

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра комп'ютерних систем та робототехніки

«Затверджую»
в.о. завідуючого кафедри
комп'ютерних систем та робототехніки
_____ к. ф.-м. н., доцент Максим Хруслов
«___» грудня 2024 р.

Пояснювальна записка

до кваліфікаційної роботи
магістра

на тему: «Модель інформаційної навчальної системи для вивчення
англійської мови»

Спеціальність 174 – Автоматизація, комп'ютерно-інтегровані технології та
робототехніка

Галузь знань 17 – Електроніка, автоматизація та електронні комунікації.

Освітня програма «Комп'ютеризовані системи управління та автоматика»

Захищено на засіданні

Екзаменаційної комісії № 44

протокол № __ від __.12.2024 р.

Оцінка _____ / _____

Голова Екзаменаційної комісії

_____ СКОБ Ю. О.

Виконав:

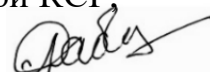
Студент групи КУ– 61

Негібов Даниїл Олегович



Керівник: к.т.н., доцент кафедри КСР,

Лабенко Дмитро Петрович



Рецензент:

к.т.н., доцент; професор кафедри ОВ
ППО СВ Харківського національного
університету Повітряних Сил імені Івана
Кожедуба

Наконечний Олександр Анатолійович



АНОТАЦІЯ

Пояснювальна записка до магістерської атестаційної роботи складається зі вступу, трьох розділів, висновків, списку використаних джерел і двох додатків. Загальний обсяг роботи складає 88 сторінки, із яких 70 сторінок основної частини з 34 рисунками, 3 таблицею, списку використаних джерел із 20 найменувань та трьома додатками на 12 сторінках.

Метою кваліфікаційної роботи є модифікація моделі інформаційної навчальної системи для вивчення англійської мови.

Об'єкт дослідження – процес вивчення іноземної (англійської) мови з використанням новітніх технологій.

Предмет дослідження – методи та засоби вивчення англійської мови за допомогою інформаційної системи.

Проблема, яка вирішується в кваліфікаційній роботі полягає в тому, щоб скориставшись існуючими програмними засобами та підходами до створення програмного коду мінімізувати витрати часу на створення програмного забезпечення, його налагодження та оптимізацію, запобігти дуплікації коду та зменшити його об'єм.

Область застосування – програмний продукт може широко використовуватися при вивченні чи удосконаленні знань з іноземної мови в будь якій сфері діяльності.

Ключові слова: модель, база даних, вивчення англійської мови, інформаційна системи, проектування, KivyMD.

ABSTRACT

The explanatory note to the master's thesis consists of an introduction, three chapters, conclusions, a list of references and two appendices. The total volume of the work is 88 pages, including 70 pages of the main part with 34 figures, 3 tables, a list of references of 20 titles and three appendices on 12 pages.

The purpose of the qualification work is to modify the model of an information learning system for learning English.

The object of research is the process of learning a foreign (English) language using the latest technologies.

The subject of research is methods and means of learning English with the help of an information system.

The problem that is solved in the qualification work is to use existing software tools and approaches to creating program code to minimize the time spent on creating software, debugging and optimizing it, preventing code duplication and reducing its volume.

Scope of application - the software product can be widely used in learning or improving knowledge of a foreign language in any field of activity.

Keywords: model, database, learning English, information system, design, KivyMD.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	6
РОЗДІЛ 1. ОГЛЯД ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Основні поняття предметної області.....	9
1.2 Існуючі аналогічні проекти та їх аналіз.....	11
1.2.1 Babbel – електронна платформа вивчення мов.....	12
1.2.2 Busuu – електронна платформа вивчення англійської мови.....	15
1.2.3 Pimsleur – електронна платформа вивчення англійської мови.....	18
1.3 Аналіз інтерфейсу та функціональності існуючої моделі.....	20
Висновки за розділом 1.....	23
РОЗДІЛ 2. ПРОЕКТУВАННЯ БАЗ ДАНИХ ТА ОНОВЛЕННЯ МОДЕЛІ ...	25
2.1 Дослідження основних принципів роботи моделі	25
2.2 Проектування та розробка баз даних мобільного додатку.....	30
2.3 Розробка нового інтерфейсу.....	35
2.4 Розробка алгоритмів роботи та програмна реалізація моделі.....	40
Висновки за розділом 2.....	60
РОЗДІЛ 3. ТЕСТУВАННЯ МОДЕЛІ.....	62
3.1 Результати, які були отримані при тестуванні моделі та їх аналіз...	62
3.2 Рекомендації щодо використання розробленої моделі.....	70
Висновки за розділом 3.....	71
ВИСНОВКИ.....	72
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	74
ДОДАТКИ.....	77
Додаток А.....	77
Додаток Б.....	79
Додаток В.....	81

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ

MVC – Model-View-Controller

iOS – iPhone operating system

EF – Education First

IT – Information Technology

NLP – Natural Language Processing

MPNet – Multi-Projection Network

ORM – Object-Relational Mapping

ВСТУП

Актуальність роботи. Одним з безперечних ключів до успіху в сучасному світі є вільне володіння другою або третьою мовою. Без сумніву, глобалізація є однією з характерних рис нашого суспільства, що робить мультилінгвістичних працівників просто необхідними. Як суспільство, ми також зараз більш мобільні, ніж будь-коли в нашій історії. Здатність спілкуватися іншою мовою дає нам більше можливостей вибору, коли йдеться про те, де ми хочемо жити або працювати

Глобальна англійська мова є унікальною, нею розмовляють майже два мільярди людей у світі, з яких лише близько 400 мільйонів використовують її як рідну мову. За даними журналу «Ethnologue», англійська посідає перше місце серед найпоширеніших мов у світі, включаючи недержавні та державні мови. Як стверджує провідний світовий фахівець з англійської мови Девід Кристал, NNS (носії мови не з народження) англійської мови зараз переважають над NS (носії мови) у співвідношенні приблизно 4:1 [4].

Англійська є найвпливовішою академічною мовою, а знання ділової англійської є обов'язковим для роботи в глобальному діловому світі. Швидко зростаюча глобалізація означає, що сьогодні англійська мова стала основною мовою бізнесу. Кількість людей, які розмовляють англійською мовою і при цьому не є носіями мови з народження, стрімко зростає з кожним роком.

Варто також зазначити, що більше половини всіх веб-сайтів написані англійською мовою. Підприємці, які хочуть успішно запустити новий бізнес, повинні добре володіти розмовною англійською, а також фразами ділової англійської мови. Вивчення англійської мови зараз є більш важливим, ніж будь-коли; країни з кращими навичками англійської мови не тільки більш конкурентоспроможні, але й більш відкриті для нових бізнес-можливостей.

Згідно міжнародного рейтинга рівня володіння англійською мовою EF EPI Україна зайняла 45 місце з 113 країн, та 31 місце з 34 країн Європи [13].

Скільки балів набрали міста та області України можна подивитись у таблиці нижче.

Таблиця В.1

Рейтинг міст та областей України за знанням англійської мови

Місто	Бали	Область	Бали
Луцьк	548	Тернопільська область	548
Київ	547	Київська область	547
Тернопіль	547	Львівська область	544
Львів	545	Закарпатська область	543
Чернівці	544	Волинська область	541
Івано-Франківськ	537	Чернівецька область	540
Полтава	532	Хмельницька область	538
Харків	531	Чернігівська область	536
Хмельницький	531	Івано-Франківська область	534
Вінниця	530	Харківська область	530
Дніпро	529	Вінницька область	528
Чернігів	529	Полтавська область	526
Запоріжжя	528	Дніпропетровська область	526
Миколаїв	527	Миколаївська область	526
Суми	524	Запорізька область	523
Рівне	522	Рівненська область	520
Житомир	520	Житомирська область	517
Кривий Ріг	517	Київська область	515
Севастополь	516	Кіровоградська область	515
Черкаси	508	Сумська область	513
Херсон	506	Черкаська область	511
Маріуполь	503	Херсонська область	510
Одеса	478	Крим	509
		Донецька область	508
		Одеська область	479
		Луганська область	448

Отримані результати рейтингу EF EPI свідчать про те, що рівень володіння англійською мовою в Україні має значний потенціал для покращення. Це підкреслює необхідність активних заходів для вдосконалення знань мови серед населення. Згідно з висновками глобального мовного дослідження, проведеного організацією Education First (EF), країни з високим рівнем володіння англійською мовою демонструють також сильнішу економіку. Їхні громадяни мають вищий рівень доходу на душу населення та вищу якість життя. Тому інвестування в вивчення англійської мови є важливим кроком для розвитку нашої країни та покращення добробуту кожного українця.

Мета дослідження є модифікація існуючої моделі інформаційної навчальної системи для вивчення англійської мови.

Об'єкт дослідження – процес вивчення іноземної (англійської) мови з використанням новітніх технологій.

Методи дослідження: у ході написання роботи використовувались метод системного підходу та аналізу, порівняльного аналізу та моделювання.

Предмет дослідження – методи та засоби вивчення англійської мови за допомогою інформаційної системи.

Завдання дослідження:

- Аналіз літератури за темою
- Аналіз та порівняння аналогічних існуючих систем
- Дослідження основних принципів роботи моделі
- Проектування баз даних мобільного додатку
- Розробка оновленої структури та алгоритмів роботи
- Проектування та розробка оновленого інтерфейсу
- Програмна реалізація моделі
- Тестування моделі

РОЗДІЛ 1

ОГЛЯД ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основні поняття предметної області

Предметною областю дослідження є процес вивчення англійської мови за допомогою інформаційних навчальних систем. Вона охоплює такі ключові компоненти, як англійська мова та інформаційна система, які забезпечують інтерактивне та ефективне навчання. Ці системи поєднують методи дистанційного навчання, інтерактивні вправи та засоби автоматизованого оцінювання знань.

Згідно рейтинга на 2023 рік англійська мова займає перше місце серед найпоширеніших мов у світі [12]. У таблиці 1.1 можна розглянути загальну популярність мов у світі.

Таблиця 1.1

Кількість носіїв мов у світі

Мова	Кількість людей
Англійська	1,500,000,000
Мандарин	1,100,000,000
Хінді	609,500,000
Іспанська	559,100,000
Французька	309,800,000
Стандартний арабський	274,000,000
Бенгальська	272,800,000
Португальська	263,600,000
Урду	231,700,000
Індонезійська	199,100,000
Німецька	133,200,000

Англійська мова є однією з найпопулярніших мов у світі і має величезне значення в багатьох сферах, таких як бізнес, наука, технології, культура та подорожі.

Багато ІТ-компаній та технологічних стартапів в Україні працюють на глобальних ринках, де англійська є основною мовою спілкування. Також англійська відкриває доступ до найновіших наукових досліджень, технологій та культурних ресурсів, що сприяє розвитку інтелектуального і професійного рівня громадян. Окрім того, англійська допомагає українцям бути більш інтегрованими в глобальну спільноту, зокрема через доступ до новин, медіа та інформаційних ресурсів.

На сьогоднішній день інформаційні технології пронизують усі аспекти нашого життя, суттєво змінюючи як особисте, так і професійне спілкування. Це основа комунікації, технологічного прогресу, інновацій, сталого розвитку та відпочинку. У повсякденному житті ми активно використовуємо інформаційні технології для зв'язку з близькими, ігрових розваг, обміну медіа-контентом, онлайн-покупок та соціальної активності. Ці технології створюють нові можливості для самовираження, взаємодії та організації вільного часу, що веде до формування глобальної спільноти.

З професійної точки зору, інформаційні технології стали невід'ємною частиною бізнес-процесів. Вони охоплюють практично всі галузі: від охорони здоров'я, де електронні медичні картки і телемедицина покращують доступ до лікування, до харчової промисловості, де автоматизація і системи управління запасами забезпечують ефективність. У виробництві інформаційні технології сприяють оптимізації процесів, підвищуючи продуктивність і якість. Сфера продажів, в свою чергу, використовує CRM-системи для кращого розуміння потреб клієнтів та адаптації своїх пропозицій.

Один з важливих напрямків розвитку інформаційних технологій — їх застосування в освітній сфері, зокрема у вивченні іноземних мов. Інформаційні системи для вивчення англійської мови значно еволюціонували завдяки використанню автоматизації та комп'ютерно-інтегрованих

технологій, що дозволяє підвищити ефективність навчального процесу та адаптувати його до потреб кожного користувача. Використання сучасних технологій, таких як штучний інтелект, автоматичне тестування, та інтерактивні елементи робить ці платформи надзвичайно гнучкими та доступними для широкого кола користувачів.

Одним із ключових аспектів розвитку інформаційних систем у галузі вивчення мов є можливість автоматизованого тестування рівня знань користувачів. Використовуючи алгоритми обробки даних та аналізу відповідей, системи можуть точно визначити мовну компетентність користувача та адаптувати навчальні завдання відповідно до його рівня знань. Ця автоматизація допомагає заощадити час та ресурси, які в іншому випадку витрачалися б на індивідуальне тестування викладачами.

Інший важливий елемент інформаційних систем для вивчення англійської мови – це інтеграція мультимедійних інструментів для вдосконалення навичок аудіювання та сприйняття інформації на слух. Використання аудіо- та відеофайлів разом із автоматизованими системами перевірки правильності відповідей дозволяє студентам одночасно розвивати кілька мовних навичок. Крім того, автоматизація у таких системах може забезпечити миттєвий зворотний зв'язок, що є важливою частиною інтерактивного навчання.

Таким чином, інформаційні системи, побудовані на принципах автоматизації та інтегрованих технологій, забезпечують сучасні методи вивчення англійської мови. Вони не лише дозволяють індивідуалізувати навчальний процес, але й роблять його більш інтерактивним та ефективним за рахунок автоматизації тестування, використання мультимедійних матеріалів та інтеграції технологій штучного інтелекту.

1.2 Існуючі аналогічні проекти та їх аналіз

На сучасному ринку існує багато платформ та додатків для вивчення англійської мови, які пропонують різноманітні підходи та технології, що допомагають користувачам оволодіти мовою. У цьому розділі ми детально проаналізуємо кілька найбільш популярних систем, щоб виявити їхні переваги та недоліки.

1.2.1 Babbel – електронна платформа вивчення мов

Babbel, заснований у 2007 році двома друзями, зацікавленими у вивченні іспанської мови, які не могли знайти якісних онлайн-матеріалів, окрім компакт-дисків та підручників, і у результаті він став однією з найвідоміших платформ для вивчення мов на сучасному ринку [16]. З моменту свого запуску він зібрав вражаючі 13 000 годин контенту та понад 50 000 уроків, що охоплюють 14 мов. Серед доступних мов користувачі можуть вивчати датську, голландську, французьку, німецьку, індонезійську, італійську, норвезьку, польську, португальську, іспанську, шведську та турецьку.

Особливістю платформи є можливість безкоштовного, але обмеженого, вивчення німецької, польської та англійської мов на базі української мови. Це робить Babbel доступним ресурсом для тих, хто хоче вдосконалити свої знання в цих мовах, не витрачаючи коштів на навчання.

Після встановлення програми користувачеві пропонується обрати мову інтерфейсу, що дозволяє зручно налаштувати систему під свої потреби. Далі додаток запрошує користувача авторизуватися або зареєструватися, щоб розпочати навчання (Рис.1.1).

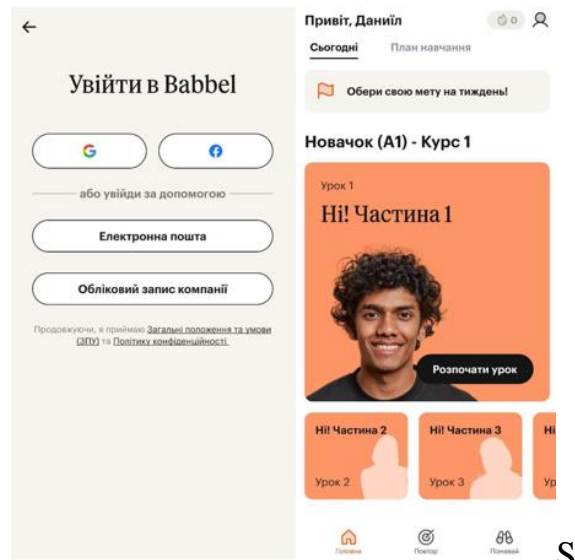


Рисунок 1.1 – Реєстрація та головний екран у Babel

Після входу у систему відображається головний екран на якому можна побачити курс для початкового рівня знань англійської мови (Рис.1.1).

Для україномовних користувачів у Babel передбачено три рівні складності завдань. Кожен рівень має певну кількість курсів: A1 містить 4 курси, A2 – 5 курсів, а B1 – 6 курсів. У межах кожного курсу користувачам пропонуються різні завдання для опанування та закріплення матеріалу. Загалом користувачам пропонують чотири типи завдань приклади яких можна побачити на рисунку 1.2 та 1.3.

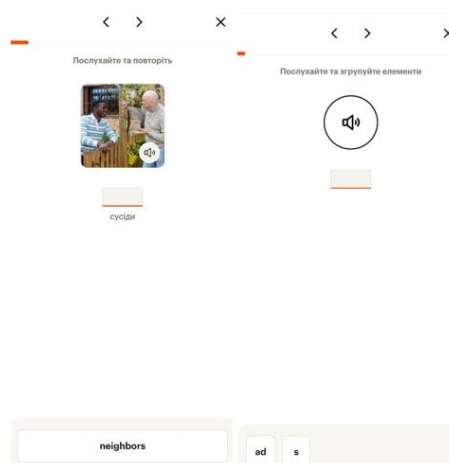


Рисунок 1.2 – Приклад завдань у Babel

На Рисунку 1.2 представлені дві вправи з курсу Babbel. У першій вправі користувач має можливість вивчити англійське слово, ознайомитися з його перекладом і прослухати правильну вимову. Це завдання допомагає покращити словниковий запас і сприйняття мови на слух.

Друга вправа побудована на аудіо: користувачу дається запис вимови слова, а також набори частин слова у вигляді кнопок. Завдання полягає в тому, щоб правильно зібрати слово, натискаючи на відповідні кнопки. Ця вправа тренує здатність до розпізнавання слів і правильного їх складання зі звукових елементів.

