

Вінницький національний технічний університет
(повне найменування вишого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему:

«Інформаційна технологія для розширення словникового запасу іноземної мови у користувача»

Виконала: студентка 2-го курсу, групи 1КН-22м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)



Карабун В.С.
(прізвище та ініціали)

Керівник к.т.н., доц. каф. КН


Барабан С. В.
(прізвище та ініціали)

« 07 » 12 2023 р.

Опонент: к.т.н., доцент каф. КСУ


Юхимчук М. С.
(прізвище та ініціали)

« 07 » 12 2023 р.

Допущено до захисту
Завідувач кафедри КН
д.т.н., проф. Яровий А. А.
(прізвище та ініціали)

« 08 » 12 2023 р.

Вінниця - 2023 року

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

ЗАТВЕРДЖУЮ
Завідувач кафедри КН

д.т.н., проф. Яровий А.А.

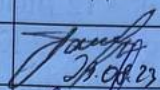
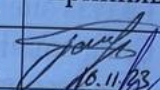
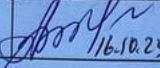
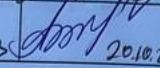
(підпис)

« 19 » 08 2023 року

**ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ**
Карабун Вікторії Сергіївни

1. Тема роботи: «Інформаційна технологія для розширення словникового запасу іноземної мови у користувача»
керівник роботи: Барабан Сергій Володимирович, к.т.н., доц.
затверджені наказом вищого навчального закладу «18» 09 2023 року №114
2. Строк подання студентом роботи 13. 11. 2023 р.
3. Вихідні дані до роботи : мова програмування – Java, середовище розробки – IntelliJ IDEA 2020; мінімальна кількість відповідей – 4; мінімальна кількість записів в базі даних – 30; бібліотеки: Swing, JDBC; база даних – MySQL Workbench 6.3.
4. Зміст текстової частини: вступ, обґрунтування доцільності розробки інформаційної технології для розширення словникового запасу іноземної мови у користувача, моделювання інформаційної технології для розширення словникового запасу іноземної мови у користувача, структурна організація та особливості програмної реалізації інформаційної технології для розширення словникового запасу іноземної мови у користувача, економічна частина висновки, перелік використаних джерел, додатки.
5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): структурна схема інформаційної технології; схема загального алгоритму функціонування інформаційної технології для розширення словникового запасу іноземної мови у користувача; алгоритм тестування для розширення словникового запасу іноземної мови у користувача; загальна структурна схема; діаграма варіантів використання інформаційної технології; ER-діаграма інформаційної технології для розширення словникового запасу іноземної мови у користувача; фрагмент UML-діаграми класів; приклад роботи програми.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Барабан С. В., к.т.н., доц. каф. КН	 28.08.23	 16.11.23
4	Адлер О. О., к.т.н, доц. каф. ЕПВМ	 16.10.23	 20.10.23

7. Дата видачі завдання 29.08.2023р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Обґрунтування доцільності розробки інформаційної технології для розширення словникового запасу іноземної мови у користувача	01.09 23р.	05.09 23р.	
2	Модельовання інформаційної технології для розширення словникового запасу іноземної мови у користувача	06.09 23р.	24.09 23р.	
3	Структурна організація та особливості програмної реалізації інформаційної технології для розширення словникового запасу іноземної мови у користувача	25.09 23р.	15.10. 23р.	
4	Підготовка економічної частини	16.10.23р.	20.10.23р.	
5	Апробація та/або впровадження результатів дослідження	21.10.23р.	05.11.23р.	
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	06.11.23р.	10.11.23р.	

Студент

(підпис)

Карабун В.С.
(прізвище та ініціали)

Керівник роботи

(підпис)

Барабан С. В.
(прізвище та ініціали)

АНОТАЦІЯ

Карабун В. С. Інформаційна технологія для розширення словникового запасу іноземної мови у користувача. Магістерська кваліфікаційна робота зі спеціальності 122 – Комп'ютерні науки, освітня програма – Системи штучного інтелекту. Вінниця, ВНТУ, 2023. 131 с.

На укр. мові. Бібліогр.: 45 назв; рис.: 25; табл.: 7.

Дана магістерська кваліфікаційна робота присвячена розробці інформаційної технології для розширення словникового запасу іноземної мови у користувача. Вона дозволяє підвищити ефективність запам'ятовування іноземних слів у користувача. У загальній частині роботи розглянуто особливості розробки інформаційної технології для розширення словникового запасу іноземної мови у користувача, а також обґрунтована доцільність її розробки. Розроблено алгоритм роботи інформаційної технології для розширення словникового запасу іноземної мови у користувача та відповідне програмне забезпечення на мові програмування Java, в середовищі IntelliJ IDEA, проведено тестування та аналіз результатів роботи.

У розділі економічної частини здійснено оцінювання комерційного потенціалу розробки інформаційної технології для розширення словникового запасу іноземної мови у користувача, проведено оцінювання комерційного потенціалу розробки, спрогнозовано витрати на виконання наукової роботи та впровадження результатів, розраховано період окупності.

Ілюстративна частина складається з 8 плакатів із результатами моделювання.

Ключові слова: інформаційна технологія, словниковий запас, тестування, адаптивний контроль знань, іноземна мова.

ABSTRACT

Karabun V. S. Information technology for for expanding the user's foreign language vocabulary. Master's work in specialty 122 – Computer science. Vinnytsia: VNTU, 2023. 131 p.

In Ukrainian language. Bibliogr.: 45 titles; Figures: 25; Table 7.

This master's qualification work is devoted to the the development of information technology for expanding the user's vocabulary of a foreign language. It makes it possible to increase the efficiency of the user's memorization of foreign words. In the general part of the root, the peculiarities of the development of information technology to expand the vocabulary of a foreign language are considered, as well as the justified feasibility of its development. An algorithm for expanding the user's foreign language vocabulary and the corresponding software in the Java programming language, in the IntelliJ IDEA environment, have been developed, and testing and analysis of the work results have been carried out.

In the section of the economic part, the commercial potential of the development of information technology for expanding the user's vocabulary of a foreign language is assessed, the commercial potential of the development is assessed, the costs of research and implementation of the results are predicted, and the payback period is calculated.

The graphic part consists of 8 posters with modelling results.

Keywords: information technology, vocabulary, testing, adaptive control of knowledge, foreign language.

ЗМІСТ

ВСТУП	4
1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ РОЗШИРЕННЯ СЛОВНИКОВОГО ЗАПАСУ ІНОЗЕМНОЇ МОВИ У КОРИСТУВАЧА	8
1.1 Роль інформаційних технологій у процесі вивчення іноземної мови	8
1.2 Дослідження особливостей використання чат-ботів для розширення словникового запасу іноземної мови у користувача	10
1.3 Огляд існуючих моделей та методів, що застосовуються для вирішення поставленої задачі	14
1.4 Аналіз відомих програмних реалізацій для розширення словникового запасу іноземної мови	20
1.5 Постановка задачі дослідження	28
1.6 Висновок до розділу 1	29
2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ РОЗШИРЕННЯ СЛОВНИКОВОГО ЗАПАСУ ІНОЗЕМНОЇ МОВИ У КОРИСТУВАЧА	30
2.1 Обґрунтування вибору методики для розширення словникового запасу іноземної мови у користувача	30
2.2 Розробка математичної моделі розширення словникового запасу іноземної мови у користувача	33
2.3 Проектування структури інформаційної технології для розширення словникового запасу іноземної мови у користувача	38
2.4 Розробка алгоритму функціонування інформаційної технології для розширення словникового запасу іноземної мови у користувача	42
2.5 Розробка бази даних для інформаційної технології для розширення словникового запасу іноземної мови у користувача	46
2.6 Висновок до розділу 2	50

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ РОЗШИРЕННЯ СЛОВНИКОВОГО ЗАПАСУ ІНОЗЕМНОЇ МОВИ У КОРИСТУВАЧА	51
3.1 Обґрунтування вибору мови програмування та середовища розробки	51
3.2 Обґрунтування вибору месенджера Telegram у якості платформи для взаємодії з користувачем	57
3.3 Опис бібліотек та фреймворків.....	63
3.4 Розробка UML-діаграми класів інформаційної технології для розширення словникового запасу іноземної мови у користувача	66
3.5 Тестування та дослідження результатів роботи інформаційної технології для розширення словникового запасу іноземної мови у користувача	70
3.6 Висновок до розділу 3	78
4 ЕКОНОМІЧНА ЧАСТИНА	79
4.1 Проведення комерційного та технологічного аудиту інформаційної технології для розширення словникового запасу іноземної мови у користувача	79
4.2 Розрахунок витрат на здійснення науково-дослідної роботи.....	80
4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенціним інвестором	87
4.4 Висновок до розділу 4	92
ВИСНОВКИ.....	93
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	95
Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень.....	100
Додаток Б (обов'язковий) Лістинг програми	101
Додаток В (обов'язковий) Ілюстративна частина.....	119
Додаток Г (довідниковий) Інструкція користувача.....	127

ВСТУП

Актуальність теми дослідження. Розширення словникового запасу іноземної мови є важливою складовою процесу вивчення мови. У світі, де міжнародні зв'язки і спілкування стають все більшими, знання іноземних мов стає необхідністю. Розширення словникового запасу допомагає полегшити комунікацію та взаєморозуміння між людьми різних культур та мов, підвищує культурну освіченість, сприяє розвитку мислення. Словниковий запас людини грає важливу роль в організації процесу спілкування як специфічної форми прийому й видачі інформації, обміну поглядами, думками, відчуттями та сприяє взаєморозумінню між людьми. З ростом глобалізації та збільшенням міжнародних зв'язків вивчення іноземних мов стає ключовою компетенцією для багатьох людей. Незалежно від того, чи ставлять вони перед собою особисті, професійні чи академічні цілі, володіння іноземною мовою стає перевагою, яка відкриває нові можливості в спілкуванні, кар'єрному розвитку та освіті.

Заради розширення словникового запасу та навчання іноземним мовам люди завжди шукали нові способи та інструменти. З розвитком інформаційних технологій ця мета стала більш досяжною та цікавою, завдяки можливостям, які вони пропонують. В епоху інформаційних технологій актуальним і своєчасним є використання комп'ютеризованих програм для розширення лексикону іноземної мови. Інформаційні технології вносять суттєвий вклад у цей процес, надаючи засоби та ресурси для зручного та ефективного навчання іноземним мовам. Сучасні комп'ютерні програми, мобільні додатки, інтернет-ресурси та онлайн-платформи надають користувачам можливість вивчати мову в будь-який зручний для них час і місце, використовуючи різні методи та технології.

Використання програмних додатків для вивчення іноземної мови характеризуються своєю зручністю та доступністю, інтерактивністю, різноманіттям вправ, можливістю візуалізації, можливістю проходження

програми по декількох курсах одночасно, можливістю вибору методики, яка найбільше підходить користувачу, самостійністю при навчанні, економією фінансових ресурсів за рахунок використання безкоштовних програм та можливістю використовувати оновлення та розширення.

Актуальність інформаційної технології для розширення словникового запасу іноземної мови у користувача полягає у потребі створення інструменту для ефективного запам'ятовування слів, що забезпечить підвищення рівня словникового запасу іноземної мови у користувача. Сьогодні на ринку представлена велика кількість програм, завдяки яким можна вивчати мову та вдосконалювати комунікаційні навички. Актуальність теми обумовлена попитом на такі програми, які можна використовувати у навчальному процесі.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська кваліфікаційна робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Моделі, методи, технології та пристрої інтелектуальних інформаційних систем управління, економіки, навчання та комунікацій» та плану наукової та навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою роботи є розширення функціональних можливостей програмного додатку по вивченню іноземної мови за допомогою інформаційної технології, що дозволить підвищити рівень словникового запасу у користувача.

Об'єкт дослідження – це процес розширення словникового запасу іноземної мови у користувача.

Предмет дослідження – це програмні засоби для розширення словникового запасу іноземної мови у користувача.

Для досягнення наведеної мети були поставлені та вирішені наступні задачі:

– проаналізувати предметну область розширення словникового запасу іноземної мови у користувача;

- провести аналіз сучасних систем для розширення словникового запасу іноземної мови у користувача;
- розробити структуру інформаційної технології для розширення словникового запасу іноземної мови у користувача;
- розробити математичну модель інформаційної технології для розширення словникового запасу іноземної мови у користувача;
- здійснити програмну реалізацію інформаційної технології для розширення словникового запасу іноземної мови у користувача;
- провести тестування розробленого програмного забезпечення для розширення словникового запасу іноземної мови у користувача;
- економічно обґрунтувати доцільність розробки інформаційної технології програмного забезпечення для розширення словникового запасу іноземної мови у користувача.

Методи дослідження. У роботі використано такі методи наукових досліджень: нейромережеві методи; методи експертних систем; метод адаптивного контролю знань; методи об'єктно-орієнтованого програмування для автоматизації розрахунків.

Наукова новизна одержаних результатів полягає в наступному: удосконалено інформаційну технологію для розширення словникового запасу іноземної мови у користувача, яка відрізняється від існуючих удосконаленим методом, який поєднує в собі методи адаптивного навчання, що дозволяє підвищити ефективність запам'ятовування іноземних слів, тестуючи користувача та оцінюючи його знання.

Практичне значення одержаних результатів полягає у такому:

1. Удосконалено алгоритм розширення словникового запасу у користувача.
2. Здійснено програмну реалізацію інформаційної технології для розширення словникового запасу іноземної мови у користувача.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним

застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими та збіжністю результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

Особистий внесок здобувача. Результати даної магістерської кваліфікаційної роботи отримані самостійно. В публікації у співавторстві здобувачу належить дослідження перспектив інформаційної технології для розширення словникового запасу іноземної мови у користувача [1, 2].

Апробація результатів роботи. Результати роботи були апробовані на ІІ науково-технічній конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2023) [1, 2].

Публікації. За основними результатами досліджень опубліковано тези доповіді на ІІ науково-технічній конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2023) [1, 2].

1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ РОЗШИРЕННЯ СЛОВНИКОВОГО ЗАПАСУ ІНОЗЕМНОЇ МОВИ У КОРИСТУВАЧА

1.1 Роль інформаційних технологій у процесі вивчення іноземної мови

Вивчення іноземної мови – багатогранний процес, який має створювати умови для культурного та особистісного розвитку учнів. Головною метою в освітньому процесі (а в нашому випадку – вивченні іноземної мови) на сучасному етапі розвитку освіти є особистість учня, його мотивація до участі в міжкультурному спілкуванні, здатність самостійно здобувати знання та вдосконалювати їх. Щоб учень міг здобувати знання та вдосконалювати їх, йому слід запропонувати такий підхід до вивчення іноземної мови, який буде сучасним, зручним і певною мірою індивідуальним [3].

Новітні інформаційні технології дають змогу активніше використовувати науково-освітній потенціал провідних університетів та інститутів, залучати кращих викладачів до створення дистанційних курсів, розширювати аудиторію студентів.

Інформаційні технології надають можливість [4]:

- раціонально організувати пізнавальну діяльність учнів у процесі навчання;
- зробити навчання ефективнішим, залучаючи сприйняття учня в мультимедійний контекст і озброюючи інтелект новим концептуальним інструментарієм;
- створити відкриту систему освіти, яка забезпечує кожній людині власну траєкторію навчання;
- залучати до процесу активного навчання категорії дітей з різними здібностями та стилями навчання;

- використовувати специфічні властивості комп'ютера для індивідуалізації навчального процесу;
- активізувати всі етапи навчально-виховного процесу.

В епоху активного розвитку новітніх технологій наше життя неможливо уявити без використання мобільного телефону. Серед усіх сучасних комунікаційних пристроїв це найпотужніший засіб спілкування, навіть багатший за електронну пошту чи чат, де він також може виступати в якості навчального пристрою, незважаючи на свої технічні обмеження. При такій організації навчання вчитель контролює процес і прогрес у власному просторі, виходячи зі свого когнітивного стану.

Якісне навчання іноземної мови неможливо уявити без залучення технічних засобів, адже вони використовуються для підвищення ефективності навчання, всебічного розвитку знань, умінь і навичок учня. Раніше для використання ТЗ на уроці німецької мови необхідно було знайти спеціально обладнану аудиторію. З появою смартфонів ситуація кардинально змінилася. Тепер у кожного учня в кишені є персональний міні-комп'ютер. Зараз мобільні пристрої є практично у всіх, це зручний інструмент для доступу до будь-якої інформації, але не завжди учні використовують свої смартфони, комунікатори та планшети з навчальною метою.

Виділяють такі переваги впровадження смартфонів у навчальний процес [5]:

- мобільний пристрій робить навчання дійсно індивідуальним, учні мають можливість обирати зміст навчання з урахуванням своїх інтересів;
- гнучкість, миттєвий доступ до інформації, необхідної для конкретної роботи, за допомогою мобільних пристроїв дозволяє підвищити продуктивність людини;
- самостійне навчання та негайне надання контенту на вимогу є характерними рисами мобільного навчання.

Існує багато програм, які допомагають вчителю англійської мови та учням в опануванні іноземної мови.

Навчальні програми, що використовують інформаційні технології, мають багато переваг перед традиційними методами навчання. Вони дозволяють тренувати різні види мовної діяльності та поєднувати їх у різних комбінаціях, допомагають усвідомлювати мовні явища, формувати мовні здібності, створювати комунікативні ситуації, автоматизувати мовні дії, а також забезпечують можливість обліку провідної репрезентативної системи, реалізації індивідуального підходу, активізації самостійної роботи користувачів.

1.2 Дослідження особливостей використання чат-ботів для розширення словникового запасу іноземної мови у користувача

Вивчення іноземної мови за допомогою комп'ютера проходить більш ефективно порівняно із традиційними методами навчання, оскільки такий підхід поєднує в собі переваги використання книг, аудіо та відео матеріалів, а також, завдяки доступу до Інтернету, відкриває безмежні можливості для використання автентичних іноземних текстів.

Платформи обміну повідомленнями вже зробили революцію в індустрії освіти – тепер студенти можуть отримувати відповіді на запитання набагато швидше. Проте вчителі не завжди встигають на кожного учня, а допомагати в позаурочний час вони не зобов'язані. Тож можна зробити висновок, що використання чат-ботів, які можуть бути доступні в будь-який час, буде корисним.

Чат-бот (від англ. «chatterbot») – комп'ютерна програма, яка взаємодіє з користувачем за допомогою аудіальних або текстових методів [6]. Чат-боти, або віртуальні співрозмовники, використовуються в діалогових системах для досягнення різних практичних цілей, включаючи обслуговування клієнтів і збір інформації. Деякі чат-боти використовують складні системи обробки природної мови, але в той же час існує багато простих систем сканування введених даних на наявність ключових слів і вибору відповіді, яка відповідає

більшості ключових слів або найбільш схожа на шаблон формулювання з бази даних.

На сьогоднішній день сфера застосування чат-ботів дуже обширна, а впровадження їх у навчальний процес початкової школи, особливо при вивченні іноземної мови, є принципово новим явищем, малопоширеним, а тому потребує розвитку, подальших досліджень і вдосконалень. Створення та використання чат-ботів з метою модернізації навчального процесу, надання йому ознак інноваційності, допоможе уникнути буденності в процесі навчання, підтримає інтерес до вивчення англійської мови, а також буде креативним як для учнів, так і для вчителів.

Переваги використання ботів для вивчення іноземної мови наступні:

Персональне навчання. У класі у вас обмежені можливості практикувати розмову один на один з викладачем і отримувати від нього зворотний зв'язок. Але з роботом-чатом ви отримуете індивідуальну англійську практику.

Ігрове навчання і мотивація. Чат-програми – це така сама гейміфікація (кумедні функції і нагороди, які змушують вас відчувати себе в грі). Наприклад, користувачі виграють дорогоцінне каміння чи значки, якщо використовують бот для вивчення англійської. Ці функції роблять додаток цікавим і забавним. А коли нам цікаво, ми краще навчаємося.

Доступність 24/7. Всякий раз, коли ви відчуваєте, що хочете поговорити англійською, ваш чат-бот буде до ваших послуг. Ви отримаєте миттєву відповідь на будь-яке питання і можете брати участь декілька годин поспіль.

Немає критики, стресу, збентеження. Боти для вивчення англійської не будуть вас критикувати. Вони не зачочують очі, не качають докірливо головою, коли ви робите помилку. Крім того, ви самі не будете відчувати себе збентеженим, якщо ви неправильно використовуєте або вимовляєте слово. Без критики і стресу ви із задоволенням будете практикувати вивчення англійської онлайн.

Отже, з огляду на викладене, можна визначити значний навчальний потенціал розроблюваного навчального чат-бота для вивчення іноземної мови, зокрема:

- підвищення ефективності навчання;
- підвищення рівня мотивації до навчання;
- забезпечення нових можливостей інклюзивної освіти;
- прискорення процесу засвоєння навчального матеріалу;
- розширення можливостей для відпрацювання необхідних практичних навичок у вивченні іноземної мови тощо.

Вивчення іноземної мови за допомогою чат-ботів може бути ефективним і цікавим способом покращити свої навички мовлення, розуміння та граматики. Чат-боти здатні надати користувачам можливість практикувати мову у реальному часі, отримувати зворотній зв'язок та корисні поради.

Наведемо кілька способів, як можна використовувати чат-боти для вивчення іноземної мови [7]:

Розмовний партнер: Багато чат-ботів спрограмовані для здійснення розмовних практик. Ви можете проводити діалоги з чат-ботом, відповідаючи на його запитання та задавати свої питання. Це допоможе вам покращити навички мовлення та граматику.

Вправи на лексику: Деякі чат-боти надають словникові бази даних або можуть пропонувати щоденні вправи на поповнення словникового запасу. Ви можете використовувати ці вправи, щоб вивчити нові слова та фрази.

Граматичні вправи: Чат-боти можуть також надавати граматичні вправи та перевіряти ваші відповіді. Ви можете тренувати правильне використання граматичних правил та отримувати зворотний зв'язок щодо своїх помилок.

Аудіо та відео матеріали: Деякі чат-боти можуть надсилати вам аудіо- або відеофайли для прослуховування та перегляду. Це допоможе вам покращити навички розуміння на слух.

Корекція текстів: Ви можете використовувати чат-бота для перевірки своїх написаних текстів. Він може виправляти помилки та надавати рекомендації щодо поліпшення вашого письма.

Основні переваги вивчення іноземної мови з чат-ботами:

Навчання на основі реальних ситуацій: Чат-боти можуть створювати вигадані розмови або ситуації, що допомагають вам практикувати мову у контексті, подібному до реального спілкування.

Неструктурований досвід: Ви можете експериментувати з мовою без страху зробити помилку, що дозволяє вам набути впевненості в своїх навичках мовлення.

Наявність зворотного зв'язку: Чат-боти можуть надати негайний зворотний зв'язок, виправляючи ваші помилки, роз'яснюючи правильну вимову та граматику.

Нестримне навчання: Завдяки доступності та зручності чат-ботів ви можете вчитися в будь-який зручний для вас час і в будь-якому місці з використанням мобільного телефону або комп'ютера.

Персоналізований підхід: Багато чат-ботів використовують штучний інтелект для адаптації до вашого рівня мовлення та вибору підходящих завдань та матеріалів.

Використання месенджерів для розширення словникового запасу іноземної мови є чудовим способом навчання і практики. Деякі месенджери мають ботів, які надають переклади, визначення слів або пропонують нові слова щодня. Використання чат-ботів для розширення словникового запасу іноземної мови користувача може бути дуже корисним і ефективним. Розглянемо кілька способів використання чат-ботів з цією метою [8]:

1. Словникові боти: Багато мовних сервісів та словників мають своїх чат-ботів. Ви можете використовувати їх, щоб швидко перекласти слова або отримати визначення певного слова. Деякі з них також надають приклади вживання, синоніми та антоніми.

2. Словникові вправи: Деякі чат-боти пропонують різні словникові вправи, які допоможуть вам вивчати нові слова та їх вживання. Ви можете отримувати щоденні вправи, заповнювати прогалини або виконувати інші завдання, щоб практикувати та запам'ятовувати нові слова.

3. Словникові повідомлення: Деякі чат-боти можуть надсилати вам щоденні повідомлення з новими словами або виразами для вивчення. Вони можуть надавати переклад, вживання в контексті та інші корисні відомості про слова.

4. Практика діалогів: Деякі чат-боти можуть моделювати розмову з вами, пропонуючи діалоги, в яких ви повинні використовувати нові слова. Ви можете тренувати свої навички мовлення та використовувати нові слова у реальних ситуаціях.

5. Навчання за запитом: Багато чат-ботів дозволяють вам запитувати переклад або визначення певного слова, яке ви зустріли під час читання або слухання. Вони можуть надати швидку відповідь та пояснення.

1.3 Огляд існуючих моделей та методів, що застосовуються для вирішення поставленої задачі

Моделі і методи для розширення словникового запасу іноземної мови у користувачів можуть здаватися несподіваними, але вони можуть бути корисними для покращення ефективності навчання та розширення словникового запасу. Математичні методи, такі як статистичний аналіз тексту, можуть використовуватися для визначення слів та виразів, що найчастіше зустрічаються у текстах іноземної мови. Це допомагає ідентифікувати ключові слова, які важливі для освоєння.

Методи машинного навчання, такі як нейронні мережі, можуть бути використані для створення програм, які розпізнають слова, вивчені користувачем, та надають завдання та вправи, спрямовані на покращення навичок використання цих слів [9].

Математична кластеризація допомагає групувати слова за семантичними схожістями. Це дозволяє користувачам вивчати слова у контексті та швидше розуміти їхнє використання. Деякі програми використовують алгоритми, які враховують частоту вживання слів та послідовно навчають користувачів нових слів, поступово підвищуючи їхній словниковий запас.

Математичні тести, такі як тести на розширення словникового запасу, можуть допомагати користувачам визначити, на які слова їм варто звернути увагу, а якими вони вже володіють. Аналіз контексту допомагає розуміти, як вживаються слова в реальних реченнях та текстах, що полегшує їхнє запам'ятовування. Математичні методи візуалізації даних можуть бути використані для створення графіків та діаграм, які демонструють зв'язки між словами та їхню вживання.

Ці математичні методи можуть бути використані як в окремих програмах для вивчення мови, так і в більш загальних інформаційних системах для підтримки навчання іноземних мов. Вони допомагають користувачам більше розуміти мовну структуру, розширювати свій словниковий запас та ефективно навчатися новим мовам.

Розглянемо методи машинного навчання для системи розширення словникового запасу іноземної мови у користувача. Методи машинного навчання стали все більш ефективними та популярними для розширення словникового запасу іноземної мови у користувачів [10]. Вони дозволяють створити індивідуалізовані програми навчання та надати користувачам спеціальні завдання та ресурси для ефективного вивчення мови.

Машинне навчання може використовуватися для аналізу тексту на іноземній мові та виділення ключових слів, які користувач може вивчати. Класифікація тексту також допомагає виділити різні типи текстів, такі як новини, романи або наукові статті, що дозволяє користувачам вибирати тексти відповідно до їхніх інтересів. Також за допомогою машинного навчання можна створити рекомендаційну систему, яка пропонуватиме

користувачам тексти, відповідні їхнім інтересам та рівню знань. Такий тип системи аналізуватиме історію користувача та його попередні вибори для створення індивідуалізованих рекомендацій.

Вивчення слів на основі контексту може також відбуватися за допомогою машинного навчання. Воно може аналізувати контекст, у якому вживаються слова в тексті, що допомагає користувачам краще розуміти значення та використання слів.

Машинне навчання дозволяє створювати інтерактивні тести та вправи для вивчених слів. Вони можуть бути адаптовані до рівня знань користувача та автоматично коригуватися. З його допомогою можна побудувати моделі, які можуть автоматично перевіряти правильність відповідей. Деякі програми використовують машинне навчання для аналізу та навчання користувачів вимові іноземних слів. Для створення більш точних моделей навчання та рекомендацій машинне навчання користується великими наборами текстів.

Наступними розглянемо нейронні мережі для розширення словникового запасу іноземної мови у користувача. Використання нейронних мереж є принципово новим підходом до вирішення завдань перевірки та контролю знань, а також ідеально підходить для аналізу текстів природною мовою [11].

Використання нейронних мереж для розширення словникового запасу іноземної мови у користувача стало активно досліджуваною та впроваджуваною практикою в галузі навчання. Нейронні мережі, зокрема глибокі нейронні мережі, виявилися дуже ефективними у вирішенні завдань, пов'язаних із засвоєнням іноземних мов.

Нейронні мережі можуть створювати вправи та завдання для вивчення слів. Наприклад, вони можуть генерувати різні типи питань та вправ, що спрямовані на вивчення вимови, правопису, значення тощо.

За допомогою аналізу користувачів нейронні мережі можуть створювати персоналізовані плани навчання. Вони враховують рівень знань та індивідуальні потреби кожного користувача.

Нейронні мережі можуть автоматично відстежувати прогрес користувача, аналізуючи його відповіді на завдання та вправи. Вони надають зворотній зв'язок та рекомендації щодо подальших кроків у навчанні.

Нейронні мережі можуть створювати візуальні моделі слів та їх семантичних зв'язків, що полегшує їх розуміння.

Наступними розглянемо експертні системи. Експертні системи представляють собою різноманітні комп'ютеризовані системи, які сприяють у прийнятті рішень. Вони комбінують в собі експертні оцінки та результати аналітичної обробки даних [12].

Використання експертних систем для розширення словникового запасу іноземної мови у користувача може значно полегшити і покращити процес навчання. Експертні системи – це програми, які використовують бази даних і правила, щоб надавати рекомендації та вирішувати завдання на рівні, що близький до рівня експерта в конкретній галузі.

Наведемо можливості експертних систем для вивчення іноземної мови.

