



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

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**  
на тему:  
«Інформаційна технологія автоматизації онлайн-навчання»


Виконала: студентка 2-го курсу,  
групи 1КН-21м,  
спеціальності 122 – «Комп'ютерні  
науки»

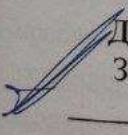
  
Дідик Я. В.  
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

  
Крилик Л. В.  
(прізвище та ініціали)  
« 15 » 12 2022 р.

Опонент: к.т.н., доцент каф. АІТ

  
Богач І. В.  
(прізвище та ініціали)  
« 15 » 12 2022 р.

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

« 16 » 12 2022 р.

Вінниця - 2022 року

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

ЗАТВЕРДЖУЮ  
 Директор ГОВ «АТЛАС»  
 Пашенко О.П.  
 (підпис)  
 « 15 » 12 2022 року



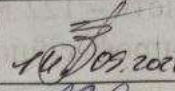
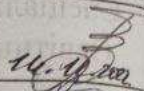
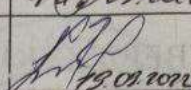
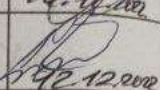
ЗАТВЕРДЖУЮ  
 Завідувач кафедри КН  
 д.т.н. проф. Яровий А.А.  
 (підпис)  
 « 14 » 09 2022 року

### ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Дідик Яні Вікторівні  
 (прізвище, ім'я, по батькові)

- Тема роботи: «Інформаційна технологія автоматизації онлайн-навчання»  
 Керівник роботи к.т.н., доц., доц. каф. КН, Крилик Л. В.  
 затверджені наказом вищого навчального закладу від «14» 09 2022 року №203
- Строк подання студентом роботи 18 листопада 2022 року
- Вихідні дані до роботи: мова програмування – об'єктно-орієнтована, кількість питань до користувача – не менше 7, мінімальна кількість записів в базі даних – 50 од., максимальна кількість символів опису заявки – 270 од.; час від подачі заявки до збереження її в базі даних не перевищує 45 секунд; тип інтерфейсу користувача - інтуїтивно зрозумілий.
- Зміст текстової частини: вступ, обґрунтування доцільності розробки інформаційної технології автоматизації онлайн-навчання, моделювання інформаційної технології автоматизації онлайн-навчання, програмна реалізація інформаційної технології автоматизації онлайн-навчання, економічна части висновки, список використаних джерел, додатки.
- Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): схема алгоритму функціонування клієнтського модуля інформаційної технології автоматизації онлайн-навчання; схема алгоритму функціонування модуля адміністратора інформаційної технології автоматизації онлайн-навчання; ER-діаграма автоматизованого онлайн-навчання; загальна структурна схема функціонування системи; діаграма діяльності сервера для проведення реєстрації; діаграма діяльності сервера для вибору онлайн-курсу; діаграма діяльності сервера для роботи адміністратора з навчальними матеріалами; загальна UML-діаграма класів; загальний вигляд інтерфейсного вікна роботи програми.

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-3	Крилик Л.В., к.т.н., доц. каф. КН	 10.09.2022	 14.09.2022
4	Нікіфорова Л. О., к.е.н, доц. каф. ЕПВМ	 19.09.2022	 02.12.2022

7. Дата видачі завдання 14.09 2022 року

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Обґрунтування доцільності розробки	14.09.2022р — — 1.10.2022р	Розділ 1
2	Моделювання інформаційної технології автоматизації онлайн-навчання	2.10.2022р — — 16.10.2022р	Розділ 2
3	Програмна реалізація інформаційної технології автоматизації онлайн-навчання	17.10.2022р — — 07.11.2022р	Розділ 3
4	Підготовка економічної частини	08.11.2022р — — 21.11.2022р	Розділ 4
5	Апробація результатів дослідження	23.11.2022р — 01.12.2022р	тези доповіді, довідка про впровадження розробки
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	02.12.2022р — — 14.12.2022р	Пояснювальна записка, графічний матеріал, презентація

Студентка

Керівник роботи

(підпис)

(підпис)

(прізвище та ініціал)

Дідик Я.

(прізвище та ініціал)

Крилик Л.

## АНОТАЦІЯ

УДК 004.42

Дідик Я. В. Інформаційна технологія автоматизації онлайн-навчання. Магістерська кваліфікаційна робота зі спеціальності 122 – Комп'ютерні науки, освітня програма – Системи штучного інтелекту. Вінниця: ВНТУ, 2022. 124 с.

Укр. мовою. Бібліогр.: 46 назв; рис.: 21; табл. 8.

У магістерській кваліфікаційній роботі проведено аналіз предметної галузі онлайн-навчання. Досліджено основні підходи щодо автоматизації онлайн-навчання. Виконано аналіз аналогів, розглянуто методи та технології, які використовуються для вирішення подібних задач. Розроблено структурну модель інформаційної технології, охарактеризовано основні та другорядні вимоги до створення подібних систем.

Дана робота допомагає організувати такі процеси, як набір та реєстрація на онлайн-курси та безпосереднього онлайн-навчання студентів, допомога з вибором потрібного онлайн-курсу та визначення існуючого рівня знань потенційного студента, при цьому не маючи прив'язки до дня чи періоду доби, графіку викладачів, особистих обставин студентів.

В роботі приведені результати досліджень та аналіз роботи інформаційної технології автоматизації онлайн-навчання. Доведена економічна доцільність розробки.

Ілюстративна частина складається з 9 плакатів із результатами моделювання.

Ключові слова: онлайн-навчання, автоматизація реєстрації та оплати, інформаційна система, чат-бот, штучний інтелект, експертна система, месенджер Telegram, Viber, Facebook Messenger, інструментальні засоби.

## ABSTRACT

Didyk Ya.V. Information technology for online learning automation. Master's work in specialty 122 – Computer science. Vinnytsia: VNTU, 2022. 124 p.

In Ukrainian language. Bibliogr .: 46 titles; Fig.: 21 ; 8 table.

In the master's qualification thesis, an analysis of the subject area of online education was carried out. The main approaches to the automation of online learning have been studied. An analysis of analogues was carried out, methods and technologies used to solve similar problems were considered. A structural model of information technology has been developed, the main and secondary requirements for the creation of such systems have been characterized.

This work helps to organize such processes as recruitment and registration at online courses and direct online training of students, help with choosing the right online course and determining the existing level of knowledge of a potential student, while not being tied to the day or period of the day, the schedule of teachers, personal circumstances of students.

The work presents the results of research and analysis of the information technology of online learning automation. The economic feasibility of the development has been proven.

The illustrative part consists of 8 posters with simulation results.

Keywords: online training, registration and payment automation, information system, chatbot, artificial intelligence, expert system, Telegram messenger, Viber, Facebook Messenger, tools.