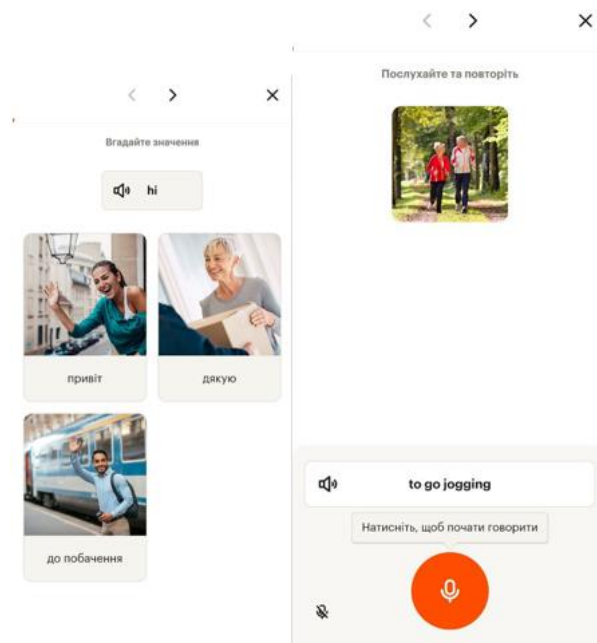


Рисунок 1.3 – Приклад вправ у Babbel

На Рисунку 1.3 зображені дві вправи з курсу Babbel. У першій вправі користувачу пропонується набір картинок з підписами українською мовою і відповідний аудіофайл, який озвучує слово. Завдання полягає в тому, щоб вибрати картинку, яка підписана правильним перекладом, тобто співвідноситься з почутим англійським словом.

У другій вправі користувач має можливість прослухати словосполучення, а потім самостійно повторити його вголос. Це завдання

спрямоване на розвиток вимовних навичок і вміння правильно будувати словосполучення англійською мовою.

Також в Babbel доступні живі онлайн-уроки з носіями мови, але для участі в них потрібна платна підписка. Ці уроки можуть бути як групові, так і приватні уроки з учителем.

Тепер можна виділити переваги та недоліки цієї системи:

Переваги:

- Різні рівні складності завдань
- Різноманітність завдань
- Живі онлайн-уроки

Недоліки:

- Обмежений доступ до безкоштовного контенту
- Відсутність гнучкості у виборі тем
- Недостатня увага до граматики
- Не дуже зручна для продвинутих учнів

1.2.2 Busuu – електронна платформа вивчення англійської мови

Busuu — це одна з найвідоміших платформ для вивчення мов у світі, яка вже привернула увагу понад 100 мільйонів користувачів [3]. Ця система пропонує як безкоштовний, так і платний план, проте варто зазначити, що безкоштовний варіант має суттєві обмеження. Він надає доступ лише до словника та п'яти уроків, що, звичайно, може виявитися недостатнім для серйозного навчання. Тому більшість користувачів обирають план Busuu Premium, ціна якого коливається від \$7 до \$14 на місяць, залежно від обраного терміна підписки — на один місяць, шість місяців або рік.

Після реєстрації на Busuu нові користувачі можуть розпочати курс з нуля або пройти короткий вступний тест, що допоможе визначити їхні початкові

знання з обраної мови. Далі користувачі вибирають бажаний рівень володіння мовою та відповідний графік навчання, що буде зручний для них.

Після цього користувач потрапляє на інформаційну панель Busuu, де всі уроки організовані в лінійній, схожій на шлях, послідовності. На верхній навігаційній панелі розміщені різні інструменти, доступні в рамках підписки, що дозволяє користувачам легко орієнтуватися у своїх навчальних матеріалах (Рис.1.4).

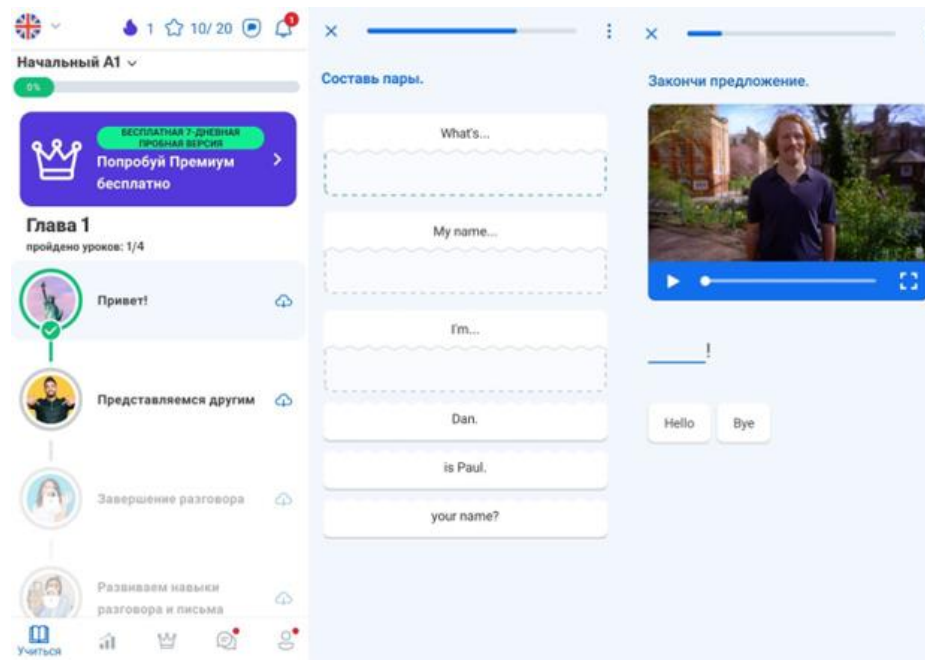


Рисунок 1.4 – Головний екран та приклади вправ в Busuu

З цього екрану ви можете вибрати урок. Кожен урок триває в середньому близько 2-5 хвилин і складається з кількох окремих вправ, які охоплюють одну й ту саму тему або цільові слова. У кожному уроці використовується поєднання зображень та аудіозаписів, а також різноманітні практичні завдання, вправи та вікторини, які охоплюють аудіювання, читання, письмо та говоріння.

Ці вправи включають заповнення пропусків, запитання з декількома варіантами відповідей, підбір пар, складання речень та перегляд відеозаписів розмов носіїв мови. На Рисунку 1.4 можна побачити приклади вправ що

даються користувачу на проходження. У першій вправі користувачу потрібно доповнити діалог по змісту, а у другій потрібно вставити слово що каже людина на відео.

У системі також передбачено вкладку «Повторювати» (Рис.1.5), яка є важливим інструментом для закріплення пройденого матеріалу. Кожні кілька уроків програма пропонує користувачам повторити слова та фрази, які вони вивчили до цього моменту. Ця функція допомагає тримати активний словниковий запас у тонусі.

Busuu, на вкладці “Словник” відображає прогрес графічно, де слова поділяються на три категорії: «слабкі», «середні» та «сильні». Це дозволяє користувачам швидко оцінити, які терміни потребують додаткового повторення. Зі вкладки «Словник» можна перейти безпосередньо до сеансу повторення цих слів, що допомагає закріпити слабкі місця та підтримувати активний словниковий запас на високому рівні.

Крім того, у вкладці “Граматика” доступні граматичні вправи, де можна повторити пройдений матеріал, зокрема артиклі, прислівники, прикметники та інші граматичні категорії. Такий підхід сприяє не тільки запам'ятовуванню окремих слів, але й розумінню правил їх використання в реченнях, що допомагає будувати міцнішу мовну базу для подальшого навчання.

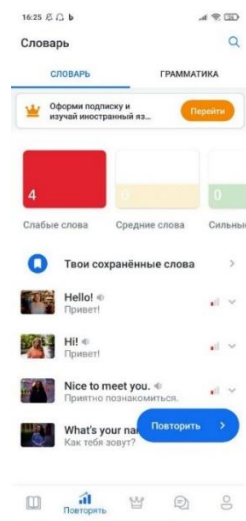


Рисунок 1.5 – Вкладка словник

Так після аналізу платформи можна виділити наступні переваги та недоліки:

Переваги:

- Різноманітні інтерактивні вправи
- Функція повторення
- Граматичні вправи

Недоліки:

- Обмежена кількість вправ у безкоштовній версії
- Неможливість самостійного обрання вправ
- Відсутність мотиваційних механізмів

1.2.3 Pimsleur – електронна платформа вивчення англійської мови

Pimsleur — це платформа, яка спеціалізується на аудіоуроках для вивчення мов і пропонує курси близько 50 мов. Система навчання заснована на повторенні та поступовому збільшенні складності мовного матеріалу. Кожен курс поділений на п'ять рівнів, кожен з яких включає 30 уроків, що дозволяє учням поступово вдосконалювати свої навички.

Основоположник методу Pimsleur, доктор Пол Пімсleur, був видатним лінгвістом і психологом, який розробив аудіоуроки, ґрунтуючись на наукових дослідженнях щодо пам'яті та вивчення мов. Перші курси Pimsleur були запропоновані споживачам у 1980 році [10].

Так після реєстрації користувачу пояснюють що додаток засновано на аудіоуроках тобто користувач слухає розмови, і його просять відповідати на запитання і повторювати фрази вголос.

Користувачі мають можливість безкоштовно пройти перший урок для будь-якої мови, однак подальші матеріали доступні лише за умови оформлення платної підписки. На початку інструкції в уроках надаються рідною мовою користувача, але з поступовим прогресом інструкції починають

подаватися мовою, яку користувач вивчає. Це створює занурення у мовне середовище та сприяє ефективному засвоєнню нової мови.

Основними уроками в системі є аудіоуроки, тобто це 30-хвилинні сесії, в яких ви слухаєте і повторюєте фрази за диктором. Також у Pimsleur є вправи де програма надає фрази, які потрібно повторити вголос, після чого користувач записує свій голос, натиснувши на кнопку запису. А також є вправа де користувачу подається фраза рідною мовою, і завдання полягає в тому, щоб вибрати правильний варіант перекладу цієї фрази мовою, яку користувач вивчає. Це допомагає закріпити нові слова та фрази через швидке порівняння, тренуючи розуміння та переклад між двома мовами. Приклади уроків у Pimsleur можна побачити на Рисунку 1.6.

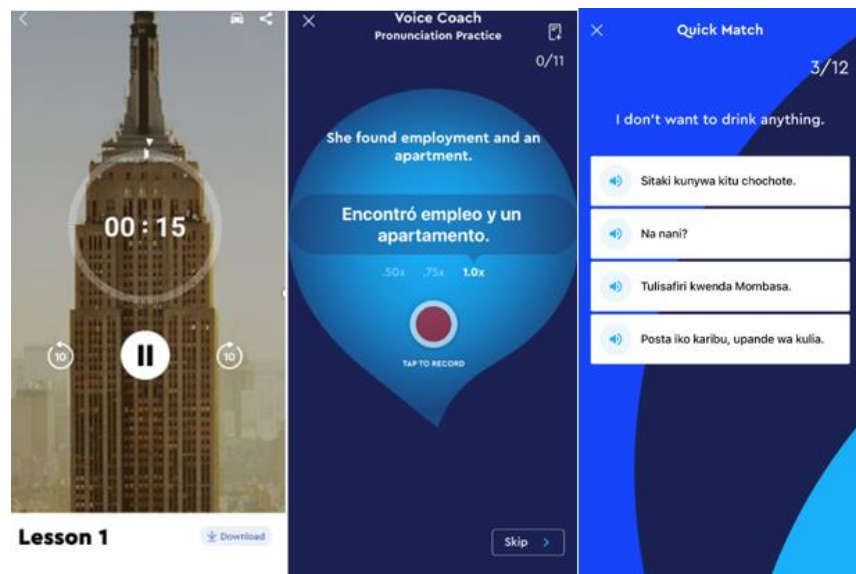


Рисунок 1.6 – Приклади вправ у Pimsleur

Тож тепер можна виділити наступні переваги та недоліки:

Переваги:

- Розвиток розмовних навичок
- Науково обґрунтований метод навчання

Недоліки:

- Зосереджується лише на говорінні та аудіюванні

- Часто буває нудним і повторюваним
- Не призначений для просунутих учнів

Для наочності та спрощення порівняння між системами можна зробити порівняльну таблицю, яка буде містити основні переваги та недоліки систем(таблиця 1.2).

Таблиця 1.2

Аналіз переваг та недоліків аналогічних систем

Назва	Основні переваги	Основні недоліки
Babbel	<ol style="list-style-type: none"> 1.Різні рівні складності завдань 2.Різноманітність завдань 3.Живі онлайн-уроки 	<ol style="list-style-type: none"> 1.Обмежений доступ до безкоштовного контенту 2.Відсутність гнучкості у виборі тем 3.Недостатня увага до граматики 4.Не дуже зручна для продвинутих учнів
Busuu	<ol style="list-style-type: none"> 1.Різноманітні інтерактивні вправи 2.Функція повторення 3.Граматичні вправи 	<ol style="list-style-type: none"> 1.Обмежена кількість вправ у безкоштовній версії 2.Неможливість самостійного обрання вправ 3.Відсутність мотиваційних механізмів
Pimsleur	<ol style="list-style-type: none"> 1.Розвиток розмовних навичок 2.Науково обґрунтований метод навчання 	<ol style="list-style-type: none"> 1.Зосереджується лише на говорінні та аудіюванні 2.Часто буває нудним і повторюваним 3.Не призначений для просунутих учнів

1.3 Аналіз інтерфейсу та функціональності існуючої моделі

Під час написання дипломної роботи на бакалавра була розроблена модель інформаційної навчальної системи для вивчення англійської мови. Система була реалізована з використанням мови програмування Python та бібліотеки Kivy, що дозволило створити сучасний та інтуїтивно зрозумілий графічний інтерфейс. Основними функціональними компонентами системи стали вправи, спрямовані на перевірку знань з форм неправильних дієслів та

складання речень. Вправи надавали можливість користувачам тренувати свої навички у різних режимах складності, що забезпечувало адаптацію до рівня знань кожного індивідуального користувача.

Після запуску програми відображався головний екран, де користувачу пропонувалося обрати одну з двох доступних вправ (Рис.1.7). У разі вибору режиму перевірки знань неправильних дієслів, першочергово необхідно було обрати рівень складності. У разі відсутності вибору система не дозволяла розпочати вправу, наочно підсвічуючи необхідність цієї дії зміною кольору кнопки на червоний, що інтуїтивно спонукало користувача до правильних дій.

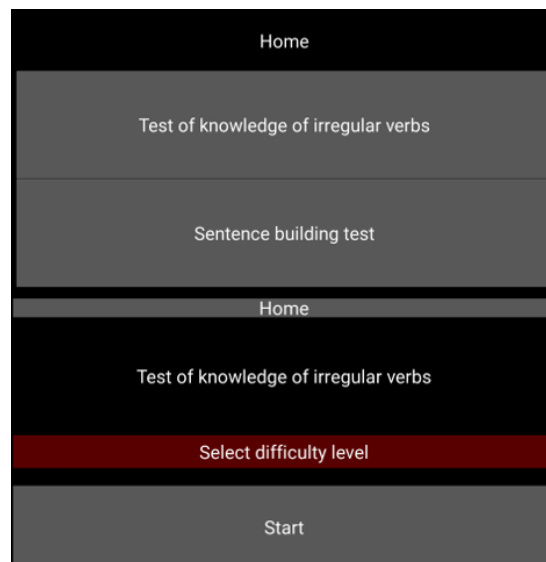


Рисунок 1.7 – Головний екран та екран вибору рівня складності

Після вибору рівня складності система переходила до екрану із завданням, де користувачеві пропонувалося ввести правильну форму дієслова на основі заданого перекладу. У разі помилкової відповіді система одразу надавала зворотний зв'язок, відображаючи правильний варіант, після чого користувач міг перейти до наступної спроби. Аналогічний підхід застосовувався у вправах зі складання речень. Після вибору рівня складності користувачу надавався набір слів, з яких необхідно було сформулювати правильне англійське речення відповідно до українського перекладу. У цьому режимі система також відслідковувала успішність виконання вправ, надаючи

лічильник, який підраховував кількість правильних і неправильних відповідей, що дозволяло користувачеві стежити за своїм прогресом. Візуальні приклади вправ представлені на Рисунку 1.8.

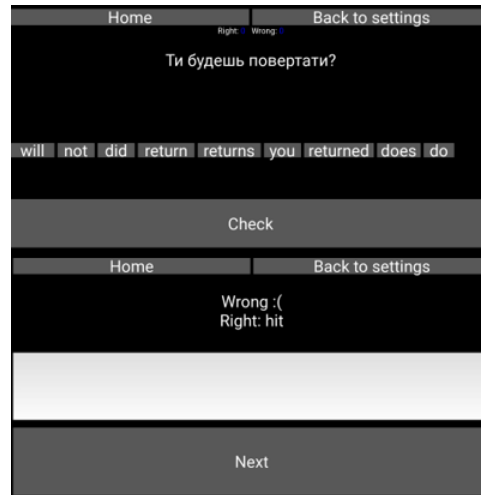


Рисунок 1.8 – Приклади вправ

База даних, яка лежала в основі цієї системи, складалася з трьох основних таблиць. Таблиця Words зберігала інформацію про неправильні дієслова, включаючи базову форму, минулу форму та третю форму дієслова, а також їх значення. Таблиця DifficultyLevels містила дані про рівні складності. А також була створена проміжна таблиця WordDifficulty, що встановлювала зв'язок між словами та рівнями складності, забезпечуючи багатоваріантність і гнучкість системи. Така структура бази даних дозволяла легко оновлювати та додавати нові слова, а також адаптувати вправи під різні рівні складності, що значно підвищувало функціональність системи. На Рисунку 1.9. можна побачити зв'язок між таблицями у базах даних.

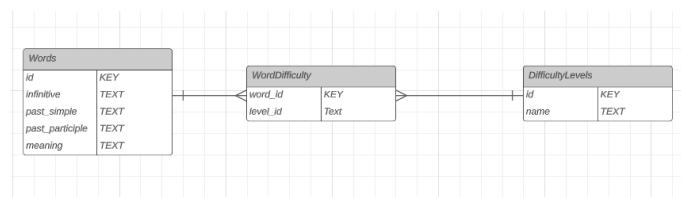


Рисунок 1.9 - Зв'язок між таблицями

Після аналізу існуючих навчальних систем стало зрозуміло, в якому напрямку слід модифікувати створену модель. Вона повинна намагатися інтегрувати переваги, які були в системах, що розглядалися, а також активно усувати їхні недоліки. Це передбачає розробку розширеної бібліотеки вправ, яка задовольнить потреби як початківців, так і просунутих учнів, надаючи їм можливість самостійно обирати теми навчання для певних вправ. Інтерактивні елементи та мотиваційні механізми також відіграватимуть важливу роль у підвищенні залученості користувачів, створюючи більш динамічну та привабливу атмосферу для навчання. Завдяки цим нововведенням нова версія системи зможе запропонувати більш ефективний та адаптивний інструмент для вивчення англійської мови, суттєво покращуючи навчальний досвід всіх користувачів.

Висновки за розділом 1

У цьому розділі було проведено всебічний аналіз предметної області вивчення англійської мови через інформаційні навчальні системи. Визначено основні компоненти даної області, зокрема роль англійської мови в глобальному контексті, а також значення інформаційних технологій у навчальному процесі.