- можуть аналізувати рівень знань користувача та його потреби, і відповідно надавати персоналізовані навчальні матеріали та завдання для вивчення іноземної мови.
- можуть оцінювати знання користувача шляхом проведення тестів та вправ. Вони можуть визначити рівень користувача та запропонувати відповідний навчальний матеріал;
- можуть аналізувати граматичну структуру текстів та надавати рекомендації щодо правильного вживання слова в контексті;
- можуть враховувати контекст, у якому користувач намагається вивчити слово, та надавати відповідні приклади та вирази для розуміння його використання;
- можуть надавати рекомендації щодо нових слів для вивчення, враховуючи рівень знань користувача та його мовні цілі;
- можуть надавати інструкції та вправи для поліпшення вимови іноземних слів;

- можуть надавати користувачам пояснення стосовно значень слів, виразів та граматичних правил;
- можуть створювати мовні тести, які допомагають користувачам перевірити свої знання та навички;
- можуть вести статистику про прогрес користувача та надавати зворотній зв'язок щодо досягнень;
- можуть бути використані для відповідей на питання користувачів та надання інформації на іноземній мові.

При розробці експертних систем для оцінки якості засвоєння знань враховуються такі ключові принципи:

- роль вчителя як консультанта-фахівця, який консультиує студентів у навчально-виховному процесі;
- відхід від традиційних методів навчання на користь індивідуального підходу, що сприяє самостійній роботі студентів;
- перенесення більшої частини навчального процесу на самостійну роботу студентів;
- підготовка навчально-методичного комплексу з використанням комп'ютерних технологій навчання;
- індивідуальний кумулятивний індекс, що зміцнює роль поточного, транскордонного та підсумкового контролю знань, умінь і навичок.

Незважаючи на переваги, експертні системи мають свої недоліки:

- Здоровий глузд. Експертні системи, поки що, не здатні враховувати аспекти здорового глузду, що є важливим для прийняття рішень.
- Творчий потенціал. Вони не можуть виявляти творчість в незвичайних ситуаціях, як це робить людина-експерт.
- Навчання. Експертні системи потребують явних змін при адаптації до нових умов.
- Сенсорний досвід. Люди-експерти мають багатий сенсорний досвід, що важливо при прийнятті рішень.

- Обмежена компетенція. Експертні системи не ефективні, якщо стикаються з проблемами або ситуаціями за межами їхньої компетенції [13].

Далі розглянемо адаптивний автоматизований контроль знань. Адаптивний автоматизований контроль знань – це потужний інструмент для розширення словникового запасу іноземної мови у користувача. Здійснення автоматизованого контролю знань, умінь учнів, включає розв’язання задачі визначення сукупності необхідних якостей знань, без яких неможливо визначити критерії оцінювання знань та способи визначення рівня їх засвоєння.

Ключовими аспектами моделі адаптивного автоматизованого контролю знань є здатність створення персоналізованих тестів на завдань, що враховують рівень знань користувача та його потреби; відстеження прогресу користувача, фіксуючи вивчені слова; відстеження помилок; використання методів повторення та підсилення пам’яті; автоматичне оновлення завдань; аналіз здібностей та слабких сторін для надання індивідуальних рекомендацій.

Тест-система, базована на моделі Раша, перетворює результати вимірювань, які взяті на дихотомічних і порядкових шкалах, у лінійні величини. Це призводить до можливості аналізувати якісні дані за допомогою кількісних методів, розширюючи спектр статистичних процедур, які можна застосовувати [14].

Модель Раша – це статистична модель, яка використовується для аналізу результатів тестів та питань, що оцінюють знання та навички користувачів. Вона може бути використана для створення тест-системи для розширення словникового запасу іноземної мови у користувача. Модель Раша дозволяє створити систему навчання, яка оптимально пристосована до потреб та рівня знань кожного користувача. Вона допомагає ефективно вивчати нові слова та покращувати володіння іноземною мовою через індивідуалізований та адаптивний підхід до навчання.

1.4 Аналіз відомих програмних реалізацій для розширення словникового запасу іноземної мови

Існує кілька відомих програмних реалізацій, які можуть допомогти вам розширити словниковий запас іноземної мови. Наведемо огляд кількох популярних програмних рішень:

Anki: Anki є програмою для повторення та запам'ятовування слів, фраз та виразів за допомогою флешкарток. Ви можете створювати свої власні колоди карток або завантажувати готові набори від інших користувачів. Anki використовує алгоритми повторення, що допомагають вам вивчати нові слова в оптимальний час.

Quizlet: Quizlet дозволяє створювати та використовувати набори карток для вивчення слів, фраз, термінів тощо. Ви можете робити тестування, грати у вікторини та використовувати інші інтерактивні методи для навчання. Quizlet також надає доступ до широкого спектру вже існуючих наборів, створених іншими користувачами.

LingQ: LingQ дозволяє вивчати мови шляхом читання та слухання різноманітного автентичного контенту, такого як книги, статті, подкасти та інші. Ви можете позначати незнайомі слова, додавати їх у свій словник і практикувати їх з вправами. LingQ також надає доступ до спільноти користувачів для спілкування та обговорення мовних тем.

Memrise: Memrise пропонує інтерактивні уроки та флешкарти для вивчення слів, фраз та виразів. Платформа використовує гру та асоціативні методики для запам'ятовування матеріалу. Memrise також має велику спільноту користувачів, яка допомагає обмінюватися вивченою інформацією та надає можливість практикувати мову з носіями мови.

Duolingo: Duolingo є популярною платформою для вивчення мов, яка пропонує ігровий підхід до навчання. Ви можете виконувати різноманітні вправи, проходити рівні та отримувати очки за правильні відповіді. Duolingo

покриває широкий спектр мов і має розширені курси для практикування читання, письма, аудіювання та говоріння.

Розглянемо деякі з наведених програм, що використовуються в даний час для розширення словникового запасу іноземної мови, більш детально.

Першим розглянемо розширення словникового запасу з іноземної мови за допомогою освітньої онлайн платформи «Duolingo» [15]. Ця безкоштовна платформа надає унікальну можливість вивчати різні мови, включаючи англійську, німецьку, французьку, іспанську, португальську та інші (рис. 1.1).

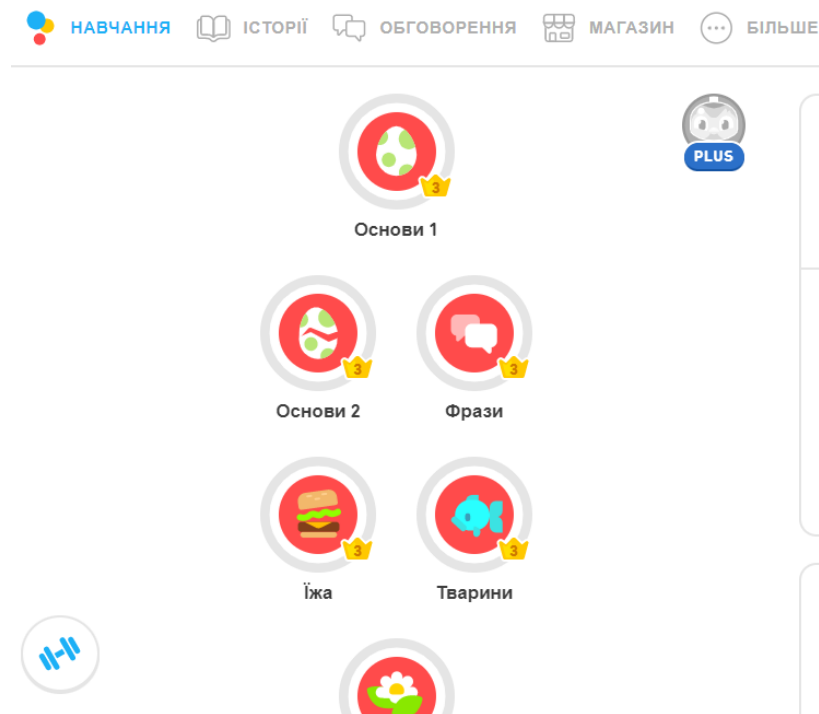


Рисунок 1.1 – Програма «Duolingo»

Метод вивчення на «Duolingo» ґрунтується на принципі гейміфікації, де користувач просувається від рівня до рівня, заробляючи бали. Платформа пропонує наступні можливості:

- «Duolingo» дозволяє користувачам розвивати навички читання, прослуховування та говоріння. Уроки включають завдання на практику розмовної мови, аудіювання, переклад та різноманітні тести для комплексного опанування мовних навичок.

- Платформа здійснює миттєву перевірку правильно виконаних завдань та вказує на помилки. Якщо завдання не виконано правильно, «Duolingo» автоматично повертає користувача до нього та надає рекомендації для вирішення проблем.

- Платформа допомагає користувачам скласти графік вивчення та веде моніторинг часу, витраченого на вивчення конкретної мови.

Таким чином, «Duolingo» не лише робить процес вивчення мови цікавим, а й забезпечує ефективний контроль та навчання в режимі реального часу.

Ще одним ефективним рішенням для поповнення словникового запасу є використання сервісу «Memrise», доступного як веб-сервіс та мобільний додаток (рисунок 1.2) [16]. Цей сервіс сприяє вивченню та запам'ятовуванню нових слів іноземної мови завдяки ігровому інтерфейсу в стилі космосу, що ідеально поєднує естетику та практичність.

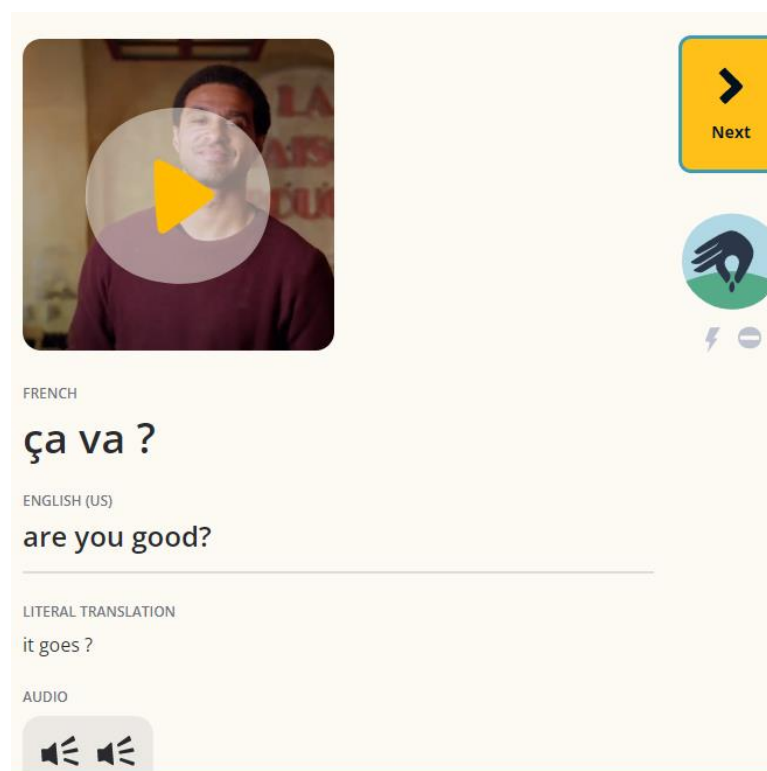


Рисунок 1.2 – Приклад роботи програми «Memrise»

Процес навчання в «Memrise» схожий на інші програми, створені за методом інтервального повторення, але має численні позитивні особливості. По-перше, більшість уроків містять чітко вимовлені слова та фрази, що надає можливість не лише бачити, але й чути нові слова, що є значущим стимулом для їх запам'ятовування. По-друге, режим навчання в «Memrise» організований так, щоб чергувати різні види завдань, таких як вибір правильного перекладу слова, відтворення слова за наявним звуком, відтворення порядку слів у реченні, складання слова з наявного переліку літер, а також запис власної вимови для порівняння з вимовою носія мови:

Режим повторення в «Memrise» доповнюється автоматичним додаванням до поточної сесії навчання слів з попередніх уроків, які, згідно з внутрішніми алгоритмами, можуть бути вже забуті.

Однак, важливо відзначити кілька недоліків програми для вивчення іноземних мов у «Memrise», що може впливати на навчання користувачів:

- в основній, безкоштовній версії додатку «Memrise» доступ до деяких функцій може бути обмежений, що може обмежити можливості користувача в повному обсязі вивчення мови;
- доступ до повного функціоналу може вимагати покупки преміум-версії за високу ціну, \$40 на рік, що може стати неоптимальним вибором для деяких користувачів;
- програма може надавати недостатню кількість граматичних вправ, що може впливати на розвиток коректної мовної структури та використання правильних граматичних форм;
- для користувачів, які вивчають українську мову, може бути обмежений вибір мовних курсів, що може вплинути на доступність різноманітних матеріалів для навчання;
- веб-версія програми може бути менш функціональною порівняно з мобільною версією, що може вплинути на зручність користувачів, особливо тих, хто активно користується веб-браузерами.

Ці аспекти варто враховувати при виборі програми для вивчення іноземних мов, адже вони можуть впливати на загальний досвід користувача та результативність навчання.

Ще однією ефективною альтернативою для розширення словникового запасу іноземної мови є програма «Lexicon By Like Thought, LLC» [17]. Це програмне забезпечення спрямоване на ефективне освоєння нових слів та мовних конструкцій. У програмі існує можливість вводити слова, які ви бажаєте запам'ятати, або додавати потрібні слова зі словника. Додаток пропонує різноманітні методи вивчення, такі як використання флеш-карт, проходження тестів, ігри, диктування та відтворення слів тощо (рисунок 1.3).



Рисунок 1.3 – Програма «Lexicon By Like Thought, LLC»

Особливості програми включають:

- Застосування флеш-карт. Зображення та визначення слів представлені у формі флеш-карт, що полегшує їх запам'ятовування.

- Тестування з вибором відповідей. Програма пропонує тести з різними варіантами відповідей для покращення розуміння та запам'ятовування слів.
- Аудіофункції. Користувач може записувати та прослуховувати вимову слів у форматі аудіо, що сприяє кращому їх засвоєнню.
- Сортування за групами. Можливість групувати слова за певними критеріями для систематизації навчального процесу.
- Підтримка більше ніж 100 мов. Програма надає можливість вивчення широкого спектру мов, що робить її універсальним інструментом для різних користувачів.

Програма Lexicon обіцяє значно збільшити словниковий запас користувача в короткий термін, надаючи різноманітні та ефективні методи вивчення слів іноземної мови.

Іншим досить ефективним засобом для опанування іноземних мов є додаток «Busuu» [18]. Ця програма створена з метою навчання однієї з 11 доступних мов, включаючи, зокрема, англійську. Вона дозволяє користувачам освоювати базову лексику за допомогою різноманітних вправ, словника, відтворення діалогів і інтерактивного тестування (рисунок 1.4). «Busuu» не лише інструмент для навчання, але й соціальна платформа для обміну досвідом та спілкування в сфері вивчення мов.

Однією з особливостей «Busuu» є можливість взаємодії та обміну досвідом у галузі вивчення мов. Платформа створює унікальний простір для спілкування і обговорення результатів, що робить процес вивчення більш інтерактивним. Неабияке значення в цьому контексті має інтерфейс «Busuu», який забезпечує можливість вивчати основні стандарти іноземного словникового запасу, використовуючи різноманітні матеріали, що доступні в бібліотеці додатка. Серед них – різні озвучені діалоги та інтерактивні тексти, які підтримують різні рівні складності та стилі навчання.

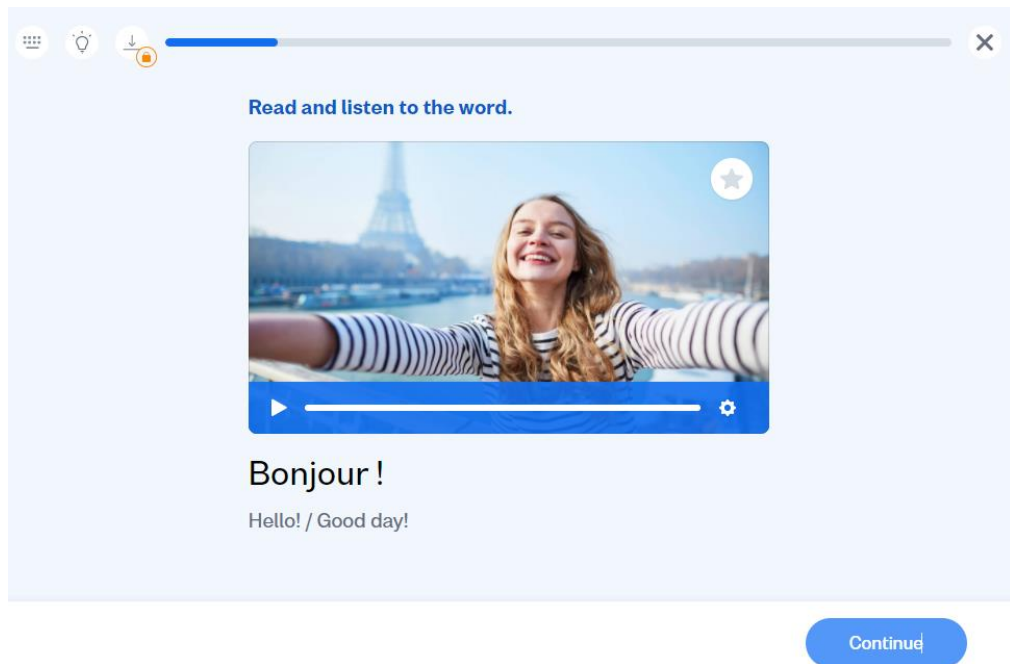


Рисунок 1.4 – Програма «Busuu»

Окрім цього, важливо відзначити можливість відправити виконане завдання на перевірку носію мови. Ця функція забезпечує зовнішню оцінку результатів і дозволяє користувачеві отримати фідбек від носія, що стимулює поліпшення мовних навичок. Такий інтерактивний підхід до вивчення мови робить «Busuu» привабливим інструментом для тих, хто прагне не лише отримати теоретичні знання, а й застосувати їх у реальних ситуаціях.

Особливу нішу в розвитку навичок іноземної мови займають програми, які працюють через боти та канали у популярному месенджері Telegram. Один із таких ботів – «Andy English Bot» [19]. Боти є великою перевагою Telegram Messenger, використання яких допомагає поліпшити мовні навички (рисунок 1.5). Бот Andy не лише створений для спілкування англійською мовою, а й дозволяє практикуватися за допомогою виконання різноманітних вправ. Користувачам доступні завдання, можливість задавати питання про нові цікаві слова або навіть брати участь у цікавих іграх.

Але важливо відзначити, що у бота є свої недоліки. Наприклад, його інтерфейс не є абсолютно інтуїтивно зрозумілим, і він дещо перевантажений функціями, що може здаватися деяким користувачам надто складним. Також

слід відзначити, що, незважаючи на збір статистики, бот не забезпечує можливості оцінювання рівня знань. Тобто йому бракує системи, яка дозволяла б користувачам отримувати конкретний фідбек щодо їхнього прогресу в опануванні англійської мови.

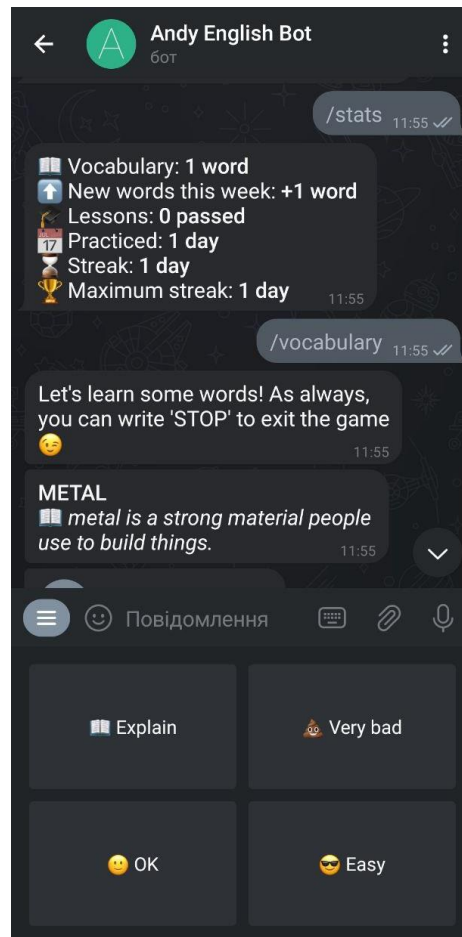


Рисунок 1.5 – Приклад роботи програми «Andy English Bot»

Після аналізу функціональних можливостей аналогічних програм, таких як «Duolingo», «Memrise», «Lexicon By Like Thought, LLC», «Busuu» та «Andy English Bot» можна виділити кілька загальних тенденцій та недоліків. Перш за все, у більшості розглянутих програм обмежений функціонал безкоштовної версії, і ціна за преміум доступ вища, ніж може бути доступно для багатьох користувачів. Також, інтерфейс більшості додатків може викликати складнощі у користувачів, особливо для тих, хто тільки починає

ознайомлюватися з ними. Це може вплинути на загальний комфорт використання та вивчення мов.

Серед розглянутих програм сервіс «Memrise» вирізняється своєю пристосованістю до вивчення словникового запасу і може бути найбільш приближеним за характеристиками до розроблюваної системи розширення словникового запасу іноземної мови. Однак, враховуючи вигоди використання ботів у Telegram, які були проілюстровані на прикладі «Andy English Bot», потрібно взяти до уваги переваги платформи для розробки інформаційної технології. Це рішення може виявитися більш гнучким та інтуїтивно зрозумілим, адже бот може надавати конкретний фідбек щодо прогресу користувача та адаптуватися до його потреб у навчанні мов. Отже, в подальшому обиремо як прототип до розроблюваної системи саме ці дві інноваційні платформи – «Memrise» і телеграм-бота «Andy English Bot».

1.5 Постановка задачі дослідження

Задача цієї роботи полягає у розробці інформаційної технології для розширення словникового запасу іноземної мови у користувача.

Вимоги до користувача:

- можливість вибору режиму роботи;
- можливість вибору режиму тестування;
- перегляд варіантів відповіді;
- можливість виходу під час тестування;
- завершення сесії (вихід).

Вимоги до адміністратора:

- можливість перегляду статистики;
- можливість додавання нової інформації (слів);
- можливість проходження тестування.

Вимоги до серверу:

- зберігання даних користувача через зв'язок з базою даних;
- видача необхідних даних по запиті від користувача.

Отже, потрібно розробити інформаційну технологію, в якій буде реалізовано алгоритм розширення словникового запасу іноземної мови у користувача. В контексті даної роботи фінальним набором є оцінка здібностей учня, доступ до якої отримає вчитель. Для забезпечення можливості вчителю перевіряти знання учнів вводяться дві ролі: адміністратора та користувача. У режимі адміністратора вчитель може додавати нові слова, а також переглядати статистику та оцінки знань учнів. Для учнів створюється режим користувача, де вони мають можливість тренувати свій словниковий запас.

1.6 Висновок до розділу 1

В даному розділі було розглянуто важливість ролі інформаційних технологій у процесі вивчення іноземної мови. Досліджено та обґрунтовано переваги використання чат-ботів для розширення словникового запасу іноземної мови у користувача. Було проведено аналіз сучасних програм-аналогів, допомагають у вивченні іноземної мови, забезпечують перевірку та тестування знань, а також сприяють розширенню словникового запасу іноземної мови, наведено короткий опис основних функцій, які виконують дані програми, наведено їх переваги та недоліки. Під час вибору прототипу було вирішено обрати платформу «Memrise» і телеграм-бота «Andy English Bot», оскільки вони відзначаються своєю пристосованістю до збільшення словникового запасу і можуть бути найбільш приближеним за характеристиками до розроблюваної інформаційної технології розширення словникового запасу іноземної мови. Також було досліджено методи, що можуть бути використані для розширення словникового запасу іноземної мови у користувача, для чого запропоновано використати адаптивний контроль знань, що забезпечить персональний підхід до тестування кожного користувача, впровадить інтервальні повторення для кращого запам'ятовування слів, а також проведе оцінювання ефективності розширення словникового запасу у користувача.

2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ РОЗШИРЕННЯ СЛОВНИКОВОГО ЗАПАСУ ІНОЗЕМНОЇ МОВИ У КОРИСТУВАЧА

2.1 Обґрунтування вибору методики для розширення словникового запасу іноземної мови у користувача

Впровадження комп'ютерних систем для тестування є перспективним напрямком сучасного навчального процесу. Використання тестів для запам'ятовування слів та лексичних конструкцій іноземної мови має численні переваги та може бути дуже ефективним інструментом навчання [20]. Перевагами підходу є:

- автоматизація індивідуалізованого навчання (Комп'ютерні системи тестування можуть створювати індивідуалізовані тести та завдання, які враховують рівень знань іноземної мови кожного користувача. Це дозволяє надавати завдання, які відповідають конкретним потребам та навичкам кожного користувача);
- миттєвий зворотній зв'язок (негайне надання зворотнього зв'язку сприяє негайній корекції помилок та покращенню навичок користувача);
- зручність та доступність (не залежність від часу та місця, використовуючи комп'ютер, планшет або смартфон);
- аналіз прогресу та статистики (дозволяє як користувачам, так і вчителям відстежувати навчання та визначати слабкі місця);
- адаптивність (мають можливість адаптувати рівень складності завдань до рівня знань користувача);
- ефективне використання часу (дозволяють користувачам вчити слова власним темпом та не витратити час на дорогу до навчальних закладів);
- глобальний доступ до ресурсів (можуть надавати доступ до широкого спектру навчальних ресурсів, включаючи словники, електронні підручники та онлайн-курси).

Тести в контексті розширення словникового запасу іноземної мови дають змогу швидко та легко встановити результати тестування, а також використовувати результати для діагностики труднощів у вивченні предмета.

Адаптивне тестування – це широка категорія методів тестування, здатних змінювати порядок завдань у самому процесі тестування, зважаючи на відповіді в попередніх завданнях [21].

Відомо, що значне збільшення кількості завдань призводить до зниження ефективності самого тесту, так як студенти втомлюються в кінці тесту і можуть не правильно виконати завдання, що могло б виконатися правильно, якби тест починався з цього питання. Однак зменшення кількості завдань також не призведе до покращення якості тестування, тому що мала кількість питань не зможе охопити весь матеріал, необхідний для перевірки.

Використання завдань, що відповідають рівню підготовки, значно підвищує точність вимірювання та зменшує час на проведення індивідуальних тестових робіт. У свою чергу, зменшення кількості завдань у тесті та скорочення часу тестування знижує вартість тесту.

Для розширення словникового запасу іноземної мови доцільно вибрати варіант адаптивного тестування Stradaptive (рис. 2.1).

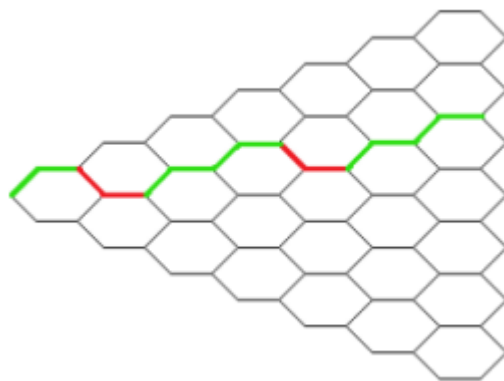


Рисунок 2.1 – Модель тестування Stradaptive

Він пропонує набір тестових завдань різного рівня складності. Якщо відповідь правильна, користувачеві пропонується наступне завдання вищого

рівня складності (зелений шлях), якщо відповідь неправильна – нижчого (червоний шлях).

Метод Раша був розроблений Германом Рашем в 1960-х роках. Він базується на ідеї повторення слів у різних інтервалах, що допомагає їх кращому засвоєнню і запам'ятовуванню [14].

Алгоритм методу включає наступні кроки:

1. Вивчення нових слів. Користувач вивчає нові слова та їх переклади.

2. Повторення через короткий інтервал. Після вивчення нових слів, користувач повторює їх протягом короткого періоду, наприклад, через декілька хвилин після вивчення.

3. Повторення через середній інтервал. Після успішного повторення на короткому інтервалі, слова повторюються через більший інтервал, наприклад, через годину або декілька годин.

4. Повторення через довгий інтервал. Якщо слова правильно повторено на середньому інтервалі, вони повторюються через більший період, наприклад, наступного дня або через декілька днів.

5. Завдання «востаннє перед забуттям». Якщо слова були успішно повторені на довгому інтервалі, їх знову вивчають перед тим, як вони можуть бути забуті.

6. Регулярні повторення: Цей процес повторюється регулярно для всіх вивчених слів.

Цей метод базується на принципах психології пам'яті, зокрема на принципі «кривої забуття». Згідно цього принципу, людина забуває інформацію з часом, але регулярне повторення допомагає підтримувати інформацію в пам'яті. Повторення на різних інтервалах допомагає закріпити інформацію в пам'яті і підвищує ймовірність успішного запам'ятовування слів. Важливо систематично виконувати повторення і регулярно оновлювати список слів для вивчення, додавати нові слова і розширювати свій словниковий запас.

2.2 Розробка математичної моделі розширення словникового запасу іноземної мови у користувача

Ключовими моделями, що можна використати для розробки інформаційної технології для розширення словникового запасу іноземної мови у користувача є модель інтерактивного навчання та модель формування рейтингу та оцінки успішності.

Модель інтерактивного навчання дозволяє боту взаємодіяти з користувачем у формі вправ та тестів, аналізувати їхні відповіді та надавати додатковий матеріал залежно від успішності користувача. Модель рейтингу та оцінки успішності може допомогти боту визначити, наскільки добре користувач розуміє іноземну мову та які аспекти йому слід покращити.

Модель інтерактивного навчання складається з кількох етапів:

1. Визначення рівня користувача.
2. Персоналізований план навчання.
3. Створення набору вправ і тестів.
4. Підвищення рівня складності.
5. Оцінка та корекція.

Для персоналізації навчання було вибрано використання адаптивного контролю знань на основі моделі Раша.

