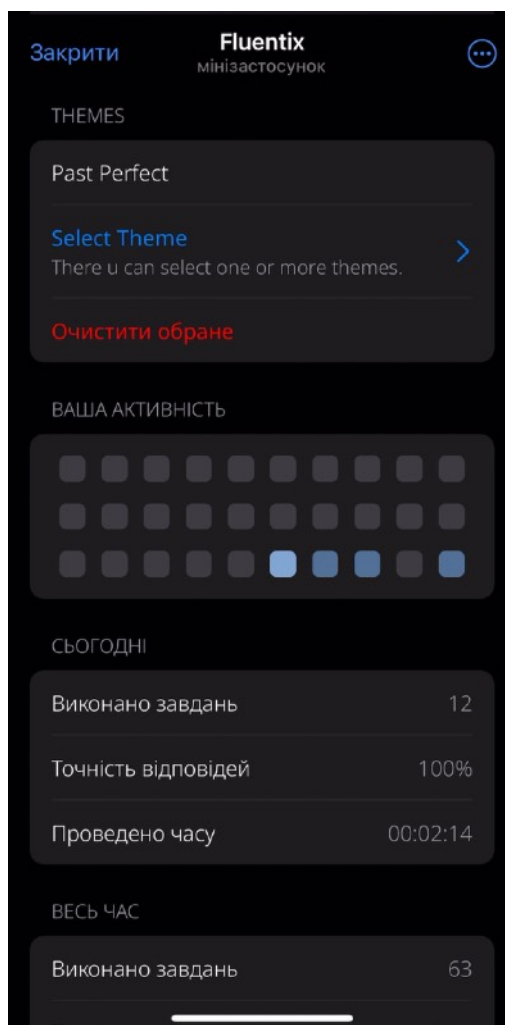


Рисунок 3.9 — Сторінка профілю користувача

Аналогічна інформація, але за весь період використання бота, міститься в блоці «Весь час» (рисунок 3.10). Ці дані дозволяють користувачеві оцінити власний прогрес, виявити тенденції у навчанні та, за потреби, скоригувати підхід до практики.

У наступному блоці відображається інформація щодо активної підписки користувача, зокрема: кількість днів до завершення терміну її дії, дата придбання, дата завершення підписки, а також доступні кнопки для її придбання та скасування (остання доступна протягом перших 15 хвилин після купівлі).

Після натискання кнопки «Придбати Premium» відкривається інтегроване

вікно застосунку Telegram, у якому користувачу пропонується оформити підписку за певну кількість зірок.

Розроблений інтерфейс профілю користувача забезпечує зручний доступ до ключових функцій інформаційної системи, що сприяють ефективному навчанню англійської мови. Користувач має змогу керувати темами для практики, відстежувати власну динаміку навчання, аналізувати статистику виконання завдань, а також керувати підпискою. Інтуїтивно зрозумілий інтерфейс і наочне подання інформації сприяють підвищенню мотивації до регулярного використання системи та покращенню навчального процесу загалом.

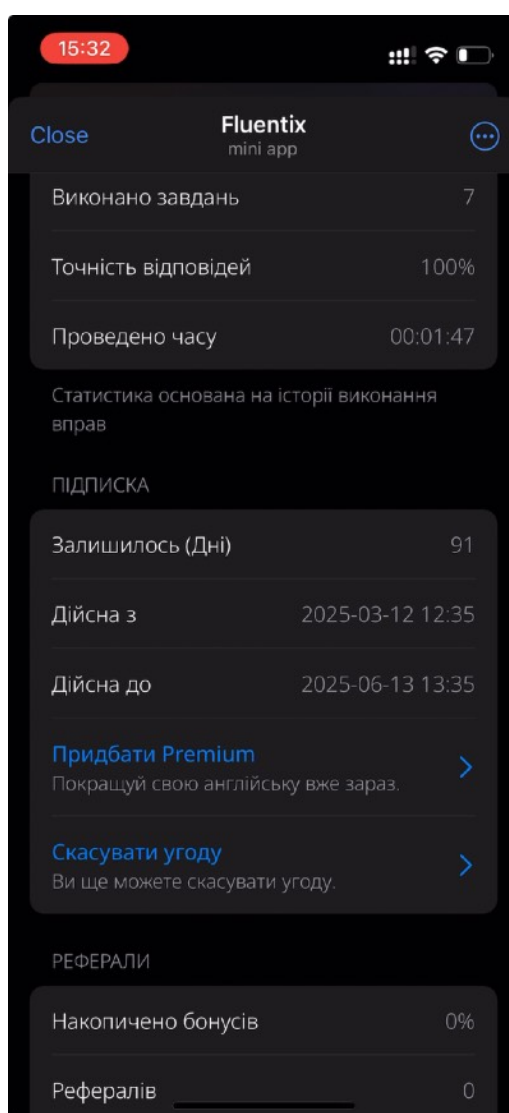


Рисунок 3.10 — Сторінка профілю користувача

Висновок до розділу 3

У розділі було детально розглянуто структуру та основні компоненти реалізації інформаційної системи для закріплення англійської мови у вигляді Telegram-бота. Запропонована модульна архітектура з чітким розмежуванням відповідальностей забезпечує гнучкість, масштабованість та простоту підтримки проекту. Використання об'єктно-орієнтованих принципів і патернів проектування, зокрема шаблону «Команда» та фасаду, дозволило структурувати код таким чином, щоб забезпечити зручність розширення функціоналу та полегшити інтеграцію нових можливостей.