Аналіз існуючих платформ, таких як Babbel, Busuu та Pimsleur, продемонстрував широкий спектр підходів до навчання. Актуальність проведеного аналізу полягає в тому, що він підкреслює необхідність модифікації системи, розробленої під час бакалаврату, враховуючи виявлені переваги та недоліки існуючих платформ. Впровадження новітніх технологій автоматизації та інтеграції інформаційних систем у навчальний процес дозволить створити більш ефективний інструмент для вивчення англійської мови. Це не лише відповідає вимогам сучасного освітнього середовища, але й підкреслює важливість адаптації навчання до потреб користувачів. Таким

чином, реалізація даного проекту має потенціал стати важливим кроком у розвитку інноваційних рішень для вивчення іноземних мов.

РОЗДІЛ 2. ПРОЕКТУВАННЯ БАЗ ДАНИХ ТА ОНОВЛЕННЯ МОДЕЛІ

2.1 Дослідження основних принципів роботи моделі

Принципи роботи системи ґрунтуються на необхідності розвитку мовних навичок користувачів через інтерактивні вправи, орієнтуючись на найбільш ефективні методики, аналіз яких був проведений у попередньому розділі. Основним завданням системи є надання користувачам можливості не лише перевірити свої знання, а й активно їх удосконалювати, сприймаючи мову через різноманітні типи вправ, що сприяють розвитку всіх мовних аспектів: лексики, граматики, слухових навичок та здатності до побудови правильних речень.

Одним з основних принципів є використання вправ для перевірки знань лексики та граматики. Цей тип вправ дозволяє користувачу активно взаємодіяти з мовними одиницями та формувати у нього навички правильного використання лексики та граматичних конструкцій у різних контекстах. Вправи, орієнтовані на вивчення лексики, допомагають користувачу збільшити словниковий запас, що є основою для побудови усвідомлених комунікативних навичок, а граматичні завдання сприяють розвитку здатності правильно структурувати речення. Цей підхід є важливим для розвитку мовленнєвих навичок, адже дозволяє користувачу закріпити знання через практичні завдання.

Іншим важливим аспектом роботи системи є завдання на аудіювання, оскільки сприймання мови на слух є необхідною частиною вивчення іноземної мови. Вправи, що передбачають аудіювання, дозволяють користувачу поліпшити здатність розпізнавати і розуміти мову в її живому звучанні, що є критично важливим для реального використання мови у повсякденних ситуаціях. Вправи, засновані на прослуховуванні аудіо, допомагають

користувачу тренувати увагу та навички розуміння на слух, а також поліпшують сприйняття інтонації, акценту та інших особливостей мови.

Ще одним важливим елементом є вправи на формування речень з набору слів. Ці завдання сприяють розвитку граматичних навичок, оскільки дозволяють користувачу вправлятися в побудові зв'язних та граматично правильних конструкцій. Подібні вправи стимулюють користувача до активного застосування своїх лексичних і граматичних знань на практиці, що є важливим етапом для розвитку комунікативних навичок.

Крім того, що система ґрунтується на принципах, проаналізованих у попередньому розділі, важливо також внести удосконалення, які дозволяють забезпечити більш ефективне та мотиваційне навчання для користувачів. Одним із таких удосконалень є надання користувачеві можливості перевірити свій рівень знань. Завдяки інтеграції тестів для визначення рівня знань, користувач може обрати відповідний рівень складності вправ, що дозволяє йому працювати саме з тим матеріалом, який відповідає його поточному рівню.

Ще одним важливим принципом є тестування на час, яке включає елементи гейміфікації. Враховуючи, що в багатьох проаналізованих системах увага була акцентована на звичайних завданнях без застосування додаткових механізмів мотивації, наша система вводить рейтинг, що дозволяє користувачам порівнювати свої результати з іншими. Введення часових обмежень для тестів не лише робить процес навчання динамічнішим, але й стимулює користувачів покращувати свої результати. Гейміфікаційний підхід з рейтингами дозволяє створити елементи змагання, що підвищують залученість користувачів та мотивують до досягнення кращих результатів. Таким чином, цей принцип не лише забезпечує інтерактивність процесу навчання, а й сприяє підвищенню рівня змагання між учасниками, що підвищує ефективність засвоєння матеріалу.

Окрім цього, у системі потрібно передбачити доступ до теоретичних матеріалів, що є суттєвим удосконаленням порівняно з іншими системами, де

теоретичні знання зазвичай обмежуються лише певними вправами. Важливість цього компонента полягає в тому, що користувачі зможуть вивчати граматичні та лексичні правила англійської мови в зручному форматі та у зручний час, що дозволяє їм поглиблювати свої знання та застосовувати їх на практиці. Це сприятиме комплексному підходу до навчання, де теоретична частина поєднується з практичними вправами, що забезпечить більш цілісне засвоєння мови.

І, зрештою, важливим елементом є інтерактивний компонент у вигляді чату. Це дасть користувачам можливість ставити запитання та отримувати відповіді у реальному часі, що сприяє активній взаємодії з системою та дозволяє вирішувати питання, які можуть виникнути під час проходження вправ. Такий підхід підвищує рівень залученості користувачів, оскільки вони можуть отримати допомогу без необхідності виходити за межі системи або шукати додаткові ресурси. Чат також стане корисним інструментом для розвитку комунікативних навичок, оскільки дає змогу користувачам ставити запитання на практичні теми, а також отримувати пояснення щодо складних граматичних або лексичних ситуацій.

Таким чином, принципи роботи системи включають не лише базові методики навчання через вправи, а й елементи, що сприяють мотивації та зворотному зв'язку. Завдяки введенню можливості перевірки рівня знань, гейміфікації через рейтинги та тестування на час, а також доступу до теоретичних матеріалів та інтерактивного чату, система створює умови для всебічного розвитку мовних навичок, зосереджуючи увагу на підвищенні залученості та ефективності навчання.

Для розробки системи було обрано архітектурний шаблон MVC (Model-View-Controller) (Рис.2.1), який є одним з найпоширеніших підходів до розробки програмного забезпечення, особливо в контексті створення мобільних і веб-додатків [8]. Цей шаблон дозволяє розділити додаток на три основні компоненти: модель, представлення та контролер, кожен з яких має чітко визначену відповідальність у роботі системи.

Модель відповідає за обробку даних, бізнес-логіку та взаємодію з базою даних. Вона отримує, обробляє та передає дані, але не має безпосереднього відношення до того, як ці дані будуть представлені користувачу. У випадку нашої системи модель опрацьовує запити до бази даних за допомогою PostgreSQL і SQLAlchemy, зберігає результати тестів користувачів, управляє даними вправ, рівнями складності та іншими внутрішніми компонентами системи. Завдяки такому підходу кожен елемент даних і його обробка знаходиться окремо від інтерфейсу користувача, що дозволяє знижувати складність зміни та підтримки бази даних і логіки системи.

Представлення, у свою чергу, відповідає за інтерфейс користувача, тобто за те, як дані будуть відображатися на екрані. В системі представлення побудовано за допомогою фреймворку KivyMD, який забезпечує можливість створення інтерфейсів у стилі Material Design. Всі елементи інтерфейсу, такі як кнопки, списки, текстові поля та інші, забезпечують інтерактивність і надають користувачеві зручний спосіб взаємодії з системою. Всі дані, отримані від моделі, передаються через контролер до представлення, яке відображає їх користувачеві.

Контролер є центральною частиною системи, оскільки він займається обробкою запитів користувача та взаємодією між моделлю та представленням. Контролер отримує вхідні дані від користувача, наприклад, натискання кнопок або введення тексту, обробляє їх, викликає відповідні методи в моделі для обробки даних і потім передає результат представленням для відображення. Він забезпечує ефективну комунікацію між користувачем і системою, виконуючи роль посередника між інтерфейсом та бізнес-логікою.

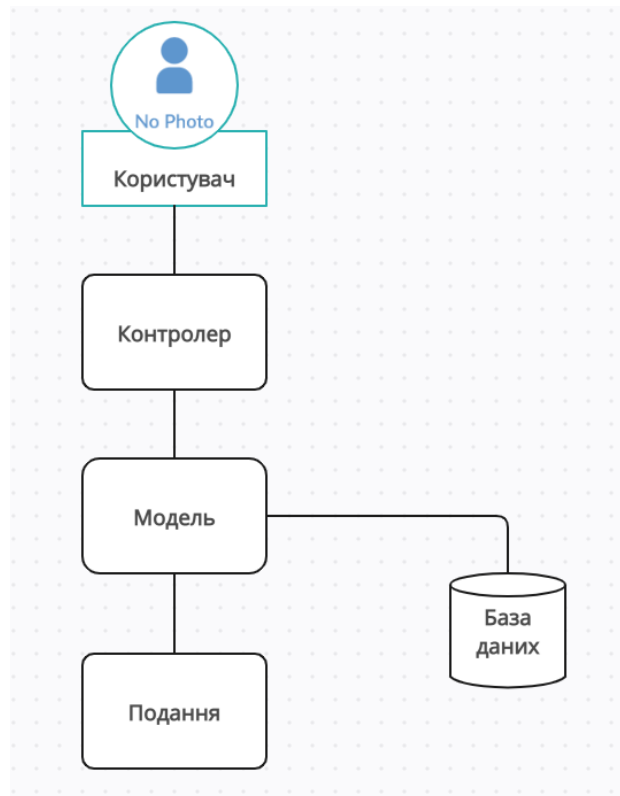


Рисунок 2.1 – Архітектура MVC

Однією з головних переваг архітектури MVC є чітке розділення обов'язків між компонентами, що дозволяє розробникам працювати над різними частинами системи незалежно один від одного [18]. Архітектура MVC також сприяє високій масштабованості та гнучкості системи. Розробка нових функціональних можливостей, таких як додавання нових типів вправ чи рівнів складності, може бути здійснена без зміни основної структури додатку. Цей підхід є особливо важливим у нашій системі, де необхідно забезпечити широкий функціонал для користувачів без ускладнення основної архітектури. Завдяки такій архітектурі можна реалізувати зручний інтерфейс, при цьому зберігаючи чистоту та ефективність коду в середині системи.

Одним із недоліків архітектури MVC є те, що для невеликих або простих додатків використання цього шаблону може бути надмірним, оскільки для реалізації навіть простих функцій необхідно створювати окремі компоненти для кожної частини системи. Водночас, у разі складних систем, таких як наша, де є велика кількість компонентів і взаємодій, використання MVC є доречним.

2.2 Проектування та розробка баз даних мобільного додатку

Проектування та розробка бази даних є одним із ключових етапів створення системи, оскільки саме база даних забезпечує збереження, обробку та доступ до інформації, необхідної для функціонування додатку.

Для роботи з базою даних було обрано PostgreSQL. PostgreSQL є потужною об'єктно-реляційною системою керування базами даних, яка підтримує складні запити, транзакції та зберігання великих обсягів даних. Вибір PostgreSQL був зумовлений його стабільністю, масштабованістю та підтримкою розширених функціональних можливостей [19]. Ця система керування базами даних дозволяє ефективно зберігати й обробляти дані користувачів, інформацію про виконані справи, результати тестувань та інші необхідні елементи системи. Також треба зазначити що для зберігання медіафайлів, таких як аудіо та зображення, було обрано Amazon S3. Цей сервіс забезпечує високу надійність, масштабованість і швидкий доступ до даних з будь-якої точки світу [15]. Використання S3 дозволяє зберігати файли в хмарі з високою безпекою та можливістю автоматичного масштабування в міру збільшення обсягу даних.

Для взаємодії з базою даних обрано бібліотеку SQLAlchemy, яка є ORM (Object Relational Mapper) для Python. ORM — це технологія, яка дозволяє взаємодіяти з реляційними базами даних, використовуючи об'єктно-орієнтований підхід [1]. Замість написання SQL-запитів програміст працює з об'єктами мови програмування, що представляють записи в таблицях бази даних. SQLAlchemy забезпечує автоматичне перетворення цих об'єктів у SQL-запити та навпаки. Це спрощує розробку, знижує ризик помилок у запитах і полегшує розширення системи в майбутньому.

База даних системи була спроектована таким чином, щоб забезпечити підтримку всіх основних функцій системи, зокрема зберігання інформації про користувачів, результати тестувань, справи різних рівнів складності,

медіафайли та інші елементи. Основними елементами структури бази даних є таблиці, кожна з яких виконує специфічну роль у забезпеченні функціональності системи.

Оскільки у системі передбачено функціонал авторизації та реєстрації, необхідно було створити таблицю users (Рис.2.2), яка забезпечує зберігання даних про користувачів системи. Ця таблиця містить унікальний ідентифікатор користувача id, який автоматично генерується та забезпечує однозначну ідентифікацію кожного користувача. Поле name містить ім'я користувача, що він вводить при реєстрації. Для виконання функцій авторизації використовується поле email, де зберігається унікальна електронна адреса кожного користувача, та поле password, яке містить хешований пароль, що гарантує безпечне зберігання даних. Додатково, ця таблиця слугує точкою зв'язку з іншою таблицею бази даних, а саме ratings про яку буде сказано далі.

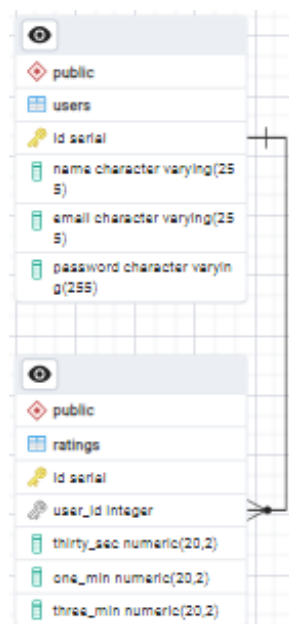


Рисунок 2.2 – Таблиці users та ratings

У системі реалізована вправа, де користувач обирає певну тему, після чого йому пропонуються чотири зображення з підписами, і він має визначити правильний підпис до одного із зображень. Для цієї вправи були створені таблиці categories та image_choice_questions (Рис.2.3). Таблиця categories

використовується для класифікації завдань за темами. Вона містить поле `id`, яке є унікальним ідентифікатором кожної категорії, та поле `name`, яке визначає назву категорії, наприклад, «Подорож», «Хоббі» тощо. Таблиця `image_choice_questions` містить дані про конкретні питання цієї вправи. У цій таблиці зберігається поле `id`, яке ідентифікує кожне питання, поле `image_url`, де міститься посилання на зображення, збережене на сервісі Amazon S3, поле `correct_caption`, яке містить правильний підпис до зображення, та три поля `wrong_caption1`, `wrong_caption2` і `wrong_caption3`, у яких зберігаються неправильні варіанти підписів. Для зв'язку питання з відповідною категорією використовується поле `category_id`, яке відповідає за зв'язок з таблицею `categories`.

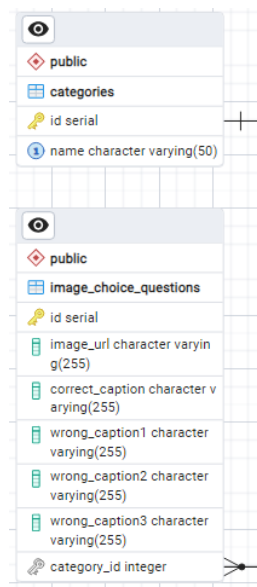


Рисунок 2.3 – Таблиці `categories` та `image_choice_questions`

У системі також передбачено функціонал тестування, за допомогою якого користувач може визначити свій рівень знань з англійської мови, наприклад, A1, B1 або C1. Для цього були створені таблиці `levels`, `test_questions` та `test_answers` (Рис.2.4). Таблиця `levels` містить унікальний ідентифікатор рівня `id` та поле `name`, яке вказує на конкретний рівень складності. Таблиця `test_questions` зберігає дані про питання тесту, серед яких

унікальний ідентифікатор питання `id`, текст питання `text` та поле `level_id`, яке відповідає за зв'язок з таблицею `levels`. Відповіді на тестові питання зберігаються у таблиці `test_answers`, яка містить поля `id`, варіанти відповідей `first_option`, `second_option` та `third_option`, а також поле `is_correct`, що вказує, яка з відповідей є правильною. Ці самі питання використовуються і в іншій вправі, де користувач обирає часовий ліміт (наприклад, 30 секунд, 1 хвилина чи 3 хвилини), після чого розпочинається тест. Користувачу задається одне питання за іншим доти, доки не завершиться відведений час. Результати цього тесту зберігаються у таблиці `ratings`, яка пов'язана з таблицею `users` (Рис.2.2). У таблиці `ratings` зберігається унікальний ідентифікатор запису `id`, поле `user_id`, що відповідає за зв'язок із користувачем, та результати трьох типів тестів: `thirty_sec`, `one_min` та `three_min`, які містять результати тестів з відповідними часовими обмеженнями.

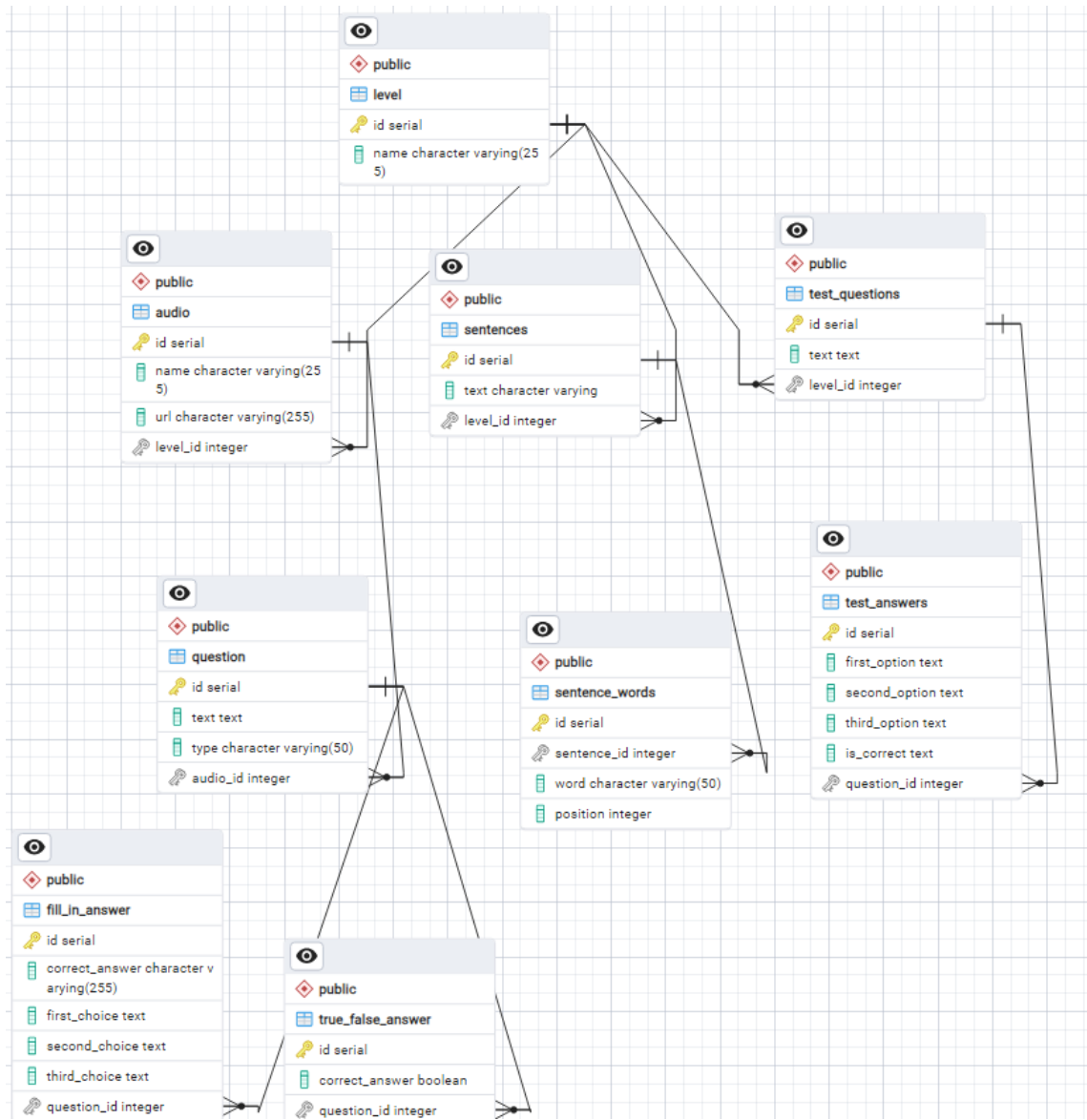


Рисунок 2.4 – Таблиці levels, test_questions, test_answers, sentence, sentence_word, audio, question, fill_in_answer, true_false_answer

Для забезпечення функціонування вправи, де користувачу надається аудіофайл та набір слів, з яких необхідно скласти речення у правильному порядку, були створені таблиці sentences та sentence_word (Рис.2.4). Таблиця sentences містить унікальний ідентифікатор речення id, текст речення text та поле level_id, яке вказує на рівень складності, пов'язаний із таблицею levels. Таблиця sentence_word містить кожне слово речення та одне слово що не входить в це речення, при цьому таблиця включає його позицію слова в реченні, що дозволяє визначити правильну послідовність.