Адаптивний контроль знань використовує теорію IRT (Item Response Theory) у сполученні з дидактичним принципом індивідуального навчання. Основні переваги IRT включають вимірювання параметрів студентів і завдань тесту на одній шкалі. Це дозволяє встановити взаємозв'язок між рівнем знань студента та складністю кожного завдання тесту [22].

Тобто Item Response Theory (IRT) – це математична теорія для параметричної оцінки тестових завдань і студентів.

Для ефективного вивчення нової інформації важливо повторно повертатися до неї. Концепція «кривої забуття» Еббінгауза вказує, що людина забуває близько двох третин матеріалу через день та більше 80% за

тиждень. Для подолання цього ефекту використовується система інтервальних повторень. Слова показуються зростаючими інтервалами в залежності від рівня їх згадки. При високому рівні згадки інтервали вибираються значно рідше, а при низькому рівні – слово показується наступного дня [23].

Частота виведення кожного тесту залежить від розрахунку пріоритету для певного слова, яке приймає значення від 0 до 1. Високий пріоритет вказує на складність вивчення слова користувачем. Пріоритет можна обчислити за формулою:

$$P_i = \frac{F_i}{N_{ti}}, \#(2.1)$$

де P_i – пріоритет іноземного слова;

N_{ti} – кількість тестових завдань, що містять дане слово;

F_i – кількість невиконаних тестових завдань з цим словом [24].

Завершивши визначення пріоритетів, слова сортуються за спаданням пріоритету. Слова з нульовим пріоритетом вважаються вивченими, і їх виключають зі списку. Решта слів додаються до початкового списку, за винятком слова з найнижчим пріоритетом. Таким чином, формується список слів, в якому слова з найвищим пріоритетом відображаються частіше, а слова з найнижчим пріоритетом – лише одноразово.

Умовна ймовірність надання правильної відповіді на j -е завдання з δ_j рівнем важкості різними студентами

$$P_j(\theta_i) = \frac{e^{1.7(\theta_i - \delta_i)}}{1 + e^{1.7(\theta_i - \delta_i)}}, \#(2.2)$$

де θ_i – латентний параметр, що визначає рівень знань студента, $i=1, \dots, n$;

δ_i – латентний параметр, який відповідає рівню складності завдань тесту, $j=1, \dots, m$

Розглянута формула є однопараметричною моделью Дж. Раша.

Ця модель може бути удосконалена за допомогою двопараметричної моделі А. Бірнбаума:

$$P_j(\theta_i) = \frac{e^{1.7a_j(\theta_i - \delta_i)}}{1 + e^{1.7a_j(\theta_i - \delta_i)}}, \#(2.3)$$

А також трипараметричної моделі А. Бірнбаума:

$$P_j(\theta_i) = c_j + (1 - c_j) \frac{e^{1.7a_j(\theta_i - \delta_i)}}{1 + e^{1.7a_j(\theta_i - \delta_i)}}, \#(2.4)$$

де a_j – який вказує на диференційовану здатність завдання при зміні різних значень параметрів θ та δ ;

c_j – імовірність того, що користувач вгадав правильну відповідь на j -те завдання.

У системі адаптивного контролю знань вирішено використовувати систему ланцюга запитань. В цьому випадку, кілька запитань об'єднуються певною послідовністю, яку визначає вчитель. Кожне наступне запитання в отриманій послідовності отримує певний коефіцієнт важливості в ланцюжку, при цьому сума цих коефіцієнтів дорівнює 1. Важливість кожного запитання в послідовності визначається при обробці результатів тесту, де оцінка відповідей групується залежно від важливості питань, на які отримано правильні відповіді [25].

Відносною важливістю питань розраховують за їх вагою, що визначається при підведенні результатів тесту. Кожне запитання послідовності ставиться тільки після отримання відповіді на попереднє запитання.

Кожне j -те питання в i -му ланцюжку отримує ваговий коефіцієнт K_{ij} , що вказує на його важливість в рамках цього ланцюжка. Сума всіх

коефіцієнтів в кожній послідовності повинна бути одиницею. Послідовність може об'єднувати необмежену кількість тематично близьких запитань.

Процедура визначення кількісної оцінки знань, отриманих в результаті тестування, містить три етапи. На першому етапі відбувається розрахунок балів за кожне окреме тестове завдання, представлене у послідовності тесту. Кожне завдання оцінюється відповідно до відповідей студента:

$$S_i = \sum_{j=1}^{L_i} K_{ij} Z_{ij}, \#(2.5)$$

де S_i – це бал, поставлений за вірні відповіді на i -у тестову послідовність;

K_{ij} – ваговий коефіцієнт j -го запитання i -ї послідовності;

$Z_{ij} = 1$, якщо на j -е питання в i -ій послідовності отримана правильна відповідь та $Z_{ij} = 0$ – у протилежному випадку;

L_i – число запитань i -ої послідовності.

Другий етап передбачає обчислення сумарного балу S_Σ за відповіді на всі тестові завдання, включаючи усі послідовності тестових питань, на які користувач встиг відповісти за певний час. Це враховується в формулі:

$$S_\Sigma = \sum_{i=1}^N S_i * \frac{K_t}{N^2}, \#(2.6)$$

де N – кількість тестових завдань;

K_t – кількість послідовностей тестових завдань, на які користувач встиг відповісти за певний час t .

На останньому етапі визначається підсумкова оцінка знань студента, коли сумарний бал S_Σ проектується на шкалу оцінювання. Це робиться за допомогою шкали $[0; I_1; I_2; I_3; 1]$, де $0 < I_1 < I_2 < I_3$ – границі діапазонів оцінок, що задає вчитель для організації тестів.

Ці етапи спроектовані для використання у системі адаптивного контролю знань, де процес тестування динамічно адаптується до відповідей студента, надаючи більше або менше важливості певним тестовим завданням залежно від його результатів.

Використовуючи дану модель тестування, яка базується на процедурі вибору та подання користувачу наступного тестового завдання на кроці $(n+1)$, з урахуванням попередніх відповідей, можна запропонувати наступну математичну модель.

Нехай $U = \{u_j\}$, $j = \overline{1:N_{max}}$ – послідовність питань тесту, необхідна для формування тестових завдань, а $U_T = \{u_i\}$, $U_T \subseteq U$, $i = \overline{1:N}$ – питання тесту, здатні оцінити рівень володіння мовою конкретного користувача, до того ж $N \ll N_{max}$.

Нормована сума балів S_t може бути визначена відповідно до кількості балів підсумкової оцінки знань на поставлені питання конкретного студента, за відповіді на пройдених кроках тесту, де $n \leq N$, а $0 \leq S_t \leq 1$. Шкала підсумкової оцінки знань Z_I може бути представлена у спрощеній формі як $Z_I = [0; I_1/O_1; I_2/O_2; \dots; I_n/O_n; 1/O_{n+1}]$, де $0 < I_1 < I_2 < \dots < I_n < 1$ – границі інтервалів оцінювання; O_i , $i = \overline{1:n+1}$ – бал, яким описують знання користувача, який проходить тестування, при накопиченні ним суми балів S_t , що потрапить в інтервал $[I_{i-1}, I_i]$.

У моделі враховується інтервалу невизначеності $\pm \Delta I > 0$, що має істинне значення $S_t : S_t \in [S_t - \Delta I; S_t + \Delta I]$ з імовірністю $P_t < 1$ при виведенні остаточної оцінки знань, показуючи результати $t \leq N$ кроків проходження тесту. З практичної точки зору це означає, що при проектуванні значення S_t на шкалу Z_I необхідно враховувати не лише окремий сегмент цієї шкали $[I_i, I_{i+1}]$, який включає значення S_t , але також його зонування, враховуючи інтервал невизначеності ΔI [26].

У випадку, коли сума балів опиняється в інтервалі $I_i + \Delta I \Delta S_N < I_{i+1} - \Delta I$ сума балів S_N абсолютно не здатна потрапити до зони невизначеності $(I_i \pm \Delta I)$ і $(I_{i+1} \pm \Delta I)$. Дана ситуація має досить підстав для

того, щоб поставити оцінку O_{i+1} безпосередньо після завершення першого етапу тестування. Таким чином, подальше введення додаткових або уточнюючих питань не видається доцільним.

Цю модель слід впроваджувати для оцінки розширення словникового запасу іноземної мови у користувача.

2.3 Проектування структури інформаційної технології для розширення словникового запасу іноземної мови у користувача

Здійснення автоматизованого контролю знань користувачів – це процес використання технологій та інформаційних систем для оцінки та вимірювання рівня знань та навичок користувачів в певній області або предметі. Він включає рішення проблеми визначення сукупності необхідних якостей знань, без яких критерії оцінки знань та способи визначення рівня їх засвоєння виявити не можна [27].

Для здійснення автоматизованого контролю знань необхідно створити набір тестів, запитань або завдань, які відповідають за розширення словникового запасу користувача. Ці тести можуть бути у різних форматах, включаючи питання з вибором однієї або кількох відповідей, завдання з короткими відповідями, відкриті питання.

Авторизовані користувачі проходять тести або виконуватимуть завдання відповідно до визначених правил та умов. Результати тестування автоматично зберігаються в системі. Після проходження тесту користувачеві надається оцінка або зворотній зв'язок щодо його відповідей. Система також може аналізувати результати тестів для визначення рівня знань та навичок користувача. Здійснення автоматизованого контролю знань дозволяє збирати важливі дані та статистику, які можна використовувати для аналізу та вдосконалення процесу навчання. Тести адаптуються до рівня та потреб кожного користувача, надаючи індивідуальний підхід до контролю знань.

Отримана інформація дозволяє вчителям або викладачам надавати зворотній зв'язок користувачам та розробляти стратегії для покращення навчання.

Структуру програмного забезпечення інформаційної технології для розширення словникового запасу іноземної мови у користувача наведено на рисунку 2.2.



Рисунок 2.2 – Структура інформаційної технології для розширення словникового запасу іноземної мови у користувача

Модульний підхід до проектування інформаційної технології для розширення словникового запасу іноземної мови у користувача передбачає поділ системи на окремі функціональні модулі, які виконують конкретні завдання та функції. Цей підхід дозволяє розробляти систему більш гнучкою,

масштабованою та піддається змінам. Система включатиме в себе кілька ключових модулів для ефективного функціонування. Розроблена структурна схема функціонування інформаційної технології зображена на рисунку 2.3.

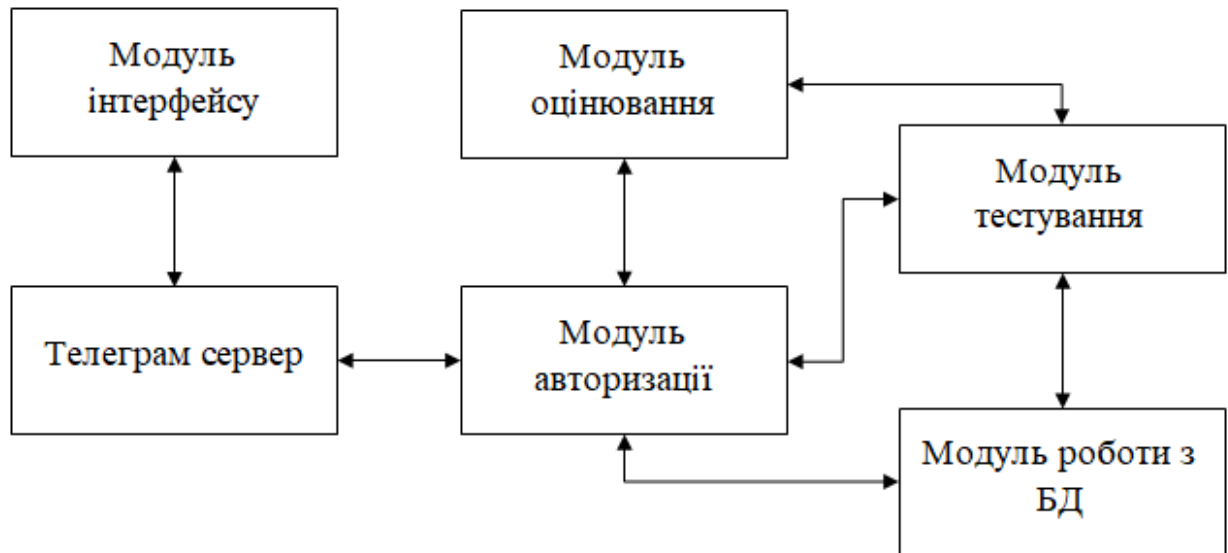


Рисунок 2.3 – Структурна схема функціонування інформаційної технології

Модуль інтерфейсу взаємодіятиме з користувачем через телеграм сервер, надаючи зручне вікно чату для попередніх бесід та поле для введення інформації. Для взаємодії з системою передбачено кнопки з пропонованими варіантами.

Модуль авторизації передбачає вхід користувача до створеного профіля, що зберігає його особисті дані та індивідуальні потреб. Модуль авторизації пов'язаний з усіма іншими модулями, адже залежно від авторизації користувача залежить його подальша робота, доступ до якої він має. На цьому етапі відбувається перевірка користувача на доступ до ролі адміністратора.

Модуль тестування є ключовим елементом, який співпрацює із модулями оцінювання та роботи з базою даних. Він відповідає за вибір тестів для кожного користувача з використанням адаптивної моделі контролю

знань. Система, аналізуючи відповіді, розробляє індивідуальні плани для подальшого навчання. Модуль оцінювання визначає оцінку знань користувача, доступ до якої має адміністратор системи.

Модуль роботи з базою даних взаємодіє з модулем тестування та забезпечує зв'язок із базою даних. В базі даних містяться завдання для розширення словникового запасу, створені вчителями. Цей модуль включає запити на мові SQL для перегляду, видалення та додавання завдань. В базі даних міститься весь навчальний контент: слова, фрази, вправи. Також в ній зберігається інформація про користувачів, їхні логіни та паролі, а також результати проходження ними тестування.

Отже, запустивши додаток система перевіряє ідентифікаційні дані користувача. Якщо користувач адміністратор, відкриваються додаткові можливості. Далі запускається модуль інтерфейсу через телеграм сервер. Користувач має доступ до вікна чату для введення інформації та кнопок для взаємодії. Модуль тестування визначає потреби користувача та вибирає тести. Користувач отримує запитання, відповідає, і система аналізує результати. Модуль оцінювання використовує інтелектуальний алгоритм для визначення рівня знань користувача. Результати зберігаються в базі даних. Модуль роботи з базою даних взаємодіє з базою для отримання тестових завдань та збереження результатів.

Інформаційна технологія розширення словникового запасу іноземної мови у користувача матиме два види користувачів: адміністратор і користувач.

Діаграма варіантів використання наведена на рисунку 2.4.

Користувач матиме можливість виконувати наступні дії:

- вибір режиму тестування;
- перегляд варіантів відповідей;
- проходження тестування.

Модуль, доступний для адміністратора, буде обмежений етапом авторизації, після чого йому будуть надані додаткові можливості:

- перегляд статистики;
- додавання слів.

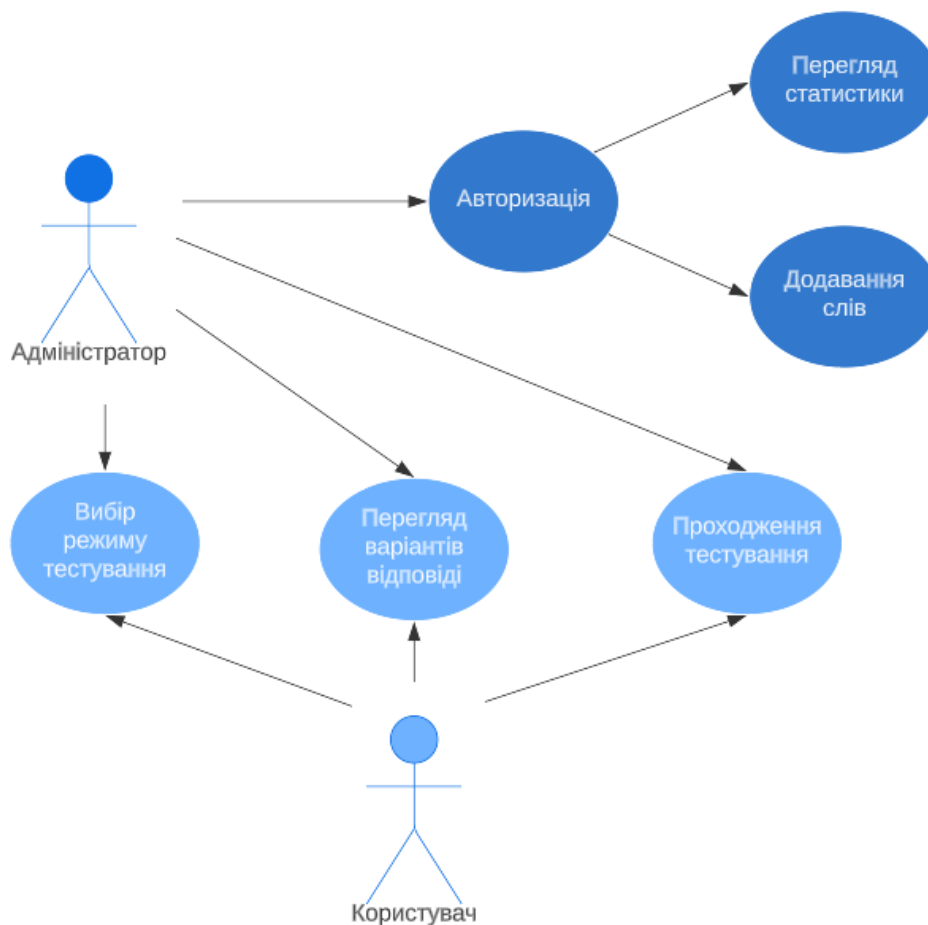


Рисунок 2.4 – Діаграма варіантів використання

Отже, визначено види користувачів інформаційної технології, на основі чого було створено діаграму варіантів використання інформаційної технології.

2.4 Розробка алгоритму функціонування інформаційної технології для розширення словникового запасу іноземної мови у користувача

Алгоритм інформаційної технології для розширення словникового запасу іноземної мови у користувача можна поділити на загальний алгоритм функціонування модуля та алгоритм функціонування модуля тестування користувача.

Графічне представлення загального алгоритму функціонування інформаційної технології для розширення словникового запасу іноземної мови у користувача представлено на рис. 2.5.

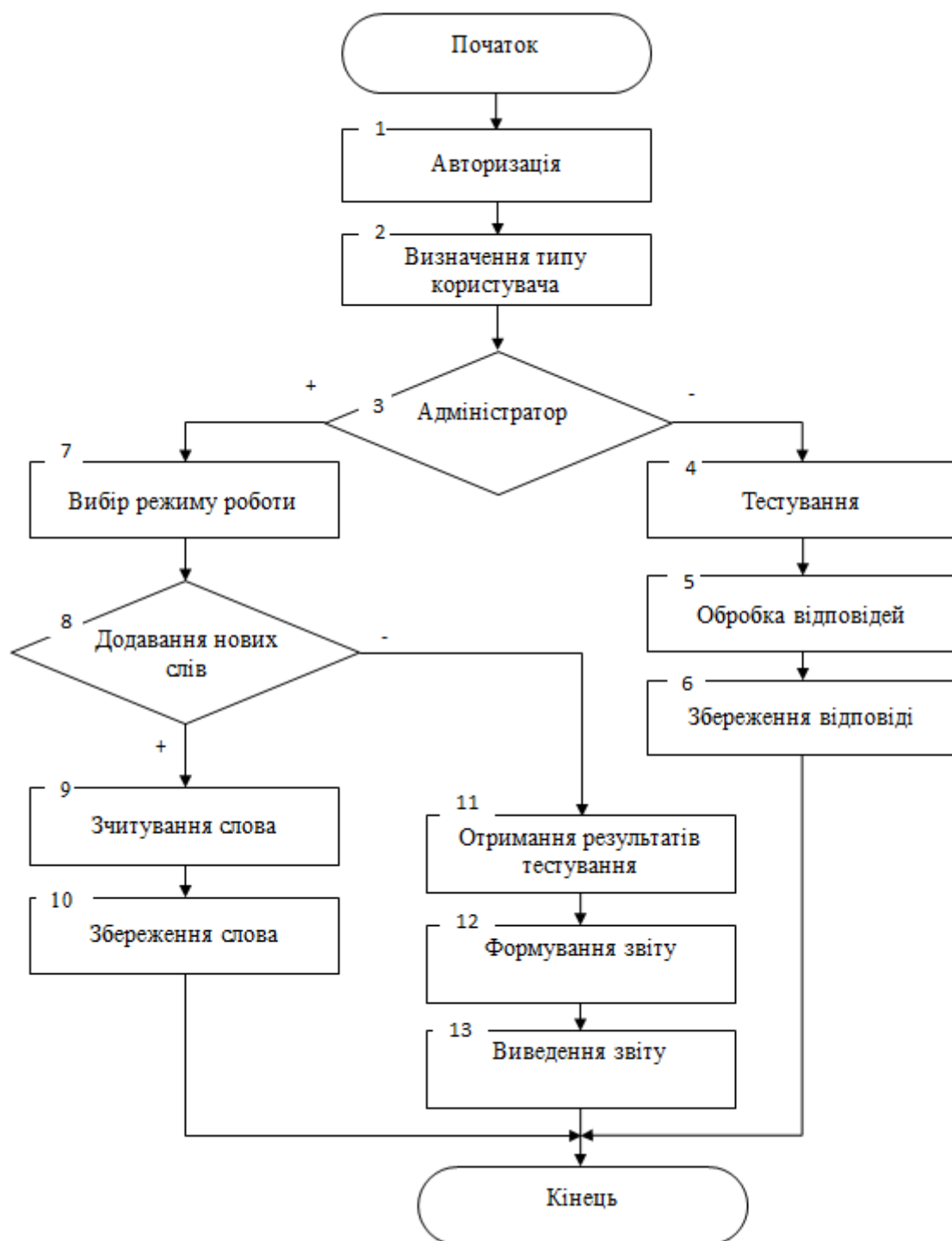


Рисунок 2.5 – Схема алгоритму функціонування інформаційної технології для розширення словникового запасу іноземної мови у користувача

Даний алгоритм складається з наступних етапів:

1. Початок роботи з програмою. Користувач авторизується, вводячи свій номер телефону.
2. Визначення типу користувача: користувачі можуть бути двох типів – користувач та адміністратор.
3. Перевірка прав адміністратора. Якщо користувач має права адміністратора – перехід до пункту 7.
4. Починається тестування користувача.
5. Після тестування програма обробляє отримані відповіді.
6. Відповіді зберігаються програмою. Перехід до пункту 14.
7. Адміністратор обирає режим роботи серед двох варіантів: додавання нових слів або отримання результатів проходження тестування учасниками.
8. Перевірка на режим додавання нових слів. Якщо вибраний режим отримання результатів тестування – перехід до пункту 11.
9. Зчитування нового слова програмою.
10. Нове слово зберігається до бази даних. Перехід до пункту 14.
11. Отримання результатів тестування адміністратором.
12. Формується звіт з результатами тестування.
13. Звіт виводиться на екран.
14. Діалог з програмою завершується.

Наведений алгоритм є комплексною системою взаємодії користувача з програмою для розширення словникового запасу іноземної мови. Він дозволяє ефективно використовувати програму для вивчення мови, надаючи можливості тестування, редагування словникового запасу та отримання звітів про прогрес користувачів. Алгоритм є структурованим та логічно побудованим, сприяючи зручності взаємодії з програмою для користувачів та адміністраторів.

Графічне представлення алгоритму функціонування модуля тестування для розширення словникового запасу іноземної мови у користувача зображено на рисунку 2.6.

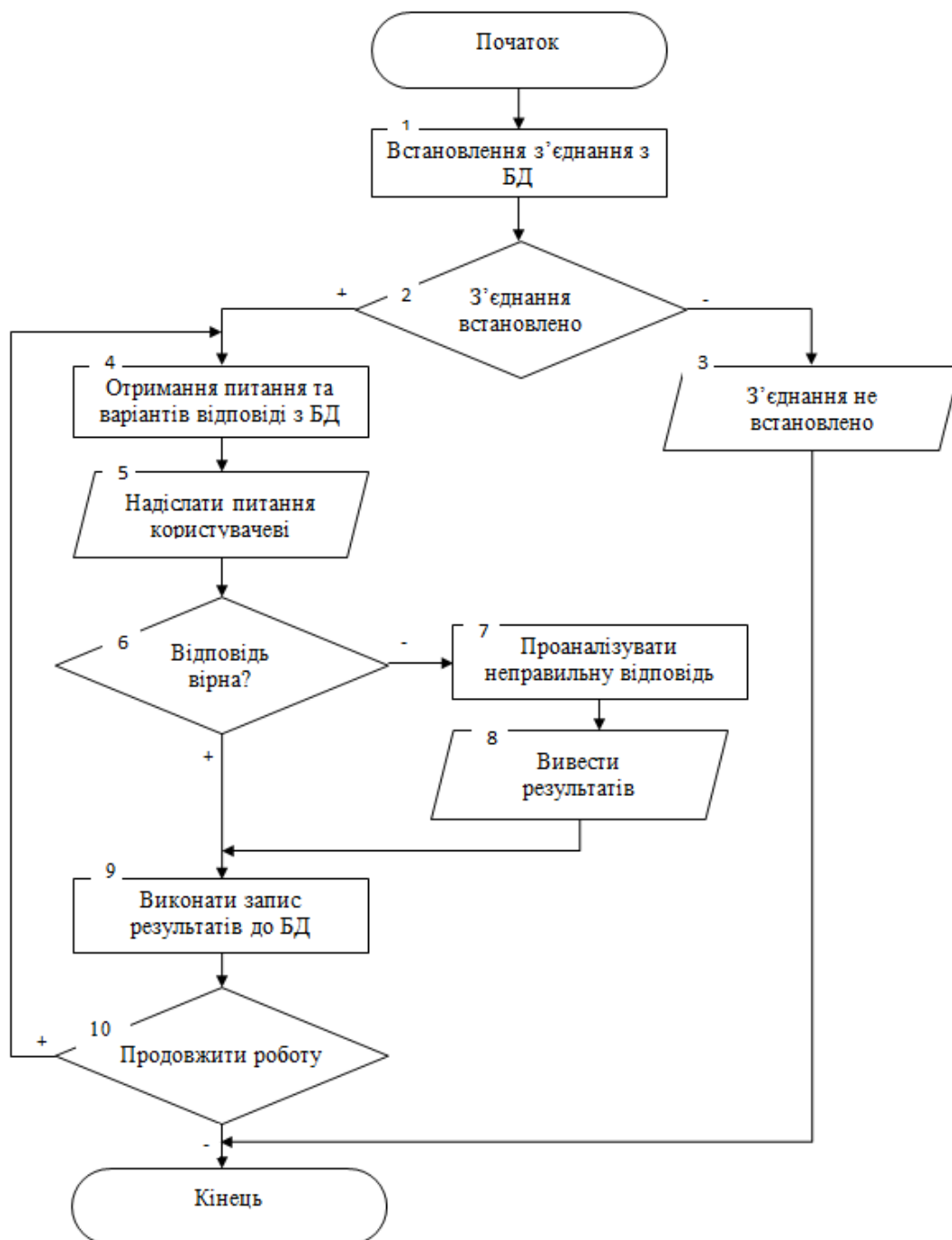


Рисунок 2.6 – Схема алгоритму функціонування модуля тестування користувача

Наведений алгоритм визначає процес тестування знань користувача з іноземної мови. Він включає наступні кроки:

1. Початок взаємодії програми, встановлення з'єднання з Базою даних.
2. Перевірка успішного встановлення з'єднання. Якщо з'єднання встановлено, переходимо до пункту 4.
3. Якщо немає з'єднання, завершення роботи програми.
4. Отримання питань та варіантів відповідей з бази даних.
5. Надсилання цих питань та варіантів відповідей користувачеві для перевірки його знань.
6. Перевірка правильності отриманої відповіді. Якщо відповідь вірна, перехід до пункту 9.
7. Аналіз неправильної відповіді для подальшого вивчення системою.
8. Виведення результату щодо неправильного вибору варіанту відповіді.
9. Запис результатів у базу даних.
10. Якщо користувач бажає продовжити роботу, повторення процесу з етапу отримання питань. У випадку небажання продовжувати роботу – закінчення взаємодії з системою.

Наведений алгоритм представляє собою ефективний процес тестування користувача з іноземної мови. Він визначає послідовність кроків, що охоплюють взаємодію з базою даних, подання тестових завдань, обробку відповідей та запис результатів. Цей алгоритм розроблено для забезпечення ефективного навчання та визначення рівня знань користувача з іноземної мови з використанням адаптивного тестування.

2.5 Розробка бази даних для інформаційної технології для розширення словникового запасу іноземної мови у користувача

Основне завдання бази даних полягає в забезпеченні зберігання значних обсягів інформації та забезпечення доступу до цієї інформації

користувачів або програмних застосунків. Основне призначення бази даних полягає в об'єднанні двох ключових функцій: ефективного зберігання даних та ефективного управління цими даними [28]. База даних допомагає зберігати, організувати та забезпечувати доступ до різних видів інформації, яка включає слова, вирази, вправи, тести, статистику користувачів тощо.

Для опису предметної області необхідні наступні сутності: студент, адміністратор, кеш, слова, неправильні відповіді. Представлення ER-діаграми для бази даних зображено на рисунку 2.7.