Реалізовані механізми генерації завдань, крокової взаємодії з користувачем, перевірки відповідей та адаптивного підбору тем практики сприяють підвищенню ефективності навчального процесу. Окрему увагу приділено зручному інтерфейсу профілю користувача, який надає доступ до персоналізованих налаштувань, статистики та управління підпискою, що підвищує мотивацію до регулярного використання системи.

Загалом, реалізована інформаційна система відповідає вимогам сучасного навчального застосунку, забезпечуючи інтерактивну, адаптивну та користувачоорієнтовану платформу для вдосконалення знань англійської мови.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1 Загальні питання охорони праці

Охорона праці являє собою комплекс організаційних і технічних заходів, які спрямовані на забезпечення безпеки працівників, збереження їхнього здоров'я та працездатності під час виконання професійних обов'язків. Головною метою охорони праці є створення безпечних і комфортних умов роботи, а також мінімізація ризику виникнення нещасних випадків чи професійних захворювань.

В Україні питання охорони праці регулюються Законом України «Про охорону праці». У цьому законі викладені основні засади державної політики в цій сфері, серед яких – пріоритетність життя та здоров'я працівників над економічними інтересами, обов'язок роботодавця забезпечувати належні умови праці та гарантії соціального захисту для працівників, які постраждали на виробництві. Окрім цього, значну роль у регулюванні охорони праці відіграють різноманітні нормативно-правові документи, санітарні правила, стандарти безпеки та гігієнічні вимоги [23].

Відповідно до законодавства, кожен працівник зобов'язаний ознайомитися з вимогами охорони праці, а роботодавець повинен організовувати навчання і проводити інструктажі з охорони праці на робочому місці. Інструктажі, зокрема первинний, повторний, позаплановий та цільовий, мають здійснюватися у встановлені строки та в повному обсязі.

4.2 Структура управління питаннями охорони праці на підприємстві

На підприємстві система управління охороною праці складається з кількох основних компонентів, які забезпечують виконання державної політики у сфері безпеки праці та дотримання всіх необхідних норм і вимог.

Головну відповідальність за організацію роботи з охорони праці несе керівник підприємства. Він визначає політику та стратегію у цій галузі, організовує фінансування заходів безпеки та забезпечує їх ресурсну підтримку. Також керівник здійснює контроль за реалізацією заходів з охорони праці, щоб гарантувати належний рівень безпеки на підприємстві.

Спеціалізованою структурною одиницею, що займається організацією та контролем виконання заходів з охорони праці, є служба охорони праці. До її обов'язків належать: аналіз поточного стану охорони праці, розроблення заходів щодо поліпшення умов праці, впровадження планів і програм у цій сфері, проведення навчальних заходів та інструктажів для працівників, а також перевірка виконання встановлених норм.

Для розгляду й узгодження питань, пов'язаних з охороною праці, на підприємстві діє комісія з охорони праці, до якої входять представники адміністрації та працівників. Завдання комісії полягають у розгляді планів заходів з охорони праці, участі в аналізі умов праці, розслідуванні нещасних випадків і профзахворювань, а також у підготовці пропозицій щодо поліпшення стану охорони праці.

Працівники обирають свого уповноваженого з питань охорони праці, який представляє їхні інтереси. Уповноважений бере участь у роботі комісії з охорони праці, контролює дотримання вимог безпеки та захищає права працівників на безпечні умови праці.

Інші посадові особи підприємства, зокрема керівники підрозділів, інженери з безпеки праці, начальники цехів та відділів, також зобов'язані дбати про охорону праці в межах своєї компетенції. Вони відповідають за реалізацію заходів безпеки, проведення інструктажів та постійний моніторинг дотримання вимог безпеки на своїх робочих місцях [24].

4.3 Шкідливі та небезпечні фактори, що супроводжують працю програміста

Робота програміста, як і будь-яка інша професійна діяльність, супроводжується певними шкідливими та небезпечними факторами, що можуть мати негативний вплив на здоров'я працівника. Зокрема, тривале перебування у сидячому положенні за комп'ютером призводить до проблем з опорно-руховим апаратом: виникають болі у спині, шийі, можуть розвиватися такі захворювання, як остеохондроз і сколіоз. Відсутність фізичної активності лише погіршує ситуацію, оскільки сприяє застійним явищам у м'язах та суглобах.

Крім того, значне навантаження на органи зору, пов'язане з роботою перед монітором, викликає втому очей, сухість, зниження гостроти зору та може призвести до розвитку короткозорості або інших порушень зору. Ситуацію ускладнюють яскраве світло екрана, відблиски та неправильне освітлення робочого місця.

Монотонність і статичність роботи також чинять негативний вплив, провокуючи психоемоційне напруження, втому та ризик виникнення стресу й депресивних станів. Додатково варто врахувати вплив електромагнітного випромінювання, яке супроводжує використання комп'ютерної техніки. Хоча сучасні прилади випромінюють значно менше, ніж раніше, загроза для нервової системи та загального стану здоров'я все ж існує.