Для вправи на аудіювання, де користувачу надається аудіофайл, після чого він відповідає на питання, були створені таблиці `audio`, `questions`, `fill_in_answer` та `true_false_answer` (Рис.2.4). Таблиця `audio` зберігає дані про аудіофайли, включаючи унікальний ідентифікатор `id`, назву файлу `name`, посилання на файл `url` та поле `level_id`, яке вказує на рівень складності. Таблиця `questions` містить питання, пов'язані з конкретним аудіофайлом, та ідентифікатор питання `id`. У таблиці `fill_in_answer` зберігаються дані про питання типу «заповніть пропуски», включаючи правильну відповідь `correct_answer` та три варіанти відповідей для вибору. Для вправи типу «правда/неправда» використовується таблиця `true_false_answer`, яка містить поле `id` та логічне значення `correct_answer`, що вказує, чи є відповідь правильною. Усі ці таблиці спільно забезпечують комплексне зберігання даних, необхідних для реалізації функціональності системи.

2.3 Розробка нового інтерфейсу

Інтерфейс є основним засобом взаємодії між користувачем та системою, і його розробка є критично важливою для забезпечення зручності та ефективності використання програмного забезпечення. У попередній версії системи для створення графічного інтерфейсу використовувався фреймворк Kivy. Kivy — це кросплатформний фреймворк для розробки додатків, що підтримує роботу на різних платформах, таких як комп'ютери, мобільні пристрої та інші [7]. Він надає можливості для швидкої розробки інтерактивних додатків, маючи в своєму складі багатий набір візуальних елементів, таких як кнопки, текстові поля, списки, а також підтримує різноманітні методи взаємодії, включаючи натискання, жести та інші види взаємодії.

Kivy забезпечує високу гнучкість при розробці інтерфейсу завдяки підтримці багатьох типів віджетів і можливості налаштування анімацій та мультимедійних ефектів.

Для попередньої версії системи було створено кілька екранів, що містять як інтерактивні, так і статичні елементи інтерфейсу. Наприклад, на головному екрані користувач мав змогу обирати між різними режимами навчання, а для кожного режиму навчання були реалізовані окремі екрани з кнопками вибору рівня складності, функціями переходу на головне меню та початку відповідних занять. Для кожного екрану використовувалися основні віджети Kivu, такі як 'Label' для відображення тексту і 'Button' для інтерактивних елементів. Також для деяких екранних елементів застосовувались інші віджети, наприклад, 'TextInput' для введення тексту користувачем, та 'StackLayout' для організації елементів у вигляді списку.

Проте, інтерфейс, розроблений у попередній версії системи, був мінімалістичним і не надто привабливим, тому виникла потреба в його вдосконаленні для підвищення естетичної привабливості та функціональності. Для цього було обрано використання бібліотеки KivumD, що є розширенням для Kivu. KivumD дозволяє створювати сучасні, красиві та зручні інтерфейси, які відповідають принципам Material Design, що був розроблений компанією Google для створення інтерфейсів із акцентом на зручність, простоту і ефективність використання.

KivumD — це надбудова над популярним фреймворком Kivu, яка додає нові елементи та компоненти, що полегшують розробку інтерактивних додатків з естетичним та зручним дизайном [6]. Вона надає готові компоненти інтерфейсу, такі як кнопки, текстові поля, меню, списки, тощо які виконують принципи Material Design, що дозволяє створювати сучасні та інтуїтивно зрозумілі інтерфейси. Однією з головних переваг KivumD є його здатність легко інтегруватися з іншими компонентами та підтримка кросплатформенності, що дозволяє одночасно створювати додатки для кількох операційних систем без значних змін у коді.

Завдяки використанню KivumD розробка інтерфейсу стала простішою та ефективнішою, оскільки бібліотека надає інструменти для створення адаптивних інтерфейсів, які автоматично налаштовуються під різні розміри

екранів і платформи. Це дозволяє зберігати високу продуктивність і функціональність на різних пристроях, зокрема мобільних телефонах, планшетах та комп'ютерах, без необхідності створення окремого коду для кожної платформи.

KivyMD також значно покращує зручність розробки, оскільки підтримує багатофункціональні компоненти, такі як навігаційні панелі, плавні анімації, інтерактивні елементи та інші елементи, що дозволяють створювати сучасні додатки з мінімальними зусиллями. Завдяки цьому KivyMD дозволяє зосередитись на логіці додатку, не витрачаючи багато часу на ручне налаштування інтерфейсів для різних платформ.

У новій версії інтерфейсу було використано кілька ключових компонентів KivyMD, щоб створити зручний та сучасний інтерфейс для користувачів. Для встановлення фону на екрані була використана компонента Image, що дозволяє розмістити картинку як фон, забезпечуючи привабливий і зручний вигляд системи.

Для відображення тексту на екрані, зокрема для назви "LangVoyage", був використаний компонент MDLabel (Рис.2.5). MDLabel — це компонент KivyMD, який використовують для відображення тексту в стилі матеріального дизайну. Це розширена версія стандартного Label з Kivy, яка додає підтримку стилізації, шрифтів та інших елементів, характерних для Material Design.

Компонент MDTextField (Рис.2.5) був використаний для створення полів вводу, де користувачі можуть вводити свої дані, такі як логін та пароль для авторизації або текст для чат-повідомлень. Цей компонент дозволяє налаштувати зовнішній вигляд та поведінку полів вводу, що дає змогу зручно вводити інформацію.

Для створення кнопок, які виконують дії, такі як "Увійти", "Не маєте облікового запису? Зареєструйтесь" тощо, застосовувались компоненти MDRaisedButton та MDTextButton відповідно (Рис.2.5). MDRaisedButton має піднятий вигляд, що робить їх більш виразними та інтуїтивно зрозумілими для користувачів. MDTextButton надає текстовий стиль, без піднятого ефекту, що

робить її менш виразною порівняно з `MDRaisedButton`, але при цьому вона залишається інтуїтивно зрозумілою для користувача. `MDTextButton` ідеально підходить для таких ситуацій, де кнопка виконує другорядну, але важливу роль, наприклад, пропонуючи альтернативну дію, як у випадку реєстрації нового облікового запису. Текстова кнопка має стиль, що гармонійно вписується в інтерфейс і дозволяє утримати простоту та елегантність дизайну. Кнопки були розташовані на відповідних екранах та виконували необхідні функції, наприклад, для переходу до процесу входу чи реєстрації.



Рисунок 2.5 – Приклад використання `MDRaisedButton`, `MDTextButton`, `MDTextField`, `MDLabel`

Для налаштування екранів в новій версії системи були використані компоненти `FloatLayout` та `MDScrollView`, які дозволяють організувати елементи інтерфейсу з урахуванням адаптивності та зручності перегляду на різних пристроях.

`FloatLayout` є віджетом, що дає змогу вільно розміщувати елементи на екрані за допомогою вказівки їхніх відносних координат. Це дозволяє розміщувати елементи інтерфейсу з фіксованими пропорціями або позиціями, що особливо корисно при розробці адаптивних інтерфейсів для різних екранів. У нашій системі цей компонент використовувався для розміщення основних елементів інтерфейсу, таких як текстові поля, кнопки, та інші компоненти, що потребують певного розташування на екрані.

`MDScrollView` був використаний для забезпечення прокручуваності вмісту екранів. Цей віджет дозволяє додавати вертикальну та горизонтальну прокрутку, що особливо важливо для екранів з великою кількістю елементів

або для забезпечення адаптивності на мобільних пристроях з обмеженим розміром екрана. `MDScrollView` допомагає організувати елементи таким чином, щоб користувач міг комфортно переглядати весь вміст, не виходячи за межі екрану, особливо при великій кількості інформації або кнопок.

Також були використанні такі компоненти, як `MDBottomNavigation`, `MDIconButton` та `MDCard` (Рис.2.6). `MDBottomNavigation` — це компонент, що дозволяє реалізувати навігацію між різними екранами програми за допомогою вкладок, що розташовані внизу екрана. Це дозволяє користувачам швидко перемикатися між основними екранами системи. В нашій системі `MDBottomNavigation` використовувався для зручного перемикання між різними екранами такими як “Головний екран”, “Бібліотека”, “Повідомлення”, “Рейтинг”. “Чат”.

`MDIconButton` — це кнопка, яка дозволяє додавати значки (іконки) у інтерфейс, замінюючи стандартні текстові кнопки на більш компактні та зрозумілі графічні елементи. Цей компонент часто використовувався для додавання кнопок повернення на попередній екран, що особливо важливо для забезпечення зручної навігації в межах програми. `MDIconButton` має різні налаштування для управління його виглядом і поведінкою, зокрема, можна налаштувати колір іконки, розмір, а також її позицію на екрані. У нашій системі ця кнопка дозволяла користувачам повернутися до попереднього екрана за допомогою іконки зі стрілкою назад.

`MDCard` — це віджет, який дозволяє створювати картки з різним вмістом, що може включати текст, зображення, кнопки та інші елементи інтерфейсу. `MDCard` надає можливість красиво організувати інформацію та зробити інтерфейс більш привабливим і зручним для користувача. Вона має параметри для налаштування тіней, закруглених кутів, висоти, ширини та інших стилістичних елементів. В нашій системі `MDCard` використовувався для створення значків вправ, кнопок для вибору категорій або рівня складності завдань, а також для виведення різних питань і варіантів відповідей на екранах

тестів та опитувань. Використання MDCard дозволило організувати інформацію в компактний, естетичний та зручний вигляд.



Рисунок 2.6 – приклад використання MDBottomNavigation, MDIconButton, MDCard.

2.4 Розробка алгоритмів роботи та програмна реалізація моделі

Для розробки системи як основну мову програмування було обрано Python. Python є високорівневою мовою програмування загального призначення, яка набула популярності завдяки своїй простоті, читабельності та широким можливостям [11]. Основною причиною вибору Python стала його універсальність, що дозволяє використовувати цю мову для вирішення найрізноманітніших задач, а також широкий вибір готових бібліотек та фреймворків, які значно спрощують процес розробки та впровадження програмного забезпечення [20].

Python має зрозумілий синтаксис, який сприяє швидкому створенню коду та зменшує ймовірність допущення помилок під час написання програм. Завдяки своїй читабельності та логічності, Python є ідеальною мовою для реалізації складних проектів, оскільки з його допомогою можна легко створювати модульний та структурований код, який у подальшому може бути масштабований і модифікований. Мова підтримує кілька парадигм програмування, включаючи об'єктно-орієнтоване, процедурне та функціональне, що робить її універсальною для вирішення задач різного рівня складності [5].

Серед ключових переваг Python, які зумовили його вибір для розробки системи, варто відзначити широкий вибір готових бібліотек, які охоплюють

майже всі аспекти програмування. Наприклад, для машинного навчання існують бібліотеки TensorFlow та PyTorch, для створення графічного інтерфейсу — Kivy та KivyMD. Використання таких інструментів дозволяє значно скоротити час розробки, зосередившись на реалізації специфічних функцій, необхідних для конкретного проекту.

Таким чином, Python став оптимальним вибором для реалізації даного проекту завдяки своїй універсальності, зручності використання, доступності готових інструментів та підтримці з боку спільноти розробників. У нашому випадку ця мова забезпечила зручне та ефективне середовище для реалізації алгоритмів роботи системи, дозволяючи створити програмний продукт, який відповідає сучасним вимогам функціональності та продуктивності.

Під час розробки алгоритмів роботи системи було вирішено створити кілька ключових функціональних елементів, що забезпечують ефективну взаємодію користувачів із системою та сприяють досягненню навчальних цілей. Екран авторизації та реєстрації є одним із ключових елементів розробленої системи, оскільки він забезпечує індивідуальний доступ користувачів до функціоналу додатку (Рис.2.7). На цьому екрані користувачі можуть створити новий обліковий запис, надавши свої дані, або увійти до вже існуючого, ввівши логін та пароль.

Особлива увага була приділена безпеці користувацьких даних. Для збереження паролів використовувався метод хешування, що виключає можливість зберігання паролів у відкритому вигляді. Хешування здійснюється за допомогою функції `pbkdf2_hmac` з бібліотки `hashlib`, яка забезпечує високий рівень криптографічної стійкості. Даний метод включає використання криптографічного алгоритму SHA-256 та додавання випадкової "солі" до кожного пароля, що захищає від атак із заздалегідь підготовленими таблицями відповідностей. Сіль (salt) — це випадковий набір даних, який додається до пароля перед хешуванням, щоб забезпечити унікальність результату хешування навіть для однакових паролів [17].

Процес хешування пароля починається з перетворення введеного користувачем текстового пароля в байтовий формат. Потім генерується випадкова сіль розміром 16 байт. Комбінація пароля і солі проходить через хеш-функцію 100000 ітерацій, створюючи унікальний хеш для кожного пароля. У базі даних зберігаються тільки хешовані паролі разом із відповідною сіллю, що унеможлиблює зловмисникам доступ до фактичних паролів навіть у разі компрометації бази даних.

Усі процеси авторизації та реєстрації в системі автоматизовані, що забезпечує зручність і швидкість взаємодії користувачів із системою. Після введення даних, таких як логін і пароль, система автоматично перевіряє коректність введених даних та здійснює відповідні операції — або реєструє нового користувача, або здійснює вхід до вже існуючого облікового запису. Таким чином, процес реєстрації та авторизації користувачів є не тільки автоматизованим, але й безпечним завдяки використанню сучасних методів криптографії, таких як хешування паролів.



Рисунок 2.7 – Приклад екранів авторизації та реєстрації

Асоціативне навчання — це методика, що базується на встановленні зв'язків між різними поняттями, об'єктами або явищами. Вона сприяє кращому засвоєнню інформації, оскільки дозволяє учням асоціювати нові знання з уже відомими або візуальними образами. Цей підхід широко використовується у вивченні мов, оскільки допомагає розвивати зв'язок між словами і відповідними їм поняттями, зображеннями чи ситуаціями. Основною метою асоціативного навчання є підвищення ефективності запам'ятовування через створення міцних зв'язків у пам'яті, що забезпечує швидкий доступ до інформації.

У розробленій системі асоціативне навчання реалізоване через вправу на вибір зображення з правильним підписом, що має назву “Знайди пару” (Рис.2.8). Ця вправа є інтерактивним завданням, яке сприяє розвитку мовних навичок, зокрема здатності ідентифікувати слова у контексті та співвідносити їх із реальними об'єктами. Під час виконання вправи користувач обирає одну з шести доступних категорій: Подорож, Хобі, Освіта, Мистецтво, Тварини або Їжа. Кожна категорія включає набір завдань, у яких користувачу пропонуються чотири зображення та відповідні підписи до них.

Дана вправа спрямована на одночасне залучення зорової, когнітивної та лінгвістичної пам'яті, що є ключовим для ефективного засвоєння лексики. Крім того процес надання завдання та перевірки правильності вибраного варіанту відповіді є автоматизованим, що забезпечує зручність та швидкість виконання вправи для користувача. Користувачеві автоматично надається завдання з вибору картинки з правильним підписом, залежно від вибраної категорії. Після того, як користувач обирає одну з шести доступних категорій (Подорож, Хобі, Освіта, Мистецтво, Тварини або Їжа), система автоматично генерує завдання для цієї категорії, надаючи чотири зображення і відповідні підписи. Після того як користувач вибирає картинку та натискає на кнопку “Перевірити”, система автоматично перевіряє, чи відповідає вибране зображення правильному підпису. Це здійснюється шляхом порівняння вибраного варіанту з правильно заданою відповіддю, яка є частиною

програмної логіки вправи. У разі правильного вибору замість кнопки “Перевірити” з’являється зелена кнопка “Далі” і обраний користувачем варіант відповіді стає також зеленим, а у разі помилки кнопка “Далі” і обраний користувачем варіант відповіді стає також червоним, а зеленим кольором стає правильний варіант відповіді.



Рисунок 2.8 – Приклад вправи “Знайди пару”

Ще однією важливою складовою системи є вправа "Аудіювання" (Рис.2.9), оскільки вона сприяє розвитку слухових навичок, що є ключовими для вивчення будь-якої мови. Включення завдань на аудіювання додає аудіальний аспект до процесу навчання, що значно підвищує ефективність засвоєння матеріалу. Аудіальне сприймання дозволяє учням не лише покращити розуміння мовлення на слух, але й зміцнити асоціативні зв'язки між словами та реальними ситуаціями. Це важливо, адже сприйняття мови в природному контексті допомагає краще адаптуватися до реальних умов її використання, таких як спілкування з носіями мови або прослуховування матеріалу.

У цій вправі користувач має змогу вибрати рівень складності аудіо матеріалу, починаючи від А1 (початковий) до С1 (просунутий), що дозволяє системі адаптувати завдання відповідно до рівня знань учня. Після вибору

рівня складності, система автоматично підтягує випадковий аудіофайл, що відповідає обраному рівню, а також тестові завдання для перевірки розуміння почутого матеріалу. Завдання включають два типи тестів: питання на "правда/неправда" (5 питань) та завдання на "заповнення пропусків" (5 питань).