## ЗМІСТ

ВСТУП.....	6
<b>1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АВТОМАТИЗАЦІЇ ОНЛАЙН-НАВЧАННЯ .....</b>	<b>10</b>
1.1 Аналіз відомих програмних систем-аналогів.....	13
1.2 Методи та програмні засоби для створення інформаційної технології автоматизації онлайн-навчання .....	16
1.3 Постановка задачі дослідження.....	21
1.4 Висновок до розділу 1.....	24
<b>2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АВТОМАТИЗАЦІЇ ОНЛАЙН-НАВЧАННЯ.....</b>	<b>24</b>
2.1 Обґрунтування вибору методів обробки природньої мови для інформаційної технології автоматизації онлайн-навчання.....	24
2.2 Розробка ER-моделі бази даних автоматизації онлайн-навчання та обґрунтування вибору бази даних.....	28
2.3 Математична модель прийняття рішень у діалозі агента з користувачем.....	35
2.4 Висновки до розділу 2.....	38
<b>3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АВТОМАТИЗАЦІЇ ОНЛАЙН-НАВЧАННЯ.....</b>	<b>39</b>
3.1 Обґрунтування вибору інструментів для реалізації інформаційної технології автоматизації онлайн-навчання.....	39
3.2 Розробка алгоритму функціонування інформаційної технології автоматизації онлайн-навчання.....	55
3.3 Обґрунтування вибору інструменту для реалізації експертної системи для інформаційної технології автоматизації онлайн-навчання .....	59
3.4 Реалізація програмних модулів інформаційної технології автоматизації онлайн-навчання .....	62

3.5 Тестування інформаційної технології автоматизації онлайн-навчання та аналіз отриманих результатів .....	69
3.6 Висновок до розділу 3 .....	75
4 ЕКОНОМІЧНА ЧАСТИНА .....	77
4.1 Комерційний та технологічний аудит науково-технічної розробки .....	77
4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько-технологічної роботи.....	81
4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором.....	86
4.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.....	87
4.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.....	88
4.4 Висновок до розділу 4 .....	91
ВИСНОВКИ.....	92
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	94
Додаток А (обов'язковий) Результат перевірки на плагіат в онлайн-системі UNICHECK.....	99
Додаток Б (обов'язковий) Лістинг програми .....	100
Додаток В (обов'язковий) Ілюстративна частина.....	110
Додаток Г (довідниковий) Інструкція користувача.....	116
Додаток Д (довідниковий) Довідка про участь в конкурсі студентських наукових робіт.....	121
Додаток Е (довідниковий) Довідка про впровадження розробки.....	122

## ВСТУП

**Актуальність теми.** Станом на сьогоднішній час, Всесвітня мережа Інтернет, все більше бере на себе комунікаційні функції. Глобальна мережа стає своєрідним середовищем спілкування, яке займає важливе місце в усіх сферах життєдіяльності суспільства.

З розвитком сучасних технологій, функції Всесвітньої мережі, як комунікатора, набувають нових форм. Особливо актуальними зараз є комунікаційні технології, які базуються на месенджерах і чат-ботах.

Згідно з дослідженнями, користувачами використовуються смартфони для комунікації через месенджери частіше, ніж для пошуку інформації, телефонних дзвінків чи інших можливих цілей. Тому компанії прагнуть збільшення потоку онлайн-клієнтів створюють чат-боти, щоб інтегрувати їх в месенджери. На сьогодні, такі засоби спілкування глобально використовуються користувачами для вирішення різних задач, а також для клієнтської взаємодії з компаніями, пошуку товарів, використання контенту тощо [1].

У теперішній час боти створюють для ділового та особистого користування. Вони використовуються в розважальних або інформаційних цілях (чат-боти можуть надати інформацію про погоду, зорієнтувати користувача з приводу курсу валют, записати на прийом чи провести реєстрацію), для спілкування з клієнтами чи користувачами та в службах підтримки.

Фундаментом такої інтернет-комунікаційної програми, як чат-бот, є технології штучного інтелекту та нейронних мереж, що дозволяє зробити комунікації чат-ботів ближчими до людського віртуального спілкування і вирішувати з їх допомогою комерційні та маркетингові задачі.

Звідси можна зробити висновок, що створення інформаційної технології автоматизації онлайн-навчання, основним компонентом якої є інтелектуальний чат-бот, в сучасних реаліях є достить **актуальним**, оскільки кількість людей, які надають перевагу віртуальному навчанню, зросла порівняно зі звичайним стаціонарним.



**Зв'язок роботи з науковими програмами, планами, темами.** Магістерська кваліфікаційна робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Розробка прикладних інтелектуальних інформаційних технологій та систем».

**Мета і завдання дослідження.** Метою магістерської кваліфікаційної роботи є розширення функціональних можливостей інформаційної технології автоматизації онлайн-навчання, що дозволить автоматизувати процес навчання, який підвищить ефективність освітнього процесу.

Для досягнення поставленої мети потрібно розв'язати наступні **завдання**:

- провести аналіз технічного рівня чат-ботів для онлайн-курсів та обґрунтувати доцільність розробки інформаційної технології автоматизації онлайн-навчання;
- розробити математичну модель прийняття рішень для діалогу агента з користувачем;
- розглянути методи та технології для розробки чат-ботів;
- спроектувати архітектуру та структуру програмного модуля чат-бота для навчальних онлайн-курсів;
- проаналізувати та обґрунтувати вибір інструментів для розробки;
- виконати програмну реалізацію програмного модуля чат-бота для онлайн-курсів;
- виконати тестування програми та провести аналіз отриманих результатів;
- економічно обґрунтувати доцільність розробки інформаційної технології автоматизації онлайн-навчання.

**Об'єкт дослідження** – процес автоматизації онлайн-навчання.

**Предметом дослідження** є програмні засоби проектування та реалізації, алгоритми спілкування та отримання інформації в месенджерах.

**Методи дослідження.** У роботі використані такі методи наукових досліджень: аналіз інформаційної технології автоматизації онлайн-навчання, методи обробки цифрової інформації, методи реалізації експертної системи для підбору курсу та визначення поточного рівня знань, методи взаємодії між серверами, методи

комунікації з базою даних, методи інтеграції з месенджерами, методи об'єктно-орієнтованого програмування для отримання подальших результатів роботи програми.

**Наукова новизна одержаних результатів** полягає в наступному: удосконалено інформаційну технологію автоматизації онлайн-навчання, яка відрізняється від аналогів наявністю інтегрованої експертної системи для допомоги у підборі виду навчального онлайн-курсу та визначенні рівня знань користувача. Це збільшує ефективність набору студентів; наявність двох гілок для роботи із системою: гілки користувача та гілки адміністратора, що сприятиме проведенню потрібних маніпулювань даними.

**Практичне значення одержаних результатів** полягає у наступному:

1. Розроблено алгоритм функціонування інформаційної технології автоматизації онлайн-навчання.
2. Розроблено діаграми діяльності сервера для гілок адміністратора та користувача інформаційної технології автоматизації онлайн-навчання.
3. Здійснено програмну реалізацію інформаційної технології автоматизації онлайн-навчання.

**Достовірність теоретичних положень магістерської кваліфікаційної роботи** підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, чітким та лаконічним виведенням аналітичних співвідношень, порівнянням результатів з відомими аналогами та збіжністю результатів математичного моделювання з результатами, які отримані під час впровадження розроблених програмних засобів.

**Особистий внесок магістранта.** Результати магістерської кваліфікаційної роботи отримані самостійно. В публікації у співавторстві здобувачу належить дослідження особливостей розробки інтелектуального чат-бота інтегрованого з месенджером Telegram для онлайн-курсів та удосконалення функціоналу інтелектуального чат-бота для автоматизації онлайн-навчання [1, 2].

**Апробація результатів роботи.** Результати досліджень було апробовано на LI науково-технічній конференції підрозділів Вінницького національного

технічного університету (НТКП ВНТУ–2022) м. Вінниці у 2022 р. [2], а також в I турі Всеукраїнського конкурсу студентських наукових робіт у ВНТУ 2021/2022 н.р. За підсумками I-го туру наукова робота була рекомендована для участі у II-му турі конкурсу і направлена до Центральноукраїнського національного технічного університету, м. Кропивницький 2022 р.