Неправильне або недостатнє освітлення може викликати головний біль, створювати додаткове навантаження на зір і підвищувати ризик травматизму через погану видимість. Окрім цього, високий темп роботи, часті дедлайни, постійне оновлення знань і необхідність освоєння нових технологій сприяють розвитку психологічного стресу та професійного вигорання.

З метою зниження шкідливого впливу цих чинників доцільно організовувати робоче місце з урахуванням ергономіки, робити регулярні перерви для фізичної активності, застосовувати засоби захисту зору та обов'язково проходити профілактичні медичні огляди.

4.4 Ергономічні вимоги до робочого місця при роботі за комп'ютером

Ергономічна організація робочого місця програміста є ключовим фактором для забезпечення комфортних і безпечних умов праці. Це не лише підвищує продуктивність, а й знижує ризик виникнення професійних захворювань. Одним з основних аспектів є правильне розташування монітора: верхній край екрана має знаходитись на рівні очей або трохи нижче, а відстань від очей до екрана — 50-70 см. Це дозволяє уникнути надмірної напруги зору та шийного відділу хребта.

Стіл і стілець повинні бути підібрані таким чином, щоб забезпечувати правильне положення тіла. Регульований стілець із підтримкою попереку, можливістю нахилу спинки та наявністю підлокітників допомагає зменшити навантаження на спину та плечі. Стіл має бути достатньо просторим для зручного розташування обладнання.

Клавіатура та миша повинні розташовуватись на рівні ліктів або трохи нижче, щоб руки залишалися у природному положенні без зайвого напруження. Сьогодні популярні спліт-клавіатури, які допомагають знизити навантаження на зап'ястки та запобігти розвитку тунельного синдрому. Також доцільно використовувати підставки під ноутбук, які дозволяють розмістити екран на потрібній висоті та зручному куті огляду.

Щодо освітлення, варто надавати перевагу природному світлу, а для додаткового освітлення — настільним лампам зі спеціальним регулюванням яскравості та кольорової температури. Для роботи з монітором часто рекомендують спеціальні лампи, які допомагають зменшити відблиски та навантаження на очі. Крім того, важливо звернути увагу на налаштування герцовки (частоти оновлення) екрана: чим вона вища, тим менше навантаження на зір.

Необхідно пам'ятати і про важливість регулярних перерв. Рекомендується робити короткі паузи кожні 60 хвилин для розслаблення м'язів шиї, спини, рук та очей. Також варто виконувати прості вправи для підтримки кровообігу та зняття м'язової напруги.

Висновок до розділу 4

Узагальнюючи викладене, можна зробити висновок, що забезпечення належного рівня охорони праці в сфері ІТ, зокрема для фахівців, які працюють за комп'ютером, має важливе значення для збереження здоров'я працівників, підтримання їхньої працездатності та підвищення ефективності роботи. Попри відсутність фізичних навантажень, діяльність програміста супроводжується впливом низки шкідливих чинників — статичним положенням тіла, високим навантаженням на зір, психоемоційною напругою тощо.

Рациональна ергономічна організація робочого місця, використання сучасних засобів для зниження навантаження (наприклад, спліт-клавіатур, підставок під ноутбук, спеціального освітлення, моніторів із високою герцовкою), а також дотримання режиму праці та відпочинку, є необхідними умовами для мінімізації негативного впливу цих факторів.

Таким чином, впровадження системного підходу до охорони праці у сфері інформаційних технологій є не лише вимогою законодавства, а й запорукою довгострокового збереження здоров'я та професійної придатності працівників.

ВИСНОВОК

У процесі виконання дипломної роботи було здійснено повний цикл розроблення інформаційної системи для закріплення граматики англійської мови. На основі аналізу предметної області та вивчення існуючих програмних рішень сформульовано вимоги до інформаційної системи, що дозволило створити ефективну концепцію. Розроблено архітектуру системи, структуру бази даних, алгоритми обробки користувацьких запитів і механізми навчання, що включають граматичні вправи, повторення матеріалу відповідно до рівня користувача.

Під час реалізації було впроваджено функціонал підписки та реферальну систему, які розширюють можливості монетизації й стимулюють залучення нових користувачів. Проведене тестування підтвердило стабільність роботи системи та відповідність розробленого програмного забезпечення поставленим вимогам. Також було підготовлено супровідну документацію та враховано питання безпеки праці при роботі з комп'ютерною технікою.