У завданнях "правда/неправда" користувач перевіряє своє розуміння інформації з прослуханого матеріалу, визначаючи, чи є твердження вірним чи хибним. У завданнях на "заповнення пропусків" користувач має вставити правильне слово або фразу в пропущене місце на основі того, що він почув під час прослуховування.

Процес перевірки відповідей також є автоматизованим: після виконання завдання користувач натискає кнопку "ПЕРЕВІРИТИ ВІДПОВІДІ". Після цього система автоматично підсвічує правильні відповіді зеленим кольором, а неправильні – червоним. Такий зворотний зв'язок дає можливість користувачеві швидко зрозуміти, де він припустився помилок, і допомагає йому сконцентруватися на тих аспектах, які потребують покращення.

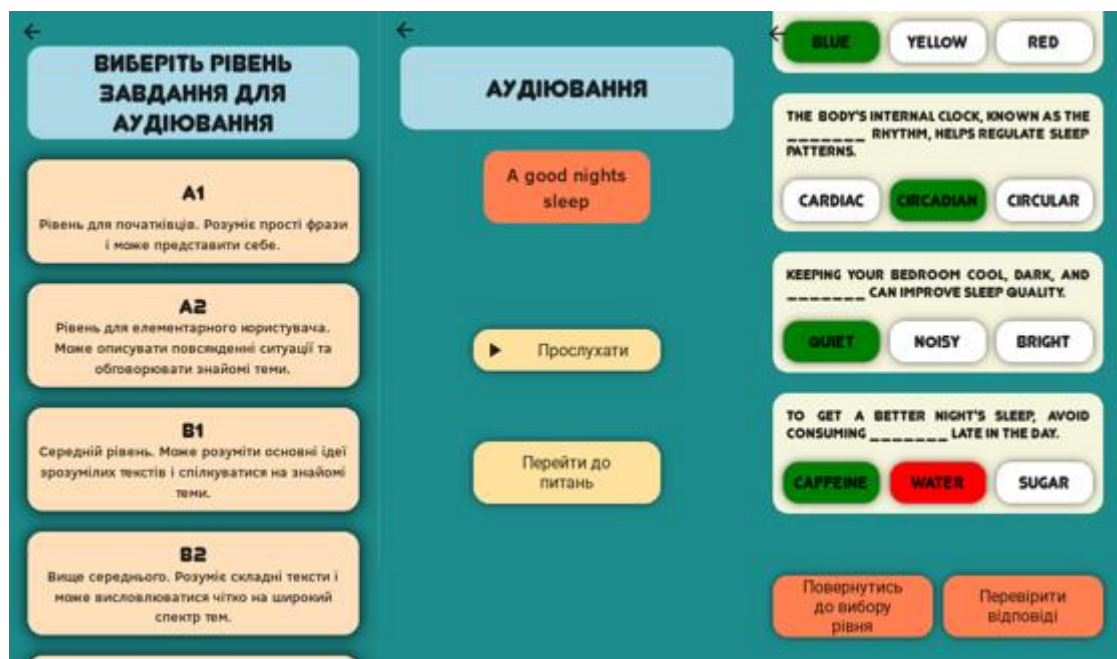


Рисунок 2.9 – Приклад вправи "Аудіювання"

Наступною важливою вправою у системі є "Збери речення" (Рис.2.10), яка також має на меті розвиток граматичних навичок та формування правильної мовної структури. У цій вправі користувачеві надається аудіофайл, що містить певне речення, а також набір слів, які необхідно правильно розташувати, щоб скласти граматично коректне речення. Це завдання дозволяє користувачеві не тільки практикуватися в слуханні і розумінні мови, а й тренуватися у побудові правильних структур з набраних слів. Така вправа є ефективною для розвитку синтаксичних навичок, оскільки користувач навчається розуміти не лише окремі слова, а й те, як ці слова взаємодіють між собою в контексті.

Процес надання завдання і перевірки відповідей у цій вправі також автоматизований. На початку користувачу автоматично надається речення випадкової складності і після того як користувач збере речення, система автоматично перевіряє правильність розташування слів. У разі правильного формулювання речення, замість кнопки "Перевірити" з'являється зелена кнопка "Далі", а у разі неправильного формулювання речення з'являється червона кнопка "Далі". Цей автоматизований підхід дозволяє не тільки зберігати час на перевірку, а й забезпечує зручність для користувача, оскільки він може отримувати миттєвий зворотний зв'язок.

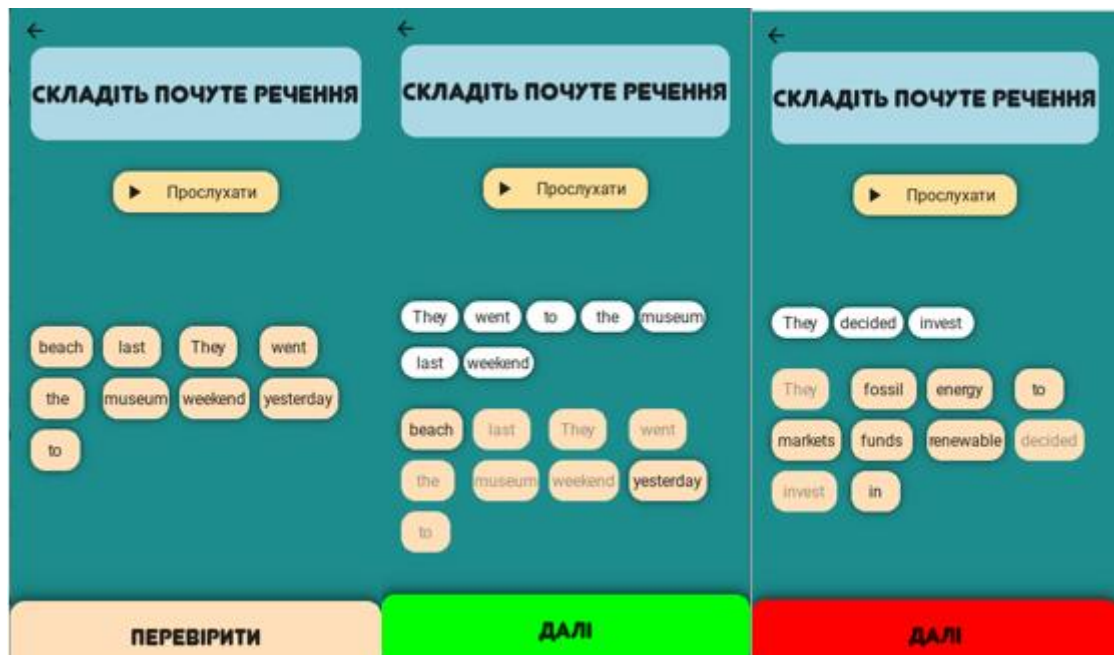


Рисунок 2.10 – Приклад вправи "Збери речення"

Також було створено вправу для визначення рівня знань користувача через тестування (Рис.2.11), що включає 35 випадкових питань, розподілених за рівнями складності: 14 питань для рівнів А1-А2, 12 — для В1-В2 та 9 — для рівня С1. Користувач має 20 хвилин для виконання тесту, що дає можливість перевірити його знання на різних етапах володіння мовою.

Тестування в системі побудоване на основі класичного методу тестування, де кожному питанню присвоєний певний ваговий коефіцієнт залежно від рівня складності. Таким чином, кожна правильна відповідь на питання з вищим рівнем складності має більший ваговий вплив на кінцевий результат. Алгоритм для розрахунку результатів виглядає таким чином:

1. Для кожного рівня (А1-А2, В1-В2, С1) визначається кількість правильних відповідей.
2. Далі кожна кількість правильних відповідей множиться на відповідний ваговий коефіцієнт для кожного рівня (наприклад, для рівня А1-А2 — 1, для В1-В2 — 2, для С1 — 3).
3. Результат тесту обчислюється за такою формулою:

$$S = \left(\frac{N_{A1-A2}}{N_{total,A1-A2}} * W_{A1-A2} \right) + \left(\frac{N_{B1-B2}}{N_{total,B1-B2}} * W_{B1-B2} \right) + \left(\frac{N_{C1}}{N_{total,C1}} * W_{C1} \right) \quad (2.1),$$

де N_{A1-A2} , N_{B1-B2} , N_{C1} — кількість правильних відповідей відповідно для кожного рівня, N_{total} — загальна кількість питань для кожного рівня, W_{C1} — ваговий коефіцієнт для кожного рівня складності.

Враховуючи розрахунки, рівень знань користувача визначається на основі отриманого результату:

- Якщо результат менший за 1.8 — рівень A1.
- Від 1.8 до 3.2 — рівень A2.
- Від 3.2 до 4.6 — рівень B1.
- Від 4.6 до 5.2 — рівень B2.
- Більше 5.2 — рівень C1.

Процес подання питань та перевірка відповідей є автоматизованими. Після того, як користувач завершить тестування, будь то через закінчення часу або натискання кнопки «ЗАКІНЧИТИ СПРОБУ», система автоматично підраховує кількість правильних відповідей, розраховує загальний бал і визначає рівень знань користувача. На основі цього виводиться діалогове вікно з результатами тесту, де показується рівень та кількість правильних відповідей.



Рисунок 2.11 – Приклад вправи для визначення рівня знань користувача

Також у системі реалізовано вправу з таймером під назвою “Тест на час” (Рис.2.12), яка передбачає вибір користувачем одного з трьох часових лімітів: 30 секунд, 1 хвилина або 3 хвилини. Метою вправи є перевірка швидкості реакції та закріплення знань у режимі обмеженого часу. Після вибору часу користувачеві пропонуються питання випадкової складності, які автоматично подаються одне за одним. Підхід, який використовується у вправі, базується на активному навчанні через регулярне повторення матеріалу, що сприяє закріпленню знань та розвитку вміння швидко орієнтуватися в ситуаціях.



Рисунок 2.12 – Приклад вправи “Тест на час”

Система обраховує результати на основі кількості правильних відповідей, враховуючи бонуси за серії правильних відповідей поспіль, а також за залишок часу при кожній правильній відповіді. Щоразу, коли користувач відповідає правильно, до загального бонусу додається значення, рівне 10% від поточного часу, що залишився до завершення вправи. Це стимулює відповідати не тільки правильно, але й швидко. Формула розрахунку виглядає так:

$$S = (\text{Correct} \times 10) + (\text{ConsecutiveCorrect} \times 10) + \text{TotalRemainingTime} \quad (2.2),$$

де *Correct* — кількість правильних відповідей; *ConsecutiveCorrect* — кількість

правильних відповідей поспіль, помножена на 10, як бонус за послідовність; TotalRemainingTime — підсумковий бонус за час, який формується як сума всіх бонусів RemainingTime, отриманих за кожну правильну відповідь.

Важливо зазначити, що формула розрахунку результату використовується лише після завершення часу, відведеного на вправу. Тобто користувач бачить фінальний результат тільки тоді, коли вичерпується обраний часовий ліміт, після чого результат автоматично записується до рейтингової таблиці.

Як вже можна було зрозуміти, у системі також створений екран з рейтинговою таблицею (Рис.2.13), що є важливим елементом гейміфікації та сприяє мотивації користувачів заробляти більші бали. Вона реалізована таким чином, щоб кожен користувач бачив свій найвищий результат для кожного з доступних часових інтервалів: 30 секунд, 1 хвилина та 3 хвилини. Такий підхід дозволяє підтримувати дух змагання між користувачами, спонукаючи їх вдосконалювати свої навички та прагнути досягти максимально можливого результату в межах визначеного часу.

30 секунд			1 хвилина			3 хвилини		
№	Ім'я	Бали	№	Ім'я	Бали	№	Ім'я	Бали
1.	Mary	102.58	1.	John	392.12	1.	Jennifer	945.67
2.	Jennifer	88.76	2.	Jennifer	349.78	2.	Elizabeth	799.99
3.	Даниїл	80.50	3.	Mary	301.56	3.	Michael	688.45
4.	Elizabeth	75.34	4.	Michael	275.34	4.	Mary	537.21
5.	James	70.12	5.	Elizabeth	227.89	5.	James	459.78
6.	Michael	67.89	6.	James	213.45	6.	John	412.34
7.	John	50.25	7.	Даниїл	0.00	7.	Даниїл	0.00

Рисунок 2.13 – Екран з рейтинговою таблицею

Також через те, що жодна з розглянутих систем не пропонувала відкритого доступу до структурованих правил з вивчення англійської мови, було вирішено створити екран бібліотеки, де буде представлено структуровану інформацію з основними правилами англійської мови (Рис.2.14). Цей екран забезпечує користувачів доступом до ключових аспектів граматики, що дозволяє поглиблено вивчати правила мови. У бібліотеці виділено кілька основних підрозділів.



Рисунок 2.14 – Екран бібліотеки

Перший підрозділ “Час і часові конструкції” пропонує детальний огляд правил побудови різних часів, пояснює їх використання, а також надає приклади для кращого розуміння. Тут розглядаються такі часові форми, як Present Simple, Present Continuous, Present Perfect, Past Simple, Future Simple та інші. Цей розділ допомагає зрозуміти, як і в яких випадках використовувати ту чи іншу часову форму, що є важливим аспектом у процесі формування правильного мовлення.

Другий підрозділ “Грамаптичні структури” охоплює основні елементи граматики, зокрема іменники, займенники, прикметники, прийменники,

сполучники та інші мовні категорії. Цей розділ забезпечує розуміння базових граматичних принципів, необхідних для побудови простих і складних речень.

Підрозділ “Вираження уподобань” включає пояснення щодо використання таких виразів, як *I want to*, *I prefer*, *I would* та інших подібних конструкцій. Цей розділ акцентує увагу на словах та фразах, які дозволяють висловлювати свої бажання, вподобання чи переваги, що є важливим у повсякденному спілкуванні.

Окремий підрозділ “Спеціальні фрази” присвячений конструкціям, які часто використовуються у розмовній англійській мові. Сюди входять фрази, такі як *I’m going to*, *I’m used to*, *I have to*, які дозволяють передавати наміри, звички чи обов’язки.

Таким чином, екран бібліотеки став цінним інструментом для систематизації знань, що дозволяє користувачам не лише швидко знайти необхідну інформацію, але й глибше зрозуміти тонкощі англійської мови.

У системі також створено інтерактивний чат, що дозволяє користувачам задавати питання та отримувати відповіді (Рис.2.15). Це було реалізовано наступним чином: зібрано текстовий файл із питаннями та відповідями, які стосуються різних аспектів вивчення англійської мови. Після цього файл було перетворено у формат ембеддингів — `question_embeddings.pt`, що слугує базою даних для пошуку найбільш відповідного запиту.

Для реалізації логіки чату використовувалась нейронна мережа `SentenceTransformer`, модель `'paraphrase-multilingual-mpnet-base-v2'`, яка є однією з найбільш потужних моделей для завдань семантичного пошуку і класифікації текстів. Ця модель базується на трансформерах, зокрема на архітектурі MPNet [2], і забезпечує високу якість обчислення семантичної подібності між текстами, що є ключовим для точного пошуку відповідей.

Завдання семантичного пошуку та класифікації текстів — це підгалузь обробки природної мови (NLP), яка спрямована на розуміння та аналіз текстів [9]. Семантичний пошук зосереджений на тому, щоб знайти інформацію, яка не лише точно відповідає запиту за текстом, але й враховує значення та

контекст слів. Наприклад, замість пошуку точного збігу слова "автомобіль", семантичний пошук також може знайти релевантні результати, пов'язані з "машина", "транспорт" або "авто". Це досягається через створення векторних подань текстів (ембеддингів), які дозволяють оцінювати схожість текстів на основі їх семантики. Класифікація текстів — це завдання автоматичного розподілу текстових даних за певними категоріями. Наприклад, визначення, чи є текст позитивним чи негативним (аналіз настрою), або ж класифікація новин за темами (спорт, політика, економіка).

Трансформери — це тип нейронної мережі, розроблений для обробки послідовностей даних, зокрема текстів. Архітектура трансформерів базується на механізмі уваги (attention mechanism), який дозволяє мережі фокусуватися на релевантних частинах тексту при обробці кожного слова [14]. Трансформери ефективно враховують контекст слів у реченні, що особливо важливо для багатозначних слів або складних граматичних конструкцій. Найвідомішими моделями на основі трансформерів є BERT, GPT, RoBERTa та інші. Трансформери стали стандартом в NLP через їхню здатність ефективно працювати з великими наборами даних і вирішувати широкий спектр завдань.

Архітектура MPNet (Masked and Permuted Pre-training Network) є удосконаленням попередніх моделей трансформерів, таких як BERT і RoBERTa. MPNet поєднує дві основні стратегії попереднього навчання:

1) Masked Language Modeling (MLM) — слова у тексті замінюються масками, і модель навчається передбачати, що там має бути, враховуючи контекст зліва та справа.

2) Permuted Language Modeling (PLM) — моделі подаються слова у випадковому порядку, і вона повинна передбачити, яке слово підходить у контексті.

Ця архітектура дозволяє MPNet краще вивчати взаємозв'язки між словами у тексті, що забезпечує високу точність у багатьох завданнях, таких як класифікація текстів, семантичний пошук або генерація тексту.

Процес підготовки системи розпочався зі збору текстового файлу з питаннями та відповідями. Ці дані були конвертовані у формат DataFrame для зручності роботи з ними. Потім за допомогою моделі SentenceTransformer згенеровано ембеддинги для кожного питання у наборі даних. Ембеддинг — це багатовимірне представлення тексту у векторному просторі, яке дозволяє ефективно порівнювати семантичну схожість між текстами. Усі ембеддинги були збережені у файл `question_embeddings.pt` для подальшого використання.

Для обчислення відповідей в реальному часі було розроблено функцію `find_best_answer`, яка виконує такі дії:

- 1) Створює ембеддинг для введеного користувачем питання.
- 2) Визначає ступінь схожості між введеним питанням та ембеддингами з бази даних, використовуючи функцію `util.pytorch_cos_sim` з бібліотеки `sentence-transformers`.
- 3) Знаходить індекс найбільш схожого питання у базі.
- 4) Перевіряє, чи значення схожості перевищує заданий поріг (наприклад, 0.7). Якщо ні, користувачу пропонується переформулювати питання.
- 5) Якщо поріг перевищено, повертається відповідь, яка відповідає знайденому найбільш схожому питанню.

Відображення відповіді відбувається з невеликою затримкою (2 секунди), імітуючи природну поведінку. Зараз чат володіє можливістю відповідати на широкий спектр питань, що стосуються вивчення англійської мови. Він здатний пояснювати побудову різних граматичних часів, розкривати їхнє використання в різних контекстах та надавати практичні приклади. Чат також дає чіткі пояснення щодо використання прийменників, таких як `on`, `in`, `at`, крім того, він розкриває відмінності між словами `some` та `any`, пояснюючи, де і як вони вживаються у стверджувальних, заперечних і питальних реченнях. Аналогічно, чат пояснює використання слів `much` і `many`, допомагаючи користувачам зрозуміти, коли застосовувати ці слова залежно від контексту.