**Публікації.** За результатами магістерської кваліфікаційної роботи опубліковано тези доповідей на науково-технічних конференціях [1, 2] та отримано свідоцтво про реєстрацію авторського права на твір у формі комп'ютерної програми – «Інформаційна технологія автоматизації онлайн-навчання з допомогою інтелектуального чат-бота» [3].

Результати роботи пройшли експериментальне випробування та впровадженні для подальшого використання ТОВ «ATLAS.UA» м. Київ.

## **1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АВТОМАТИЗАЦІЇ ОНЛАЙН НАВЧАННЯ**

У сучасному світі застосування чат-ботів для забезпечення віртуального спілкування бізнесу зі своїми потенційними та існуючими користувачами є актуальною задачею, що надає можливість швидкого обміну інформацією між бізнесом і клієнтами в сучасних месенджер-чатах для розв'язання будь-яких питань консультування, можливості одночасного обслуговування десятків тисяч клієнтів, доступності 24/7 та інших переваг. Для різних компаній, що існують у бізнес-середовищі, важливим є своєчасне та швидке забезпечення користувачів необхідною інформацією. Саме тут у нагоді стає технологія чат-ботів, яку багато експертів вважають майбутнім сфери обслуговування та управління клієнтами.

Крім того, найкращі чат-боти будують модель поведінки, максимально наближену до людської, що дозволяє зробити комунікації чат-ботів ближчими до людського віртуального спілкування і вирішувати з їх допомогою комерційні, маркетингові задачі та підвищувати свій професійний рівень.

Чат-боти є найшвидшим інструментом для того, щоб повідомляти користувачів про все, що відбувається чи має відбуватися в ході навчання. Боти виступають як віртуальні співрозмовники, з допомогою яких реалізується автоматизація спілкування з клієнтами, а також виконуються аналітичні функції.

Основною сферою застосування є підтримка та супровід користувачів переважно однієї чи декількох обраних предметних сфер. Використання штучного інтелекту у чат-ботах дозволяє створювати інтелектуальні додатки, які розуміють природню мову та взаємодіють з користувачами. Ми можемо навчити бота вивчати вхідні дані і відповідати на різного роду запитання. Інтелектуальна служба розпізнавання мови призначена не тільки для синтаксичного аналізу, вона дає пряму відповідь на деякі сценарії програми, пов'язані з розумінням мови і дозволяє використовувати програмний код.

Розробка автоматизованої системи реєстрації, навчання та їх безпосередній моніторинг потребує якісного деплоюменту програмного продукту, який використовуватиметься користувачами, не зважаючи на тип пристрою чи якість інтернет-з'єднання. Вагомою перевагою віртуального асистента над іншими системами-аналогами – це можливість використання інтелектуального чат-бота одночасно на декількох популярних месенджерах та розподілу ролей для різних типів користувачів [2].

Розроблюваний чат-бот орієнтований на користувачів або компанії, які займаються організацією та проведенням курсів. З допомогою інтелектуального чат-бота вони можуть організувати процес набору та реєстрації на курси та безпосереднього навчання студентів, при цьому не маючи прив'язки до дня чи періоду доби, графіку викладачів, особистих обставин студентів.

Розроблювана інформаційна технологія автоматизації онлайн-навчання дасть можливість користувачам використовувати особистого навчального асистента на основі таких відомих месенджерів як Telegram, Viber, Facebook Messenger. При чому, навчання не прив'язане до одного месенджера, користувач в будь-який момент може продовжити навчання у будь-якому із запропонованих вище обмінників, так боти у кожному месенджері оперують даними однієї бази.

Чат-бот – інформаційна система, переважно у вигляді програми із можливостями обробки вхідної інформації та наданням на її основі необхідної інформації. Основною сферою застосування є підтримка та супровід користувачів переважно однієї чи декількох обраних предметних галузей.

Застосування штучного інтелекту у чат-ботах дає можливість розробляти інтелектуальні додатки, які здатні розуміти природню мову та колаборувати з користувачами. Боти здатні аналізувати вхідні дані і давати відповіді на запитання різного роду. Інтелектуальна служба розпізнавання мови призначена не тільки для синтаксичного аналізу, вона дає пряму відповідь на деякі сценарії програми, пов'язані з розумінням мови, і дозволяє використовувати програмний код.

Боти на основі штучного інтелекту здатні активно навчатися, завдяки чому з'являється можливість вивчати та коригувати невідомі для системи речення. Вони розуміють не тільки слова, а й контекст речення.

При виборі типу чат-бота для більш глобальної інтелектуальної інформаційної системи, основна увага приділяється предметній галузі в якій має розумітися чат-бот. Відповідно до цього ступінь розуміння та складність імплементації сховища даних прямо-пропорційно залежить від термінологічної бази предметної галузі.

Розробка чат-бота зі штучним інтелектом вимагає вагомого людського внеску в роботу на кожному її етапі. Основна задача при створенні ботів – навчити розуміти контекст звернень користувача і будувати модель діалогу [3].

До основних переваг чат-ботів відносять:

- робота без прив'язки до періоду доби чи дня тижня;
- спілкування з клієнтами без участі працівників, що допомагає компаніям мінімізувати витрати;
- зменшення рівня навантаження на менеджерів-консультантів, які спілкуються з клієнтами на гарячих лініях і в онлайн-чатах;
- одночасне паралельне спілкування з деякою кількістю користувачів;
- оперативний відгук на всі запити;
- спроможність ведення діалогів різними мовами;
- вміння виявити індивідуальний підхід до користувачів завдяки збереженню діалогу.

З недоліків чат-ботів можна виділити таке:

- можливі проблеми з розумінням суті запиту користувача. Боти здатні розпізнавати лише прості і коректно сформульовані запити;
- відсутність емоцій. Віртуальні помічники не можуть забезпечити клієнта відчуттям «живого» діалогу, на відмінну від менеджера, який проявляє людську небайдужість;

– у випадку, коли користувач допускає орфографічні чи синтаксичні помилки, бот може некоректно розпізнавати контекст запиту і, як наслідок, неправильно відреагувати на нього [4].

### **1.1 Аналіз відомих програмних систем-аналогів**

У роботі чат-бота для автоматизації онлайн-навчання можливі такі сценарії:

- вибір курсу для навчання;
- реєстрація користувача на онлайн-курс;
- вибір тарифу та проведення оплати;
- навчання та тестування;
- інформування про навчальні курси;
- збереження прогресу навчання;
- авторизація користувача;
- вибір гілки користувача чи адміністратора для роботи з чат-ботом.

Розглянемо декілька прикладів чат-ботів, вже існуючих на ринку та їх основні сценарії.

HomeOnlineEnglishBot – це чат-бот створений для вивчення англійської мови з можливістю використовувати аудіозаписи, графічні матеріали, виконувати завдання з подальшою їх перевіркою. Чат-бот дає можливість оплатити навчання, зв'язатися з викладачем, ознайомитися з відгуками. Розрахований на два рівні: для дітей та для дорослих. Матеріали у платформі оновлюються один раз у 1,5 місяці (рис. 1.1).

Алгоритм зв'язку:

1. Привітання. Вивчення можливостей роботи з чат-ботом;
2. Користувач має можливість обрати онлайн-курс, який його цікавить, отримати пробне заняття по ньому або відразу оплатити навчання;
4. Користувач отримує доступ до уроків, які розділені на теми. Після вивчення матеріалу користувач має пройти загальний тест.
5. При необхідності користувач може зв'язатися з викладачем у чаті.