У результаті виконаної роботи підтверджено актуальність використання Telegram як платформи для реалізації інтерактивного освітнього інструменту. Розроблений чат-бот забезпечує зручний і доступний спосіб регулярного повторення англійської граматики, що сприяє підвищенню ефективності самостійного навчання. Отримані результати можуть бути основою для подальшого розвитку системи, зокрема доповнення її новими навчальними матеріалами, адаптивними методиками навчання та інструментами аналітики успішності користувачів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. ПРОЄКТ ЗАКОНУ ПРО ЗАСТОСУВАННЯ АНГЛІЙСЬКОЇ МОВИ В УКРАЇНІ. ВЕРХОВНА РАДА УКРАЇНИ. [Електронний ресурс] — Режим доступу до ресурсу: <https://itd.rada.gov.ua>
2. ПРОЄКТ FUTURE PERFECT - МОН. [Електронний ресурс] — Режим доступу до ресурсу: <https://mon.gov.ua>
3. РІВЕНЬ ВОЛОДІННЯ АНГЛІЙСЬКОЮ ТА ІНШИМИ ІНОЗЕМНИМИ МОВАМИ В УКРАЇНІ - КМІС. [Електронний ресурс] — Режим доступу до ресурсу: <https://kiis.com.ua>
4. ІНСТРУКТИВНО-МЕТОДИЧНІ РЕКОМЕНДАЦІЇ МОН У 2024/2025 НАВЧАЛЬНОМУ РОЦІ. [Електронний ресурс] — Режим доступу до ресурсу: <https://osvita.ua>
5. ЗАСВОЄННЯ МОРФОЛОГІЇ АНГЛІЙСЬКОЇ МОВИ: ДОСЛІДЖЕННЯ НА ПРИКЛАДІ СІМ'Ї. ЯНІНЬ ЧЖАН, ІМА ВІДЯСТУТІ
6. ГРАМАТИЧНИЙ АСПЕКТ ВИВЧЕННЯ ІНОЗЕМНОЇ МОВИ. ЛАЗУТКІНА Ю.
7. ДОСЛІДЖЕННЯ ФАКТОРІВ, ЩО ВПЛИВАЮТЬ НА ЗБЕРЕЖЕННЯ МАТЕРІАЛУ В ПАМ'ЯТІ. ПАШКІВСЬКА Л.П., ГІДА Г.П.
8. МОБІЛЬНІ ДОДАТКИ ДЛЯ ВИВЧЕННЯ АНГЛІЙСЬКОЇ МОВИ ЯК ЗАСІБ ОРГАНІЗАЦІЇ САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ. ШЕВЧУК А.О.
9. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ. МАРЧЕНКО А.В. 2016
10. ДОКУМЕНТАЦІЯ APP STORE: ПОКУПКИ ВСЕРЕДЕНІ ЗАСТОСУНКІВ. [Електронний ресурс] — Режим доступу до ресурсу: <https://developer.apple.com>
11. ДОКУМЕНТАЦІЯ GOOGLE PLAY: ПОКУПКИ ВСЕРЕДЕНІ ЗАСТОСУНКІВ. [Електронний ресурс] — Режим доступу до ресурсу: <https://support.google.com>
12. ІНФОРМАЦІЯ ПРО POSTGRE SQL. [Електронний ресурс] — Режим доступу до ресурсу: <https://www.postgresql.org>
13. СИСТЕМА УПРАВЛІННЯ БАЗАМИ ДАНИХ REDIS: ПЕРЕВАГИ ТА НЕДОЛІКИ. А. С. ПРОЦЕНКО, В. М. ГОНЧАР.

14. ВВЕДЕННЯ ДО ER ДІАГРАМ. [Електронний ресурс] — Режим доступу до ресурсу: <https://www.geeksforgeeks.org>
15. ЛОКАЛЬНІ СИСТЕМИ КОНТРОЛЮ ВЕРСІЙ. [Електронний ресурс] — Режим доступу до ресурсу: <https://git-scm.com>
16. ANDROID ANNOUNCES SUPPORT FOR KOTLIN. [Електронний ресурс] — Режим доступу до ресурсу: <https://android-developers.googleblog.com>
17. SPRING. SPRING BOOT OVERVIEW. [Електронний ресурс] — Режим доступу до ресурсу: <https://spring.io>
18. DOCKER. WHAT IS DOCKER? [Електронний ресурс] — Режим доступу до ресурсу: <https://docs.docker.com/>
19. SVELTE. ФРЕЙМВОРК ДЛЯ ВЕБ-РОЗРОБКИ. [Електронний ресурс] — Режим доступу до ресурсу: <https://svelte.dev/>
20. GOOGLE APP STORE - TELEGRAM MOBILE. [Електронний ресурс] — Режим доступу до ресурсу: <https://play.google.com>
21. APPLE APP STORE - TELEGRAM DESKTOP. [Електронний ресурс] — Режим доступу до ресурсу: <https://apps.apple.com>
22. APPLE APP STORE - TELEGRAM MOBILE. [Електронний ресурс] — Режим доступу до ресурсу: <https://apps.apple.com>
23. ЗАКОН УКРАЇНИ ПРО ОХОРОНУ ПРАЦІ. [Електронний ресурс] — Режим доступу до ресурсу: <https://zakon.rada.gov.ua>
24. КОДЕКС ЗАКОНІВ УКРАЇНИ ПРО ОХОРОНУ ПРАЦІ. [Електронний ресурс] — Режим доступу до ресурсу: <https://zakon.rada.gov.ua>

ДОДАТОК А — ТЕХНІЧНЕ ЗАВДАННЯ

Вступ

Завданням даної дипломної роботи є розроблення інформаційної системи для практики англійської граматики. У таблиці 1 наведено загальні відомості про систему.

Таблиця 1 — Загальні відомості про систему

Повна назва системи	Інформаційна система для практики англійської граматики
Розроблювач	Мозжухін Данііл Андрійович
Замовник	Кафедра інформаційних управляючих систем та технологій

1. Підстави для розроблення

Інформаційна система для практики англійської граматики розробляється на підставі завдання для дипломного проектування за спеціальністю 122 — «Комп'ютерні науки».