Також чат допомагає з дієсловами, надаючи їхні 1-шу, 2-гу та 3-тю форми. Наприклад, на запит "Назвіть 2-гу та 3-тю форму дієслова 'hear'", відповідь буде: "2-га форма дієслова 'hear' — 'heard', 3-тя форма також 'heard'".

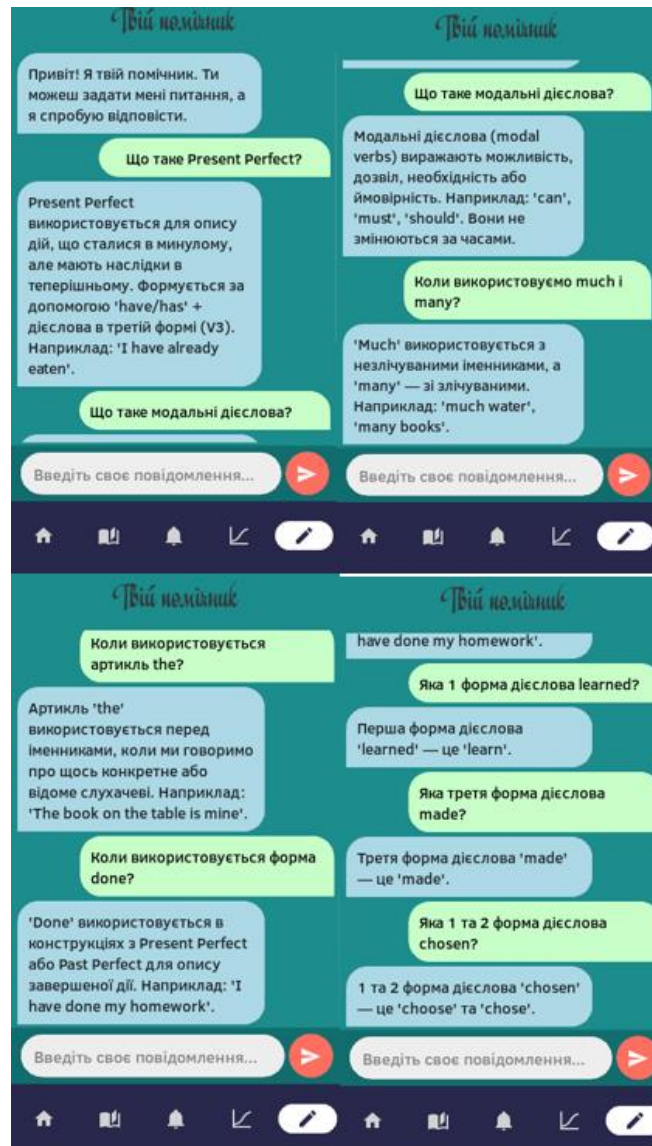


Рисунок 2.15 – Екран з “Чатом”

У результаті система має наступну схему використання як показано на Рисунку 2.16:

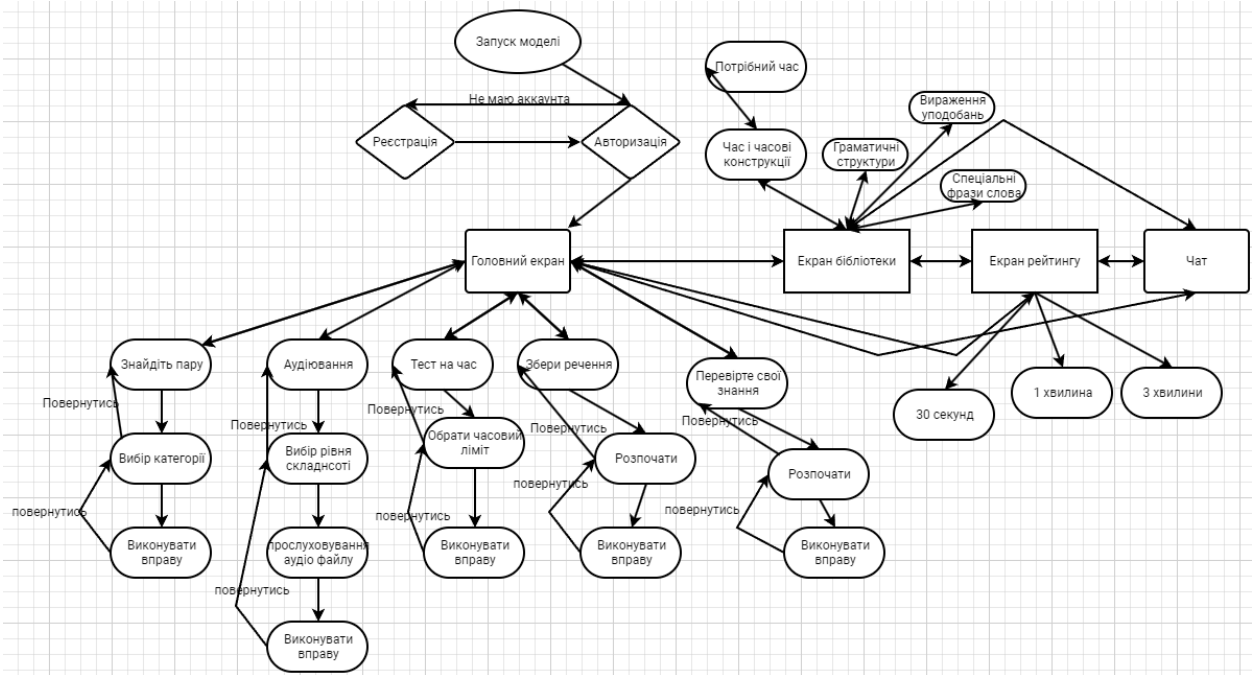


Рисунок 2.16 – Схема використання моделі

Після запуску моделі відображається екран авторизації або реєстрації. У випадку, якщо користувач не має облікового запису, йому пропонується перейти до реєстрації, після чого він повертається на екран авторизації для входу в систему. Після успішного входу користувач потрапляє на головний екран, який слугує центральним вузлом для доступу до різних функцій системи.

З головного екрану користувач може обрати будь-який із доступних розділів екрану або вправ. Перша вправа, яка пропонується, – вправа «Знайдіть пару». У цій вправі користувач вибирає категорію завдань, після чого йому надається можливість виконати вправу. Після завершення вправи або впродовж її виконання користувач може повернутися до екрану вибору категорій, а з нього на головний екран.

Другою вправою є аудіювання. Користувач спочатку обирає рівень складності завдання, після чого отримує доступ до аудіофайлу, прослуховує його та відповідає на тестові запитання. Упродовж виконання вправи або після її завершення користувач може повернутися до вибору рівня складності, а з нього до головного екрана.

Третьою вправою є «Тест на час». У цій вправі користувач вибирає часовий ліміт, серед запропонованих варіантів на 30 секунд, 1 хвилину або 3 хвилини. Після вибору часу починається тест, який складається із серії запитань, що потрібно вирішити до закінчення встановленого ліміту часу. Після завершення тесту або упродовж його виконання користувач має можливість повернутися на екран вибору часового ліміту, а з нього на головний екран.

Четвертою вправою є «Збери речення». У цій вправі користувач отримує набір слів і аудіофайл, який диктує правильну послідовність слів. Завдання полягає у формуванні речення відповідно до почутого аудіо. Після завершення вправи або під час її виконання користувач може повернутися на екран пояснення вправи, а з нього на головний екран.

П'ята вправа – «Перевірте свої знання». Ця вправа дозволяє визначити рівень знань користувача (від A1 до C1). Користувач розпочинає тестування, проходить серію завдань і отримує результат. Після завершення вправи або під час його виконання користувач може повернутися на екран пояснення вправи, а з нього на головний екран.

Крім вправ, у системі користувач може швидко переключатися між такими екранами як Головний екран, екран Бібліотеки, Рейтинговий екран, Чат. Екран бібліотеки містить зібрані правила англійської мови, які користувач може переглядати для вивчення та повторення, переходячи на наступні екрани такі як “час і часові конструкції”, “граматичні структури”, “вираження уподобань”, “спеціальні фрази слова”. Екран рейтингу дозволяє користувачеві переглядати результати інших користувачів і порівнювати свої досягнення переключуючись між відображенням рейтингу по часовому ліміту. Останнім функціональним екраном є чат, який дає можливість задавати питання та отримувати відповіді.

Далі буде представлена діаграма класів (Рис.2.17), яка зображатиме лише одну з вправ, що входять до складу системи. Вибір конкретної вправи для цієї діаграми обумовлений тим, що для демонстрації достатньо одного

прикладу, щоб показати, як організовано управління даними, взаємодія з користувачем і обробка логіки. Повна діаграма класів включала б усі можливі варіанти вправ, екрани, а також взаємодію з іншими компонентами системи. Однак, для наочності та спрощення викладу, в даному випадку буде показано лише одну вправу, що дозволить зрозуміти загальну структуру класів та принципи їх взаємодії в рамках основної функціональності програми.

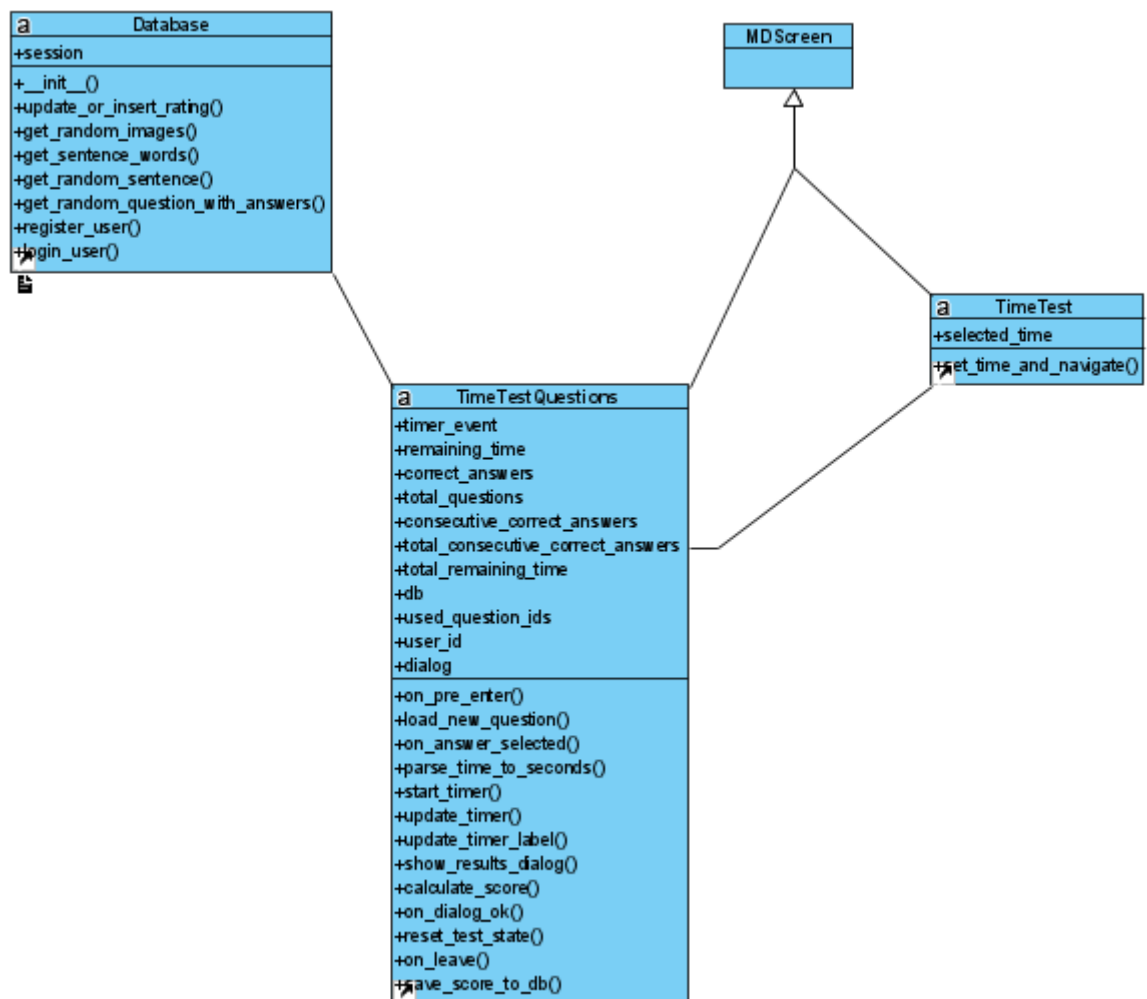


Рисунок 2.17 – Діаграма класів для вправи тест на час

Як видно з діаграми, майже всі класи успадковують клас MDScreen з бібліотеки Kivumd. Це обумовлено тим, що кожен екран у програмі представляє собою інтерфейс для користувача, що є основною частиною системи. Клас MDScreen служить базовим класом для всіх екранів, надаючи

необхідні функції для взаємодії з елементами інтерфейсу, управління подіями та обробки переходів між екранами. Завдяки цьому всі екрани успадковують від нього базову функціональність і можуть бути налаштовані відповідно до вимог конкретного екрану.

Клас `TimeTest` є екраном, на якому користувач може обрати час для виконання тесту. Він успадковує клас `MDScreen` з `KivyMD` і містить властивість `selected_time`, яка зберігає обраний користувачем час. Метод `set_time_and_navigate` дозволяє встановити обраний час і переключити екран на екран з питаннями тесту. Цей клас забезпечує інтерфейс для вибору часу тестування, після чого користувач буде перенаправлений до екрану з питаннями тесту.

Клас `TimeTestQuestions` є екраном, що успадковує `MDScreen` і відповідає за логіку тестування. Він містить кілька важливих властивостей, таких як `remaining_time` (час, що залишився для відповіді), `correct_answers` (кількість правильних відповідей), а також інформацію про питання та їх варіанти. Крім того, в класі ініціалізується об'єкт класу `Database`, який дозволяє використовувати функцію `get_random_question_with_answers` для отримання випадкових питань і варіантів відповідей із бази даних. Метод `on_pre_enter` викликається перед тим, як екран стає активним, і відповідає за ініціалізацію стану тесту, скидаючи лічильники правильних відповідей, загальної кількості питань, а також очищаючи список вже використаних питань. Крім того, цей метод ініціалізує таймер і завантажує перше питання.

Метод `load_new_question` відповідає за завантаження нового питання з бази даних. Він перевіряє, чи було вже використано це питання, щоб уникнути дублювання, і завантажує нове, якщо це необхідно. Після завантаження питання цей метод оновлює текст питання та варіанти відповідей на екрані, а також зв'язує події вибору відповідей з методами для обробки вибору. Метод `on_answer_selected` обробляє вибір користувача, перевіряючи, чи є вибрана відповідь правильною. Якщо відповідь правильна, кількість правильних відповідей збільшується, а також додається бонусний час до результату. Якщо

відповідь неправильна, то правильні відповіді підряд скидаються, і користувач отримує нове питання.

Таймер, що відраховує залишковий час, запускається за допомогою методу `start_timer`, який оновлює час кожну секунду через виклик методу `update_timer`. Коли час вичерпано, викликається метод `show_results_dialog`, який відображає діалог з результатами та обчислює підсумкові бали на основі кількості правильних відповідей і бонусного часу. Підсумковий бал залежить від правильних відповідей, кількості правильних відповідей підряд і часу, що залишався під час кожної правильної відповіді. Клас також містить методи для збереження результатів у базі даних і відображення діалогу з результатами тесту. Метод `save_score_to_db` відповідає за збереження або оновлення рейтингу користувача в базі даних, в залежності від результатів тесту та обраного часу.

Усі вправи в системі тісно пов'язані з базою даних, оскільки кожна з них потребує доступу до інформації, збереженої в базі, для коректного виконання. База даних забезпечує збереження та отримання таких даних, як питання, варіанти відповідей та інші важливі параметри, що дозволяє системі надавати користувачу персоналізований досвід і зберігати статистику та прогрес.

Оскільки логіка інших вправ та функціоналу є досить стандартною і схожою з тією, що описана для класу `TimeTestQuestions`, детальне описання інших вправ та функціоналу не є необхідним. Ці вправи взаємодіють з базою даних аналогічним чином, без необхідності додаткових складних алгоритмів чи обчислень. Враховуючи це, доцільно було не деталізувати їхні описи і діаграми, оскільки вони не додають суттєвих нових аспектів до загальної концепції роботи системи.

Висновки за розділом 2

У другому розділі дипломної роботи було проведено детальне дослідження основних принципів роботи моделі системи для вивчення

англійської мови. Було розроблено та спроектовано базу даних, яка використовується для зберігання та управління необхідною інформацією, що дозволяє ефективно зберігати дані про результати вправ, статистику користувачів та іншу релевантну інформацію. Крім того, був розроблений інтерфейс системи, який відповідає за візуальну частину програми та забезпечує інтерактивну взаємодію з користувачем. Інтерфейс включає елементи для вибору вправ, доступу до граматичних правил, перегляду результатів тестів, можливість задавати питання в чат тощо.

Також було проведено роботу над алгоритмами роботи системи та їх реалізацією. Було розроблено алгоритми для обробки запитів користувачів, включаючи алгоритми автоматичної перевірки відповідей на тести, обчислення результатів користувачів у виконанні вправ, а також алгоритми для семантичного пошуку відповіді на питання в чаті. Було розглянуто схему використання моделі, а також діаграму класів для вправи “Тест на час”.

РОЗДІЛ 3. ТЕСТУВАННЯ МОДЕЛІ

3.1 Результати, які були отримані при тестуванні моделі та їх аналіз

Тестування є важливим етапом життєвого циклу розробки програмного забезпечення, який забезпечує перевірку коректності роботи функціоналу, виявлення помилок та підтвердження відповідності програмного продукту технічним вимогам. Основна мета тестування полягає у верифікації реалізованих функціональних можливостей, включаючи взаємодію з інтерфейсом, роботу вправ, перевірку відображення результатів та інших компонентів системи.

У процесі тестування було реалізовано кілька сценаріїв, кожен із яких передбачав виконання послідовності дій користувача з подальшою перевіркою результатів. Для наочності наведемо результати виконання трьох основних тестів.

Тест 1:

Після запуску програми система відобразила екран авторизації, який містив поля для введення логіна та пароля, а також кнопку для реєстрації нового користувача. Оскільки акаунт у системі не був створений раніше, було обрано опцію реєстрації. Відобразився екран реєстрації, де ввели необхідні дані, такі як ім'я, електронна пошта та пароль. Після завершення реєстрації система автоматично повернулася до екрану авторизації, де були введені щойно створені дані. Після успішної авторизації потрапив на головний екран програми. Інтерфейс головного екрана пропонував перелік доступних вправ, включаючи такі варіанти: "Знайдіть пару", "Перевірте свої знання", "Тест на час", "Аудіювання" та "Збери речення" (Рис.3.1).