Недоліками додатку є відсутність можливості тестування рівня знань перед вибором курсу та відсутність збереження прогресу навчання. Також описаний вище навчальний чат-бот функціонує лише на месенджері Telegram.

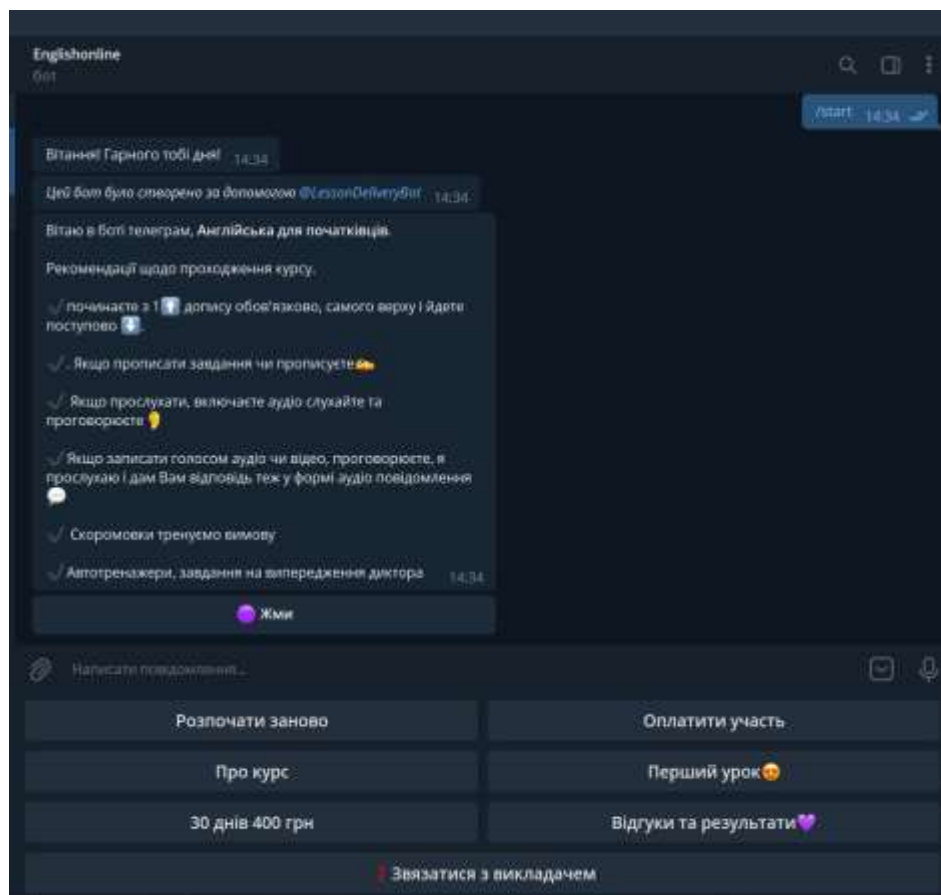


Рисунок 1.1 – Загальний вигляд інтерфейсного вікна чат-боту HomeOnlineEnglishBot

ProgAcademyBot – це навчальна академія, яка орієнтована на вивчення ІТ-професій. Ними був розроблений чат-бот, який виступає у ролі техпідтримки. Його основною метою є ознайомлення користувача з наявними курсами академії та зацікавлення у придбанні курсу (рис. 1.2).

Алгоритм зв'язку:

1. Привітання. Опис роботи та функцій чат-бота;
2. Чат-бот надсилає перелік курсів, які надає компанія та короткий опис до них;
3. Після уточнення технології, користувач отримує декілька відео-уроків по обраній технології;



4. Після ознайомлення з відео-уроками користувачеві пропонується купити навчання в академії.

Серед недоліків є те, що цей бот не дає повної автоматизації навчання, а лише ознайомлює з його можливостями, даючи тільки декілька уроків в автоматизованому режимі. Про подальшу прив'язку до часу та групи чат-бот не повідомляє, чим вводить в оману користувача, даючи привід думати, що навчання буде продовжуватися в першочерговому форматі. Також бот не дає можливості провести оплату навчання.

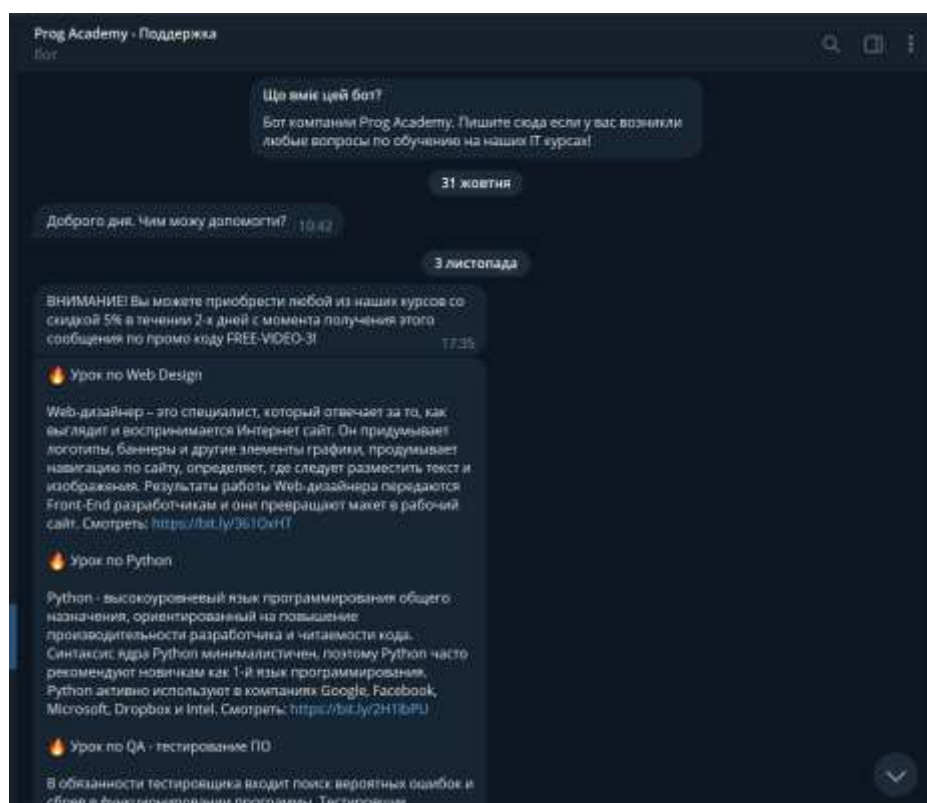


Рисунок 1.2 – Загальний вигляд інтерфейсного вікна чат-боту ProgAcademyBot

InMind – чат-бот месенджера Telegram, для вивчення слів англійської мови з можливістю налаштування рівнів складності (рис. 1.3).

Алгоритм зв'язку:

1. Стартове вікно з описом роботи чат-бота;
2. Після натискання кнопки «Start» надається доступ до функцій чат-бота;
3. Користувач може обрати рівень складності вивчення серед варіантів

«Легкий», «Середній», «Високий рівень»;

4. Користувач має можливість переглянути статистику, змінити налаштування.