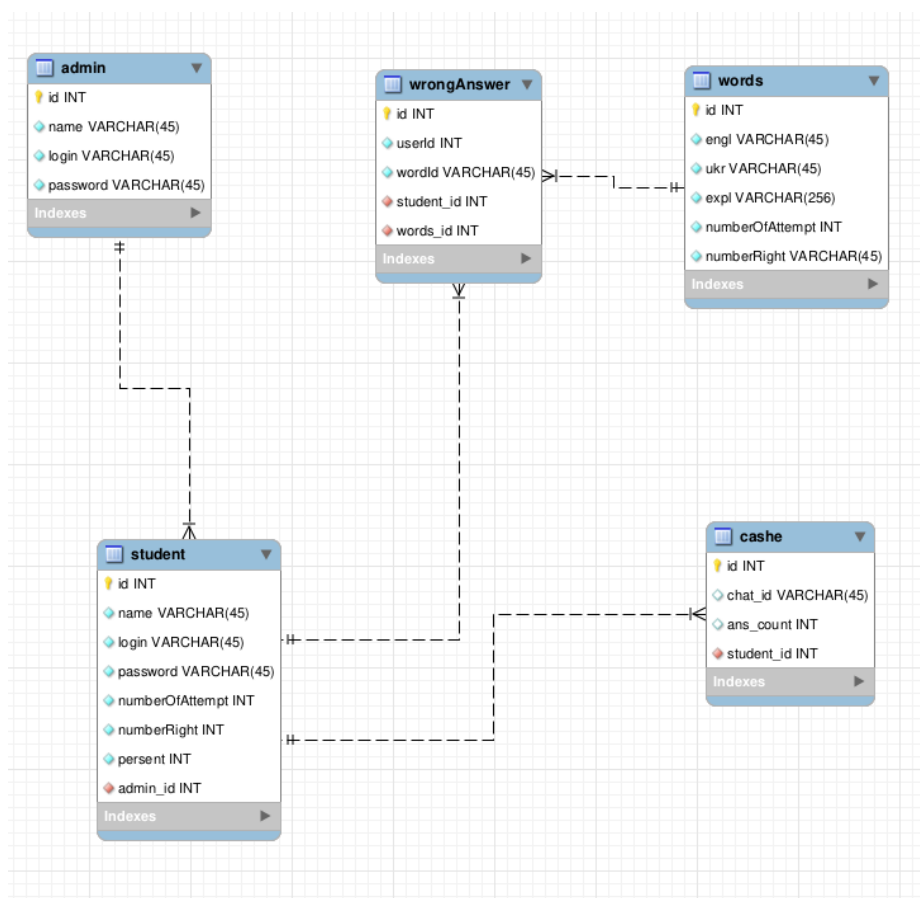


Рисунок 2.7 – ER-діаграма для бази даних розширення словникового запасу

Для зберігання словникового запасу іноземної мови потрібно мати ефективну структуру даних. Реляційні бази даних, такі як MySQL або PostgreSQL, можуть бути корисними, якщо існує необхідність зберігати слова, їх переклади, вимову, приклади вживання тощо в табличному форматі. При розробці інформаційної технології для розширення словникового запасу

іноземної мови необхідно зберегти інформацію, що міститься в програмі, і програмно реалізується. Тож для цього необхідно використати інструмент для роботи з базами даних MySQL Workbench.

Реляційна модель даних – це така модель подання даних, у якій уся інформація, яка може бути цифровою чи текстовою, подається за допомогою таблиць [29]. Кожна з таблиць, які називаються відношеннями, складається з кортежів (рядки таблиці) та атрибутів (стовпці таблиці). Реляційна модель визначає структуру (представлення даних), цілісність даних (захист від некоректних змін) та операції з даними.

Дані в реляційній моделі зберігаються у вигляді таблиць. Кожна таблиця містить записи, які представляють конкретний тип даних, такі як користувачі, слова або тести. Рядки таблиці представляють окремі записи, а стовпці визначають атрибути цих записів. Наприклад, таблиця «Користувачі» може містити рядки для кожного користувача зі стовпцями, такими як «ім'я», «прізвище», «номер» тощо. В реляційній моделі ключі використовуються для ідентифікації унікальних записів у таблицях. Головний ключ (Primary Key) визначає унікальний ідентифікатор для кожного запису, і зовнішні ключі (Foreign Key) використовуються для встановлення зв'язків між таблицями.

Реляційна модель підтримує нормалізацію даних, яка допомагає уникнути аномалій та забезпечує консистентність даних. Нормалізація включає розбиття даних на більше малих таблиць та визначення зв'язків між ними. Дані в реляційній моделі є атомарними, що означає, що вони не можуть бути розбиті на менші фрагменти. В реляційній моделі дані можна отримувати за допомогою структурованих запитів, таких як SQL (Structured Query Language). Запити дозволяють витягти, модифікувати, видаляти або вставляти дані у таблиці, використовуючи різноманітні умови та операції. Дана модель підтримує цілісність даних, включаючи перевірку даних на правильність, обмеження і контроль доступу до даних.

Реляційна модель надає спрощений та легко зрозумілий спосіб для організації даних, що полегшує розробку та адміністрування баз даних. Вона

є основою для багатьох сучасних систем управління базами даних (СУБД) і широко використовується в бізнесі, науці, технологіях та інших галузях для організації та збереження даних.

Представити базу даних необхідно використовуючи реляційну модель даних, що має переваги простоти для розуміння та використання, що робить її популярною серед розробників і адміністраторів баз даних; єдина використовувана інформаційна структура – «таблиця»; уникнення дублювання даних і підтримка консистентності; суворі правила проектування, засновані на математичному апараті; повна незалежність даних; може бути легко розширена та підтримує високий рівень надійності; можливість розширення бази даних шляхом додавання нових елементів, записів без зміни існуючих підсхем і додатків; оптимізація для виконання запитів, що забезпечує швидкий доступ до даних [30].

SQL є стандартною мовою запитів для реляційних баз даних, таких як MySQL, PostgreSQL, Oracle, Microsoft SQL Server і багато інших [31]. Вона використовується для створення, зчитування, оновлення та видалення даних. SQL дозволяє виконувати запити для вибору певних даних з бази, а також створювати таблиці та зв'язки між ними. Мова SQL має наступні переваги для розробки інформаційної технології розширення словникового запасу іноземної мови у користувача:

- SQL побудована на реляційній моделі даних, яка добре підходить для зберігання інформації про слова, їх переклади, вимову, синоніми та інші лінгвістичні дані.
- SQL надає потужну мову запитів для отримання даних з бази. Є змога швидко і просто вибирати слова за певними критеріями, виконувати запити на пошук подібних слів або отримувати дані, впорядковані за алфавітом.
- Реляційні бази даних, які використовують SQL, зазвичай оптимізовані для швидкого пошуку і доступу до даних. Це особливо важливо, якщо інформаційна технологія повинна надавати миттєві відповіді на запити користувачів.

- Реляційні бази даних зазвичай надають засоби для забезпечення цілісності даних і відмовостійкості. Це важливо, коли мова йде про зберігання цінної інформації про словниковий запас іноземної мови.

- SQL бази даних можуть використовувати індекси для швидкого пошуку даних. Це робить можливим ефективний доступ до даних, навіть коли база даних містить великий обсяг інформації.

SQL має свої недоліки, і до них відносяться деякі обмеження. Наприклад, при створенні запитів за допомогою цієї мови необхідно чітко визначити структуру даних. Важливо враховувати, що зміни в структурі даних можуть мати негативний вплив на всю систему.

2.6 Висновок до розділу 2

У другому розділі було обґрунтовано вибір методики розширення словникового запасу іноземної мови у користувача. Метод адаптивного навчання характеризується найбільшою ефективністю для даної задачі, тому його доцільно використовувати для створення математичної моделі розширення словникового запасу іноземної мови у користувача, що забезпечить підвищення ефективності запам'ятовування іноземних слів. У даному розділі було спроектовано структуру інформаційної технології, розроблено загальну структурну схему, наведено діаграму варіантів використання інформаційної технології. Розроблено загальний алгоритм функціонування інформаційної технології та алгоритм тестування користувача. Також розроблено базу даних для інформаційної технології для розширення словникового запасу у користувача та вибрано сутності для опису предметної області, наведено ER-діаграму бази даних для розширення словникового запасу у користувача.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ДЛЯ РОЗШИРЕННЯ СЛОВНИКОВОГО ЗАПАСУ ІНОЗЕМНОЇ МОВИ У КОРИСТУВАЧА

3.1 Обґрунтування вибору мови програмування та середовища розробки

Процес вибору мови програмування та середовища розробки є критичним у розробці будь-якого програмного продукту, і він визначає не тільки зручність розробки, але й ефективність та успішність реалізації проекту в цілому. Вирішення цього питання вимагає комплексного підходу, враховуючи різні аспекти, такі як функціональність мови програмування, інструменти для розробки, швидкодія та масштабованість, вимоги до безпеки та цілісності даних, а також специфіку завдань, які стоять перед інформаційною технологією для розширення словникового запасу іноземної мови. Вирішуючи це питання, необхідно врахувати різні аспекти, а також провести дослідження та порівняння можливих варіантів. Обрані мова програмування та середовище розробки мають бути оптимальними для досягнення цілей та завдань цього проекту. На сьогодні найбільш актуальними мовами для вирішення поставленої задачі є C# і Java.

Об'єктно-орієнтована мова програмування C#, створена між 1998 та 2001 роками, здобула популярність для розробки різних видів програм, включаючи веб-додатки, десктоп-додатки, мобільні застосунки та багато інших [32]. C# – це мова зі строгою типізацією, де змінні мають статично визначений тип даних, що сприяє виявленню помилок на етапі компіляції. Вона підтримує багатопотоковість, розширення методів та делегатів, багатофункціональність; має автономатичне управління пам'яттю. C# має синтаксис, який схожий до C++ та Java, сприяючи легкому переходу розробникам з досвідом в цих мовах до C#. C# використовується для

розробки широкого спектру програм, і його популярність постійно зростає завдяки широкому спектру можливостей та підтримці з боку Microsoft.

Перевагами C# для розробки інформаційної технології для розширення словникового запасу іноземної мови у користувача є:

- використання проміжного коду, що дозволяє виконуватися на будь-якому комп'ютері, незалежно від апаратного та системного програмного забезпечення. Багато коду на C# може бути перенесено між різними платформами завдяки підтримці .NET Core

- повна об'єктно-орієнтованість, що спрощує створення зрозумілого та підтримуваного коду, завдяки використанню об'єктів та класів;

- можливість взаємодії з іншими мовами програмування та бібліотеками, роблячи її гнучкою для розширення та використання сторонніх ресурсів.

- висока масштабованість та можливість модифікації програм для розширення функціональності.

Однак варто враховувати недоліки, такі як орієнтованість на системи Windows, обмежені можливості на інших операційних системах, залежність від .NET і відсутність деяких низькорівневих можливостей [33].

Мова програмування Java є іншим важливим варіантом для розробки інформаційної технології для розширення словникового запасу іноземної мови у користувача. Java володіє своїми перевагами та особливостями, які роблять її привабливою для таких проектів.

Java – це об'єктно-орієнтована мова програмування, розроблена компанією Sun Microsystems (пізніше придбана Oracle). Розроблена на основі C++, вона мала враховувати всі недоліки попередників, характеризуючись такими особливостями, як відсутність покажчиків та адресної арифметики, в результаті чого мала б істотно підвищитися живучість нового середовища; усунення перевантаження, множинного спадкування та автоматичного узгодження типів даних, які вважалися джерелом проблем і помилок в

програмах; виділення об'єктів як єдиних виразних одиниць мови; інтерпретованість мови; близький до C++ синтаксис.

Програми на основі Java зазвичай скомпільовані в спеціальний байт-код, який дозволяє їм працювати на будь-якій віртуальній машині Java (JVM), незалежно від архітектури комп'ютера. Перевагою цього методу компіляції програми є повна незалежність від операційної системи та апаратного байт-коду, що дозволяє запускати Java-додатки на будь-якому пристрої, який має відповідну віртуальну машину [34].

Іншою ключовою особливістю технології Java є її гнучка система безпеки, яка дозволяє запускати програму, що повністю керується віртуальною машиною. Будь-які операції, що виходять за межі встановленого дозволу програми, автоматично припиняються. Також важливо зазначити, що Java використовує механізм збору сміття для ефективного керування пам'яттю протягом життєвого циклу об'єкта. У цій технології розробник визначає момент створення об'єктів, а віртуальна машина відповідає за вивільнення пам'яті, коли об'єкт більше не потрібен. По суті, коли ви втрачаєте інтерес до конкретного об'єкта, механізм збору сміття може автоматично видалити його з пам'яті. Ця операція може бути викликана в будь-який момент, і в ідеалі вона відбувається в періоди неактивності програми. Також важливо відзначити, що Java не підтримує використання покажчиків у стилі C/C++. Це рішення прийнято з міркувань безпеки та надійності, щоб уникнути проблем, які можуть виникнути від використання покажчиків, і для того, щоб механізм дампу міг безпечно переміщувати покажчики в пам'яті..

Отже, Java має переваги у вигляді платформонезалежності, безпеки, великої спільноти та багатьох ресурсів, зручності розробки, широкого вибору бібліотек і фреймворків, об'єктно-орієнтованого підходу, високої продуктивності, автономного управління пам'яттю, можливості багатопоточності, легкості для навчання та використання. Ці переваги

роблять Java дуже потужним інструментом для розробки інформаційної технології для розширення словникового запасу іноземної мови користувача.

Беручи до уваги всі вищенаведені переваги та недоліки, зазначимо, що мова програмування Java є найбільш доцільною для реалізації інформаційної технології для розширення словникового запасу у користувача. Розробка програмного додатку на Java буде здійснюватися за допомогою інтегрованого середовища розробки IntelliJ IDEA 2021.

IntelliJ IDEA – це інтелектуальна, контекстно-залежна IDE призначена в основному для роботи з Java, але також підтримує різні інші мови програмування та технології. Крім того, IntelliJ IDEA Ultimate може допомогти розробляти повноцінні веб-додатки завдяки своїм потужним інтегрованим інструментам, підтримці JavaScript і пов'язаних технологій, а також розширеній підтримці популярних фреймворків, таких як Spring, Spring Boot, Jakarta EE, Micronaut, Quarkus, Helidon. IntelliJ IDEA високо цінується за свої функції та можливості, що робить його одним із найпопулярніших варіантів серед розробників [35].

Інтерфейс середовища розробки IntelliJ IDEA добре організований і інтуїтивно зрозумілий, надаючи розробникам доступ до різних інструментів та опцій. Він є досить налаштовуваним, і розробники можуть адаптувати його до своїх потреб та робочого процесу. Кожен аспект IntelliJ IDEA створений для забезпечення безперебійного досвіду роботи out-of-the-box. На рисунку 3.1 наведено інтерфейс середовища розробки IntelliJ IDEA.

Однією з найкращих переваг IntelliJ IDEA є можливість налаштування. Користувач можете налаштувати практично все: зовнішній вигляд IDE, розташування вікон інструментів і панелей інструментів, підсвічування коду тощо. Є також багато способів, за допомогою яких можна налаштувати редактор і налаштувати його поведінку, щоб пришвидшити навігацію та позбутися будь-яких додаткових функцій, які відволікають від коду. Існує можливість налаштувати кольори та шрифти для вихідного коду, виводу консолі, інформації налагоджувача, результатів пошуку тощо. Можна

вибрати з кількох попередньо визначених колірних схем або налаштувати схему, щоб створити унікальне робоче середовище.

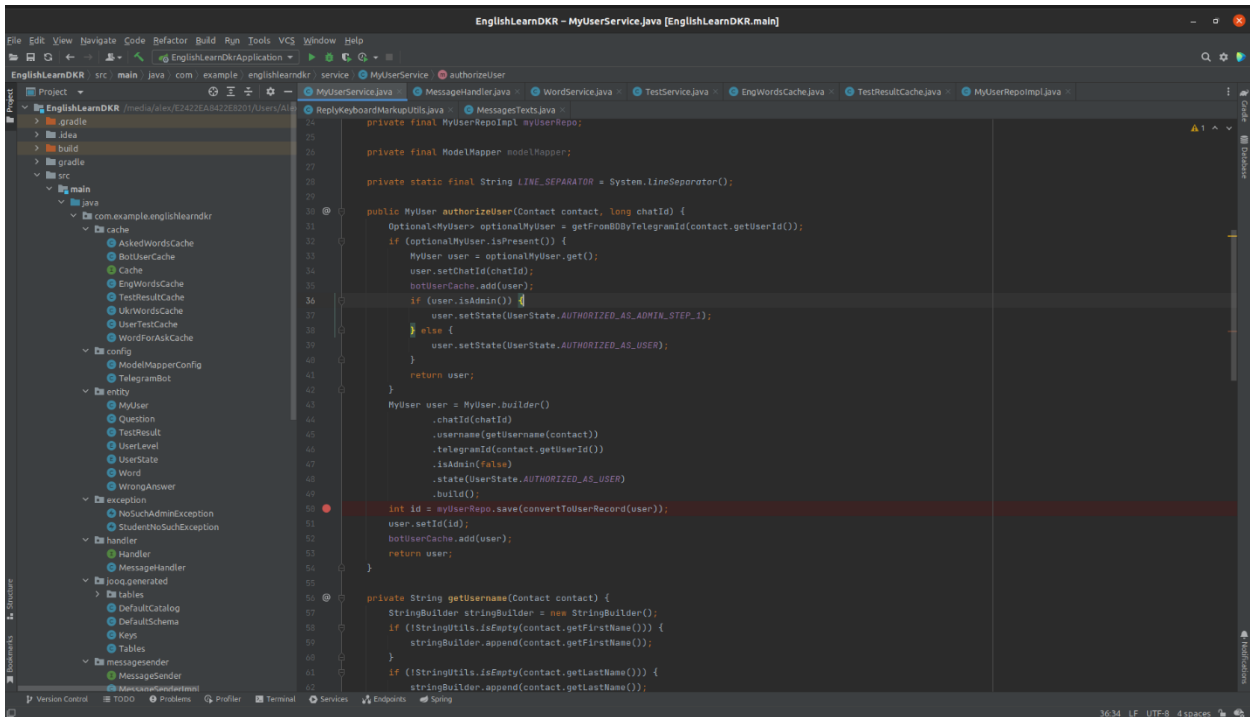


Рисунок 3.1 – Інтерфейс середовища розробки IntelliJ IDEA

У IntelliJ IDEA є ярлики майже для кожної дії, включаючи вибір і перемикач між редактором і різними вікнами інструментів. Одним із універсальних ярликів є подвійний Shift (Шукати всюди). Це дозволяє знайти практично будь-що всередині або поза проектом, варто лише почати вводити текст, і IntelliJ IDEA шукатиме пошуковий рядок серед усіх файлів, класів і символів, які належать до проекту, і навіть серед дій IDE. Також є можливість налаштувати розкладку клавіш, щоб призначити власні комбінації клавіш для улюблених команд.

IntelliJ IDEA пропонує повний набір автоматизованих рефакторингів коду, які призводять до значного підвищення продуктивності. Наприклад, перейменовуючи клас, IDE оновить усі посилання на цей клас у проекті. Навіть немає потреби нічого вибирати, перш ніж застосувати рефакторинг. IntelliJ IDEA достатньо розумна, щоб визначити оператор, який користува

збирається рефакторити. Він запитує підтвердження, лише якщо є кілька можливих варіантів.

IntelliJ IDEA надає набір перевірок, які є вбудованими інструментами статичного аналізу коду. Вони допомагають знаходити потенційні помилки, знаходити мертвий код, виявляти проблеми з продуктивністю та покращувати загальну структуру коду. Перевірки не тільки повідомлять, де є проблема, але й пропонують швидкі рішення, які допоможуть негайно впоратися з нею [36]. Крім швидких виправлень, IntelliJ IDEA також забезпечує дії намірів, які допомагають застосувати автоматичні зміни до правильного коду.

IntelliJ IDEA надає кілька способів створення загальних конструкцій коду та повторюваних елементів, що допомагає підвищити продуктивність шляхом делегування рутинних завдань IDE. Це включає генерацію коду з попередньо визначених або користувальницьких шаблонів коду, генерацію обгортки, геттерів і сеттерів, автоматичне поєднання символів тощо. IntelliJ IDEA постачається з потужним набором інструментів для налаштування параметрів запуску та складання для вашої програми та коду налагодження, а також для застосування та розробки тестів JUnit прямо в IDE.

IntelliJ IDEA має інтеграцію з популярними системами контролю версій, такими як Git, SVN і Mercurial, полегшує роботу з кодом в команді. Можна клонувати проект із VCS прямо на екрані привітання, вивчати відмінності між двома версіями, керувати гілками, фіксувати та надсилати зміни, об'єднувати конфлікти, переглядати історію тощо.

Загалом, IntelliJ IDEA відома своєю продуктивністю, розширюваністю та зручністю в роботі. Ця IDE є важливим інструментом для розробників, які працюють з Java і іншими мовами програмування.

Отже, інформаційна технологія розроблена мовою програмування Java в середовищі розробки IntelliJ IDEA, що має переваги у вигляді глибокого розуміння коду, задоволенні усіх вимог розробника, розумного доповнення коду.

3.2 Обґрунтування вибору месенджера Telegram у якості платформи для взаємодії з користувачем

Було вирішено створити інформаційну технологію для розширення словникового запасу користувача у вигляді чат-бота, тому виникла потреба вибору найбільш підходящої платформи для підтримки чат-бота. Існує кілька платформ та інструментів, які можна використовувати для створення програмного забезпечення для розширення словникового запасу іноземної мови у вигляді чат-бота. Вибір платформи залежить від досвіду розробника, потреб проекту та мови програмування, яку він бажає використовувати.

Розглянемо інструменти для створення чат-ботів в месенджерах Telegram, Viber та Facebook Messenger для вибору найоптимальнішої платформи для взаємодії з користувачем.

Telegram Bot API – це одна з найпопулярніших платформ для створення чат-ботів [37]. Вона надає зручні інструменти для створення ботів для Telegram, включаючи можливість обробки текстових повідомлень та медіафайлів.

Telegram має велику кількість активних користувачів по всьому світу. Завдяки цьому є потенціал для залучення багатьох людей, які бажають покращити свої навички в іноземних мовах. Telegram Bot API має дуже простий та легко зрозумілий інтерфейс для розробників. Це дозволяє швидко створювати прості боти та тестувати їх. Однією з переваг є можливість налаштувати бота так, щоб він відповідав на запити користувачів у реальному часі, що дозволяє створювати інтерактивні уроки та вправи, де користувачі отримують негайний зворотний зв'язок. Telegram підтримує багатомовність, що дозволяє створювати боти для навчання різних мов.

Telegram відомий своєю високою безпекою та шифруванням повідомлень. Це важливо, коли мова йде про збереження особистих даних користувачів, таких як відомості про навчання. Telegram Bot API підтримує багато різних функціональних можливостей, включаючи клавіатури для

вибору відповідей, зображення, кнопки, оновлення розмітки та інше. Можна налаштувати поведінку свого бота, визначити типи запитів, які він обробляє, та створювати власний навчальний контент. А також розробник має можливість відстежувати активність користувачів та отримувати звіти про продуктивність свого бота.

Telegram на сьогодні пропонує широкий вибір чат-ботів що включають різні можливості, від вивчення іноземних мов до спілкування з іноземцями. Особливу популярність боти набули завдяки ініціативі творців месенджера Telegram, які одними з перших надали розробникам значний інструментарій. Незалежні програмісти мають можливість творити інтерактивні інтерфейси з дуже широким функціоналом.

Приклад чат-боту в телеграм наведено на рисунку 3.2.

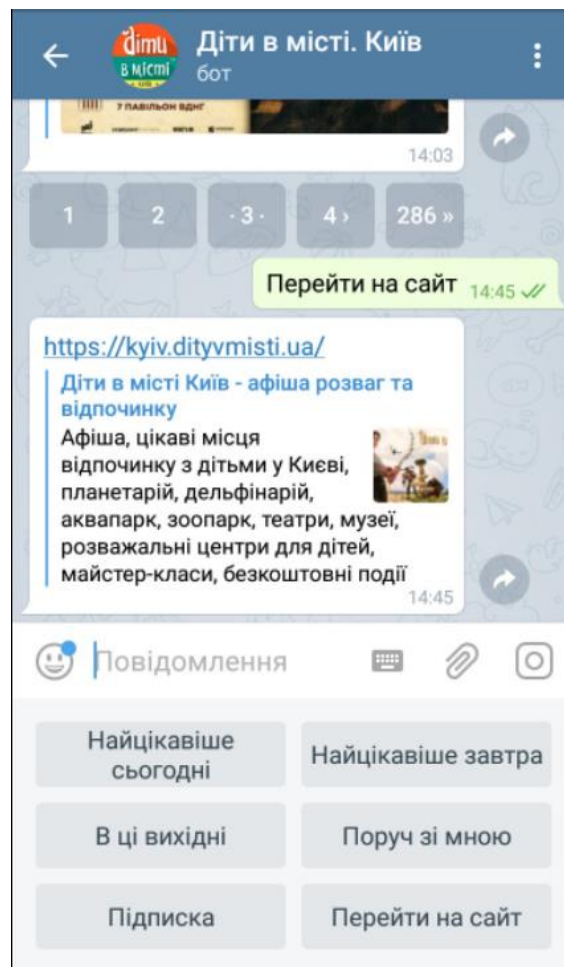


Рисунок 3.2 – Приклад чат-боту Telegram

Алгоритм функціонування ботів Telegram вкрай простий. Повідомлення, команди та запити, які користувачі надсилають, переадресовуються на сервер розробки. Анонімний проксі-сервер Telegram обробляє шифрування та обмін інформацією між утилітою та користувачем.

Взаємодія між користувачем і ботом відбувається наступним чином: користувач бота надсилає команду -> бот передає команду на сервер -> програма на сервері обробляє запит, отриманий від бота -> сервер відповідає боту -> бот показує відповідь на екрані від програми до користувача. Цей цикл повторюється, кожного разу, коли користувач взаємодіє з ботом Telegram.

Користувачі Telegram можуть створювати облікові записи та авторизуватися за допомогою коду доступу, отриманого як текстове повідомлення. Після первинної авторизації користувачі можуть налаштовувати свої облікові записи та здійснювати пошук інших користувачів. Функція двоетапної перевірки також доступна, якщо користувач бажає використовувати пароль при вході в обліковий запис.

Розглянемо месенджер Viber. Для розробки чат-бота для розширення словникового запасу іноземної мови у Viber, використовують Viber API. Веб-сайт Viber має набір інструментів і ресурсів для розробки ботів, які можуть взаємодіяти з користувачами у Viber. Viber підтримує кілька мов, включаючи Node.js, Python, PHP і багато інших. Viber надає можливість створювати бізнес-чат-ботів і використовувати їх для комунікації з клієнтами [38]. Це означає надання можливості навчати іноземній мові та взаємодіяти з користувачами, пропонуючи їм корисний контент. Якщо навчальна програма включає в себе відеокommунікацію або аудіокommунікацію для покращення навичок вимови, то можливість проводити відеодзвінки і аудіодзвінки безпосередньо в Viber, що є важливою перевагою.

Viber дозволяє створювати інтерактивні елементи в чат-ботах, такі як кнопки, каруселі зображень і інші. Це допомагає зробити навчання більш цікавим та залучити увагу користувачів. Viber відомий своєю високою

безпекою та шифруванням повідомлень, що допомагає зберегти конфіденційність даних користувачів. Також Viber надає засоби для відстеження активності користувачів та оцінки продуктивності вашого чат-бота.

Приклад чат-боту в месенджері Viber наведено на рисунку 3.3.

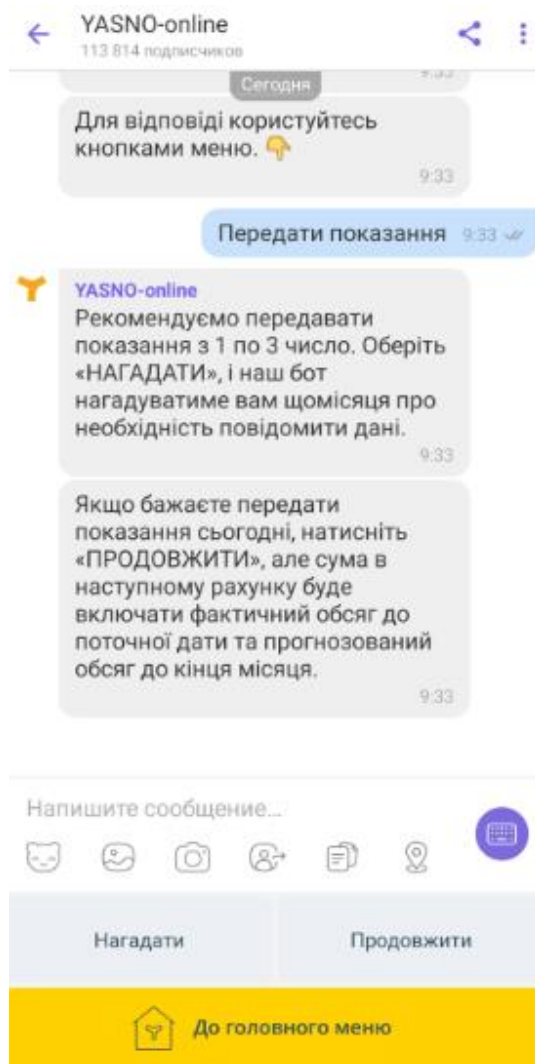


Рисунок 3.3 – Приклад чат-боту в месенджері Viber

Чат-бот Viber відкриває безліч можливостей, дозволяючи автоматично реагувати на повідомлення, надсилати текстові повідомлення, зображення, посилання, файли та інший контент на канал. Є також можливість додавати ярлики або опитування до повідомлень, щоб надати користувачам конкретний набір дій або варіантів відповідей. Боти Viber також підтримують

перевірку статусу передплатників каналу та збір доступної інформації про них. Функціональність чат-ботів Viber вражає своєю різноманітністю та можливостями, не поступаючи іншим платформам. Взаємодія з аудиторією відбувається через відкритий обліковий запис в месенджері. Це забезпечує набір стандартних функцій та додаткові можливості, такі як:

- розсилки, які включають в себе відправку повідомлень всім контактам, включаючи тих, хто відписався від каналу;
- оформлення предметів у форматі «каруселі» з товарами, що сприяє зручному споживанню інформації та взаємодії з аудиторією.