Рисунок 3.1 – Екран авторизації та реєстрації і головний екран

Спочатку була обрана вправа "Знайдіть пару" (Рис.3.2). Після натискання відкрився екран із вибором категорій. Наприклад обрав категорію "Хобі", після чого система завантажила перше завдання. Завдання полягало у виборі правильної пари (картинки та підпису) з-поміж чотирьох запропонованих варіантів. Після вибору варіанта відповіді та натискання на кнопку "Перевірити" правильний варіант система візуально відображала підсвічуючи обрану пару зеленим кольором. При виборі неправильного варіанту система підсвічувала помилку червоним кольором та одночасно вказувала правильний варіант, виділяючи його зеленим.



Рисунок 3.2 – Тестування вправи "Знайдіть пару"

Після цього для повернення на головний екран натиснув два рази на стрілку у лівому верхньому екрані. Після натискання кнопки "Перевірте свої знання" на головному екрані відкрився екран із описом вправи та кнопкою "Розпочати тест". Натиснувши цю кнопку, перейшов до виконання завдань (Рис.3.3). У рамках вправи система пропонувала серію тестових запитань, кожне з яких передбачало кілька варіантів відповідей. Після завершення всіх запитань відображається спливаюче вікно із підсумковим результатом, який включав рівень знань, як результат тестування, та кількість правильних відповідей.



Рисунок 3.3 – Тестування вправи "Перевірте свої знання"

Повертаємось до головного екрану та вибираємо вправу "Тест на час" (Рис.3.4), після вибору якої відкрився екран із можливістю обрати часовий ліміт для виконання завдання. Обираємо ліміт 30 секунд і натиснув кнопку "Почати тест". Система почала зворотний відлік часу, під час якого потрібно відповісти на серію запитань. Кожне запитання включало кілька варіантів відповідей. Відлік часу був візуально відображений у верхній частині екрана, що дозволяло слідкувати за залишковим часом. По завершенню відведеного

часу вправа автоматично завершувалася, і система відображала результати, які включали кількість запитань, на які було надано відповіді, кількість правильних відповідей, а також отриманні бали. Повертаємось на головний екран та переходимо до екрану рейтингу, і бачимо що результат додався до таблиці з 30 секундним лімітом.



Рисунок 3.4 – Тестування вправи “Тест на час”

Повертаємось на головне меню та обираємо вправу "Аудіювання" (Рис.3.5). Система відкриває екран із вибором рівня завдання. Обираємо рівень

A1, після чого з'являється екран з кнопкою "Прослухати". Натискаємо її, і система відтворює аудіофайл. Прослухавши аудіо, натискаємо кнопку "Перейти до питань". Відкривається список запитань, пов'язаних із почутим текстом. Обираємо відповіді з кількох варіантів, завершивши всі запитання, натискаємо кнопку "Перевірити відповіді". Система аналізує результати та показує помилкові відповіді, що підсвічуються червоним кольором, правильні — зеленим.



Рисунок 3.5 – Тестування вправи "Аудіювання"

Повертаємось на головне меню та обираємо вправу "Збери речення" (Рис.3.6). Система відображає екран із кнопкою "Почати". Натискаємо її, і система пропонує завдання. Натискаємо кнопку "Прослухати", після чого відтворюється аудіофайл із реченням. Збираємо почуте речення, використовуючи запропоновані слова. У разі правильного складання система підтверджує успіх, підсвічуючи кнопку "Далі" зеленим кольором якщо відповідь неправильна, система підсвічує червоним кольором.

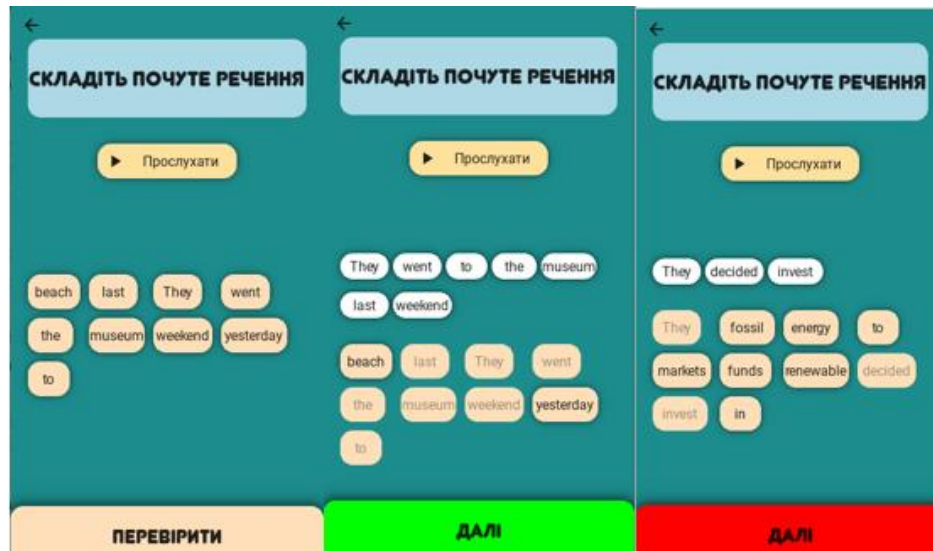


Рисунок 3.6 – Тестування вправи "Збери речення "

Тест 2:

Після запуску програми відображається екран авторизації. Вводимо дані для входу, після чого відкривається головний екран програми. На головному екрані обираємо другу кнопку в нижньому меню, що відповідає за доступ до бібліотеки правил англійської мови (Рис.3.7). Переходимо на екран, де натискаємо кнопку "Час і часові конструкції". У вікні, що з'явилося, обираємо категорію "Future Continuous". Перевіряємо коректність відображення інформації. Далі, двічі натискаючи на стрілку у верхньому лівому куті, повертаємось на екран бібліотеки. Обираємо кнопку "Граматичні структури", перевіряючи коректність відображення інформації для цієї категорії. Інформація виводиться правильно. Знову натискаємо на стрілку у верхньому лівому куті для повернення до бібліотеки та обираємо кнопку "Вираження уподобань". Аналогічно перевіряємо, що інформація відображається без помилок. Після цього повертаємось на екран бібліотеки, натиснувши стрілку у верхньому лівому куті, і натискаємо кнопку "Спеціальні фрази слова". Бачимо що інформація виводиться правильно.

Усі перевірки демонструють коректне функціонування розділу бібліотеки правил англійської мови. Інтерфейс працює стабільно, навігація інтуїтивно зрозуміла, а дані виводяться відповідно до обраної категорії.



Рисунок 3.7 – Тестування екрану “Бібліотеки”

Тест 3:

Натискаємо останню кнопку у нижньому меню, що забезпечує доступ до чату з мовною моделлю. У вікні чату вводимо перший запит: "Що таке Present Perfect?" та натискаємо кнопку із зображенням трикутника для відправлення повідомлення (Рис.3.8). Після короткої паузи отримуємо відповідь із детальним поясненням граматичної структури. Далі вводимо наступний запит: "Що таке модальні дієслова?". Відповідь з'являється на екрані у вигляді структурованого опису модальних дієслів, їх використання та прикладів. Переходимо до третього запиту: "Коли використовується much і many?". Отримуємо пояснення про відмінності між цими словами та правила їх застосування залежно від типу іменників. Далі задаємо питання про "Коли використовується артикль the?" та отримуємо відповідь про його використання. Потім питаємо про "Коли використовується форма done?" і

одержуємо відповідь. Наступним запитом вводимо: "Яка 1 форма дієслова learned?". Система надає відповідь, вказуючи початкову форму дієслова — learn. Далі тестуємо систему запитом: "Яка третя форма дієслова made?". Отримуємо правильну відповідь: made. Заключний запит: "Яка 1 та 2 форма дієслова chosen?". Модель відповідає, вказуючи першу форму — choose, і другу форму — chose.

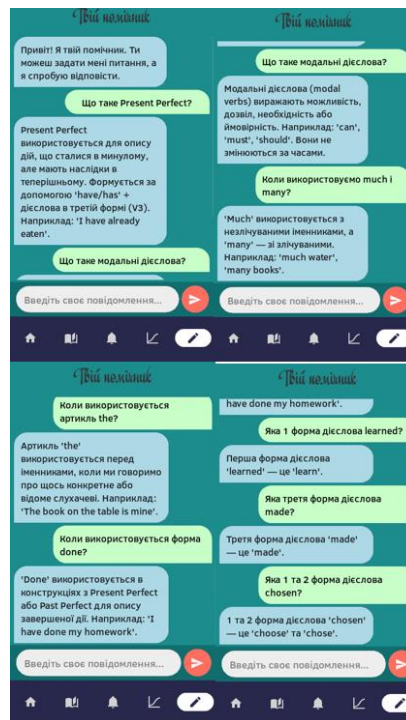


Рисунок 3.8 – Тестування чату системи

Тестування функціональності чату підтвердило, що система коректно обробляє запити, надає точні відповіді на граматичні питання та демонструє стабільність роботи. Інтерфейс забезпечує зручну навігацію, а модель адекватно реагує на різноманітні формулювання запитань.

На основі проведеного тестування можна зробити висновок, що система функціонує коректно. Усі основні модулі програми, включаючи реєстрацію та авторизацію, інтерактивні вправи, бібліотеку граматичних правил, відображення рейтингу та чат для запитань, були перевірені й продемонстрували правильну роботу. Відображення даних, взаємодія з

користувачем і виконання поставлених завдань відповідають очікуваному результату. Таким чином, система успішно пройшла тестування.

3.2 Рекомендації щодо використання розробленої моделі

Розроблена модель є універсальним інструментом для вивчення англійської мови, який може бути ефективно використаний різними категоріями користувачів. Завдяки низці вправ, що охоплюють різні аспекти мовних навичок, система підходить як для початківців, так і для користувачів, які прагнуть удосконалити свої знання на більш високому рівні. Враховуючи зручність і доступність на різних платформах, від комп'ютерів до мобільних пристроїв, модель надає користувачам можливість здійснювати навчання в будь-який час і в будь-якому місці, що є важливим для осіб з насиченим графіком, таких як студенти.

Різноманітність рівнів складності вправ дозволяє адаптувати навчальний процес до індивідуальних потреб користувачів. Вибір між завданнями, що перевіряють знання граматики, слухові навички, здатність до складання речень та тестування знань за допомогою таймера, дозволяє користувачам практикуватися у тих аспектах мови, які потребують найбільше уваги. Це створює умови для персоналізованого навчання та прогресу в засвоєнні мови.

Також варто зазначити перевагу інтегрованого чат-бота, який дозволяє користувачам швидко отримувати відповіді на питання щодо граматики та інших аспектів англійської мови. Використання нейронної мережі для пошуку найбільш схожих питань допомагає значно підвищити ефективність навчання, оскільки користувачі можуть отримати точні та вичерпні відповіді на свої запити. Цей функціонал робить систему ще більш інтуїтивно зрозумілою та корисною.

З урахуванням цих факторів, система є потужним інструментом для самостійного вивчення англійської мови, що підходить для широкої аудиторії,

незалежно від рівня знань або досвіду. Вона може бути ефективно використана в навчальних закладах, для самостійного навчання, а також в корпоративному середовищі для покращення англійської мови працівників.

Висновки за розділом 3

Під час тестування розробленої моделі було підтверджено, що всі основні функції системи працюють коректно і відповідають вимогам. Система успішно пройшла перевірку зокрема щодо реєстрації та авторизації користувачів, виконання вправ, взаємодії з користувачем через чат та відображення результатів. Усі модулі, включаючи інтерактивні вправи, бібліотеку граматичних правил і систему зворотного зв'язку, продемонстрували свою стабільну роботу, що вказує на високу якість розробки.

Розроблена модель є зручним та ефективним інструментом для вивчення англійської мови, яка адаптується до потреб різних користувачів. Завдяки різноманітності вправ, які охоплюють усі аспекти мовних навичок, а також можливості вибору між різними рівнями складності, модель підходить як для початківців, так і для більш досвідчених користувачів. Гнучкість у виборі вправ і можливість навчатися на різних платформах робить систему доступною для широкої аудиторії в будь-який час і в будь-якому місці.

З огляду на високу ефективність системи, вона може бути використана в різних навчальних контекстах — як для самостійного вивчення мови, так і в навчальних закладах чи корпоративному середовищі для покращення рівня володіння англійською мовою серед працівників. Система є надійним і гнучким інструментом для вивчення англійської мови, що відповідає вимогам сучасного навчання.

ВИСНОВКИ

В даній роботі було проведено детальний огляд предметної області вивчення англійської мови через інформаційні навчальні системи, що дозволило визначити основні компоненти цієї сфери, а також оцінити роль технологій у навчальному процесі. Аналіз існуючих платформ, таких як Babbel, Busuu та Pimsleur, показав різноманітність підходів до навчання, а також виявив переваги та недоліки цих систем. Актуальність проведеного аналізу полягає в тому, що він дозволив сформулювати вимоги до вдосконалення існуючої системи та врахувати досвід інших платформ для створення більш ефективного інструменту вивчення англійської мови.

У результаті розробки автоматизованої моделі системи для вивчення англійської мови було створено комплекс інструментів, який включає інтерактивні вправи, алгоритми обробки запитів користувачів та систему для семантичного пошуку відповідей у чаті. Спроектowana база даних ефективно зберігає інформацію про користувачів і результати їхніх вправ, забезпечуючи зручний доступ до цих даних. Інтерфейс системи дозволяє користувачам легко взаємодіяти з платформою та здійснювати вибір вправ відповідно до їхніх потреб.

Тестування розробленої системи підтвердило коректну роботу всіх основних функцій, зокрема реєстрації та авторизації користувачів, виконання вправ, взаємодії з чат-ботом та відображення результатів. Всі модулі продемонстрували стабільну роботу, що вказує на високу ефективність та надійність системи як інструменту для вивчення англійської мови.

Розроблена система є потужним інструментом для самостійного вивчення англійської мови, а також може бути використана в освітніх установах або корпоративному середовищі для підвищення рівня володіння мовою серед співробітників. Вона відповідає вимогам сучасного навчального процесу та є доступною для широкої аудиторії завдяки гнучкості в

налаштуванні рівня складності вправ і можливості навчання на різних платформах.

Проте, у майбутньому є кілька напрямків для вдосконалення цієї моделі. По-перше, варто розглянути інтеграцію додаткових технологій, таких як штучний інтелект для адаптивного навчання, що дозволить більш точно підбирати вправи відповідно до індивідуальних потреб користувачів. По-друге, можна вдосконалити алгоритми зворотного зв'язку, щоб надавати більш детальні рекомендації з покращення мовних навичок. По-третє, варто додати можливість голосового введення для вправ з аудіювання, що значно підвищить ефективність навчання. Таким чином, подальший розвиток системи може значно покращити досвід користувачів і зробити процес навчання ще більш інтерактивним та персоналізованим.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Abba I. What is an ORM – The Meaning of Object Relational Mapping Database Tools. *freeCodeCamp.org*. [Електронний ресурс]. – режим доступу: URL: <https://www.freecodecamp.org/news/what-is-an-orm-the-meaning-of-object-relational-mapping-database-tools/> (date of access: 14.11.2024).
2. All Mpnet Base V2 by Sentence Transformers | AI model details. *Your Roadmap to the AI Revolution - AIModels.fyi*. [Електронний ресурс]. – режим доступу: URL: <https://www.aimodels.fyi/models/huggingFace/all-mpnet-base-v2-sentence-transformers> (date of access: 17.11.2024).
3. Busuu Review. Test Prep Insight. [Електронний ресурс]. – режим доступу: URL: <https://testprepinsight.com/reviews/busuu-review/> (date of access: 15.10.2024).
4. English in the World Today: Status report and state of research - E L Gazette. E L Gazette. [Електронний ресурс]. – режим доступу: URL: <https://www.elgazette.com/english-in-the-world-today-status-report-and-state-of-research/> (date of access: 14.10.2024).
5. FREEhost.UA. [Електронний ресурс]. – режим доступу: URL: <https://freehost.com.ua/ukr/faq/wiki/chto-takoe-jazik-programmirovaniya-python/> (date of access: 17.11.2024).
6. GeeksforGeeks. Building a Simple Application using KivyMD in Python - GeeksforGeeks. [Електронний ресурс]. – режим доступу: URL: <https://www.geeksforgeeks.org/building-a-simple-application-using-kivymd-in-python/> (date of access: 6.11.2024).
7. GeeksforGeeks. List of Python GUI Library and Packages - GeeksforGeeks. [Електронний ресурс]. – режим доступу: URL: <https://www.geeksforgeeks.org/python3-gui-application-overview/> (date of access: 15.11.2024).

8. GeeksforGeeks. MVC Design Pattern - GeeksforGeeks. [Электронный ресурс]. – режим доступа: URL: <https://www.geeksforgeeks.org/mvc-design-pattern/> (date of access: 2.11.2024).
9. GeeksforGeeks. Understanding Semantic Analysis - NLP - GeeksforGeeks. [Электронный ресурс]. – режим доступа: URL: <https://www.geeksforgeeks.org/understanding-semantic-analysis-nlp/> (date of access: 19.11.2024).
10. I Was Trash at Learning Languages for 30 Years–Until I Tried This App. VICE. [Электронный ресурс]. – режим доступа: URL: <https://www.vice.com/en/article/pimsleur-review/> (date of access: 14.10.2024).
11. Python Code 101. Hyperskill University. [Электронный ресурс]. – режим доступа: URL: <https://hyperskill.org/university/python/python-code-101> (date of access: 18.11.2024).
12. Ranked: The Top Languages Spoken in the World. [Электронный ресурс]. – режим доступа: URL: <https://www.visualcapitalist.com/top-languages-spoken-in-the-world/> (date of access: 14.10.2024).
13. Ukraine | EF English Proficiency Index | EF Global Site (English). EF Education First | Deutschland. [Электронный ресурс]. – режим доступа: URL: <https://www.ef.com/wwen/epi/regions/europe/ukraine/> (date of access: 14.10.2024).
14. What are Transformers? - Transformers in Artificial Intelligence Explained - AWS. Amazon Web Services, Inc. [Электронный ресурс]. – режим доступа: URL: <https://aws.amazon.com/what-is/transformers-in-artificial-intelligence/> (date of access: 22.11.2024).
15. What AWS Cloud Storage Will Work Best for Your Business?. Infopulse. [Электронный ресурс]. – режим доступа: URL: <https://www.infopulse.com/blog/best-aws-cloud-storage-solution> (date of access: 22.11.2024).