2. Призначення розроблення

Система призначена для автоматизації процесу самостійного вивчення англійської граматики. Telegram-бот забезпечить інтерактивну взаємодію з користувачем, подачу теоретичного матеріалу, перевірку знань, ведення статистики прогресу та мотиваційні елементи (бали, досягнення тощо).

3. Мета створення системи

Головна мета — підвищення ефективності навчання англійської граматики шляхом регулярної практики в зручному та доступному форматі через Telegram. Система має забезпечити індивідуальний підхід, адаптацію складності завдань до рівня користувача, а також простоту використання без необхідності встановлення окремих програм.

4. Вимоги до програми

4.1 Вимоги до функціональних характеристик

Інформаційна система повинна реалізовувати такі функції:

- Авторизація користувача через Telegram;
- Вибір тем граматики (наприклад, Present Simple, Passive Voice тощо);
- Виведення короткого пояснення теми;
- Проведення тестування (вибір правильної форми, побудова речення, переклад, тощо);
- Ведення статистики результатів та історії виконаних вправ.

4.2 Вимоги до надійності

Бот повинен забезпечувати стабільну роботу 24/7. Важливим є коректне оброблення помилок та збоїв. Всі запити користувача мають мати відповідь не довше ніж за 3 секунди. Дані повинні зберігатися надійно з регулярним резервним копіюванням.

4.3 Умови експлуатації

Інформаційну систему розроблено на мові програмування Java та JavaScript, в якості СКБД використано PostgreSQL.

4.4 Вимоги до інформаційної та програмної сумісності

- Клієнтський пристрій: смартфон або ПК з встановленим додатком Telegram;
- Підключення до Інтернету обов'язкове;
- Серверна частина: хостинг або VPS з підтримкою Java (або іншої мови реалізації) та бази даних (наприклад, PostgreSQL або SQLite);

4.5 Вимоги до технічного забезпечення

Технічні вимоги до клієнтських комп'ютерів визначаються вимогами операційної систем.

5. Вимоги до програмної документації

Всі регламентуючі документи є елементами інформаційної системи, які безпосередньо входять до неї. Регламент системи описує правила її ефективного використання. В таблиці 2 наведено склад документів.

Таблиця 2 — Склад документів організаційно-методичного та експлуатаційного забезпечення

№	Назва	Стадія
1	Опис організаційної структури ІПГІА	Технічне проектування
2	Керівництво користувача	Реалізація системи

6. Стадії та етапи розроблення

В таблиці 3 представлено стадії та етапи по створенню інформаційної системи.

Таблиця 3 — Стадії та етапи робіт по створенню ІС

№	Стадія	Документи
1	Аналіз предметної області	Звіт з описом предметної області
2	Системне проектування	Технічне завдання
3	Ескізне проектування	Ескізний проект
4	Технічне проектування	Технічний проект
5	Робоче проектування	Робочий проект

ДОДАТОК Б

```

@Component
public class RemoveUsedKeyHint implements Function<String, String> {
    @Override
    public String apply(String sentence) {
        return removeAllUsedKeys(sentence);
    }

    private static String removeAllUsedKeys(String sentence) {
        var result = removeKeyHint(sentence);

        if (result.equals(sentence)) {
            return result;
        }

        return removeAllUsedKeys(result);
    }

    private static String removeKeyHint(String sentence) {
        int openParenIndex = sentence.indexOf('(');

        if (openParenIndex >= 2 && sentence.charAt(openParenIndex - 2) != '_') {
            int closeParenIndex = sentence.indexOf(')', openParenIndex);

            if (closeParenIndex != -1) {
                int start = openParenIndex - 1;
                int end = closeParenIndex + 1;

                return sentence.substring(0, start) +
                    sentence.substring(end);
            }
        }

        return sentence;
    }
}

@Component
public class AccuracyCheck implements BiFunction<String, String, Float> {

    @Override
    public Float apply(String answer, String expected) {

```



```

        return getAnswerAccuracy(answer, expected);
    }

    public static float getAnswerAccuracy(String userAnswer, String correctAnswer)
    {
        char[] userChars = userAnswer.toCharArray();
        char[] correctChars = correctAnswer.toCharArray();

        int maxLen = Math.max(userChars.length, correctChars.length);
        int counter = 0;
        int userOffset = 0;
        int correctOffset = 0;

        for (int i = 0; i < maxLen; i++) {
            int userIndex = i - userOffset;
            int correctIndex = i - correctOffset;

            if (userIndex >= userChars.length || correctIndex >=
correctChars.length)
                break;

            char userChar = userChars[userIndex];
            char correctChar = correctChars[correctIndex];

            if (userChar == correctChar) {
                counter++;
            } else if (correctChar == ' ') {
                correctOffset++;
            } else if (userChar == ' ') {
                userOffset++;
            }
        }

        return getPercentage(counter, maxLen);
    }

    private static float getPercentage(int a, int b) {
        if (b == 0) return 100;

        return ((a / (float) b) * 100);
    }
}