Дана платформа є комерційною, і її фінансування здійснюється за рахунок реклами та платних функцій. Месенджер акцентує увагу на своєму високому рівні безпеки, хоча слід зазначити, що резервні копії не підлягають шифруванню. Незважаючи на широку популярність месенджера, варто враховувати, що ця платформа може не бути найкращим вибором для розробки інтелектуального модуля, оскільки функціонал чат-ботів на цій платформі зроблено платним.

Facebook Messenger Platform - це потужна платформа для створення чат-ботів, які можуть бути використані для розширення словникового запасу іноземної мови у користувачів. Вона надає інструменти та можливості для розробки таких ботів [39]. Приклад чат-боту в Facebook Messenger наведено на рисунку 3.4.

Facebook Messenger Platform надає набір інструментів і можливостей для створення різноманітних чат-ботів, які можуть вдосконалювати взаємодію користувачів з вашим брендом або сервісом. Це може бути потужним інструментом для покращення обслуговування клієнтів, навчання, розваг та маркетингу.

Окрім того, що Facebook Messenger має велику аудиторію користувачів по всьому світу, він також відомий своєю безпекою та шифруванням повідомлень, що дозволяє зберегти конфіденційність даних користувачів.

Також вагомою перевагою є наявність активної спільноти розробників та надання різноманітних ресурсів для підтримки та розвитку чат-ботів.

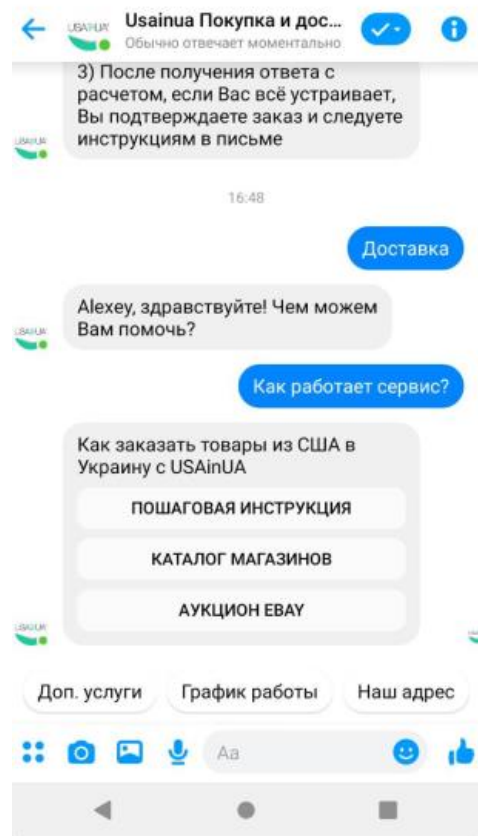


Рисунок 3.4 – Приклад чат-боту в Facebook Messenger

Як і попередньо розглянуті месенджери, Facebook Messenger надає інструменти для аналізу активності користувачів, що дозволяє оцінювати ефективність чат-бота та вдосконалювати його з часом. Тут також наявні інтерактивні елементи, такі як кнопки та каруселі, щоб зробити навчання більш цікавим та привабливим для користувачів.

Іншими суттєвими перевагами можна назвати можливість інтегрувати зовнішні сервіси, такі як онлайн-словники, перекладачі або інші освітні ресурси, щоб зробити чат-бот більш корисним. А також можливість використовувати рекламу на Facebook для просування чат-бота серед цільової аудиторії.

Хоча розробка чат-бота для розширення словникового запасу іноземної мови на Facebook Messenger Platform має численні переваги, вона також має свої недоліки. Чат-боти на Facebook Messenger можуть бути недоступні для користувачів, які не користуються цією платформою. Збір та зберігання даних користувачів повинні бути ретельно обговорені, оскільки вони можуть містити особисту інформацію.

Аналізуючи розглянуті платформи, можна зробити висновок, що найбільш оптимальною платформою є Telegram, адже він має перевагу у відкритості та безпеці, порівняно з двома іншими платформами. Telegram відомий своєю високою ступенем безпеки та шифруванням повідомлень. Це може бути важливим фактором, коли робиться акцент на освітній контент та зберіганні конфіденційних даних користувачів. Також Telegram відомий своєю доступністю в країнах, де інші платформи можуть бути заблоковані. Це дозволяє досягти аудиторії в регіонах з обмеженим доступом.

3.3 Опис бібліотек та фреймворків

В ході розробки програмного застосунку було використано стандартні бібліотеки jOOQ, Spring Framework та спеціальна бібліотека для розробки чат-ботів TelegramBots API.

jOOQ (Java Object-Oriented Querying) – це Java бібліотека, яка надає програмістам зручний спосіб взаємодіяти з реляційними базами даних за допомогою мови SQL в об'єктно-орієнтованому стилі. jOOQ спрощує створення і виконання SQL-запитів, робить код більш читабельним і покращує безпеку запитів шляхом перевірки типів даних під час компіляції [40].

jOOQ дозволяє виразити SQL-запити як об'єкти Java, що робить код більш зрозумілим та легким для супроводу; підтримує багато реляційних СУБД, включаючи PostgreSQL, MySQL, Oracle, SQL Server, та інші; SQL-запити генеруються в об'єктно-орієнтованому коді, що дозволяє виявляти помилки під час компіляції, а не під час виконання; дотримується стандартів

SQL і дозволяє використовувати всі функціональні можливості бази даних; може генерувати Java-код з схеми бази даних, включаючи таблиці, стовпці, ключі тощо, що робить роботу з базою даних більш простою; дозволяє легко викликати збережені процедури та функції бази даних з Java-коду; підтримує різні джерела даних, включаючи роботу з JDBC, JPA, та іншими технологіями.

Загалом, jOOQ створена для полегшення роботи з базами даних у Java-проектах і надає зручний та об'єктно-орієнтований інтерфейс для виконання SQL-запитів, роблячи розробку додатків, які взаємодіють з реляційними СУБД, більш ефективною та надійною.

Spring Framework – це один з найпопулярніших і розширюваних фреймворків для розробки додатків на мові програмування Java. Він надає комплексний набір інструментів та бібліотек для створення різноманітних додатків, включаючи веб-додатки, корпоративні програми, мікросервіси та інші [41].

Наведемо ключові аспекти Spring Framework. Spring використовує принципи IoC та DI для зменшення залежностей між компонентами додатку, що полегшує тестування та підтримку коду. Spring надає контейнер, який керує об'єктами. Цей контейнер дозволяє створювати, конфігурувати та управляти біном та його життєвим циклом. Spring розділений на модулі, які можуть бути використані в залежності від конкретних потреб проекту. Наприклад, Spring Core, Spring AOP, Spring Data, Spring Security, і багато інших модулів. Надає можливість реалізувати аспектно-орієнтоване програмування, що спрощує вирішення різних аспектів, таких як логування, транзакції та безпека. Може бути використаний разом з іншими технологіями Java EE, такими як JPA, JMS, і JDBC.

Spring дозволяє легко створювати RESTful веб-служби за допомогою Spring MVC або Spring WebFlux. Надає підтримку декларативного управління транзакціями та інтеграцію з різними менеджерами транзакцій. Spring надає інструменти для тестування, включаючи підтримку юніт-тестів,

інтеграційних тестів та тестування контейнера бобів. Spring Cloud та Spring Boot допомагають створювати та керувати мікросервісами в розподіленому додатку.

TelegramBots API – це офіційна бібліотека, надана Telegram для розробки ботів на Java [42]. Вона спрощує створення ботів та надає доступ до всіх можливостей Telegram API. Ця бібліотека надає програмістам інструменти для створення ботів, які можуть відправляти та отримувати повідомлення, створювати клавіатури, обробляти команди користувачів та багато інших функцій для взаємодії з Telegram API.

TelegramBots API пропонує зручний та простий API для створення та налаштування телеграм-ботів. Розробники можуть швидко почати роботу з бібліотекою та створити бота всього за кілька рядків коду. Бібліотека дозволяє обробляти текстові повідомлення, фотографії, відео, документи, голосові повідомлення та багато інших типів контенту, які користувачі можуть надсилати ботам. TelegramBots API підтримує асинхронну обробку повідомлень, що дозволяє ботам обробляти багато запитів одночасно. Бібліотека надає інструменти для створення тестів та імітації повідомлень для тестування ботів. TelegramBots API оновлюється для підтримки останніх версій Telegram Bot API, що забезпечує сумісність з новими функціями та змінами в API Telegram.

Загалом, TelegramBots API дозволяє розробникам створювати та розширювати функціональність телеграм-ботів на мові програмування Java. Вона добре підходить для створення ботів для різних цілей, включаючи розповсюдження інформації, обробку запитів користувачів, автоматизацію завдань та інше.

Ці бібліотеки забезпечують необхідними інструментами для зручної інтеграції Java-дodatка з Telegram та реалізації функціоналу для розширення словникового запасу іноземної мови у користувача.

3.4 Розробка UML-діаграми класів інформаційної технології для розширення словникового запасу іноземної мови у користувача

Створення UML-діаграм класів – це важливий етап у процесі розробки програмного забезпечення, оскільки це дозволяє візуалізувати структуру системи та взаємодію її компонентів.

UML-діаграма класів дозволяє представити структуру системи на високому рівні, показуючи класи та їх взаємодії, допомагає уточнити та визначити вимоги до системи шляхом ідентифікації ключових класів та їх взаємодій. Використовується для визначення архітектури системи, її компонентів та взаємозв'язків. Діаграми класів можуть служити основою для автоматичної генерації початкового коду, служать як частина технічної документації, яка полегшує зрозуміння структури системи іншими розробниками. Дозволяє провести аналіз системи на ранніх етапах розробки та виявити можливі проблеми в архітектурі. Сприяє спільному розумінню структури системи серед всіх учасників проекту. Вказує на можливості рефакторингу та оптимізації структури системи. Дозволяє відобразити об'єктно-орієнтовані концепції, такі як спадкування, поліморфізм, інкапсуляція та інші.

Наведемо класи, що були розроблені для інформаційній технології для розширення словникового запасу іноземної мови у користувача.

1. Класи для роботи з користувачами:
 - MyUserBuilder: Клас для побудови об'єктів користувача.
 - MyUserRepoImpl: Реалізація репозиторію користувачів.
 - MyUserService: Сервіс, що надає функції для роботи з об'єктами класу «MyUser».
2. Класи для обробки повідомлень:
 - MessageHandler: Клас для обробки повідомлень.
 - UpdateProcessorImpl: Реалізація обробника оновлень.
3. Класи для роботи з тестами:

- TestResultBuilder: Клас для побудови результатів тестів.
 - TestResult: Клас, що представляє результат тестування.
 - TestService: Сервіс для роботи з тестами.
4. Кеші та сервіси для слів:
- WordService: Сервіс для роботи із словами.
 - Cache: Загальний клас кешу.
 - WordRepository: Репозиторій для слів.
5. Інші важливі класи:
- MessageSenderImpl: Реалізація інтерфейсу для відправлення повідомлень.
 - TelegramBot: Клас для взаємодії з Telegram API.
 - ModelMapperConfig: Конфігурація бібліотеки мапінгу об'єктів, використовується для конвертації об'єктів між різними класами.
 - Question: Клас для представлення питань у тестах.
6. Винятки та статуси:
- NoSuchAdminException: Виняток для випадку, коли немає адміністратора.
 - StudentNoSuchException: Виняток для випадку, коли немає студента.
 - UserLevel: Перерахування для різних рівнів користувача.
7. Допоміжні класи:
- ReplyKeyboardMarkupUtils: Утиліта для створення клавіатур у відповідях.
 - WordBuilder: Клас для побудови об'єктів слова.
 - WrongAnswer, WrongAnswerBuilder: Клас та його білдер для представлення неправильної відповіді.

На рисунках 3.5 та 3.6 наведено фрагменти UML-діаграми класів інформаційної технології для розширення словникового запасу іноземної мови у користувача.

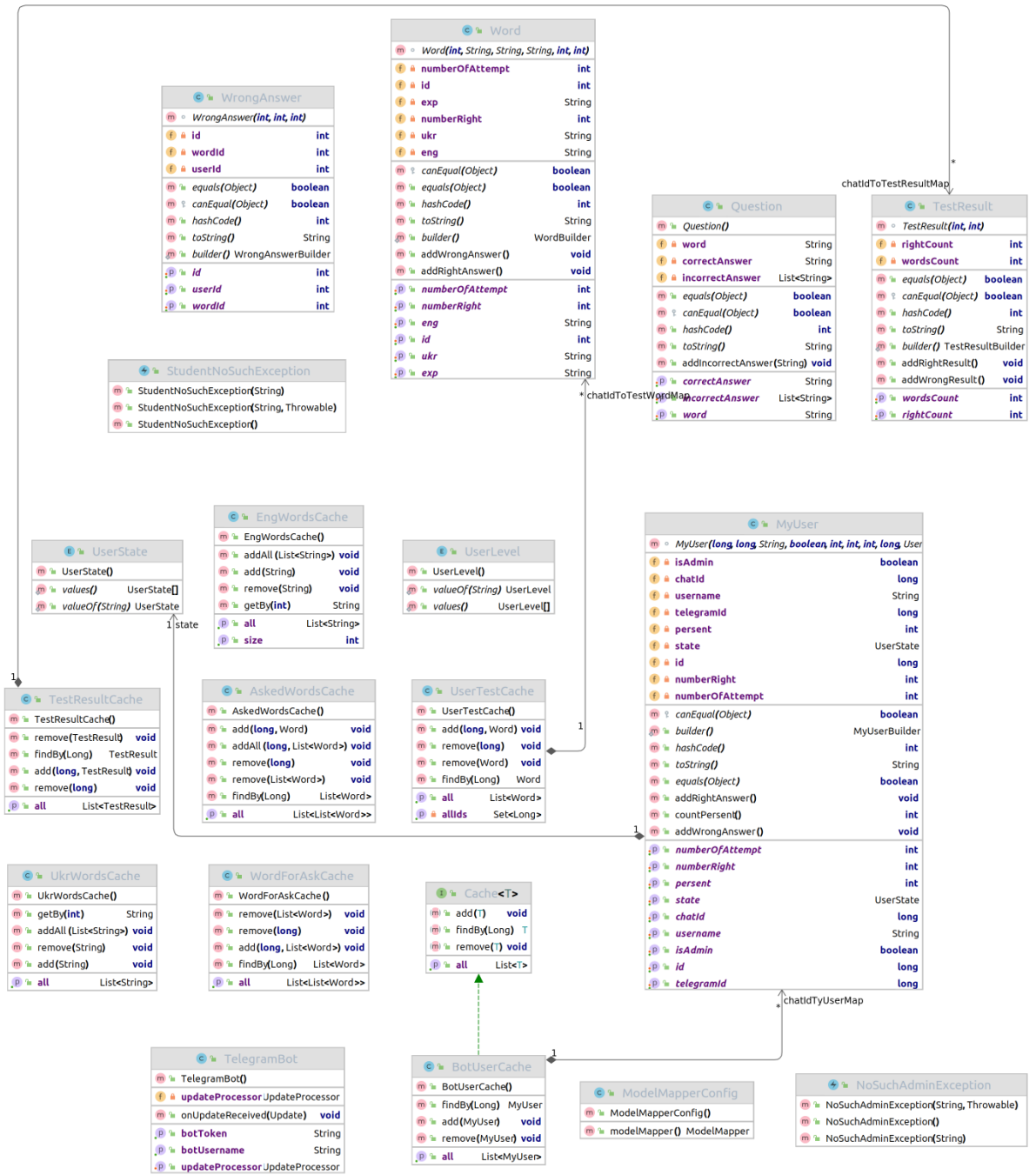


Рисунок 3.5 – Фрагмент UML-діаграму класів інформаційної технології

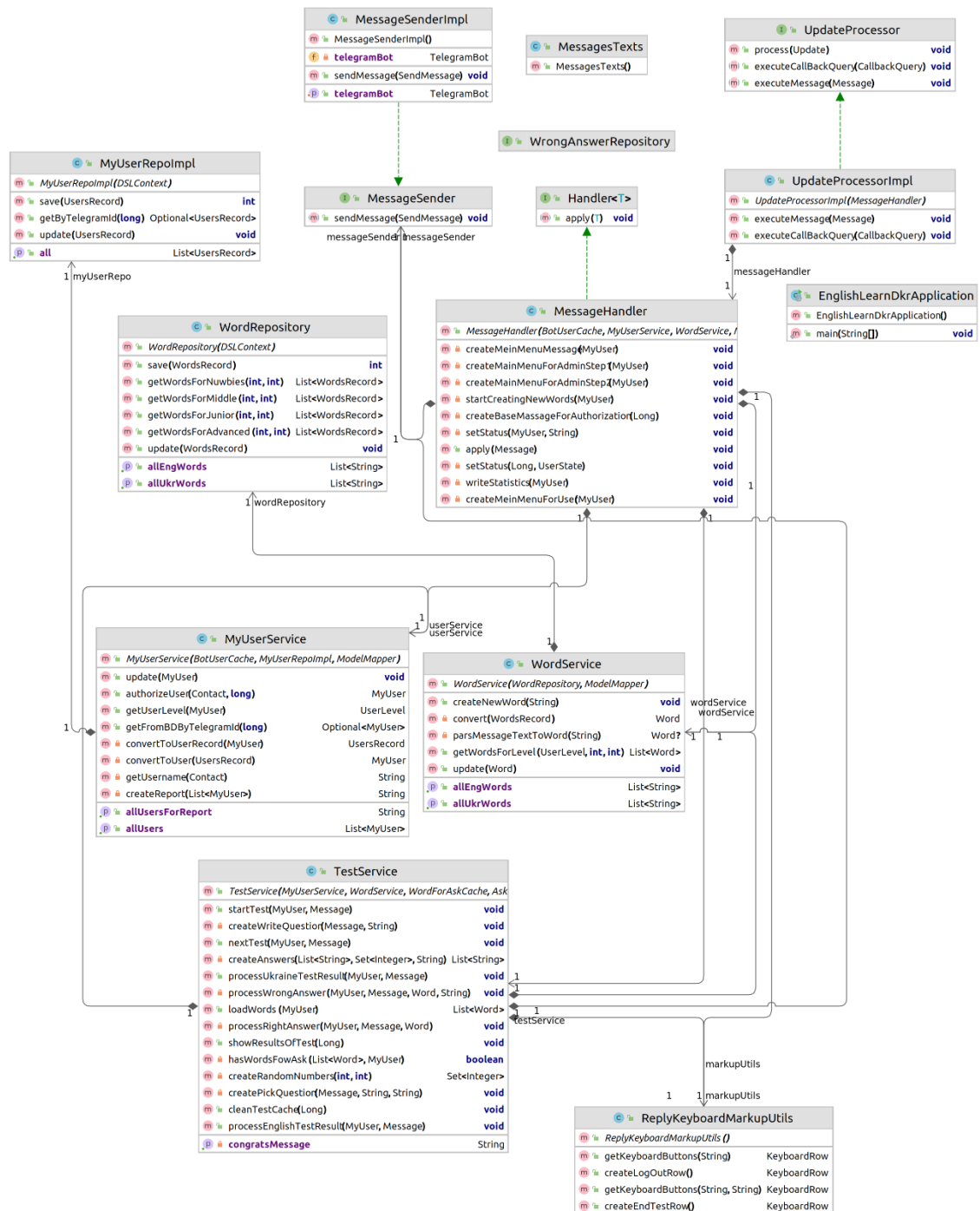


Рисунок 3.6 – Фрагмент UML-діаграму класів інформаційної технології

Деякі з розглянутих класів написані за допомогою популярних бібліотек та фреймворків. Так, класи User, Words, Wronganswer та багато інших використовують бібліотеку jOOQ. Це дозволяє ефективно взаємодіяти з базою даних, надаючи високорівневий доступ до SQL-запитів.

Деякі інші класи, зокрема `ModelMapperConfig` та `TelegramBot`, побудовані з використанням `Spring Framework`. `Spring` є високопродуктивним фреймворком для розробки Java-додатків, особливо веб-додатків. `ModelMapperConfig` може бути використаний для конфігурації мапінгу об'єктів, тоді як `TelegramBot`, використовує `Spring` для організації інфраструктури та обробки подій.

3.5 Тестування та дослідження результатів роботи інформаційної технології для розширення словникового запасу іноземної мови у користувача

Розроблена інформаційної технології для розширення словникового запасу у користувача була протестована, що підтвердило коректність її роботи. Для тестування та дослідження результатів роботи інформаційної технології для розширення словникового запасу іноземної мови у користувача, потрібно перевірити, чи відповідає інформаційної технології очікуванням, чи виконує функції, покладені на неї, визначити, наскільки добре інформаційна технологія допомагає користувачам розширювати свій словниковий запас іноземної мови. А також чи можна досягти очікуваного підвищення ефективності запам'ятовування іноземних слів за допомогою інформаційної технології для розширення словникового запасу іноземної мови у користувача.

Тестування телеграм бота проводився вручну. Після кожного етапу розробки тести продуктивності проводилися на заздалегідь підготовлених тестах.

Тестування роботи з ботом `Telegram` включає етапи [43]:

- а) правильна робота автентифікації користувача;
- б) правильна реакція на запити та повідомлення від користувача.
- в) коректна робота кнопок або меню;
- г) обробка натискання клавіші відправки повідомлення;

д) продуктивність бота, зокрема його швидкість реакції на запити користувачів та завантаженість серверів;

е) коректна обробка оновлень та нових функцій, які можуть додаватися в майбутньому.

Тестування показало високу швидкість реагування бота на повідомлення та відповіді користувача. Усі наведені вимоги виконувались підтвердили коректність роботи бота.

Запустивши бот у Telegram користувач розпочинає діалог (рис. 3.7). Надіславши повідомлення, як наприклад вітання, слід очікувати що бот привітається у відповідь та вкаже на потребу авторизації користувача для подальшої роботи.



Рисунок 3.7 – Початок діалогу з ботом

Ввівши свої дані для авторизації (поділившись своїм контактом) система визначає тип користувача. Існує два типи користувачів –

адміністратор та звичайний користувач. Користувач, що має права адміністратора, матиме дві опції продовження роботи – в ролі адміністратора, або в ролі користувача для тестування знань. Також йому пропонується завершити діалог з системою (рис. 3.8).

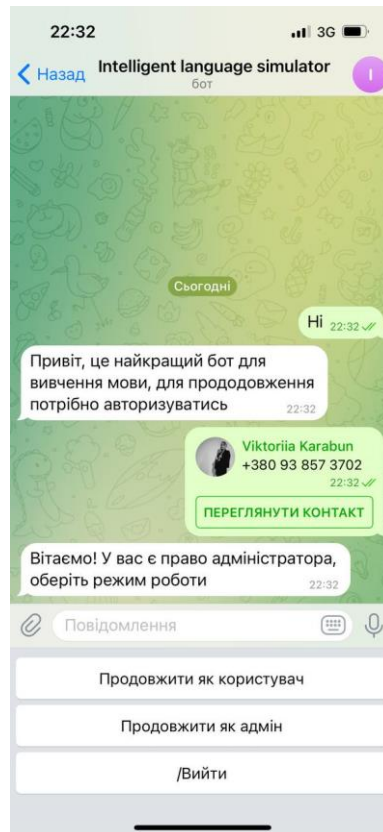


Рисунок 3.8 – Вкладка авторизації користувача

Якщо вибрана опція роботи для звичайного користувача, бот запропонує обрати режими роботи (рис. 3.9). Існує чотири види тестів: «Англійська-Українська тест», «Англійська-Українська введення відповіді», «Українська-Англійська тест», «Українська-Англійська введення відповіді». Також користувач може легко вийти з діалогу, натиснувши запропоновану кнопку.

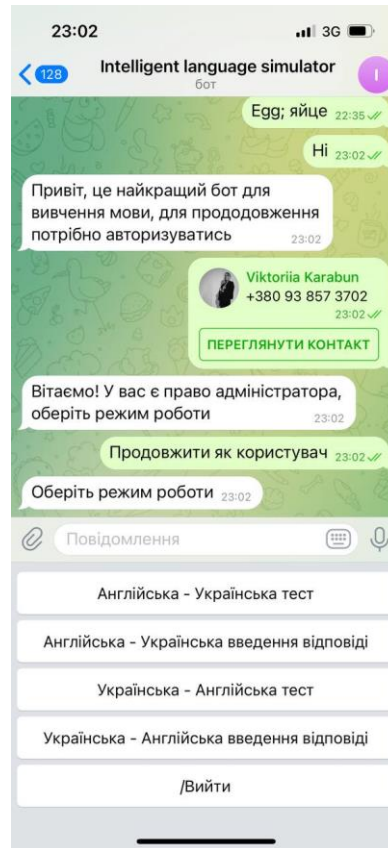


Рисунок 3.9 – Вкладка «Вибір режиму тестування»

Відповідно до обраного користувачем типу тестування відбуваються подальші дії. Так, при виборі тестування «Українська – Англійська тест» бот надсилає повідомлення з українським словом, а користувач отримує чотири варіанти відповідей (рис. 3.10).

На цьому етапі він також має змогу завершити тестування з можливістю подальшого збереження результатів. Після того, як користувач відповість на питання, відбувається перехід до наступного. Якщо відповідь була правильна – система може запропонувати складніші слова. Якщо ж невірна – простіші. Користувач проходить тестування і не відволікається на те, щоб змінити рівень складності.

Якщо авторизований користувач має права адміністратора та бажає продовжити роботу як адміністратор, система пропонує наступні режими подальшої роботи «Переглянути статистику», «Додати нове слово» та «Вийти» (рис. 3.11).

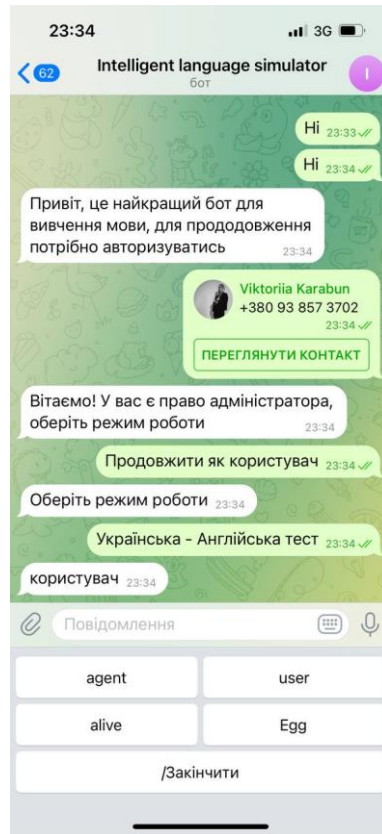


Рисунок 3.10 – Вкладка «Проходження тестування»

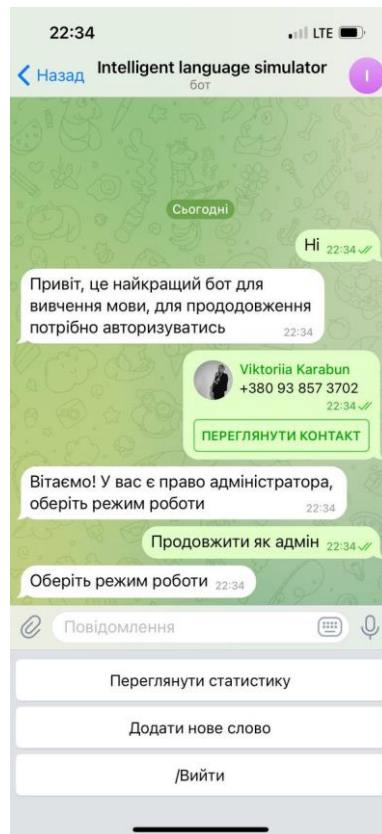


Рисунок 3.11 – Вкладка «Вибір режиму роботи»

Після вибору режиму «Переглянути статистику» адміністратору відкривається статистика, що містить результати оцінювання користувачів (рис. 3.12).

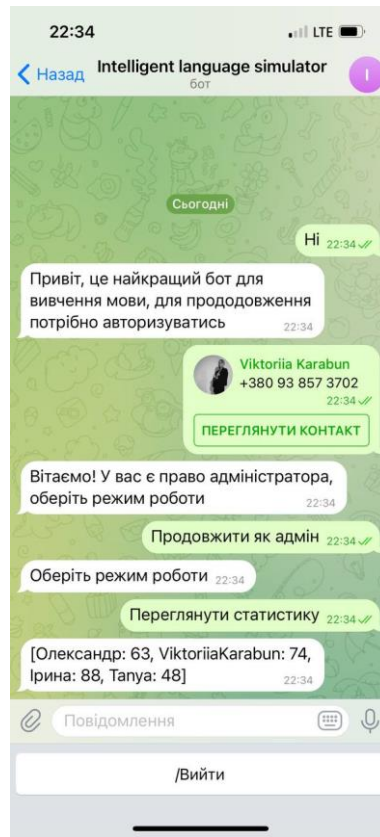


Рисунок 3.12 – Вкладка «Перегляд статистики»

Якщо адміністратор бажає додати нове слово, бот надішле повідомлення з інструкцією щодо введення нового слова «Введіть слово в форматі 'english word; українське слово'. Після чого адміністратор вводить повідомлення, наприклад, «Egg; яйце» для додавання слова (рис. 3.13).



Рисунок 3.13 – Вкладка «Додавання нового слова»

В таблиці 3.1 наведено порівняльні результати роботи програми з програмами-аналогами «Duolingo», «Memrise», «Lexicon By Like Thought, LLC», «Busuu», «Andy English Bot».