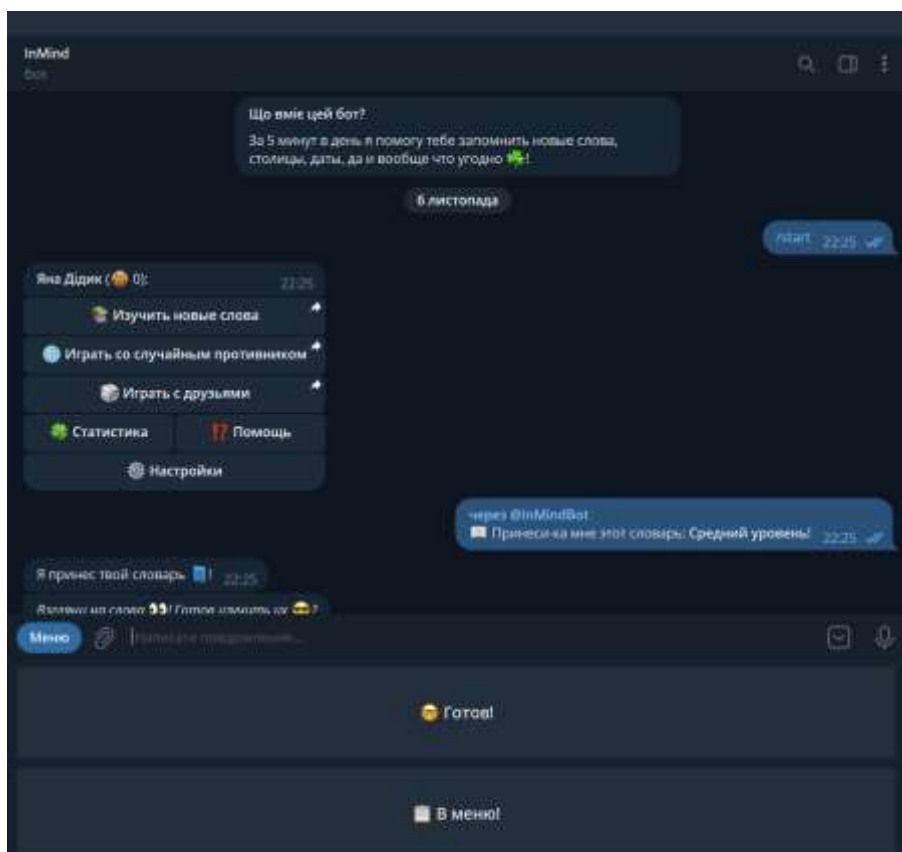


Рисунок 1.3 – Загальний вигляд інтерфейсного вікна чат-бота InMind

Недоліком чат-бота InMind є те, що бот не розпізнає в повідомленні контекст, має нагромаджений інтерфейс, який погіршує сприйняття.

## 1.2 Методи та програмні засоби для створення інформаційної технології автоматизації онлайн-навчання

Типовий цикл роботи будь-якого чат-бота можна представити ланцюжком таких дій:

- надходження запиту від користувача;

- аналіз запиту – ознайомлення з висловлюванням, його розуміння і визначення намірів клієнта;
- виконання дій за попередньо визначеним скриптом по обробці клієнтського кейса;
- генерація відповіді на природній мові;
- збереження запиту, контексту і параметрів діалогу для обробки наступних звернень;
- відправка відповіді клієнту.

Найвагомішим етапом роботи є аналіз клієнтського запиту. Чат-боти, що навчаються на базі Machine Learning використовують для цього методи NLU і NLP. Наприклад, для текстових чат-ботів процес розбору включає такі етапи:

- попередня обробка тексту – токенізація (сегментація, розбиття на слова), коректування помилок, лематизації і стемінг (визначення нормальної форми слів і частин мови), відкидання стоп-слів (артиклі, вигуки, сполучники тощо), розширення запиту шляхом роботи з наборами синонімів, доповнення інформацією про значимість окремих слів, а також визначення іменованих сутностей;
- класифікація запиту на основі прикладів фраз і ML-алгоритмів або формальних правил (шаблонів) згідно з поточним контекстом бесіди;
- отримання параметрів запиту з фрази користувача.

Необхідно зазначити, що донедавна при розробці ботів, в основному, застосовувався метод, базований на rule-based (з англійської означає «формальне правило»), суть якого заключається у виокремленні семантично важливих складових фраз та їх подальшій кодифікації. Слідом, на базі отриманих результатів створювався певний сценарій діалогу скриптовими мовами програмування, такими як наприклад, JavaScript, PHP чи Python. Однак, починаючи з 2016 року в наслідок стрімкого розвитку алгоритмів семантичної схожості текстів, методик синтезу і розпізнавання мови, а також Big Data і Machine Learning, з'явилися нові підходи до класифікації текстів та навчання NLU-систем. Так, переважна більшість теперішніх чат-ботів розроблені на базі останніх досягнень в сфері Data Science: NLU і NLP-

техніки, методи розпізнавання мови і обробки текстів за допомогою неймереж та інших інструментів штучного інтелекту [5].

Зазвичай чат-боти інтегруються месенджери, хоча використання їх на веб-сайтах в якості консультанта теж є не менш популярним. Месенджери зараз є найбільш звичним та простим засобом для комунікацій. Орієнтуючись на статистику, можна зробити висновок, що в теперішньому світі людина проводить близько 5 годин на добу зі смартфоном в руках. Більша частина цього часу припадає на комунікацію в месенджерах і соціальних мережах.

Платформа для чат-бота – це програмний додаток, який працює з використанням API-ключа з допомогою якого реалізується віртуальний бот-співрозмовник. Натепер чат-боти підтримуються більшістю популярних месенджерів, серед яких Facebook Messenger, Viber, Whatsapp, Skype, Telegram, Instagram. На рис. 1.4 показано статистику використання вказаних вище месенджерів користувачами станом на грудень 2021 року.

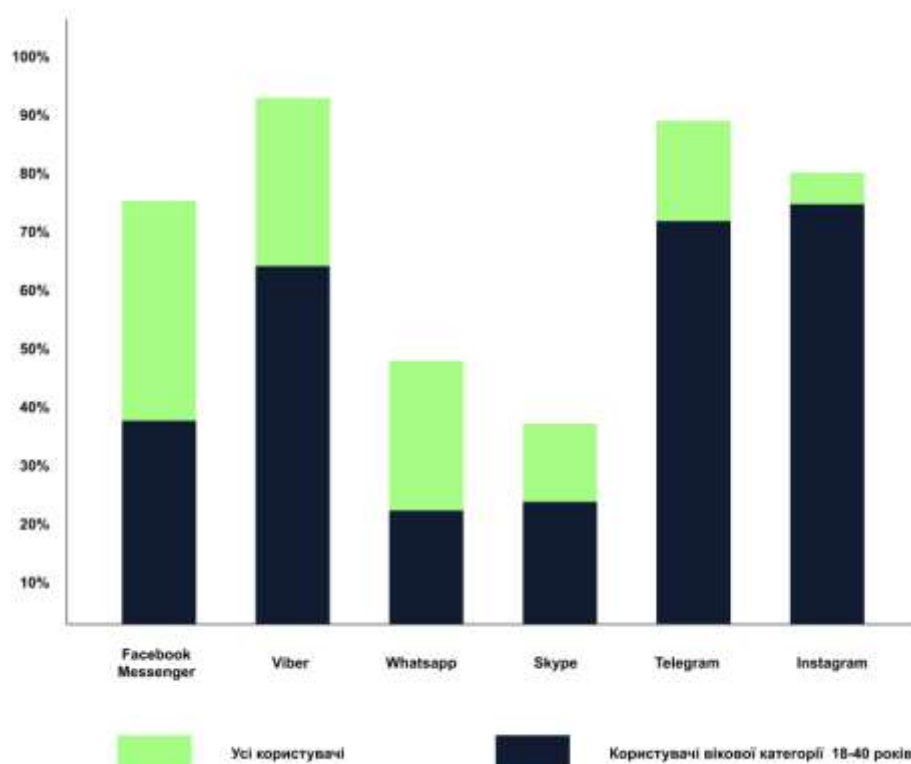


Рисунок 1.4 – Статистична діаграма використання месенджерів користувачами за 2021 рік

Facebook Messenger – популярна система для обміну миттєвими повідомленнями, створена компанією Meta, інтегрована з додатком на основному сайті Facebook і побудована на базі відкритого протоколу MQTT.