```

```

@Slf4j
@Component
public class MarkCurrentKey implements Function<String, String> {
    private static final Pattern omitWithKeyPattern = Pattern.compile("__ \\((\\w+)\\)");
    private static final Pattern keyPattern = Pattern.compile("\\((\\w+)\\)");

    @Override
    public String apply(String sentence) {
        var founded = omitWithKeyPattern.matcher(sentence);

        if (founded.find()) {
            log.info("In sentence founded keys");
            var firstFounded = founded.group(0);

            var extracted = keyPattern.matcher(firstFounded);
            if (extracted.find()) {
                log.info("In sentence extracted keys -> {}, should be replaced with -> {}", firstFounded, extracted.group(0));
                sentence = sentence.replaceFirst(Pattern.quote(firstFounded), getMarkedKey(extracted));
            }
        }

        log.info("Sentence with marked current key -> {}", sentence);

        return sentence;
    }

    @NotNull
    private static String getMarkedKey(Matcher extracted) {
        log.info("Mark key -> {}", extracted.group(0));
        return "__ <b>" + extracted.group(0) + "</b>";
    }
}

@Slf4j
@Component
public class PlainTextController implements TelegramManager {
    private final TelegramMessage ANY;
    private final Map<Integer, TelegramMessage> messages;

    public PlainTextController(List<TelegramMessage> messages) {
        this.messages = messages.stream()

```

```

        .collect(Collectors.toMap(
            TelegramMessage::getName,
            Function.identity()
        ));
        this.ANY = this.messages.getDefault("ANY".hashCode(), null);
    }

    @Override
    public BotApiMethod<?> handle(Update update, TelegramClient client) {
        String messageText = update.getMessage().getText().intern();

        var commandHandler = messages.getDefault(messageText.hashCode(), ANY);

        return commandHandler.apply(update, client);
    }
}

public interface TelegramMessage {
    Integer getName();
    BotApiMethod<?> apply(Update update, TelegramClient telegramClient);
}

```

ДОДАТОК В

При відкритті телеграм-бота з'являється екран з однією кнопкою “Розпочати” (Рисунок В-1), натиснувши на яку автоматично буде створено аккаунт користувача якщо його ще не існує.

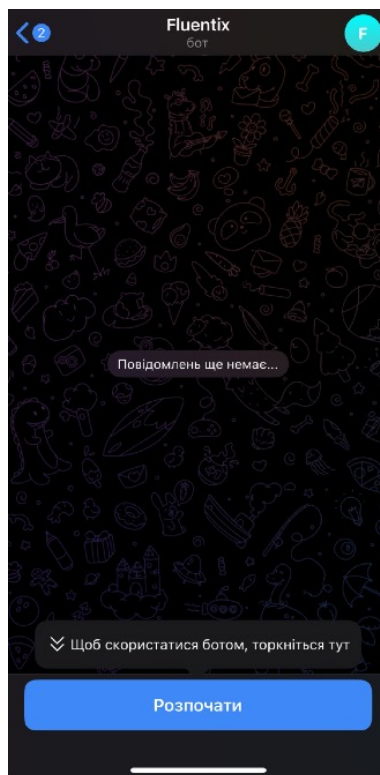


Рисунок В-1 — Початковий екран телеграм-боту

На цьому етапі взаємодії користувач має змогу натиснути одну з доступних кнопок: «Отримати завдання!» або «Профіль» (синя кнопка в лівому нижньому куті на рисунку В-2).

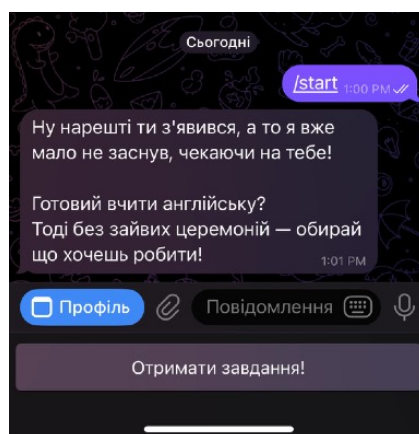


Рисунок В-2 — Початок роботи із ІС

Якщо користувач натискає кнопку «Отримати завдання!», бот надсилає завдання у вигляді повідомлення, яке містить текст із пропущеними словами. До цього повідомлення додаються кнопки з варіантами відповідей (рисунки В-3), з яких користувач повинен обрати правильну. Кожна така відповідь подана у вигляді окремої кнопки.

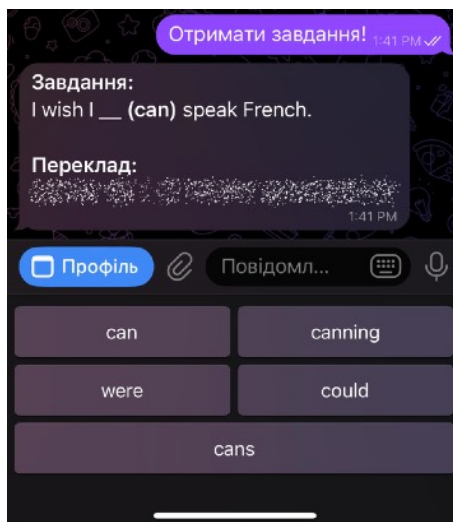


Рисунок В-3 — Отримане завдання

Після натискання на варіант відповіді бот обробляє вибір та надсилає наступний крок у межах цього ж завдання (рисунки В-4), у випадку якщо це не останній крок завдання.