Таблиця 3.1 – Порівняльний аналіз роботи розроблюваного додатку порівняно з аналогами

Програма	Платний функціонал	Функція інтервального повторення	Функція оцінювання
Розроблений додаток	-	+	+
Duolingo	+	+	-
Memrise	+	+	-
Lexicon By Like Thought, LLC	+	-	+
Busuu	+	-	-
Andy English Bot	-	-	-

З таблиці 3.1 випливає, що розроблена інформаційна технологія має переваги у вигляді безкоштовного функціоналу, наявності функції

інтервального повторення, що сприяє оптимальному запам'ятовуванню, а також забезпечує наявність функції оцінювання, що розширює функціонал програм-аналогів.

При аналізі розглянутих додатків виявлено переваги інформаційної технології для розширення словникового запасу іноземної мови у користувача, які сприяють підвищенню ефективності процесу запам'ятовування іноземних слів користувачем. Серед цих переваг виділяються наступні аспекти:

- використання методів просторового повторення для оптимального навчання. Це забезпечує систематичне і повторне представлення слів, що сприяє їх ефективному закріпленню в пам'яті користувача;
- надає широкий спектр інтерактивних тестів та завдань, що спрямовані на вивчення слів. Це стимулює активну участь користувача та покращує процес засвоєння лексичного матеріалу.
- враховує індивідуальні потреби та рівень володіння користувача. Це сприяє ефективному вивченню, орієнтованому на конкретні потреби кожного користувача.
- застосування динамічних засобів оцінки успішності та підтримки користувача. Це включає аналіз помилок, можливість оновлення матеріалів та персоналізовані рекомендації для поліпшення навчання.

Вивчаючи мову, користувачі мали можливість відстежувати власний прогрес у вивченні слів. Тестування ефективності запам'ятовування іноземних слів відбувалося на групі користувачів, що могли оцінити рівень словникового запасу до користування додатком та під час його використання. Користувачі регулярно перевіряли свої досягнення та переконалися, що вони досягають поставлених цілей. Користувачі високо оцінили, зручність технології у використанні. Так як вона відповідає їхнім потребам і уподобанням, це також мало позитивний вплив на ефективність навчання. Опитавши користувачів, вони підтвердили, що відчули, що вони

дійсно вивчають та запам'ятовують слова. Внутрішнє відчуття прогресу є добрим показником ефективності програми.

Отже, порівняно з розглянутими аналогічними рішеннями виявлено переваги у наявності функцій, які допоможуть підвищити ефективність запам'ятовування іноземних слів у користувача.

3.6 Висновок до розділу 3

Здійснено обґрунтування вибору мови та середовища програмування для реалізації інформаційної технології для розширення словникового запасу іноземної мови у користувача. Розробка інформаційної технології проводилась об'єктно-орієнтованою мовою програмування Java, що вирізняється платформонезалежністю, що надає гнучкість у використанні на різних пристроях та під різними операційними системами, а також володіє високим рівнем безпеки, надійною системою управління пам'яттю та ефективністю у роботі, в середовищі розробки IntelliJ IDEA. Обґрунтовано використання платформи Telegram, відомої своїм високим ступенем безпеки та зашифруванням повідомлень, що є критичним для забезпечення конфіденційності користувачів. Розроблено UML-діаграму класів програми, що служить важливим інструментом документації та зрозуміння архітектури програмного продукту. Основні бібліотеки та фреймворки, такі як jOOQ для роботи з базою даних та Spring Framework для підтримки об'єктно-орієнтованого підходу, були використані для покращення ефективності розробки. На основі цього було програмно реалізовано додаток. Було проведено тестування розробленої інформаційної технології, яке підтвердило коректність та працездатність її роботи. Порівняльний аналіз з існуючими аналогами визначив переваги створеної інформаційної технології. Зокрема, виявлено наявність функцій, які сприяють ефективному процесу запам'ятовування іноземних слів у користувача, що робить його привабливим і конкурентоспроможним на ринку програм для вивчення мов.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Проведення комерційного та технологічного аудиту інформаційної технології для розширення словникового запасу іноземної мови у користувача

Метою проведення комерційного і технологічного аудиту є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності, тобто під час виконання магістерської кваліфікаційної роботи.

Для проведення комерційного та технологічного аудиту залучаємо 3-х незалежних експертів, якими є провідні викладачі випускової або спорідненої кафедри.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу здійснюємо із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, а результати зводимо до таблиці 4.1 [44].

Таблиця 4.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу інформаційної технології для розширення словникового запасу іноземної мови у користувача

Критерії	Експерти		
	Барабан С.В.	Озеранський В. С.	Арсенюк І. Р.
	Бали, виставлені експертами		
Технічна здійсненність концепції	3	2	3
Ринкові переваги (наявність аналогів)	4	3	3
Ринкові переваги (ціна продукту)	2	3	2
Ринкові переваги (технічні властивості)	3	3	3
Ринкові переваги (експлуатаційні витрати)	3	2	3
Ринкові перспективи (розмір ринку)	4	3	4
Ринкові перспективи (конкуренція)	3	3	3
Практична здійсненність (наявність фахівців)	3	4	3

Продовження таблиці 4.1

Практична здійсненність (наявність фінансів)	3	3	3
Практична здійсненність (необхідність нових матеріалів)	4	4	3
Практична здійсненність (термін реалізації)	4	4	3
Практична здійсненність (розробка документів)	3	3	2
Сума балів	39	37	35
Середньоарифметична сума балів, СБ	37		

За результатами розрахунків, наведених в таблиці 4.1 робимо висновок про те, що науково-технічний рівень та комерційний потенціал інформаційної технології для розширення словникового запасу іноземної мови у користувача – вище середнього.

4.2 Розрахунок витрат на здійснення науково-дослідної роботи

Витрати на оплату праці. Належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці, також будь-які види грошових і матеріальних доплат, які належать до елемента «Витрати на оплату праці».

Основна заробітна плата дослідників. Витрати на основну заробітну плату дослідників (Z_0) розраховують відповідно до посадових окладів працівників, за формулою 4.1 [45]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \#(4.1)$$

де k – кількість посад дослідників, залучених до процесу дослідження; M_{ni} – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.; T_p – число робочих днів в місяці; приблизно $T_p = (21 \dots 23)$ дні; t_i – число робочих днів роботи розробника (дослідника).

Зроблені розрахунки зводимо до таблиці 4.2.

Таблиця 4.2 – Витрати на заробітну плату дослідників

Посада	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник	40000	1818	40	72720
Розробник	30000	1364	40	40920
Всього:				113640

Основна заробітна плата робітників. Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт розраховують за формулою 4.2:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \#(4.2)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год; t_i – час роботи робітника на виконання певної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою 4.3:

$$C_i = \frac{M_M \cdot K_i \cdot K_C}{T_p \cdot t_{зМ}}, \#(4.3)$$

де M_m – розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати (залежно від діючого законодавства), у 2023 році $M_m=6700$ грн; K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду; K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати; T_p – середня кількість робочих днів в місяці, приблизно $T_p = 21...23$ дні; $t_{зм}$ – тривалість зміни, год. Зроблені розрахунки зводимо до таблиці 4.3.

Таблиця 4.3 – Витрати на заробітну плату робітників

Найменування робіт	Трудомісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
Аналіз предметної області	16	5	51,2	1,36	819,2
Моделювання інформаційної технології	140	7	58,6	1,54	8204
Створення і наповнення БД	80	5	51,2	1,36	4096
Створення основної структури проекту	16	6	55,2	1,45	883,2
Тестування	68	3	44,9	1,18	3053,2
Всього					17056

Додаткова заробітна плата. Додаткова заробітна плата Z_d всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10...12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Z_d = 0,1 \cdot (Z_o + Z_p) = 0,1 \cdot (113640 + 17056) = 13070 \text{ грн.}$$

Відрахування на соціальні заходи. Нарахування на заробітну плату $H_{зп}$ розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$\begin{aligned} H_{зп} &= \beta \cdot (Z_o + Z_p + Z_d) = \\ &= 0,22 \cdot (113640 + 17056 + 13070) = 31628 \text{ грн.} \end{aligned}$$

де Z_o – основна заробітна плата розробників, грн.; Z_p – основна заробітна плата робітників, грн.; Z_d – додаткова заробітна плата всіх розробників та робітників, грн.; β – ставка єдиного внеску на загальнообов’язкове державне соціальне страхування, % (приймаємо для 1-го класу професійності ризику 22%).

Програмне забезпечення. До балансової вартості програмного забезпечення входять витрати на його інсталяцію, тому ці витрати беруться додатково в розмірі 10...12% від вартості забезпечення. Балансову вартість програмного забезпечення розраховують за формулою 4.4:

$$V_{\text{прг}} = \sum_1^k C_{\text{іпрг}} \cdot C_{\text{прг.і}} \cdot K_i, \#(4.4)$$

де $C_{\text{іпрг}}$ – ціна придбання програмного забезпечення і-го виду, грн.; $C_{\text{прг.і}}$ – кількість одиниць програмного забезпечення відповідного виду, шт.; K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного забезпечення, $K_i = (1,1 \dots 1,12)$; k – кількість видів програмного забезпечення.

Зроблені розрахунки зводимо до таблиці 4.4.

Таблиця 4.4 – Витрати на придбання програмного забезпечення

Найменування програмного забезпечення	Ціна за одиницю, грн.	Витрачено	Вартість програмного забезпечення, грн.
Intellij Idea	23361	1	23361
MySQL Workbench	36147	1	36147
Всього, з врахуванням коефіцієнта інсталяції та налагодження			65459

Амортизація обладнання. Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час (чи для) виконання даного етапу роботи.

У спрощеному вигляді амортизаційні відрахування A в цілому бути розраховані за формулою 4.5:

$$A = \frac{Ц_б}{T_в} \cdot \frac{t}{12}, \#(4.5)$$

де $Ц_б$ – загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, грн.; t – термін використання основного фонду, місяці; $T_в$ – термін корисного використання основного фонду, роки. Зроблені розрахунки зводимо до таблиці 4.5.

Таблиця 4.5 – Амортизаційні відрахування за видами основних фондів

Найменування	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, місяців	Сума амортизації, грн.
Ноутбук ASUS TUF Gaming A15	54999	5	2	1833
монітор Samsung Curved	5299	5	2	176
Всього	2009			

Витрати на електроенергію для науково-виробничих цілей. Ввідні дані наведено у таблиці 4.6. Витрати на силову електроенергію V_e , якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою:

Таблиця 4.6 – Витрати на електроенергію

Найменування обладнання	Потужність, кВт	Тривалість годин роботи
Ноутбук (ASUS TUF Gaming A15)	0,096	320
Монітор (Samsung Curved)	0,036	320
Освітлення	0,024	100

$$V_e = \sum \frac{W_i \cdot t_i \cdot C_e \cdot K_{\text{впі}}}{\text{ККД}}, \#(4.6)$$

$$V_e = \frac{0,096 \cdot 320 \cdot 7,5 \cdot 0,95}{0,96} + \frac{0,036 \cdot 320 \cdot 7,5 \cdot 0,95}{0,96} + \frac{0,024 \cdot 100 \cdot 7,5 \cdot 0,95}{0,96} = 331 \text{ грн.},$$

W_i – встановлена потужність обладнання, кВт; t_i – тривалість роботи обладнання на етапі дослідження, год.; C_e – вартість 1 кВт електроенергії, грн.; $K_{\text{впі}}$ – коефіцієнт використання потужності; ККД – коефіцієнт корисної дії обладнання.

Інші витрати. До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників та робітників за формулою 4.7:

$$I_{\text{в}} = (Z_o + Z_p) \cdot \frac{N_{\text{ив}}}{100\%} = (113640 + 17056) \cdot \frac{65}{100} = 84952 \text{ грн.}, \#(4.7)$$

де $N_{\text{ив}}$ – норма нарахування за статтею «Інші витрати».

Накладні (загальновиробничі) витрати. До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників та робітників за формулою 4.8:

$$V_{\text{нзв}} = (Z_o + Z_p) \cdot \frac{N_{\text{нзв}}}{100\%} = (113640 + 17056) \cdot \frac{125}{100} = 163370 \text{ грн.}, \#(4.8)$$

де $N_{\text{нзв}}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

Витрати на проведення науково-дослідної роботи. Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою 4.9:

$$V_{\text{заг}} = Z_o + Z_p + Z_{\text{дод}} + Z_{\text{н}} + K_{\text{в}} + V_{\text{спец}} + V_{\text{прг}} + A_{\text{обл}} + V_{\text{е}} + I_{\text{в}} + V_{\text{нзв}}, \#(4.9)$$

$$V_{\text{заг}} = 113640 + 17056 + 13070 + 31628 + 65459 + 2009 + 331 + 84952 + 163370 = 491215 \text{ грн.}$$

Загальні витрати. Загальні витрати ЗВ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою 4.10:

$$ЗВ = \frac{B_{\text{заг}}}{\eta} = \frac{491215}{0,9} = 545794 \text{ грн.}, \#(4.10)$$

де η – коефіцієнт, що характеризує етап виконання науково-дослідної роботи. Оскільки, якщо науково-технічна розробка знаходиться на стадії технічного впровадження, то $\eta=0,9$.

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенціним інвестором

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку.

В даному випадку відбувається розробка засобу, тому основу майбутнього економічного ефекту буде формувати: ΔN – збільшення кількості споживачів, яким надається відповідна інформаційна послуга в аналізовані періоди часу; N – кількість споживачів, яким надавалась відповідна інформаційна послуга у році до впровадження результатів нової науково-технічної розробки; Π_0 – вартість послуги у році до впровадження інформаційної системи; $\pm\Delta\Pi_0$ – зміна вартості послуги (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу.

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta\Pi$ для кожного із років, протягом яких очікується отримання позитивних

результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховується за формулою 4.11:

$$\Delta\Pi = (\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N_i)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \#(4.11)$$

де $\pm\Delta\Pi$ – зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай, таким показником може бути зміна ціни реалізації одиниці нової розробки в аналізованому році (відносно року до впровадження цієї розробки); $\pm\Delta\Pi_0$ може мати як додатне, так і від’ємне значення (від’ємне – при зниженні ціни відносно року до впровадження цієї розробки, додатне – при зростанні ціни); N – основний кількісний показник, який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної розробки; Π_0 – основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році; Π_0 – основний якісний показник, який визначає ціну реалізації існуючої (базової) науково-технічної розробки у році до впровадження результатів; ΔN – зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай таким показником може бути зростання попиту на науково-технічну розробку в аналізованому році (відносно року до впровадження цієї розробки); λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість становить 20%, а коефіцієнт $\lambda = 0,8333$; ρ – коефіцієнт, який враховує рентабельність інноваційного продукту (послуги). Рекомендується брати $\rho = 0,2 \dots 0,5$; ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\vartheta = 18\%$.

Очікуваний термін життєвого циклу розробки 1 рік, тому:

$$\begin{aligned} \Delta\Pi &= ((4128 - 2160) \cdot 10000 - 0 \cdot 2160) \cdot 0,8333 \cdot 0,3 \cdot \left(1 - \frac{18}{100}\right) = \\ &= 4203648 \text{ грн.} \end{aligned}$$

Далі розраховують приведену вартість збільшення чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки за формулою 4.12:

$$PP = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \#(4.12)$$

$$PP = \frac{4203648}{(1 + 0,1)^1} = 3821498 \text{ грн.}$$

де $\Delta\Pi$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн.; T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки (приймаємо $T=1$ рік); τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$; t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу 4.13:

$$PV = k_{\text{інв}} \cdot 3B. \#(4.13)$$

$$PV = 3 \cdot 545794 = 1637382 \text{ грн.}$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{\text{інв}}=2\dots 5$, але може бути і більшим; $ЗВ$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

Тоді абсолютний економічний ефект $E_{\text{абс}}$ або чистий приведений дохід для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки розраховується за формулою 4.14:

$$E_{\text{абс}} = \text{ПП} - PV, \#(4.14)$$

$$E_{\text{абс}} = 3821498 - 1637382 = 2184116 \text{ грн.}$$

де ПП – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн.; PV – теперішня вартість початкових інвестицій, грн.

Оскільки $E_{\text{абс}} > 0$, то можемо припустити про потенційну зацікавленість інвесторів у розробці.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність E_v або показник внутрішньої норми дохідності вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Внутрішня економічна дохідність інвестицій E_v , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, розраховується за формулою 4.15:

$$E_B = \sqrt[T_{ж}] {1 + \frac{Ea_{бс}}{PV}}, \#(4.15)$$

$$E_B = \sqrt[1] {1 + \frac{2184116}{1637382}} = 1,61.$$

де $T_{ж}$ – життєвий цикл розробки, роки.

Визначимо бар'єрну ставку дисконтування $\tau_{\text{мін}}$, тобто мінімальну внутрішню економічну дохідність інвестицій, нижче якої кошти у впровадження науково-технічної розробки та її комерціалізацію вкладатися не будуть.

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{\text{мін}}$ визначається за формулою 4.16:

$$\tau_{\text{мін}} = d + f, \#(4.16)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні $d = 0,9...0,12$; f – показник, що характеризує ризикованість вкладення інвестицій; зазвичай величина $f = 0,05...0,5$, але може бути і значно вищою.

$$\tau_{\text{мін}} = 0,9 + 0,05 = 0,95$$

Оскільки $E_B = 1,61 > \tau_{\text{мін}} = 0,95$, то потенційний інвестор може бути зацікавлений у фінансуванні впровадження науково-технічної розробки та виведенні її на ринок, тобто в її комерціалізації.

Далі розраховуємо період окупності інвестицій T_o , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки за формулою 4.17:

$$T_o = \frac{1}{E_B} \cdot \#(4.17)$$

$$T_o = \frac{1}{1,61} = 0,62 \text{ року.}$$

Оскільки $T_o=0,62 < 1 \dots 3$ -х років, то це свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

4.4 Висновок до розділу 4

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційна технологія для розширення словникового запасу іноземної мови у користувача» вище середнього.

Було спрогнозовано орієнтовну величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення та економічний ефект при використанні даної розробки. Загальні витрати на завершення науково-дослідної роботи та оформлення її результатів становить 545794 грн. Термін окупності становить 0,62 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже, можна зробити висновок про доцільність проведення науково-дослідної роботи за темою «Інформаційна технологія для розширення словникового запасу іноземної мови у користувача».

ВИСНОВКИ

Всі завдання, поставлені для реалізації інформаційної технології для розширення словникового запасу іноземної мови у користувача виконані в повному обсязі, а саме: проаналізувати предметну область розширення словникового запасу іноземної мови у користувача; провести аналіз сучасних систем для розширення словникового запасу іноземної мови у користувача; розробити структуру інформаційної технології для розширення словникового запасу іноземної мови у користувача; розробити математичну модель розширення словникового запасу іноземної мови у користувача; здійснити програмну реалізацію інформаційної технології для розширення словникового запасу іноземної мови у користувача; провести тестування розробленого програмного забезпечення для розширення словникового запасу іноземної мови у користувача; економічно обґрунтувати доцільність розробки інформаційної технології програмного забезпечення для розширення словникового запасу іноземної мови у користувача.

Під час виконання магістерської кваліфікаційної роботи було реалізовано інформаційну технологію для розширення словникового запасу іноземної мови у користувача.

Обґрунтовано доцільність розробки інформаційної технології для розширення словникового запасу іноземної мови у користувача. Проведено аналіз предметної області розширення словникового запасу іноземної мови у користувача. Розглянуто важливість ролі інформаційних технологій у процесі вивчення іноземної мови, досліджено та обґрунтовано переваги використання чат-ботів для розширення словникового запасу іноземної мови у користувача.

Було проведено аналіз сучасних програм-аналогів, які використовуються для тренування навиків іноземної мови, тестування знань та розширення словникового запасу іноземної мови. Виявлено проблему обмеженого функціоналу безкоштовної версії більшості додатків та складність інтерфейсу, що довело актуальність розробки інформаційної технології для розширення словникового запасу іноземної мови у користувача.

Було розроблено структуру інформаційної технології для розширення словникового запасу іноземної мови у користувача, структурну схему

функціонування інформаційної технології, наведено діаграму варіантів використання та спроектовано схему алгоритму роботи розробленої програми. Розроблено базу даних та наведено ER-діаграму для розробки інформаційної технології для розширення словникового запасу іноземної мови у користувача.

На основі аналізу існуючих моделей та методів, що застосовуються для вирішення поставленої задачі було обґрунтовано використання в інформаційній технології методи, що здатні забезпечити підвищення ефективності запам'ятовування іноземних слів. Згідно специфіки задачі було запропоновано математичну модель для даної інформаційної технології на основі методики адаптивного контролю знань та методів інтервального повторення, що дозволяє підвищити ефективність запам'ятовування іноземних слів.

Здійснено програмну реалізацію інформаційної технології для розширення словникового запасу іноземної мови у користувача. Обґрунтовано вибір мови та середовища програмування. Розробка інформаційної технології велася на об'єктно-орієнтованій мові програмування Java в середовищі розробки IntelliJ IDEA. Розглянуто бібліотеки та фреймворки, що доцільно використовувати в інформаційній технології для розширення словникового запасу іноземної мови у користувача.

Проведено тестування розробленої програми та підтверджено її працездатність. Порівняно з аналогічними рішеннями виявлено переваги у наявності функцій, які допоможуть підвищити ефективність запам'ятовування іноземних слів у користувача, що означає доведення поставленої мети.

Здійснено економічне обґрунтування доцільності розробки інформаційної технології для розширення словникового запасу іноземної мови у користувача. Проведено оцінювання комерційного потенціалу розробки. Згідно висновків експертів, рівень комерційного потенціалу розробки є вищим середнього. Здійснено прогнозування витрат на виконання науково-дослідної роботи. Розраховано витрати на заробітну плату та амортизаційні відрахування, витрати на силову електроенергію. Загальні витрати становлять 545794 грн.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Карабун В.С., Белзецький Р.С. Інтелектуальний модуль для розширення словникового запасу іноземної мови у користувача. Матеріали LI Науково-технічної конференції підрозділів Вінницького національного технічного університету (2022). URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2022/paper/view/15469> (дата звернення: 25.09.2023).
2. Карабун В.С., Барабан С. В. Інформаційна технологія для розширення словникового запасу іноземної мови у користувача. Матеріали LII Науково-технічної конференції підрозділів Вінницького національного технічного університету (2023). URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2023/paper/view/18773> (дата звернення: 25.09.2023).
3. Важливість вивчення іноземних мов у сучасному світі. URL: http://frutta.ucoz.ua/publ/vazhlivist_vivchennja_inozemnikh_mov_u_suchasnomu_sviti/1-1-0-3 (дата звернення: 25.09.2023).
4. Інформаційні технології в сучасній системі освіти : моногр. / Романуха О.М., Зінченко В. М., Ревуцька С. К., Чевердак П. О., Шапран Д. П. Кривий Ріг : ДонНУЕТ, 2019. 122 с.
5. Гуревич Р. С., Кадемія М. Ю., Шевченко Л. С. Інформаційні технології навчання: інноваційний підхід : навч. посіб. Вінниця : Планер, 2012, 348 с.
6. Що таке чат бот: визначення, приклади, відео. URL: <https://sendpulse.ua/support/glossary/chatbot> (дата звернення: 14.06.2023).
7. Як штучний інтелект допомагає вчити іноземні мови URL: <https://www.bbc.com/ukrainian/articles/cw0p8xjy5qko> (дата звернення: 20.09.2023).

8. Карпа І. Використання веб-сторінок та соціальної мережі Facebook у процесі навчання англійської мови // Актуальні питання гуманітарних наук. Дрогобич, 2015. С. 267-272.
9. Субботін С.О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень: навч. посіб. Запоріжжя: ЗНТУ, 2008. 341 с.
10. Баган Т. Г. Методи машинного навчання при проектуванні автоматизованих систем керування : навч. посіб. Київ: КПІ ім. Ігоря Сікорського, 2021, 28 с.
11. Шучні нейронні мережі. URL: http://www.victoria.lviv.ua/html/neural_nets/Lecture1.htm (дата звернення: 18.08.2023).
12. Яровий А. А., Арсенюк І. Р., Месюра В. І. Експертні системи. Частина 2 : навч. посіб. Вінниця : ВНТУ, 2017. 106 с.
13. Кушлик-Дивульська О. І., Кушлик Б.Р. Основи теорії прийняття рішень: навч. посіб. Київ : НТУУ «КПІ», 2014. 94с.
14. Смирнова Г.І. Розробка тезаурусу педагогічних вимірів Г.Раша: навч. посіб. Київ, 2005. 64 с.
15. Duolingo. URL: <https://www.duolingo.com/learn> (дата звернення: 25.05.2022).
16. Memrise. URL: <https://www.memrise.com/ru/> (дата звернення: 25.05.2022).
17. Lexicon By Like Thought, LLC. URL: <http://likethought.com/lexicon/> (дата звернення: 25.05.2022).
18. Busuu. URL: <https://www.busuu.com/dashboard#/timeline> (дата звернення: 25.09.2023).
19. Andy English Bot. URL: <https://andychatbot.com/> (дата звернення: 25.09.2023).
20. Герасимчук О. Б., Кирилов М. А. Підсумковий контроль у формі комп'ютерного тестування Спецпроект: аналіз наукових досліджень :

матеріали VI наук.-практ. конф. : наук. пр. з педагогіки : Дніпропетровськ. 2011. С. 92–96.

21. Мокін Б. І., Мокін О. Б. Підвищення ступеня адекватності моделі процесу забування знань. Вісник Вінницького політехнічного інституту. Вінниця, 2013. № 4, С. 116-121.

22. Єфіменко В.І., Касярум О.П., Власенко В.М. Створення критеріально-орієнтованих тестів досягнень особистості та інтерпретація результатів тестування на основі положень Item Response Theory // Вісник Черкаського університету. Педагогічні науки. Черкаси, 2001. Вип. 26. С. 40-46.

23. Основи інформаційних систем: навч. посіб. 2-ге вид, перероб. і доп. / Ситник В.Ф., Писаревська Т. А., Єр'оміна Н. В., Краєва О. С. Київ : КНЕУ, 2011. 420 с.

24. Пікуляк М. В. Методи та інструментальні засоби побудови адаптивної системи дистанційної освіти. Івано-Франківськ : ДВНЗ ПНУ ім. В. Стефаника, 2015. 128 с.

25. Федорук П. І., Масловський С. М. Адаптивна передача знань із використанням багатопараметричної моделі студента // Науковий вісник Чернівецького національного університету. Серія : Комп'ютерні системи та компоненти. – Чернівці : ЧНУ, 2011. Т. 2, вип. 2. С. 91–96.

26. Теорія ймовірностей і математична статистика : конспект лекцій / Колосов А. І., Печеніжський Ю. Є., Станішевський С. О., Якунін А. В. Харків : ХНАМГ, 2011. 128 с.

27. Петрик М.Р. Петрик О.Ю. Моделювання програмного забезпечення : навч.-метод. посіб. Тернопіль : ТНТУ імені Івана Пулюя, 2015. 200 с.

28. Савчук Т.О. Організація баз даних і знань. Частина 2. навч. посіб. Вінниця : ВНТУ, 2016. 86 с.

29. Гайна Г. А. Основи проектування баз даних: навч. посіб. Київ: КНУБА, 2002. 204 с.

30. Доценко С. І. Організація та системи керування базами даних: навч. посіб. Харків : УкрДУЗТ, 2023. 117 с.

31. Добролюбова М. В. Програмування баз даних: конспект лекцій. Київ : КПІ ім. Ігоря Сікорського, 2021. 275 с.
32. Коноваленко І.В., Марущак П.О., Савків В.Б. Програмування мовою С# 7.0 : навч. посіб. Тернопіль : ТНТУ імені Івана Пулюя, 2017. 300 с.
33. Настенко Д.В., Нестерко А. Б. Основи об'єктноорієнтованого програмування на мові С# : навч. посіб. Київ: НТУУ «КПІ», 2016. 76 с.
34. Бивойно П. Г. Об'єктно орієнтоване програмування на Java: конспект лекцій. Чернігів : ЧНТУ, 2019. 136 с.
35. IntelliJ Idea - середовище розробки для Java. URL: <https://ualinux.com/uk/ubuntu-apps-programming/idea-ic> (дата звернення: 25.09.2023).
36. Калініченко А. В. Порівняння інтегрованих середовищ розробки додатків JAVA із відкритим кодом: ECLIPSE та INTELLIJ IDE. Львів, 2013. 82 с.
37. Telegram Bot API. URL: <https://core.telegram.org/bots/api> (дата звернення: 25.09.2023).
38. Viber Java Bot API. URL: <https://developers.viber.com/docs/api/java-bot-api/> (дата звернення: 25.09.2023).
39. Facebook Messenger chatbot URL: <https://developers.facebook.com/blog/post/2016/04/12/bots-for-messenger/> (дата звернення: 25.09.2023).
40. Great Reasons for Using jOOQ URL: <https://www.jooq.org/> (дата звернення: 05.10.2023).
41. Spring Framework. URL: <https://spring.io/projects/spring-framework> (дата звернення: 05.10.2023).
42. Java Telegram Bot Tutorial. URL: <https://monsterdeveloper.gitbook.io/java-telegram-bot-tutorial/> (дата звернення: 05.10.2023).
43. Тестування api Telegram боту URL: <https://coder.org.ua/tools/telegram-chat-id> (дата звернення: 12.10.2023).

44. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

45. Кавецький В. В. Економічне обґрунтування інноваційних рішень: практикум / В. В. Кавецький, В. О. Козловський, І. В. Причепа – Вінниця : ВНТУ, 2016. – 113 с.