Додаток, крім обміну повідомленнями використовується для обміну текстовими повідомленнями, медіаданими та для групових чатів. Додаток використовують для комунікації з друзями у Facebook та з контактами, збереженими у смартфоні користувача [6]. Facebook Messenger користуються 74,4% українців, 48% з яких віком від 18 до 40 років.

Viber – мобільний додаток обміну повідомленнями. Ним активно користуються для обміну повідомленнями, файлами (фото, відео, аудіо) і посиланнями, а також для дзвінків і відеодзвінків. Програма адаптована під різні пристрої (стаціонарні ПК, ноутбуки, планшети і смартфони) і під різні операційні системи. Додаток не потрібно що разу повторно запускати, він працює у фоновому режимі і майже не споживає трафік та ресурси акумулятора телефону. Основна перевага – хороша якість зв'язку навіть у випадку низької швидкості інтернет-з'єднання [7]. Viber користуються 92,3% українців, 68% з яких віком від 18 до 40 років.

Whatsapp – безкоштовна платформа для миттєвого обміну текстовими повідомленнями, фотографіями, відео, аудіозаписами та здійснення відео та аудіодзвінків. Програма дозволяє користувачам бути постійно на зв'язку. Основною вимогою для використання функцій Whatsapp є наявність швидкого інтернету на пристрої, не приймаючи до уваги Wi-Fi чи мобільні дані. Завдяки систематичному оновленню системи, користувач отримує доступ до нових опцій. Крім звичайного функціоналу, в Whatsapp є додаткові функції, які дають можливість підвищити навички текстового спілкування. До корисних доповнень можна віднести можливість маскуванню статусу профілю, функція «невидимки» при використанні додатку, відстежування дій інших користувачів, редагування тексту повідомлення перед відправкою, використання Whatsapp на ПК та інше [8]. Whatsapp користуються 46,8% українців, 43% з яких віком від 18 до 40 років.

Skype – це система, яка дає змогу в будь-який час спілкуватися з іншими користувачами з використанням смартфона, комп'ютера або пристроїв Alexa. Програма розроблена для приватного та групового спілкування й підтримується на мобільних пристроях, ПК, а також пристроях Xbox і Alexa. У Skype є можливість обміну повідомленнями й спілкуватися з використанням аудіо і відеовикликів у якості HD з максимальною кількістю учасників без оплати користування, яка становить 50 користувачі. Здійснивши оплату на певну суму, користувач отримує більше можливостей [9]. Skype користуються 35% українців, 60% з яких віком від 18 до 40 років.

Telegram – простий, швидкий та безпечний додаток для обміну повідомленнями. Telegram входить до складу світових лідерів серед програмних додатків, які найчастіше завантажують, кількість його користувачів складає понад 500 мільйонів. Завдяки унікальній розподіленій мережі інформаційних центрів, Telegram доставляє повідомлення оперативніше за будь-які інші месенджери. Доступ до листування можна отримати з декількох пристроїв одночасно. Уся інформація передається в зашифрованому вигляді, включаючи будь-які повідомлення в чатах і групах, медіафайли, аудіозаписи тощо. Для захисту даних використовується комбінація 256-бітного симетричного шифрування AES, 2048-бітного шифрування RSA та алгоритму безпечного обміну ключами Діффі-Геллмана. Telegram стискає дані в мінімально можливу кількість байтів – це найбільш надійний спосіб обміну інформацією. Користувач може відправляти та отримувати повідомлення навіть при мінімальному інтернет-з'єднанні. Додатком можна користуватися безкоштовно без реклами [10]. Skype користуються 88,5% українців, 80% з яких віком від 18 до 40 років.

Instagram – соціальна мережа, що базується на обміні фотографіями та повідомленнями. Додаток дозволяє користувачам робити фотографії, застосовувати до них фільтри, а також поширювати їх через свій сервіс і низку інших соціальних мереж. Є одним із найпопулярніших сервісів у мистецтві айфонографії, тобто мистецтві фотографування на смартфон. В основному такий

застосунок виступає більше в якості розважального [11]. Instagram користуються 78.9% українців, 93% з яких віком від 18 до 40 років.

Найкращим способом для вибору месенджера, в якому буде реалізований чат-бот є аналіз статистики використання того чи іншого месенджера серед аудиторії, для якої і розробляється бот. Виходячи із описаної вище статистики, найактуальнішими месенджерами для користувачів є Viber, Telegram, Instagram та Facebook Messenger. Але зважаючи на те, що Instagram виступає більше розважальним месенджером і не підходить під ідею навчання, розроблювану інформаційну технологію для автоматизації онлайн-навчання буде інтегровано в додатки Viber, Telegram та Facebook Messenger.

### **1.3 Постановка задач дослідження**

У сучасних реаліях онлайн-навчання є надзвичайно актуальним, адже воно є доступним для кожного, хто має доступ до мережі Інтернет, не зважаючи на країну чи регіон проживання користувача, його вік, стан здоров'я чи рівень забезпеченості. Користувач вільно може займатися, не маючи прив'язки до конкретного місця розташування.

Варто також підкреслити, що це колосальна економія часу, що зараз не менш важливо. Як наслідок, кількість навчальних організацій, яка починає впроваджувати онлайн-навчання, стрімко зростає. Звідси можна зробити висновок, що актуальним є створення своєрідного програмного додатку, який зможе ще більше вдосконалити онлайн-навчання в таких організаціях, надавши можливість користувачам детально ознайомитися з програмою того чи іншого курсу, зареєструватися, провести оплату, пройти тестування, переглянути лекції та отримувати матеріали в будь-який зручний для нього час у форматі 24/7, а самим організаціям систематизувати роботу над навчанням їх студентів.

Загальний опис функціоналу системи автоматизації онлайн-навчання з допомогою інтелектуального чат-бота:

а) гілка користувача:

- реалізація входу до системи та можливість процесу реєстрації;
- реалізація підбору курсу за потрібними користувачеві критеріями;
- реалізація проведення оплати за навчання;
- надання доступу до матеріалів курсу, відео-лекцій;
- можливість проходити тестування після кожного розділу вивченого матеріалу;

- можливість кастомного налаштування нагадувань;
- реалізація пошуку теми за ключовими словами;
- реалізація позакласних тестів для самостійного закріплення матеріалів;
- можливість проходження тесту в режимі іспиту;

б) гілка адміністратора:

- реалізація керування профілем студента;
- реалізація додавання матеріалів, оголошень, акцій тощо;
- можливість модерації відгуків та запитань від потенційних клієнтів.

Цифровий формат вивчення навчальних матеріалів вже набув неабиякої популярності на всіх освітніх рівнях. Але, навіть не дивлячись на те, що онлайн-навчання зробило освіту загальнодоступною для мільйонів користувачів по всьому світу, статистика говорить, що лише 7% студентів від зарахованих на курс реально завершують їх.