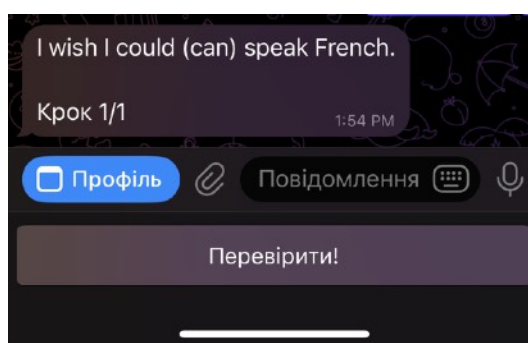


Рисунок В-4 — Наступний крок завдання

У разі, якщо користувач змінив рішення та бажає скоригувати свою відповідь, він може скористатися командою «/revert», яка повертає його до попереднього кроку (Рисунок В-5).

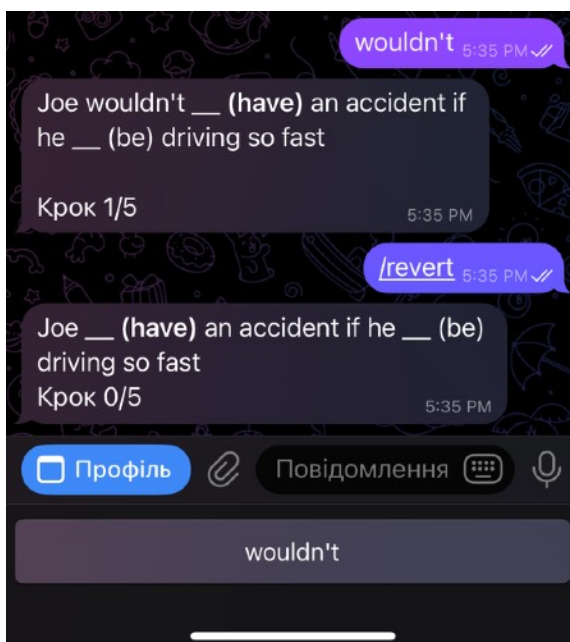


Рисунок В-5 — Скасування відповіді

Після заповнення всіх пропусків у завданні, на думку користувача, правильним чином, він має змогу надіслати відповідь на перевірку, натиснувши кнопку «Перевірити». У відповідь система надає повідомлення з результатом перевірки (чи прийнято відповідь), правильною відповіддю та показником точності (Рисунок В-6).

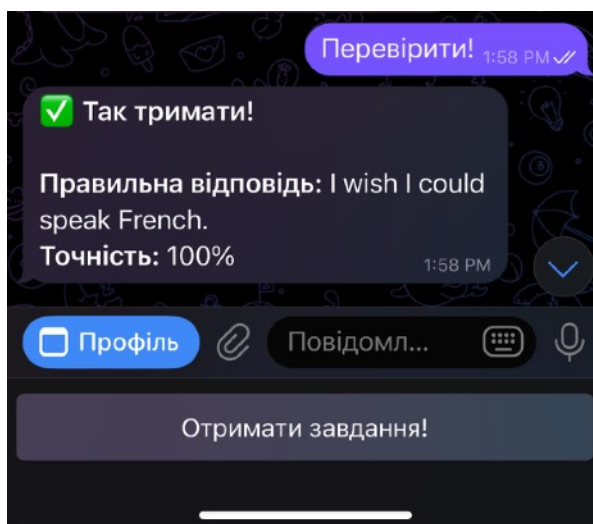


Рисунок В-6 — Результат перевірки відповіді

Натиснувши синю кнопку «Профіль», користувач переходить на вебсторінку (рисунок В-7), де може обрати теми для практики, переглянути власну статистику за попередні дні, зокрема кількість виконаних завдань, відсоток правильно

виконаних завдань, час, витрачений на їх виконання, а також інформацію щодо підписки.

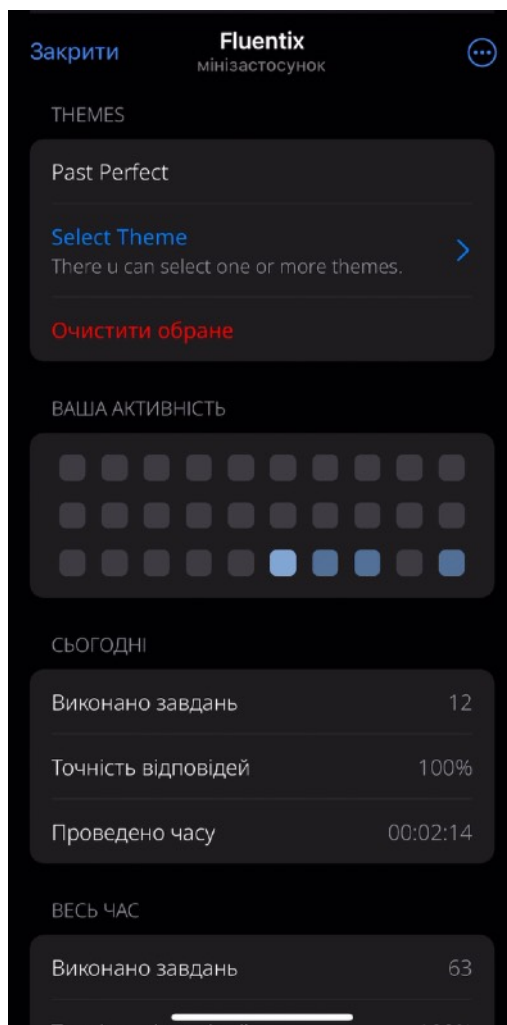


Рисунок В-7 — Сторінка профілю користувача

Користувач має змогу обрати одну або декілька тем для поточної практики. Після вибору тем, йому послідовно пропонуватимуться завдання відповідно до кожної з обраних тем. У разі натискання кнопки «Очистити обране», вибрані теми буде видалено зі списку, і надалі користувачеві пропонуватимуться рекомендовані теми.