Додаток А (обов'язковий)

Протокол перевірки кваліфікаційної роботи на наявність текстових
запозиченьПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬНазва роботи: Інформаційна технологія для розширення словникового
запасу іноземної мови у користувачаТип роботи: _____ магістерська кваліфікаційна робота _____
(БДР, МКР)Підрозділ кафедра комп'ютерних наук, ФІТА
(кафедра, факультет)

Показники звіту подібності Unichesk

Оригінальність _____ 81,3% _____ Схожість _____ 18,7% _____

Аналіз звіту подібності (відмітити потрібне):

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

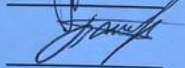
Ознайомлені з повним звітом подібності, який був згенерований системою Unichesk щодо роботи.

Автор роботи



Карабун В.С.

Керівник роботи



Барабан С.В.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку _____



Озеранський В.С.

Додаток Б (обов'язковий)**Лістинг програми****Клас EnglishLearnDkrApplication**

```

package com.example.englishlearndkr;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class EnglishLearnDkrApplication {

    public static void main(String[] args) {
        SpringApplication.run(EnglishLearnDkrApplication.class, args);
    }

}

```

Клас MyUserService

```

package com.example.englishlearndkr.service;

import com.example.englishlearndkr.cache.BotUserCache;
import com.example.englishlearndkr.entity.MyUser;
import com.example.englishlearndkr.entity.UserLevel;
import com.example.englishlearndkr.entity.UserState;
import com.example.englishlearndkr.jooq.generated.tables.records.UsersRecord;
import com.example.englishlearndkr.repository.impl.MyUserRepoImpl;
import lombok.RequiredArgsConstructor;
import org.jooq.tools.StringUtils;
import org.modelmapper.ModelMapper;
import org.springframework.stereotype.Service;
import org.telegram.telegrambots.meta.api.objects.Contact;

import java.util.List;
import java.util.Optional;
import java.util.stream.Collectors;

@Service
@RequiredArgsConstructor
public class MyUserService {

    private final BotUserCache botUserCache;
    private final MyUserRepoImpl myUserRepo;

    private final ModelMapper modelMapper;

    private static final String LINE_SEPARATOR = System.lineSeparator();

    public MyUser authorizeUser(Contact contact, long chatId) {
        Optional<MyUser> optionalMyUser = getFromBDByTelegramId(contact.getUserId());
    }

```

```

if (optionalMyUser.isPresent()) {
    MyUser user = optionalMyUser.get();
    user.setChatId(chatId);
    botUserCache.add(user);
    if (user.isAdmin()) {
        user.setState(UserState.AUTHORIZED_AS_ADMIN_STEP_1);
    } else {
        user.setState(UserState.AUTHORIZED_AS_USER);
    }
    return user;
}
MyUser user = MyUser.builder()
    .chatId(chatId)
    .username(getUsername(contact))
    .telegramId(contact.getUserId())
    .isAdmin(false)
    .state(UserState.AUTHORIZED_AS_USER)
    .build();
int id = myUserRepo.save(convertToUserRecord(user));
user.setId(id);
botUserCache.add(user);
return user;
}

private String getUsername(Contact contact) {
    StringBuilder stringBuilder = new StringBuilder();
    if (!StringUtils.isEmpty(contact.getFirstName())) {
        stringBuilder.append(contact.getFirstName());
    }
    if (!StringUtils.isEmpty(contact.getLastName())) {
        stringBuilder.append(contact.getLastName());
    }
    return stringBuilder.toString();
}

public Optional<MyUser> getFromBDBByTelegramId(long telegramId) {
    Optional<UsersRecord> usersRecord = myUserRepo.getByTelegramId(telegramId);
    if (usersRecord.isEmpty()) {
        return Optional.empty();
    }
    UsersRecord user = usersRecord.get();
    return Optional.of(convertToUser(user));
}

public List<MyUser> getAllUsers() {
    return myUserRepo.getAll().stream()
        .map(this::convertToUser)
        .collect(Collectors.toList());
}

public String getAllUsersForReport() {
    List<MyUser> allUsers = getAllUsers();
    return createReport(allUsers);
}

```

```

private String createReport(List<MyUser> allUsers) {
    List<String> collect = allUsers.stream()
        .map(user -> user.getUsername() + ": " + user.countPersent())
        .collect(Collectors.toList());
    return StringUtils.join(collect, LINE_SEPARATOR);
}
public UserLevel getUserLevel(MyUser userFromCash) {
    if (userFromCash.getNumberOfAttempt() < 30
        || userFromCash.countPersent() < 25) {
        return UserLevel.NEWBIES;
    }
    if (userFromCash.countPersent() < 50) {
        return UserLevel.JUNIOR;
    }
    if (userFromCash.countPersent() < 75) {
        return UserLevel.MIDDLE;
    }
    return UserLevel.ADVANCED;
}
public void update(MyUser userFromCash) {
    myUserRepo.update(convertToUserRecord(userFromCash));
}
private UsersRecord convertToUserRecord(MyUser user) {
    UsersRecord usersRecord = new UsersRecord();
    usersRecord.setTelegramid(user.getTelegramId());
    usersRecord.setUsername(user.getUsername());
    usersRecord.setIsadmin(user.isAdmin() ? "true" : "false");
    usersRecord.setNumberofattempt(user.getNumberOfAttempt());
    usersRecord.setNumberright(user.getNumberRight());
    return usersRecord;
}

private MyUser convertToUser(UsersRecord user) {
    return MyUser.builder()
        .id(user.getId())
        .telegramId(user.getTelegramid())
        .username(user.getUsername())
        .isAdmin("true".equals(user.getIsadmin()))
        .numberOfAttempt(user.getNumberofattempt())
        .numberRight(user.getNumberright())
        .build();
}
}

```

Класс TestService

```

package com.example.englishlearndkr.service;

import com.example.englishlearndkr.cache.*;
import com.example.englishlearndkr.entity.*;
import com.example.englishlearndkr.messagesender.MessageSender;

```

```

import com.example.englishlearndkr.messagesender.MessagesTexts;
import com.example.englishlearndkr.utols.ReplyKeyboardMarkupUtils;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;
import org.springframework.util.CollectionUtils;
import org.telegram.telegrambots.meta.api.methods.send.SendMessage;
import org.telegram.telegrambots.meta.api.objects.Message;
import org.telegram.telegrambots.meta.api.objects.replykeyboard.ReplyKeyboardMarkup;
import org.telegram.telegrambots.meta.api.objects.replykeyboard.buttons.KeyboardRow;

import java.util.*;

import static com.example.englishlearndkr.messagesender.MessagesTexts.MISTAKE;
import static com.example.englishlearndkr.messagesender.MessagesTexts.TEST_RESULT;

@Service
@RequiredArgsConstructor
public class TestService {

    private final MyUserService userService;
    private final WordService wordService;
    private final WordForAskCache wordForAskCache;
    private final AskedWordsCache askedWordsCache;
    private final EngWordsCache engWordsCache;
    private final UkrWordsCache ukrWordsCache;
    private final MessageSender messageSender;
    private final ReplyKeyboardMarkupUtils markupUtils;
    private final TestResultCache testResultCache;

    private static final int LIMIT = 15;

    public void startTest(MyUser userFromCash, Message message) {
        List<Word> words = loadWords(userFromCash);
        wordForAskCache.add(message.getChatId(), words);
        if (CollectionUtils.isEmpty(engWordsCache.getAll())) {
            engWordsCache.addAll(wordService.getAllEngWords());
        }
        if (CollectionUtils.isEmpty(ukrWordsCache.getAll())) {
            ukrWordsCache.addAll(wordService.getAllUkrWords());
        }
        nextTest(userFromCash, message);
        testResultCache.add(message.getChatId(), TestResult.builder().build());
    }

    public List<Word> loadWords(MyUser userFromCash) {
        List<Word> words = new ArrayList<>();
        int offset = 0;
        while (!hasWordsFowAsk(words, userFromCash)) {
            UserLevel userLevel = userService.getUserLevel(userFromCash);
            words = wordService.getWordsForLevel(userLevel, LIMIT, offset);
            offset = offset + LIMIT;
        }
    }

```

```

    return words;
}

private boolean hasWordsFowAsk(List<Word> words, MyUser userFromCash) {
    if (CollectionUtils.isEmpty(words)) {
        return false;
    }
    List<Word> askedWords = askedWordsCache.findBy(userFromCash.getChatId());
    words.stream()
        .filter(w -> askedWords.contains(w))
        .forEach(w -> words.remove(w));
    return !CollectionUtils.isEmpty(words);
}

public void nextTest(MyUser userFromCash, Message message) {
    if (!hasWordsFowAsk(wordForAskCache.findBy(userFromCash.getChatId()),
userFromCash)) {
        List<Word> words = loadWords(userFromCash);
        wordForAskCache.add(userFromCash.getChatId(), words);
    }
    Word word = wordForAskCache.findBy(message.getChatId()).get(0);
    if (UserState.ENGLISH_PICK_TEST.equals(userFromCash.getState())) {
        createPickQuestion(message, word.getEng(), word.getUkr());
        return;
    }
    if (UserState.ENGLISH_WRITE_TEST.equals(userFromCash.getState())) {
        createWriteQuestion(message, word.getEng());
        return;
    }
    if (UserState.UKRAINE_PICK_TEST.equals(userFromCash.getState())) {
        createPickQuestion(message, word.getUkr(), word.getEng());
        return;
    }
    if (UserState.UKRAINE_WRITE_TEST.equals(userFromCash.getState())) {
        createWriteQuestion(message, word.getUkr());
        return;
    }
}

private void createPickQuestion(Message message, String askWord, String rightAnsWord) {
    ReplyKeyboardMarkup markup = new ReplyKeyboardMarkup();
    Set<Integer> numbers = createRandomNumbers(3, engWordsCache.getSize());
    List<String> answers = createAnswers(engWordsCache.getAll(), numbers, rightAnsWord);
    KeyboardRow row = markupUtils.getKeyboardButtons(answers.get(0), answers.get(1));
    KeyboardRow row2 = markupUtils.getKeyboardButtons(answers.get(2), answers.get(3));
    KeyboardRow row3 = markupUtils.createEndTestRow();
    markup.setKeyboard(List.of(row, row2, row3));
    markup.setResizeKeyboard(true);
    SendMessage sendMessage = SendMessage.builder()
        .text(askWord)
        .chatId(String.valueOf(message.getChatId()))
        .replyMarkup(markup)
        .build();
}

```

```

    messageSender.sendMessage(sendMessage);
}
private void createWriteQuestion(Message message, String question) {
    ReplyKeyboardMarkup markup = new ReplyKeyboardMarkup();
    KeyboardRow row3 = markupUtils.createEndTestRow();
    markup.setKeyboard(List.of(row3));
    markup.setResizeKeyboard(true);
    SendMessage sendMessage = SendMessage.builder()
        .text(question)
        .chatId(String.valueOf(message.getChatId()))
        .replyMarkup(markup)
        .build();
    messageSender.sendMessage(sendMessage);
}
private List<String> createAnswers(List<String> allWords, Set<Integer> numbers, String
rightAns) {
    List<String> result = new ArrayList<>();
    numbers.forEach(n -> result.add(allWords.get(n)));
    Random r = new Random();
    int rightAnsWord = r.nextInt(4);
    result.add(rightAnsWord, rightAns);
    return result;
}
private Set<Integer> createRandomNumbers(int count, int size) {
    Random r = new Random();
    Set<Integer> result = new HashSet<>();
    while (result.size() != count) {
        result.add(r.nextInt(size));
    }
    return result;
}
public void processEnglishTestResult(MyUser userFromCash, Message message) {
    Word word = wordForAskCache.findBy(userFromCash.getChatId()).get(0);
    if (message.getText().equalsIgnoreCase(word.getUkr())) {
        processRightAnswer(userFromCash, message, word);
    } else {
        processWrongAnswer(userFromCash, message, word, word.getUkr());
    }
    askedWordsCache.add(message.getChatId(), word);
    wordForAskCache.remove(0);
}
public void processUkraineTestResult(MyUser userFromCash, Message message) {
    Word word = wordForAskCache.findBy(userFromCash.getChatId()).get(0);
    if (message.getText().equalsIgnoreCase(word.getEng())) {
        processRightAnswer(userFromCash, message, word);
    } else {
        processWrongAnswer(userFromCash, message, word, word.getEng());
    }
    askedWordsCache.add(message.getChatId(), word);
    wordForAskCache.remove(0);
}
}

```

```

private void processWrongAnswer(MyUser userFromCash, Message message, Word word,
String rightAns) {
    userFromCash.addWrongAnswer();
    word.addWrongAnswer();
    userService.update(userFromCash);
    wordService.update(word);
    testResultCache.findBy(message.getChatId()).addWrongResult();
    SendMessage sendMessage = SendMessage.builder()
        .text(String.format(MISTAKE, rightAns))
        .chatId(String.valueOf(message.getChatId()))
        .build();
    messageSender.sendMessage(sendMessage);
}

private void processRightAnswer(MyUser userFromCash, Message message, Word word) {
    userFromCash.addRightAnswer();
    word.addRightAnswer();
    testResultCache.findBy(message.getChatId()).addRightResult();
    userService.update(userFromCash);
    wordService.update(word);
    SendMessage sendMessage = SendMessage.builder()
        .text(getCongratsMessage())
        .chatId(String.valueOf(message.getChatId()))
        .build();
    messageSender.sendMessage(sendMessage);
}

private String getCongratsMessage() {
    Random r = new Random();
    int number = r.nextInt(MessagesTexts.CONGRATULATIONS.size());
    return MessagesTexts.CONGRATULATIONS.get(number);
}

public void showResultsOfTest(Long chatId) {
    TestResult testResult = testResultCache.findBy(chatId);
    SendMessage sendMessage = SendMessage.builder()
        .text(String.format(TEST_RESULT, testResult.getWordsCount(),
testResult.getRightCount()))
        .chatId(String.valueOf(chatId))
        .build();
    messageSender.sendMessage(sendMessage);
}

public void cleanTestCache(Long chatId) {
    wordForAskCache.remove(chatId);
    askedWordsCache.remove(chatId);
    testResultCache.remove(chatId);
}
}

```

Клас WordService

```
package com.example.englishlearndkr.service;
```

```

import com.example.englishlearndkr.entity.UserLevel;
import com.example.englishlearndkr.entity.Word;
import com.example.englishlearndkr.jooq.generated.tables.records.WordsRecord;
import com.example.englishlearndkr.repository.impl.WordRepository;
import lombok.RequiredArgsConstructor;
import org.jooq.tools.StringUtils;
import org.modelmapper.ModelMapper;
import org.springframework.stereotype.Service;

import java.nio.charset.StandardCharsets;
import java.util.List;
import java.util.stream.Collectors;

@Service
@RequiredArgsConstructor
public class WordService {

    private final WordRepository wordRepository;
    private final ModelMapper modelMapper;

    public void createNewWord(String text) {
        String newText = new String(text.getBytes(StandardCharsets.UTF_8),
StandardCharsets.UTF_8);
        Word word = parsMessageTextToWord(text);
        if (word != null) {
            wordRepository.save(modelMapper.map(word, WordsRecord.class));
        }
    }

    private Word parsMessageTextToWord(String text) {
        String[] splited = text.split(";");
        if (splited.length != 2
            || StringUtils.isEmpty(splited[0])
            || StringUtils.isEmpty(splited[1])) {
            return null;
        }
        return Word.builder()
            .eng(splited[0].trim())
            .ukr(splited[1].trim())
            .numberOfAttempt(0)
            .numberRight(0)
            .build();
    }

    public void update(Word word) {
        wordRepository.update(modelMapper.map(word, WordsRecord.class));
    }

    public List<String> getAllEngWords() {
        return wordRepository.getAllEngWords();
    }

    public List<String> getAllUkrWords() {

```



```

    return wordRepository.getAllUkrWords();
}
public List<Word> getWordsForLevel(UserLevel userLevel, int limit, int offset) {
    if (UserLevel.NEWBIES == userLevel) {
        return wordRepository.getWordsForNuwbies(limit, offset)
            .stream()
            .map(w -> modelMapper.map(w, Word.class))
            .collect(Collectors.toList());
    }
    if (UserLevel.JUNIOR == userLevel) {
        return wordRepository.getWordsForJunior(limit, offset)
            .stream()
            .map(w -> modelMapper.map(w, Word.class))
            .collect(Collectors.toList());
    }
    if (UserLevel.MIDDLE == userLevel) {
        return wordRepository.getWordsForMiddle(limit, offset)
            .stream()
            .map(w -> modelMapper.map(w, Word.class))
            .collect(Collectors.toList());
    }
    return wordRepository.getWordsForAdvanced(limit, offset)
        .stream()
        .map(w -> modelMapper.map(w, Word.class))
        .collect(Collectors.toList());
}
}
}

```

Клас MessageSender

```

package com.example.englishlearndkr.messagesender;

import org.telegram.telegrambots.meta.api.methods.send.SendMessage;
import org.telegram.telegrambots.meta.api.objects.Message;

public interface MessageSender {

    void sendMessage(SendMessage sendMessage);
}

```

Клас MessageSenderImpl

```

package com.example.englishlearndkr.messagesender;

import com.example.englishlearndkr.config.TelegramBot;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.telegram.telegrambots.meta.api.methods.send.SendMessage;
import org.telegram.telegrambots.meta.exceptions.TelegramApiException;

@Service
public class MessageSenderImpl implements MessageSender{

```

```

private TelegramBot telegramBot;

@Autowired
public void setTelegramBot(TelegramBot telegramBot) {
    this.telegramBot = telegramBot;
}

@Override
public void sendMessage(SendMessage sendMessage) {
    try {
        telegramBot.execute(sendMessage);
    } catch (TelegramApiException e) {
        throw new RuntimeException(e);
    }
}
}

```

Клас MessagesTexts

```

package com.example.englishlearndkr.messagesender;

import java.util.List;

public class MessagesTexts {
    public static final String LOG_OUT_BTN = "/Вийти";
    public static final String END_EX = "/Закінчити";
    public static final String STATISTICS = "Переглянути статистику";
    public static final String NEW_WORDS = "Додати нове слово";
    public static final String PICK_FLOW = "Оберіть режим роботи";
    public static final String AUTH_BTN = "Авторизуватись";
    public static final String GRITTING = "Привіт, це найкращий бот для вивчення мови, для
прододовження потрібно авторизуватись";
    public static final String USER_FLOW_1 = "user flow 1";
    public static final String USER_FLOW_2 = "user flow 2";
    public static final String USER_FLOW_3 = "user flow 3";
    public static final String USER_FLOW_4 = "user flow 4";
    public static final String RULES_FOR_NEW_WORDS = "Введіть слово в формати 'english
word; українське слово";
    public static final String YOU_ARE_ADMIN = "Вітаємо! У вас є право адміністратора,
оберіть режим роботи";
    public static final String ADMIN_FLOW = "Продовжити як адмін";
    public static final String USER_FLOW = "Продовжити як користувач";
    public static final List<String> CONGRATULATIONS = List.of("Все вірно", "Чудово",
"Ти молодець", "Ще трішки і будеш поліглотом",
"Добре", "Саме так", "Відмінно");
    public static final String MISTAKE = "На жаль ви помилились, правильна відповідь '%s";
    public static final String TEST_RESULT = "Тест завершено, кількість запитань - %s,
правильних відповідей - %s";
}

```

Класс MyUserRepoImpl

```

package com.example.englishlearndkr.repository.impl;

import com.example.englishlearndkr.entity.UserState;
import com.example.englishlearndkr.jooq.generated.tables.records.UsersRecord;
import lombok.RequiredArgsConstructor;
import org.jooq.DSLContext;
import org.springframework.stereotype.Repository;

import java.util.List;
import java.util.Optional;

import static com.example.englishlearndkr.jooq.generated.tables.Users.USERS;

@Repository
@RequiredArgsConstructor
public class MyUserRepoImpl {

    private final DSLContext context;

    public int save(UsersRecord usersRecord) {
        UsersRecord record = context.insertInto(USERS)
            .set(usersRecord)
            .returning()
            .fetchOne();
        return record.getId();
    }

    public Optional<UsersRecord> getByTelegramId(long telegramId) {
        return Optional.ofNullable(
            context.selectFrom(USERS)
                .where(USERS.TELEGRAMID.eq(telegramId))
                .fetchAny());
    }

    public List<UsersRecord> getAll() {
        return context.selectFrom(USERS)
            .fetch();
    }

    public void update(UsersRecord userRecord) {
        context.update(USERS)
            .set(USERS.NUMBEROFATTEMPT, userRecord.getNumberOfattempt())
            .set(USERS.NUMBERRIGHT, userRecord.getNumerright())
            .where(USERS.ID.eq(userRecord.getId()))
            .execute();
    }
}

```

Класс WordRepository

```

package com.example.englishlearndkr.repository.impl;

```

```

import com.example.englishlearndkr.jooq.generated.tables.records.WordsRecord;
import lombok.RequiredArgsConstructor;
import org.jooq.DSLContext;
import org.springframework.stereotype.Repository;

import java.util.List;

import static com.example.englishlearndkr.jooq.generated.Tables.WORDS;

//import static com.example.englishlearndkr.jooq.generated.tables;

@Repository
@RequiredArgsConstructor
public class WordRepository {

    private final DSLContext context;

    public int save(WordsRecord map) {
        WordsRecord wordsRecord = context.insertInto(WORDS)
            .set(map)
            .returning()
            .fetchOne();
        return wordsRecord.getId();
    }

    public List<String> getAllEngWords() {
        return context.select(WORDS.ENG)
            .from(WORDS)
            .fetch(WORDS.ENG);
    }

    public List<String> getAllUkrWords() {
        return context.select(WORDS.UKR)
            .from(WORDS)
            .fetch(WORDS.UKR);
    }

    public List<WordsRecord> getWordsForMiddle(int limit, int offset) {
        return context.selectFrom(WORDS)
            .where(WORDS.NUMBEROFATTEMPT.lessThan(30))

            .or(WORDS.NUMBERRIGHT.mul(100).div(WORDS.NUMBEROFATTEMPT).between(15,
60))
            .limit(limit)
            .offset(offset)
            .fetch();
    }

    public List<WordsRecord> getWordsForNuwbies(int limit, int offset) {
        return context.selectFrom(WORDS)

```

```

        .where(WORDS.NUMBEROFATTEMPT.lessThan(30))

.or(WORDS.NUMBERRIGHT.mul(100).div(WORDS.NUMBEROFATTEMPT).between(55,
100))
        .limit(limit)
        .offset(offset)
        .fetch();
    }

    public List<WordsRecord> getWordsForJunior(int limit, int offset) {
        return context.selectFrom(WORDS)
            .where(WORDS.NUMBEROFATTEMPT.lessThan(30))

.or(WORDS.NUMBERRIGHT.mul(100).div(WORDS.NUMBEROFATTEMPT).between(40,
85))
        .limit(limit)
        .offset(offset)
        .fetch();
    }

    public List<WordsRecord> getWordsForAdvanced(int limit, int offset) {
        return context.selectFrom(WORDS)
            .where(WORDS.NUMBEROFATTEMPT.lessThan(30))

.or(WORDS.NUMBERRIGHT.mul(100).div(WORDS.NUMBEROFATTEMPT).between(0,
45))
        .limit(limit)
        .offset(offset)
        .fetch();
    }

    public void update(WordsRecord wordsRecord) {
        context.update(WORDS)
            .set(WORDS.NUMBEROFATTEMPT, wordsRecord.getNumberofattempt())
            .set(WORDS.NUMBERRIGHT, wordsRecord.getNumberright())
            .where(WORDS.ID.eq(wordsRecord.getId()))
            .execute();
    }
}

```

Клас MessageHandler

```

package com.example.englishlearndkr.handler;

import com.example.englishlearndkr.cache.BotUserCache;
import com.example.englishlearndkr.entity.MyUser;
import com.example.englishlearndkr.entity.UserState;
import com.example.englishlearndkr.messagesender.MessageSender;
import com.example.englishlearndkr.messagesender.MessagesTexts;
import com.example.englishlearndkr.service.MyUserService;
import com.example.englishlearndkr.service.TestService;
import com.example.englishlearndkr.service.WordService;

```

```

import com.example.englishlearndkr.utols.ReplyKeyboardMarkupUtils;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Component;
import org.telegram.telegrambots.meta.api.methods.send.SendMessage;
import org.telegram.telegrambots.meta.api.objects.Message;
import org.telegram.telegrambots.meta.api.objects.replykeyboard.ReplyKeyboardMarkup;
import org.telegram.telegrambots.meta.api.objects.replykeyboard.buttons.KeyboardButton;
import org.telegram.telegrambots.meta.api.objects.replykeyboard.buttons.KeyboardRow;

import java.util.List;

import static com.example.englishlearndkr.messagesender.MessagesTexts.*;
import static java.util.Objects.isNull;

@Component
@RequiredArgsConstructor
public class MessageHandler implements Handler<Message> {

    private final BotUserCache userCache;
    private final MyUserService userService;
    private final WordService wordService;
    private final MessageSender messageSender;
    private final TestService testService;
    private final ReplyKeyboardMarkupUtils markupUtils;

    @Override
    public void apply(Message message) {
        Long chatId = message.getChatId();
        MyUser userFromCash = userCache.findBy(chatId);

        //log out
        if (LOG_OUT_BTN.equals(message.getText())) {
            userFromCash.setState(UserState.NON_AUTHORIZED);
            createBaseMessageForAuthorization(chatId);
            return;
        }

        //user is not authorized
        if (isNull(userCache.findBy(chatId))
            || UserState.NON_AUTHORIZED.equals(userCache.findBy(chatId).getState())) {
            //authorization
            if (isNull(message.getContact())) {
                createBaseMessageForAuthorization(chatId);
                return;
            }
            MyUser myUser = userService.authorizeUser(message.getContact(), chatId);
            createMeinMenuMessage(myUser);
            return;
        }

        //user can be admin
        if (UserState.AUTHORIZED_AS_ADMIN_STEP_1.equals(userFromCash.getState())) {

```

```

String text = message.getText();
if (USER_FLOW.equals(text)) {
    createMeinMenuForUser(userFromCash);
    userFromCash.setState(UserState.AUTHORIZED_AS_USER);
    return;
}
createMainMenuForAdminStep2(userFromCash);
userFromCash.setState(UserState.AUTHORIZED_AS_ADMIN_STEP_2);
return;
}

//authorized as admin
if (UserState.AUTHORIZED_AS_ADMIN_STEP_2.equals(userFromCash.getState())) {
    String text = message.getText();
    if (MessagesTexts.NEW_WORDS.equals(text)) {
        startCreatingNewWords(userFromCash);
        userFromCash.setState(UserState.ADDING_NEW_WORDS);
        return;
    }
    writeStatistics(userFromCash);
    userFromCash.setState(UserState.RESULTS_CHECK);
    return;
}

//finish test
if (MessagesTexts.END_EX.equals(message.getText())){
    testService.showResultsOfTest(message.getChatId());
    testService.cleanTestCache(message.getChatId());
    createMeinMenuForUser(userFromCash);
}

//adding newWords
if (UserState.ADDING_NEW_WORDS.equals(userFromCash.getState())) {
    wordService.createNewWord(message.getText());
    return;
}

//authorizedAsUser
if (UserState.AUTHORIZED_AS_USER.equals(userFromCash.getState())) {
    String text = message.getText();
    if (USER_FLOW_1.equals(text)
        || USER_FLOW_2.equals(text)
        || USER_FLOW_3.equals(text)
        || USER_FLOW_4.equals(text)) {
        setStatus(userFromCash, text);
        testService.startTest(userFromCash, message);
        return;
    }
    createMeinMenuForUser(userFromCash);
    return;
}
}

```

```

//english testing
if (UserState.ENGLISH_PICK_TEST.equals(userFromCash.getState())
    || UserState.ENGLISH_WRITE_TEST.equals(userFromCash.getState())){
    testService.processEnglishTestResult(userFromCash, message);
    testService.nextTest(userFromCash, message);
    return;
}
//ukranian testing
if (UserState.ENGLISH_PICK_TEST.equals(userFromCash.getState())
    || UserState.ENGLISH_WRITE_TEST.equals(userFromCash.getState())){
    testService.processUkraineTestResult(userFromCash, message);
    testService.nextTest(userFromCash, message);
    return;
}
}

private void createMeinMenuMessage(MyUser myUser) {
    if (myUser.isAdmin()) {
        createMainMenuForAdminStep1(myUser);
        myUser.setState(UserState.AUTHORIZED_AS_ADMIN_STEP_1);
        return;
    }
    createMeinMenuForUser(myUser);
    myUser.setState(UserState.AUTHORIZED_AS_USER);
}

private void createMainMenuForAdminStep1(MyUser myUser) {
    ReplyKeyboardMarkup markup = new ReplyKeyboardMarkup();
    KeyboardRow row = markupUtils.getKeyboardButtons(USER_FLOW);
    KeyboardRow row2 = markupUtils.getKeyboardButtons(ADMIN_FLOW);
    markup.setKeyboard(List.of(row, row2, markupUtils.createLogOutRow()));
    markup.setResizeKeyboard(true);
    SendMessage sendMessage = SendMessage.builder()
        .text(YOU_ARE_ADMIN)
        .chatId(String.valueOf(myUser.getChatId()))
        .replyMarkup(markup)
        .build();
    messageSender.sendMessage(sendMessage);
}

private void createMainMenuForAdminStep2(MyUser myUser) {
    ReplyKeyboardMarkup markup = new ReplyKeyboardMarkup();
    KeyboardRow row = markupUtils.getKeyboardButtons(STATISTICS);
    KeyboardRow row2 = markupUtils.getKeyboardButtons(NEW_WORDS);
    markup.setKeyboard(List.of(row, row2, markupUtils.createLogOutRow()));
    markup.setResizeKeyboard(true);
    SendMessage sendMessage = SendMessage.builder()
        .text(PICK_FLOW)
        .chatId(String.valueOf(myUser.getChatId()))
        .replyMarkup(markup)

```



```

        .build();
    messageSender.sendMessage(sendMessage);
}

private void startCreatingNewWords(MyUser myUser) {
    ReplyKeyboardMarkup markup = new ReplyKeyboardMarkup();
    markup.setKeyboard(List.of(markupUtils.createLogOutRow()));
    markup.setResizeKeyboard(true);
    SendMessage sendMessage = SendMessage.builder()
        .text(RULES_FOR_NEW_WORDS)
        .chatId(String.valueOf(myUser.getChatId()))
        .replyMarkup(markup)
        .build();
    messageSender.sendMessage(sendMessage);
}

private void writeStatistics(MyUser myUser) {
    ReplyKeyboardMarkup markup = new ReplyKeyboardMarkup();
    markup.setKeyboard(List.of(markupUtils.createLogOutRow()));
    markup.setResizeKeyboard(true);
    String userReport = userService.getAllUsersForReport();

    SendMessage sendMessage = SendMessage.builder()
        .text(userReport)
        .chatId(String.valueOf(myUser.getChatId()))
        .replyMarkup(markup)
        .build();
    messageSender.sendMessage(sendMessage);
}

private void createMeinMenuForUser(MyUser myUser) {
    ReplyKeyboardMarkup markup = new ReplyKeyboardMarkup();
    KeyboardRow row = markupUtils.getKeyboardButtons(USER_FLOW_1);
    KeyboardRow row2 = markupUtils.getKeyboardButtons(USER_FLOW_2);
    KeyboardRow row3 = markupUtils.getKeyboardButtons(USER_FLOW_3);
    KeyboardRow row4 = markupUtils.getKeyboardButtons(USER_FLOW_4);
    markup.setKeyboard(List.of(row, row2, row3, row4, markupUtils.createLogOutRow()));
    markup.setResizeKeyboard(true);
    SendMessage sendMessage = SendMessage.builder()
        .text(PICK_FLOW)
        .chatId(String.valueOf(myUser.getChatId()))
        .replyMarkup(markup)
        .build();
    messageSender.sendMessage(sendMessage);
}

private void createBaseMessageForAuthorization(Long chatId) {
    ReplyKeyboardMarkup markup = new ReplyKeyboardMarkup();
    KeyboardRow row2 = new KeyboardRow();
    row2.add(KeyboardButton.builder()
        .text(AUTH_BTN)
        .requestContact(true)

```

```

        .build());
markup.setKeyboard(List.of(row2));
markup.setResizeKeyboard(true);
SendMessage sendMessage = SendMessage.builder()
    .text(GRITTING)
    .chatId(String.valueOf(chatId))
    .replyMarkup(markup)
    .build();
messageSender.sendMessage(sendMessage);
}

private void setStatus(Long chatId, UserState state) {
    userCache.findBy(chatId).setState(state);
}

private void setStatus(MyUser user, String text) {
    if (USER_FLOW_1.equals(text)){
        user.setState(UserState.ENGLISH_PICK_TEST);
        return;
    }
    if (USER_FLOW_2.equals(text)){
        user.setState(UserState.ENGLISH_WRITE_TEST);
        return;
    }
    if (USER_FLOW_3.equals(text)){
        user.setState(UserState.UKRAINE_PICK_TEST);
        return;
    }
    if (USER_FLOW_4.equals(text)){
        user.setState(UserState.UKRAINE_WRITE_TEST);
        return;
    }
}
}
}

```

Додаток В (обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ДЛЯ РОЗШИРЕННЯ
СЛОВНИКОВОГО ЗАПАСУ ІНОЗЕМНОЇ МОВИ У
КОРИСТУВАЧА»

Виконала: студентка 2-го курсу,
групи 1КН-22м

Спеціальності 122 – «Комп'ютерні
науки»



Карабун В. С.