Попри глобальну цифрову передачу даних, студенти в «віртуальних, інтерактивних аудиторіях» стикаються з дискомфортом через відсутність підтримки та зворотного зв'язку. Чат-боти допомагають перекривати цей недолік, функціонуючи в якості навчальних асистентів.

Оскільки кількість існуючих онлайн-сервісів в сегменті eLearning є досить великою, чат-боти презентують себе досить перспективним інструментом, оскільки можуть супроводжувати кожного студента індивідуально, відповідно до його рівня і темпу засвоєння, роблячи навчання доступним практично для будь-якої людини, яка має доступ до Wi-Fi. Чат-боти не вимагають істотних витрат ресурсів і потенційно можуть допомогти мільйонам слухачів по всьому світу [12].



Інформаційна технологія має на меті використання її користувачами для ознайомлення з програмами курсів, вибором тарифів, реєстрацією, оплатою та безпосереднього вивчення матеріалів, тестування по них та спілкування з викладачами. Таким чином, додаток повністю автоматизує роботу курсів, виконуючи функції викладача, який не залежить від періоду доби та дня тижня, що дає можливість організаціям, що займаються онлайн-навчанням не втрачати клієнтів, які не мають можливості бути прив'язаними до певного часу чи місця.

Такий бот-додаток має бути зручним та інтуїтивно зрозумілим, відповідати користувацькому досвіду (User eXperience) без потреби спеціальних знань у галузі ІТ. Вхідні дані мають задаватися, імпортуючись з бази даних, вихідні дані мають бути чітко і зрозуміло подані.

Розроблювана інформаційна технологія в першу чергу орієнтована на зручність та комфорт при її застосуванні. Зручність полягає у можливості користувача в будь-який момент часу розпочати заняття, використовуючи інтуїтивно зрозумілий та простий програмний додаток.

Програмний модуль буде здійснювати підключення до сервісу DialogFlow для реалізації можливостей інтелектуального чат-бота, підключення до сервера MongoDB та обробка даних з вищезгаданих технологій.

Також до чат-бота буде інтегровано експертну систему для проведення онлайн-консультації для клієнта.

Експертною системою (ЕС) називають систему підтримки прийняття рішень, яка містить знання з певної вузької предметної галузі, а також може пропонувати користувачу рішення проблем з цієї галузі і обґрунтовувати їх [13].

Чат-бот можна розглядати як QA-система, яка має елементи Machine Learning. Маються на увазі функція розбору природної мови та логічного висновку і модуль зв'язку із зовнішніми програмами. Вагомою проблемою для таких чат-ботів є реалізація машини логічного висновку, яка розпізнає релевантність знань до питання, яке було задано. Бот займає вагоме місце в експертній системі, як і решта інтерфейсів. Він дає можливість користувачеві отримати як необхідні знання

з предметної галузі (в нашому випадку це навчання і тестування), так і проконсультуватися онлайн з допомогою зовнішнього виклику модулів.

На сьогоднішній день парадигма інтегрування експертних систем у месенджерні чат-боти стає надзвичайно популярною та актуальною. Інтелектуальні чат-боти дають змогу здійснити консультацію з такими прототипами ЕС. Спочатку відбувається виклик експертної системи для дослідження рівня знань користувача. Це необхідно для підвищення рівня якості результату підбору курсу. Опираючись на отримані результати ЕС підбере належний курс для користувача. Основною задачею такої експертної системи є пошук найоптимальнішого варіанту курсу для клієнта, зважаючи на його потреби та актуальний на даний час рівень знань. Консультативний модуль дозволяє за допомогою месенджера проводити обмін повідомленнями з базами знань через Інтернет та здійснювати онлайн-консультацію в режимі Real Time. База знань для такої ЕС формується та її організація реалізується на локальному комп'ютері. Інтеграція чат-бота та консультативного модуля ЕС являє собою обмін даними між ними без участі користувача та передачу і прийом реквестів для роботи з серверами месенджерів з використанням їх API і JSON [14].

Для вирішення описаної проблеми, необхідно вирішити такі завдання:

1. Здійснити проектування структури інформаційної технології автоматизації онлайн-навчання;
2. Розробити модуль роботи із СУБД;
3. Реалізувати програмний модуль чат-бота;
4. Реалізувати експертну систему для підбору необхідного курсу на основі бази знань;
5. Здійснити тестування програмного модуля.

#### **1.4 Висновок до розділу 1**

В першому розділі проведено аналіз сучасних технічних рішень та доведена актуальність розробки інформаційної технології автоматизації онлайн-навчання.

Розглянуто системи-аналоги: «HomeOnlineEnglish», «Programing» і система «InMind», наведений сценарій їх роботи та цільова аудиторія використання, виділено переваги та недоліки.

Розглянуті системи-аналоги є корисними та доволі актуальними, проте не в повній мірі задовольняють потреби користувачів, відчутно обмежуючи їх в процесі навчання, в порівнянні зі звичайними аудиторними курсами.

Описана та проаналізована вище інформація і стала передумовою для розробки інформаційної технології автоматизації онлайн-навчання, яка зможе повністю замінити користувачам звичайні курси, при цьому економлячи час користувача.

Наведено основні методи та програмні засоби для створення та інтеграції інтелектуальних чат-ботів. З метою розробки інформаційної технології автоматизації онлайн-навчання, обрано сервіс DialogFlow, оскільки він є доволі гнучким та допоможе реалізувати персоналізований продукт, який задовольнить усі потреби користувача. Визначено задачі дослідження, вимоги клієнта, які допоможуть в подальшому відобразити структуру системи, підібрати доцільні технології та мови для реалізації інформаційної технології автоматизації онлайн-навчання з допомогою інтелектуального чат-бота.

## 2 МОДЕЛЮВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АВТОМАТИЗАЦІЇ ОНЛАЙН-НАВЧАННЯ

### 2.1 Обґрунтування вибору методів обробки природньої мови для інформаційної технології автоматизації онлайн-навчання

Натепер питання вирішення задач обробки природньої мови набрало неабиякої актуальності. Сфера чат-ботів та голосових помічників стрімко розвивається. Обробка природньої мови спрямована на створення машин, які розуміють і реагують на текстові або голосові дані і відповідають власним текстом або мовою приблизно так само, як це роблять люди.

Обробка природньої мови (NLP) – це галузь штучного інтелекту, яка відповідає за надання комп'ютерам спроможності розуміти текст і вимовлені слова так само, як це можуть зробити люди.

NLP поєднує комп'ютерну лінгвістику (моделювання людської мови на основі правил) та статистичні моделі, моделі машинного та глибокого навчання. Об'єднавшись ці технології дозволяють комп'ютерам обробляти людську мову у формі тексту або голосових даних і розуміти її.

NLP керує комп'ютерними програмами, які займаються перекладом, сприймають голосові команди та швидко обробляють значні обсяги. NLP також відіграє все більшу роль у корпоративних рішеннях, які допомагають оптимізувати бізнес-операції, підвищити продуктивність співробітників і спростити критично важливі бізнес-процеси.

Такі віртуальні агенти, як Siri (Apple) і Alexa (Amazon), використовують розпізнавання мовлення для обробки шаблонів у голосових командах і генерації відповіді на природній мові. Чат-боти виконують таку ж функцію у відповідь на введені текстові повідомлення. Найкращі з них також навчаються розпізнавати контекстуальні підказки із запитів користувачів та використовувати їх, щоб з часом надавати ще кращі відповіді чи варіанти [15].