У блоці «Ваша активність» користувач може переглянути інтенсивність своїх занять у попередні дні: чим яскравіший колір, тим більше часу було приділено виконанню завдань у відповідний день.

У блоці «Сьогодні» відображається зведена статистика за поточну добу: кількість виконаних завдань, точність відповідей (у відсотках) та загальний час, витрачений на виконання завдань.

Аналогічна інформація, але за весь період використання бота, міститься в блоці «Весь час» (рисунок В-8). Ці дані дозволяють користувачеві оцінити власний прогрес, виявити тенденції у навчанні та, за потреби, скоригувати підхід до практики.

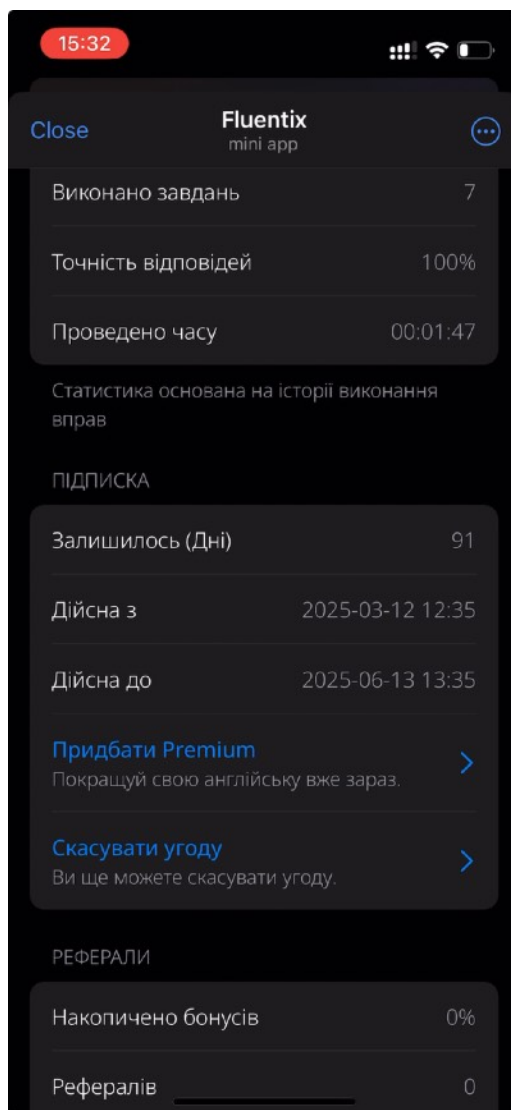


Рисунок В-8 — Сторінка профілю користувача

У наступному блоці відображається інформація щодо активної підписки користувача, зокрема: кількість днів до завершення терміну її дії, дата придбання,

дата завершення підписки, а також доступні кнопки для її придбання та скасування (остання доступна протягом перших 15 хвилин після купівлі).

Після натискання кнопки «Придбати Premium» відкривається інтегроване вікно застосунку Telegram (рисунок В-9), у якому користувачу пропонується оформити підписку за певну кількість зірок.

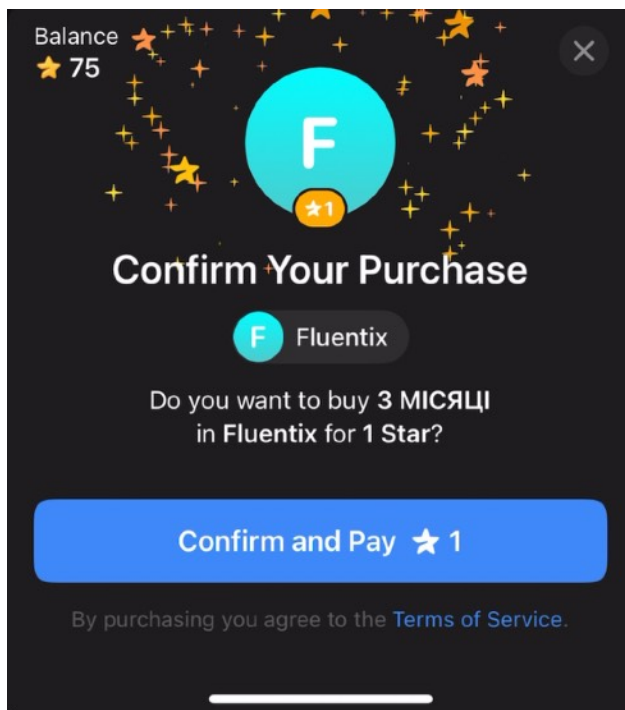


Рисунок В-9 — Вікно оформлення підписки