16. What is Babel and How to Use It?. *Neural Network Press*. [Электронный ресурс]. – режим доступа: URL: <https://neuralnetworkpress.com/what-is-babel-and-how-to-use-it> (date of access: 15.10.2024).
17. What is a salt in hashing?. Nexusgroup. [Электронный ресурс]. – режим доступа: URL: [https://www.nexusgroup.com/what-is-a-salt-hash/#:~:text=A%20salt%20is%20a%20random,SHA\)%20family%20for%20this%20purpose](https://www.nexusgroup.com/what-is-a-salt-hash/#:~:text=A%20salt%20is%20a%20random,SHA)%20family%20for%20this%20purpose). (date of access: 23.11.2024).
18. Why MVC Architecture?. *Medium*. [Электронный ресурс]. – режим доступа: URL: <https://medium.com/@socraticsol/why-mvc-architecture-e833e28e0c76> (date of access: 03.11.2024).
19. Why PostgreSQL Is a Top Choice for Enterprise Databases. *Percona Database Performance Blog*. [Электронный ресурс]. – режим доступа: URL: <https://www.percona.com/blog/why-postgresql-is-a-top-choice-for-enterprise-level-databases/> (date of access: 24.11.2024).
20. Why Use Python for Web Development?. *Imaginary Cloud: Software Development for Digital Acceleration*. [Электронный ресурс]. – режим доступа: URL: <https://www.imaginarycloud.com/blog/why-use-python-for-web-development> (date of access: 13.11.2024).

ДОДАТКИ

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра комп'ютерних систем та робототехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **Магістр**
Галузь знань: 17 – Електроніка, автоматизація та електронні комунікації
Спеціальність: 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка
Освітня програма «Комп'ютеризовані системи управління та автоматика»

ЗАТВЕРДЖУЮ

в.о. завідувача комп'ютерних
систем та робототехнікик. ф.-м. н., доц. ХРУСЛОВ М. М.
«12» вересня 2024 року


ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Негібову Даниїлу Олеговичу

(прізвище, ім'я, по батькові студента)

1. Тема роботи **Модель інформаційної навчальної системи для вивчення англійської мови**

керівник роботи **Лабенко Дмитро Петрович к.т.н., доцент**
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету № 4101-5/3657 від 12 листопада 2024 року

2. Строк подання студентом роботи ***30 листопада 2024 року***

3. Перелік питань, які потрібно розробити)

1. Постановка задачі розробки моделі інформаційної системи для вивчення англійської мови
2. Порівняльний аналіз існуючих навчальних систем
3. Аналіз інтерфейсу та функціоналу існуючої моделі, вибір нових засобів та елементів для оновленої системи
4. Проектування бази даних для оновленого мобільного додатку
5. Проектування нового дизайну інтерфейсу та розробка нових вправ.
6. Проведення тестування моделі інформаційної системи для вивчення англійської мови

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Аналіз літератури за темою	05.09.2024 — 15.09.2024
2	Аналіз та порівняння аналогічних існуючих систем	16.09.2024 — 30.09.2024
3	Дослідження основних принципів роботи моделі	01.10.2024 — 10.10.2024
4	Проектування баз даних мобільного додатку	11.10.2024 — 31.10.2024
5	Розробка нового дизайну інтерфейсу	01.11.2024 — 10.11.2024
6	Додавання нових вправ	11.11.2024 — 18.11.2024
7	Програмна реалізація моделі з урахуванням нового функціоналу.	19.09.2024 — 23.10.2024
8	Тестування моделі та розробка рекомендацій щодо її застосування	24.11.2024 - 30.11.2024
9	Представлення пояснювальної записки	05.09.2024 — 15.09.2024
10	Представлення кваліфікаційної роботи	16.09.2024 — 30.09.2024

5. Дата видачі завдання **05 вересня 2024 року.**

Студент

Д.О.Негібов

ініціали, прізвище

підпис



Керівник роботи

Д.П.Лабенко

ініціали, прізвище

підпис



**Технічне завдання
на розробку програмного виробу
«Модель інформаційної навчальної системи для вивчення англійської
мови»**

Назва розділу	Назва і зміст підрозділу
1. Введення	1.1. Назва програмного виробу Модель інформаційної навчальної системи для вивчення англійської мови 1.2. Галузь застосування Освіта
2. Підстава для розробки	2.1. Навчальний план за спеціальністю 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка 2.2. Завдання на кваліфікаційну роботу від 12 листопада 2024 року № 4101-5/3657 (Представити як Додаток А до пояснювальної записки до кваліфікаційної роботи)
3. Призначення розробки	3.1. Мета розробки програмного виробу є модифікація існуючої моделі інформаційної навчальної системи для вивчення англійської мови. 3.2. Призначення програмного виробу: Навчання англійської мови. 3.3. Вихідні дані для розробки є матеріали для вивчення англійської мови: 1) Завдання на вибір картинки з правильним підписом 2) Тестування для визначення рівня знань користувача 3) Тестування на час 4) Аудіювання 5) Збір речення з поданих слів та почутим реченням 6) Чат системи на основі нейронної мережі
4. Технічні вимоги до програмного виробу	4.1. Вимоги до функціональних характеристик є можливість надання навчального матеріалу та можливість взаємодії користувачів з системою. 4.2. Вимоги до надійності є стабільна робота системи без відмов та помилок. 4.3. Вимоги до умов експлуатації є підтримка різних платформ та пристроїв. 4.4. Вимоги до складу і параметрів технічних засобів є мінімальні вимоги до апаратного забезпечення, які дозволяють користувачам з різних пристроїв використовувати систему без перешкод. 4.5. Вимоги до інформаційної та програмної сумісності є ОС Android v5.0 та вище та iOS 12.1 та вище. 4.6. Вимоги до маркування та упаковки немає. 4.7. Вимоги до транспортування і зберігання немає. 4.8. Спеціальні вимоги немає.

5. Вимоги до програмної документації.	Програмною документацією до виробу «Модель інформаційної навчальної системи для вивчення англійської мови» вважати: 1) Справжнє Технічне завдання на розробку програмного виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи). 2) Програму і методику випробувань розробленого програмного виробу (представити у вигляді Додатку В до пояснювальної записки до кваліфікаційної роботи).	
6. Техніко-економічні показники	Оцінка економічної ефективності не потрібна.	
7. Стадії і етапи розробки	Аналіз літератури за темою	05.09.2024-15.09.2024
	Аналіз та порівняння аналогічних існуючих систем	16.09.2024-30.09.2024
	Дослідження основних принципів роботи моделі	01.10.2024-10.10.2024
	Проектування баз даних мобільного додатку	11.10.2024-31.10.2024
	Розробка нового дизайну інтерфейсу	01.11.2024-10.11.2024
	Додавання нових вправ	11.11.2024-18.11.2024
	Програмна реалізація моделі з урахуванням нового функціоналу.	19.09.2024-23.10.2024
	Тестування моделі та розробка рекомендацій щодо її застосування	24.11.2024-30.11.2024
	Представлення пояснювальної записки	05.09.2024-15.09.2024
	Представлення кваліфікаційної роботи	16.09.2024-30.09.2024
8. Порядок контролю і приймання	1) Випробування програмного виробу відповідно до Програми і методики випробувань провести на базі комп'ютерного класу. 3) захист розробленого програмного виробу провести на засіданні атестаційної комісії.	

Виконавець

студент групи КУ-61

Негібов Д.О.



Замовник

доцент

Лабенко Д.П.



Програма і методика випробувань програмного виробу

«Модель інформаційної навчальної системи для вивчення англійської мови»

1 Об'єкт випробувань

1.1 Найменування випробуваного програмного виробу: Модель інформаційної навчальної системи для вивчення англійської мови

1.2 Область його застосування: Освіта

2. Мета випробувань

2.1 Підтвердження відповідності функціональних характеристик розробленої моделі вимогам, зазначеним у Технічному завданні.

3. Загальні положення

3.1 Підстави для проведення випробувань

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

3.2 Місце і тривалість випробувань

Приймальні випробування проводяться на базі комп'ютерного класу кафедри в період роботи атестаційної комісії.

3.3 Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї Програми і методики випробувань.

3.4 Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавця та інших осіб, присутніх на засіданні.

4. Вимоги до програми або програмного виробу

Модель повинна задовольняти наступні вимоги:

- 1) Працювати на різних платформах
- 2) Система повинна мати можливість надання навчального матеріалу та можливість взаємодії користувачів з системою.
- 3) Система повинна працювати без відмов та помилок
- 4) Вимоги до маркування та упаковки немає.
- 5) Вимоги до транспортування і зберігання немає.
- 6) Спеціальні вимоги немає.

5. Вимоги до програмної документації

Програмою документацією до виробу «Модель інформаційної навчальної системи для вивчення англійської мови» вважати:

1) Справжнє Технічне завдання на розробку програмного виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).

2) Програму і методику випробувань розробленого програмного виробу (представити у вигляді Додатку В до пояснювальної записки до кваліфікаційної роботи).

6. Засоби та порядок випробувань

6.1 Засоби випробувань

Для виконання програми потрібно запустити код програми або встановити .ark файл.

Програма сумісна з різними платформами.

6.2 Порядок проведення випробувань

Як правило, випробування проводяться в два етапи:

-ознайомчий (1-й етап);

-випробування програмного виробу (2-й етап).

Перелік перевірок, що проводяться на 1 етапі випробувань, включає в себе:

1. Перевірку комплектності програмної документації.

2. Перевірка комплектності складу програмної документації здійснюється за критерієм наявності зазначеної в ТЗ документації.

3. Перевірку комплектності складу технічних і програмних засобів.

4. Методику проведення перевірок на 1 етапі випробувань.

5. Якість програмної документації перевіряється на відповідність вимогам стандартів ЕСПД.

Перелік перевірок, що проводяться на 2 етапі випробувань, включає в себе:

1. перевірку відповідності технічних характеристик програми вимогам технічного завдання;

2. перевірку ступеня виконання функціональних вимог до програми;

3. методику проведення перевірок, що входять до переліку по 2 етапу випробувань.

1. Програма працює відповідно до умов експлуатації на різних платформах.

2. Для роботи при запуску на комп'ютері необхідний компілятор мови програмування python, версії не нижчої ніж 3.9 та встановлені бібліотеки Kivy, KivyMD, sqlalchemy, pandas, torch, sentence_transformers, а також наявність PostgreSQL. Для роботи при запуску на телефоні потрібно ОС Android v5.0 та вище або iOS 12.1 та вище

3. Порядок проведення випробувань:

3.1 Запуск програми здійснюється при наявності програмного коду викликом команди `python main.py`.

3.2 Реєстрація, а після цього авторизація у системі.

3.3 Тестування кожної з наявних вправ та запис результату тестування на час у рейтингову таблицю. Тестування коректності відображення інформації у екрані “Бібліотека”. Тестування чату системи.

Для проведення випробувань пропонується тест 1, тест 2, тест 3

Тест 1:

«- Запуск здійснюється при наявності програмного коду викликом команди `python main.py`

- Відображається екран реєстрації та авторизації
- Натискаємо на кнопку “Не маєте облікового запису? Зареєструйтесь”
- Відображається екран реєстрації де вводимо свої дані
- Знову відображається екран авторизації де вводимо дані що ввели при реєстрації
- Відображається головний екран з наявними правами
- Натискаємо на вправу “Знайдіть пару”
- Обираємо категорію “Хобі”
- Обираємо 1 правильну з 4 наданих картинок з відповідними підписами
- Бачимо що система візуально показує що відповідь правильна
- Натискаємо на кнопку “Далі”
- Обираємо 1 з 4 картинок що має не правильний підпис
- Бачимо що система візуально показує що відповідь не правильна та зеленим кольором виділила правильний варіант
- Натискаємо два рази на стрілку у лівому верхньому куті, що повертає нас на головний екран
- Натискаємо на кнопку “Перевірте свої знання”
- Натискаємо на кнопку “Розпочати тест”
- Відповідаємо на надані питання
- Натискаємо на кнопку “Закінчити спробу” та бачимо відповідний результат роботи
- Повертаємось на головний екран
- Натискаємо на кнопку “Тест на час”
- Обираємо часовий ліміт 30 секунд
- Відповідаємо на наданні питання
- Бачимо відповідний результат
- Повертаємось на головний екран та переходимо на екран рейтингу натискаючи на передостанню кнопку у нижні плашці екрану
- Бачимо що результат додався до рейтингової таблиці
- Повертаємось до головного екрану
- Натискаємо на кнопку “Аудіювання”
- Обираємо рівень завдання A1
- Натискаємо на кнопку прослухати та чуємо аудіо файл
- Після закінчення аудіо файлу натискаємо на кнопку “Перейти до питань”
- Відповідаємо на подані питання та натискаємо на кнопку “Перевірити відповіді”
- Бачимо що система візуально показує наявні помилки та правильні варіанти відповідей
- Повертаємось до головного екрану

- Натискаємо на вправу “Збери речення”
- Натискаємо на кнопку “Почати”
- Натискаємо на кнопку “Прослухати” та чуємо аудіо файл
- Збираємо коректно почуте речення з поданих слів та натискаємо на кнопку “Перевірити”
- Бачимо що система візуально показує що відповідь правильна
- Натискаємо на кнопку “Далі”
- Натискаємо на кнопку “Прослухати” та чуємо аудіо файл
- Збираємо неправильно почуте речення з поданих слів та натискаємо на кнопку “Перевірити”
- Бачимо що система візуально показує що відповідь неправильна»

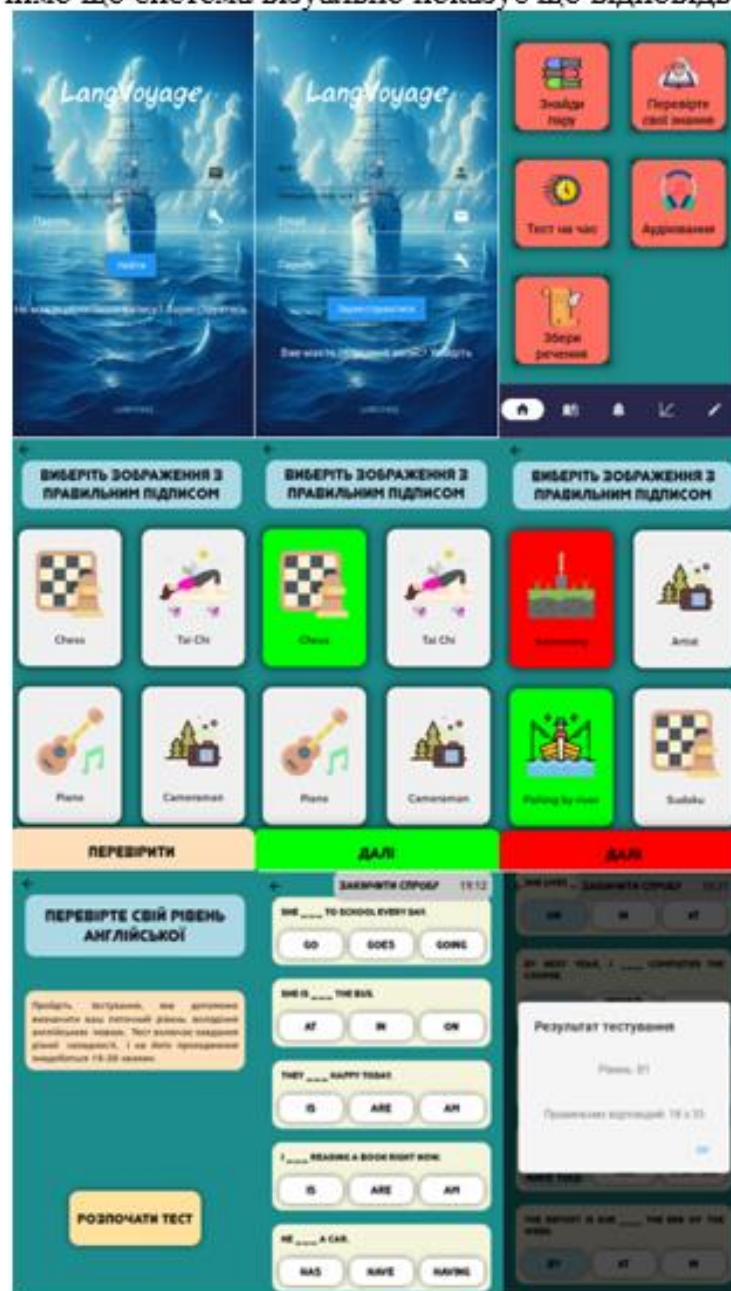


Рисунок В.1 – Тест 1



Рисунок В.2 – Тест 1

Тест вважається пройденим, якщо відбуваються вказані операції і їх відображення у програмному продукті.

Тест 2:

«- Запуск здійснюється при наявності програмного коду викликом команди `python main.py`

- Відображається екран реєстрації та авторизації
- Натискаємо на кнопку “Не маєте облікового запису? Зареєструйтесь”
- Відображається екран реєстрації де вводимо свої дані

- Знову відображається екран авторизації де вводимо дані що ввели при реєстрації
- Відображається головний екран з наявними правами
- Натискаємо на другу кнопку на нижній плашці екрану
- Натискаємо на кнопку “Час і часові конструкції”
- Обираємо “Future Continuous”
- Бачимо коректне відображення інформації
- Натискаємо два рази на стрілку у лівому верхньому куті, що повертає нас на екран “Бібліотеки”
- Натискаємо на кнопку “Граматичні структури”
- Бачимо коректне відображення інформації
- Натискаємо на стрілку у лівому верхньому куті, що повертає нас на екран “Бібліотеки”
- Натискаємо на кнопку “Вираження уподобань”
- Бачимо коректне відображення інформації
- Натискаємо на стрілку у лівому верхньому куті, що повертає нас на екран “Бібліотеки”
- Натискаємо на кнопку “Спеціальні фрази слова”
- Бачимо коректне відображення інформації»



Рисунок В.3 – Тест 2

Тест вважається пройденим, якщо відбуваються вказані операції і їх відображення у програмному продукті.

Тест 3:

«- Запуск здійснюється при наявності програмного коду викликом команди `python main.py`

- Відображається екран реєстрації та авторизації
- Натискаємо на кнопку “Не маєте облікового запису? Зареєструйтесь”
- Відображається екран реєстрації де вводимо свої дані
- Знову відображається екран авторизації де вводимо дані що ввели при реєстрації
- Відображається головний екран з наявними правами
- Натискаємо на останню кнопку на нижній плашці екрану
- Пишемо запит “Що таке Present Perfect?” та натискаємо на кнопку з трикутником щоб відправити
- Отримуємо відповідь та пишемо запит “Що таке модальні дієслова?”
- Отримуємо відповідь та пишемо запит “Коли використовується much і many?”
- Отримуємо відповідь та пишемо запит “Яка 1 форма дієслова learned?”
- Отримуємо відповідь та пишемо запит “Яка третя форма дієслова made?”
- Отримуємо відповідь та пишемо запит “Яка 1 та 2 форма дієслова chosen?”
- Отримуємо відповідь»



Рисунок В.4 -Тест3

Тест вважається пройденим, якщо відбуваються вказані операції і їх відображення у програмному продукті.

Висновки: тест 1 успішно пройшов випробування, тест 2 успішно пройшов випробування і тест 3 успішно пройшов випробування. Випробування пройшло успішно.

Виконавець: студент групи КУ-61, Негібов Д.О