(прізвище та ініціали)

Керівник: д.т.н., доц. каф. КН



Барабан С. В.

(прізвище та ініціали)

« 04 » 12 2023 р.

Вінниця ВНТУ - 2023 рік



Рисунок В.1 – Структурна схема інформаційної технології

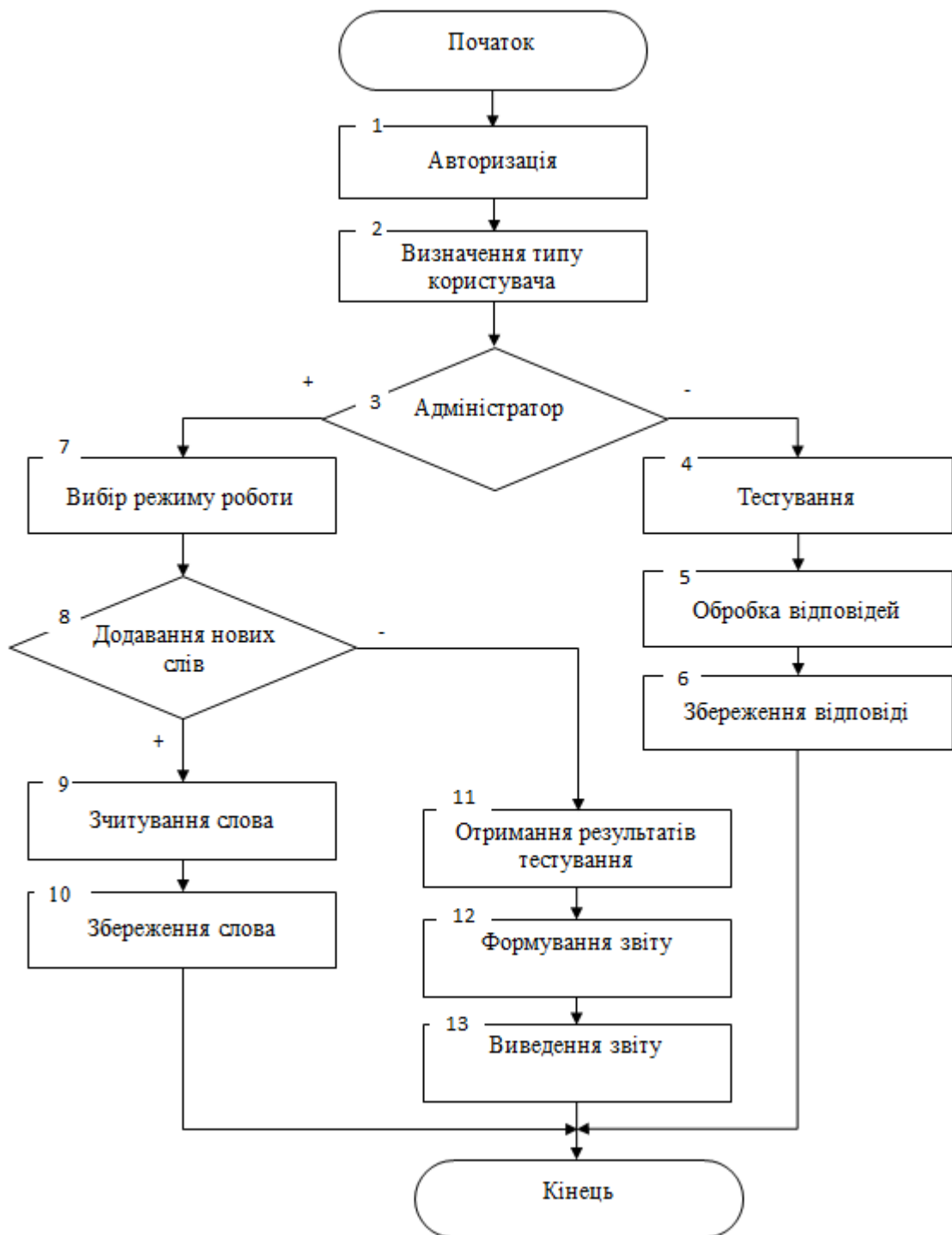


Рисунок В.2 – Схема загального алгоритму функціонування інформаційної технології для розширення словникового запасу іноземної мови у користувача

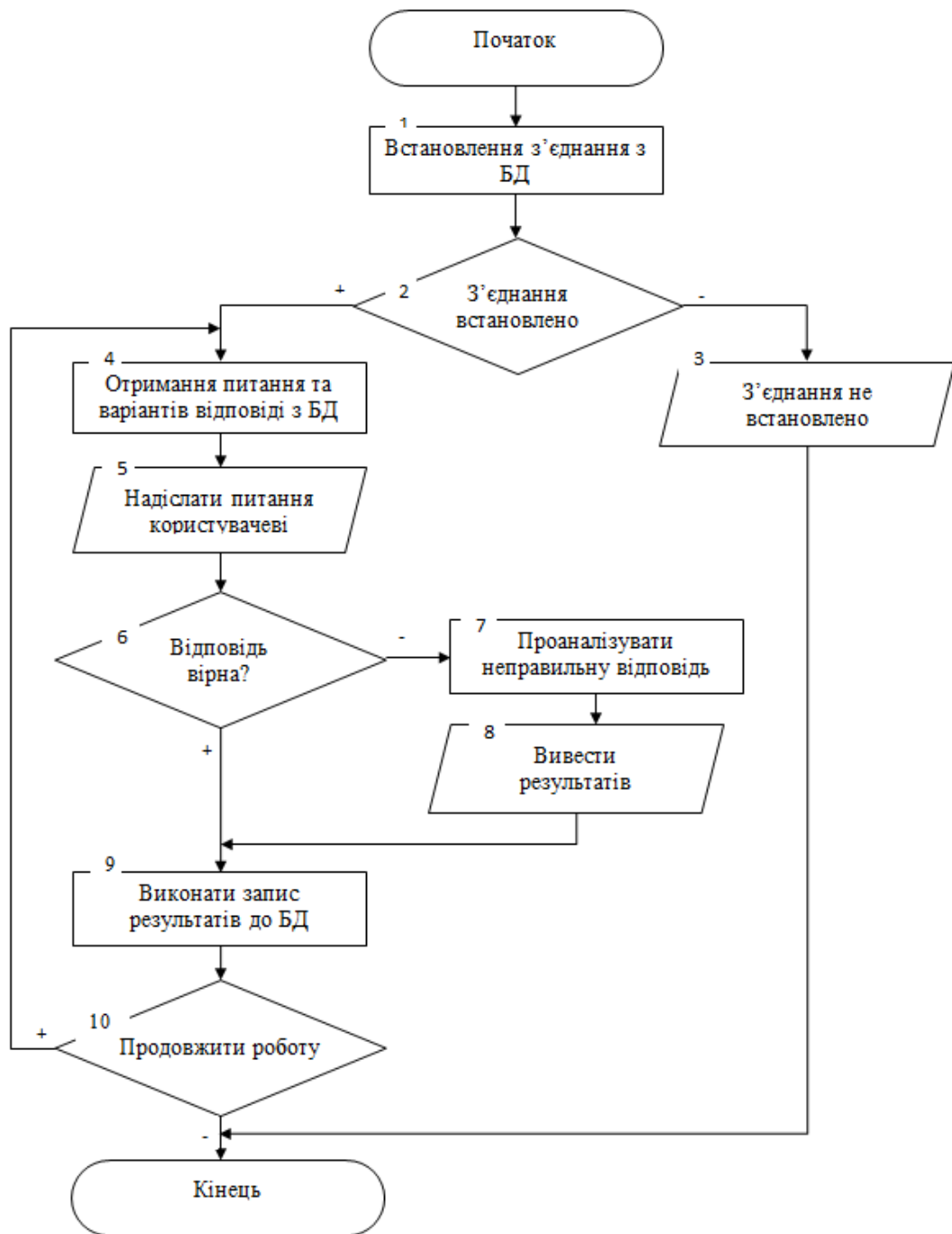


Рисунок В.3 – Алгоритм тестування для розширення словникового запасу іноземної мови у користувача

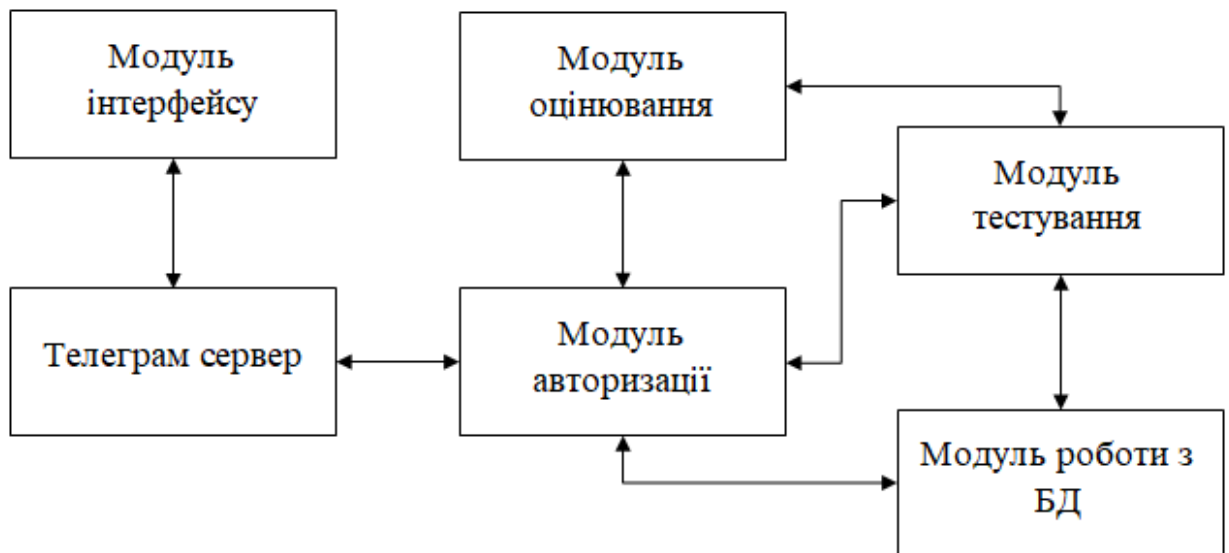


Рисунок В.4 – Загальна структурна схема

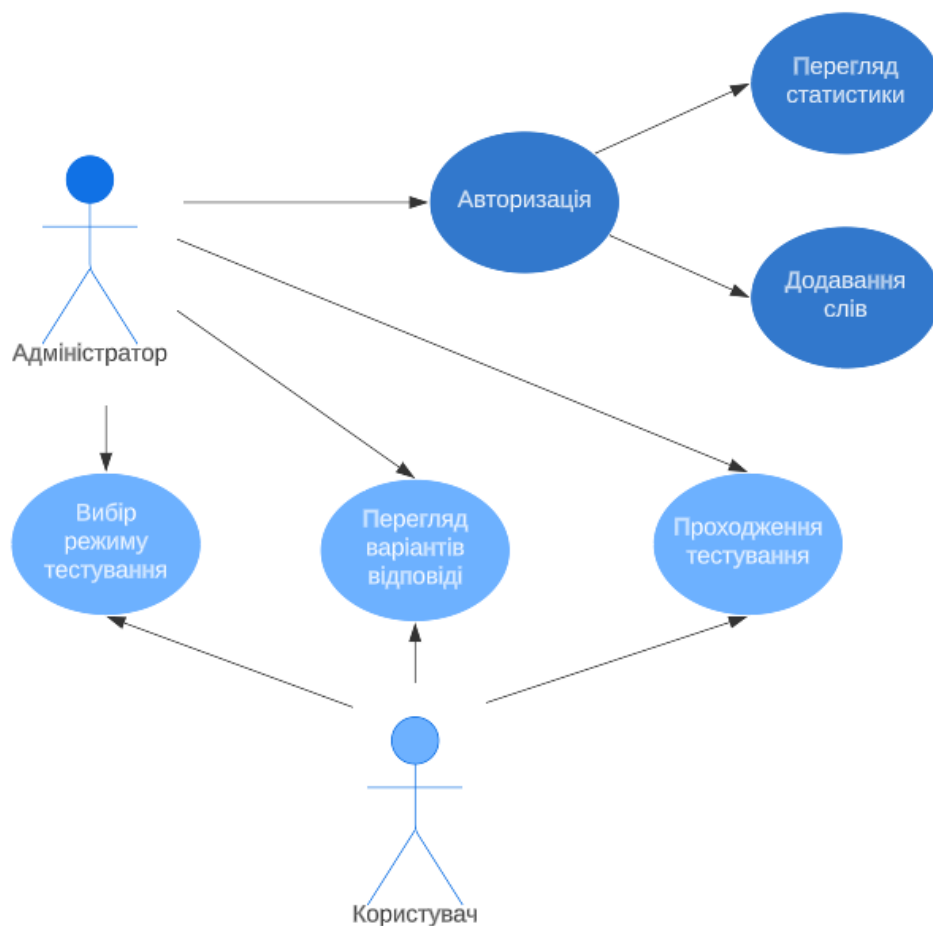


Рисунок В.5 – Діаграма варіантів використання інформаційної технології

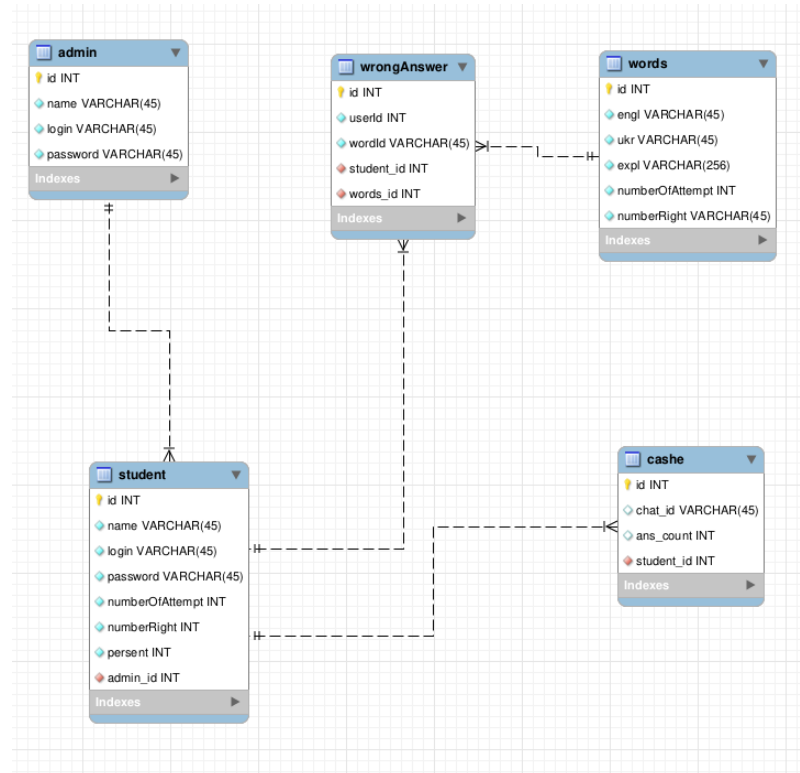


Рисунок В.6 – ER-діаграма інформаційної технології для розширення словникового запасу іноземної мови у користувача

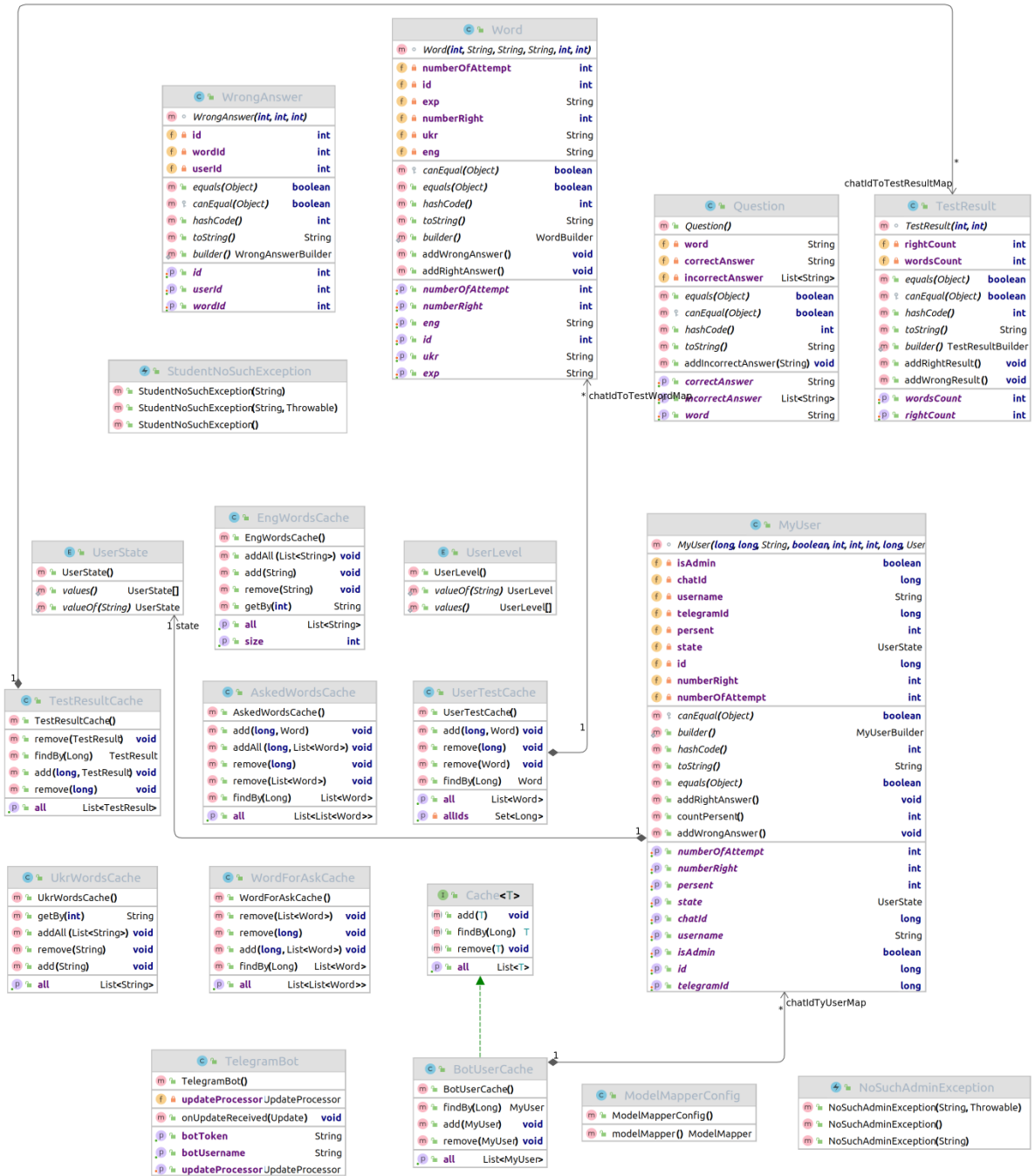


Рисунок В.7 – Фрагмент UML-діаграму класів

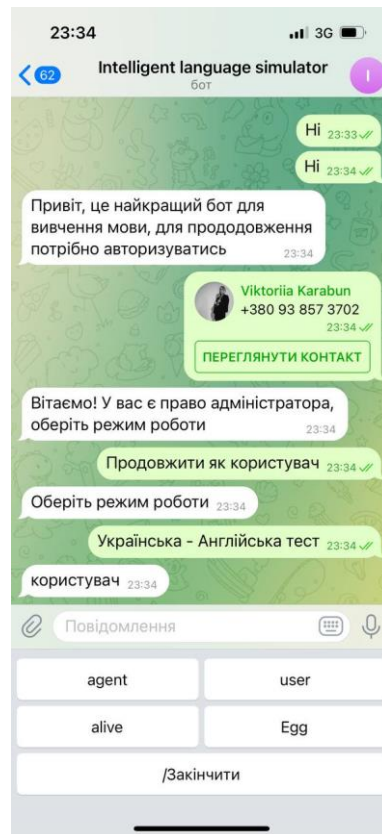


Рисунок В.8 – Приклад роботи програми

Додаток Г (довідниковий)

Інструкція користувача

Після запуску телеграм бота необхідно розпочати діалог (рис. Г.1). Привітавшись з ботом він привітається у відповідь та запропонує пройти авторизацію для подальшої роботи. Ввівши свої дані для авторизації система визначає тип користувача. Якщо користувач має права адміністратора, він матиме опції продовжити роботу як адміністратор, тестувати знання як звичайний користувач або завершити діалог з системою (рис. Г.2).



Рисунок Г.1 – Початок діалогу з ботом

Якщо авторизований адміністратор та бажає продовжити роботу як адміністратор – система пропонує наступні опції «Переглянути статистику», «Додати нове слово» та «Вийти» (рис. Г.3).

Після вибору опції «Переглянути статистику» адміністратору відкривається статистика, що містить результати оцінювання користувачів (рис. Г.4).

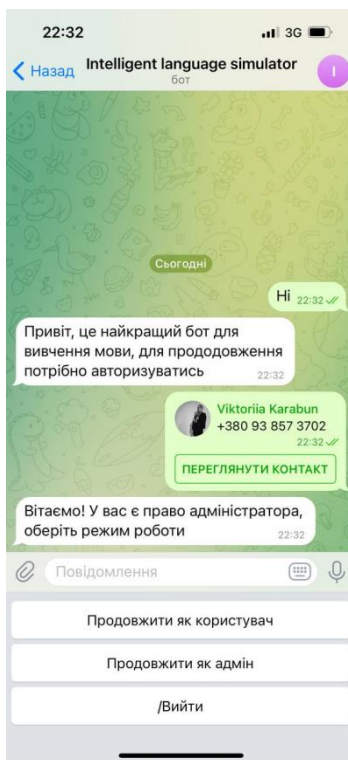


Рисунок Г.2 – Вкладка «Авторизація користувача»

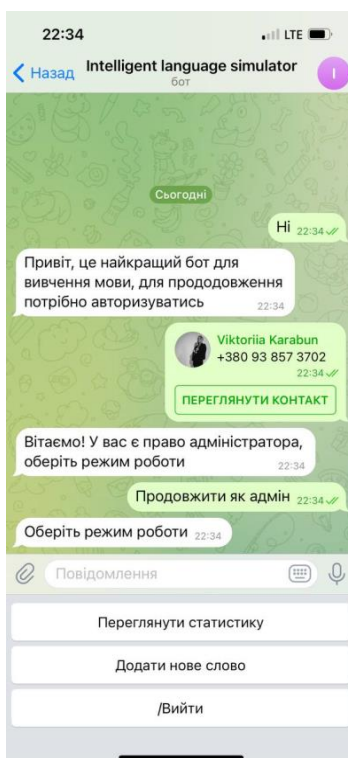


Рисунок Г.3 – Вкладка «Вибір режиму роботи»

Якщо адміністратор бажає додати нове слово, бот надішле повідомлення з інструкцією щодо введення нового слова «Введіть слово в

формати 'english word; українське слово'. Після чого адміністратор вводить повідомлення, наприклад «Egg; яйце», для додавання нового слова (рис. Г.5).

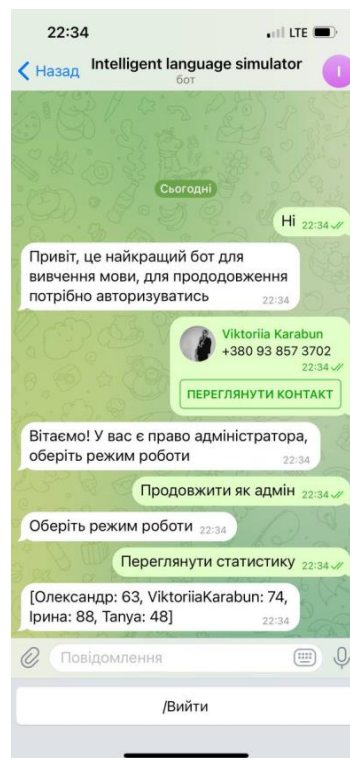


Рисунок Г.4 – Вкладка «Перегляд статистики»



Рисунок Г.5 – Вкладка «Додавання нового слова»

Якщо вибрана опція роботи для звичайного користувача, бот запропонує чотири види тестів (рис. Г.6). Користувач обирає необхідну опцію для проходження тестування. Відповідно до обраного типу тестування відбуваються подальші дії. Так, при виборі тестування «Українська – Англійська тест» бот надсилає повідомлення з українським словом, а користувач отримує чотири варіанти відповідей (рис. Г.7).

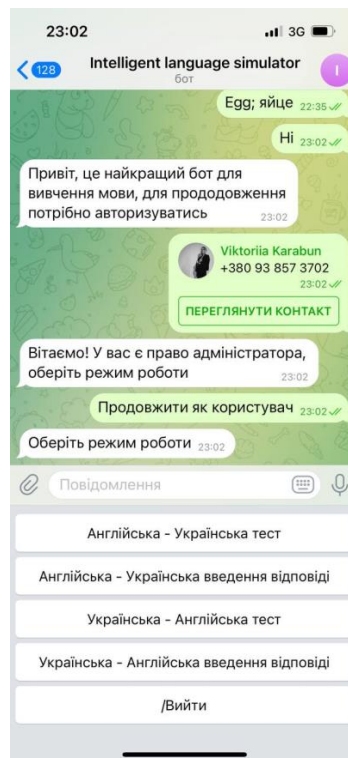


Рисунок Г.6 – Вкладка «Вибір режиму тестування»

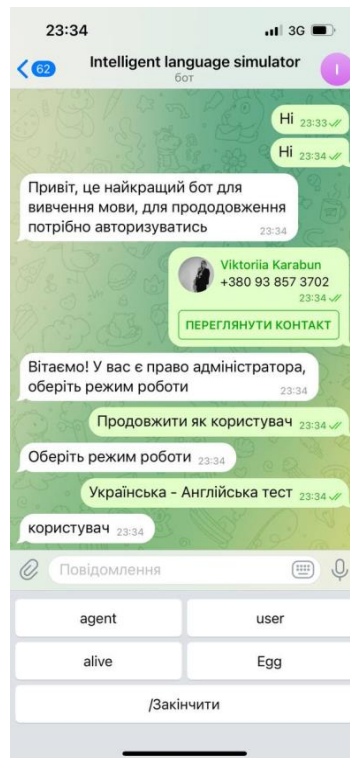


Рисунок Г.7 – Вкладка «Проходження тестування»