Розглянемо методи навчання Natural Language Processing (NLP).

Переважає більшість технологій на основі NLP реалізована з допомогою глибинного навчання (Deep Learning – DL) – це є галуззю машинного навчання, що оснований на певному наборі алгоритмів, завданням яких є моделювання абстракцій високого рівня. Ці алгоритми використовують глибинні графи з обробними шарами, які побудовані з лінійних чи нелінійних перетворень. Серед переваг Deep Learning можна виділити таке:

- наявність даних для тренування у значних обсягах;
- використання багатоядерних CPU і GPU у якості обчислювальних потужностей;
- використання удосконалених моделей і алгоритмів;
- гнучке навчання.

Алгоритми DL ставлять за мету автоматично вилучити найсуттєвіші ознаки з отриманих вхідних даних. Кастомні ознаки зазвичай дуже спеціалізовані, обмежені та неповні. Для їх створення та затвердження потрібен час. В свою чергу, ознаки, виділені глибинним навчанням, легко та швидко адаптуються. Глибинне навчання пропонує гнучкий, універсальний та фреймворк, який навчає для подання інформації за допомогою візуального та лінгвістичного представлень.

Зазвичай, у NLP усі слова позиціонуються як деякі дискретні символи, потім вони представляються у вигляді векторів.

Метод векторного представлення (Text Embedding) – це фактично клас методів, у яких окремі слова представлені у вигляді дійсних векторів у попередньо визначеному векторному просторі. Кожне слово зіставляється з одним вектором, а векторні значення вивчаються у спосіб, який нагадує нейронну мережу, і тому цю техніку часто зводять до сфери глибинного навчання.

Саме такий підхід до представлення слів можна вважати одним із ключових проривів глибинного навчання обробки природної мови.

Ключем до підходу є ідея використання щільного розподіленого представлення для кожного слова [16].

Моделі «Question-Answering» (QA) можуть отримувати відповідь на запитання із заданого тексту, що корисно для пошуку відповіді в документі. Деякі моделі відповідей на запитання можуть генерувати відповіді без контексту.

Використовувати моделі QA можна для автоматизації відповіді на поширені запитання, використовуючи певну базу знань у якості контексту. З цих документів можна отримати відповіді на запитання клієнтів. Таким чином, вони корисні для розумних віртуальних помічників.

У моделі QA застосовується спеціально оптимізована архітектура глибинного навчання Dynamic Memory Network (мережа динамічної пам'яті).

Dynamic Memory Network (DMN) – це архітектура нейронної мережі, яка обробляє вхідні дані та запитання, формує епізодичні спогади та генерує відповіді. Запитання запускають ітеративний процес, який дозволяє моделі акцентувати свою увагу на вхідних даних і результатах попередніх ітерацій. Потім ці результати обґрунтовуються в ієрархічній повторюваній моделі послідовності для отримання відповідей [17].

Отже, для обробки природної мови в інформаційній технології автоматизації онлайн-навчання буде застосовано такі методи як Deep Learning, Text Embedding та Question-Answering.

## **2.2 Розробка ER-моделі бази даних автоматизації онлайн-навчання та обґрунтування вибору бази даних**

Для вибору БД для зберігання даних інформаційної технології автоматизації онлайн-навчання проведемо порівняльний аналіз таких мов для управління організованою базою даних, як QBE та SQL.

QBE – це графічна мова запитів, у якій клієнт отримує інтерфейс користувача, який потім заповнює, щоб отримати належний результат. Така мова запитів використовується в реляційних базах даних, щоб дозволити користувачам шукати інформацію, що зберігається в таблицях і полях. Цей метод забезпечує простий інтерфейс користувача. Принцип QBE полягає в тому, що він діє як

абстракція між користувачем і вихідним запитом, який отримує система бази даних. Спочатку користувач заповнює інтерактивний інтерфейс користувача. Потім після надсилання, запит перетворюється на мовну форму, наприклад SQL у фоновому режимі. QBE — це надзвичайно потужний інструмент, який дає користувачеві можливість отримати доступ до необхідної інформації без знання будь-якої мови програмування.

Серед основних переваг такої мови запитів можна виділити такі:

- QBE підтримується більшістю програмного забезпечення баз даних;
- Використовується мова графічних запитів;
- QBE створюється паралельно з розробкою SQL;
- QBE працює за принципом «point and clicks» і дозволяє користувачеві надати приклад того, як вони хочуть, щоб виглядала відповідь на їхній запит;
- Користувачеві QBE не потрібно вивчати код SQL, щоб генерувати багато корисних запитів.

SQL – це мова структурованих запитів, яка використовується для зберігання, обробки та отримання даних, що зберігаються в реляційній базі даних. SQL використовуються для виконання таких завдань, як оновлення або отримання даних у базі даних. SQL використовує такі системи керування реляційними базами даних, як Oracle, Sybase, Microsoft SQL Server, Access, Ingres тощо.

SQL використовує набір команд для маніпулювання даними в базах даних. Приклади включають SQL INSERT, який використовується для додавання даних до таблиць бази даних; SQL SELECT, який отримує дані з таблиць бази даних; і SQL UPDATE, який змінює існуючі записи бази даних.

До основних переваг цієї мови для управління базою даних можна віднести такі:

- Можливість отримувати доступ до даних у системах керування реляційними базами даних;
- Можливість опису та маніпулювання даними;
- Можливість вбудовувати в інші мови за допомогою модулів, бібліотек і прекомпіляторів SQL;

- Можливість встановлення дозволів на таблиці, процедури та подання;
- Портативність. Можливість використання на ПК, серверах, ноутбуках і деяких мобільних пристроях.
- Швидка та ефективна обробка запитів [18].

Зважаючи на представлений аналіз, для зберігання даних інформаційної технології автоматизації онлайн-навчання було обрано мову для управління організованою базою даних SQL.

Створення будь-якого запиту в SQL починається з опису дії, що має виконуватися. Після команди слідує речення з описом даних чи інформацією про запит, який виконується.

Для опису предметної галузі «Онлайн-навчання» варто виділити такі сутності: користувач, курс, тариф, рівень знань, куратор, навчальні матеріали, адміністратор, тестування. До універсального відношення будуть входити атрибути, що описують дані сутності. Отже, універсальне відношення для бази даних онлайн-навчання матиме такий вигляд: «R» (ПІБ користувача, Email, назва курсу, рівень знань, загальний тест, назва тарифу, ціна тарифу, особливості тарифу, ПІБ куратора, спеціалізація, відео-лекція, література, тест, домашнє завдання, ПІБ адміністратора, відгуки).

Ступінь універсального відношення – 16.

ER-діаграма для бази даних онлайн-навчання зображено на рис. 2.1. Базу даних представлено, використовуючи реляційну модель даних.

Реляційна база даних – це тип бази даних, яка зберігає та надає доступ до даних, пов'язаних одна з одною. Реляційні бази даних базуються на реляційній моделі, інтуїтивно зрозумілому, простому способі представлення даних у таблицях. У реляційній базі даних кожен рядок у таблиці є записом з унікальним ідентифікатором, який називається ключем. Стовпці таблиці містять атрибути даних, і кожен запис зазвичай має значення для кожного атрибута, що полегшує встановлення зв'язків між точками даних. Проста, але потужна реляційна модель використовується організаціями всіх типів і розмірів для широкого спектру інформаційних потреб. Реляційні бази даних використовуються для відстеження



записів, обробки транзакцій електронної комерції, керування величезними обсягами критично важливої інформації про клієнтів і багато іншого.

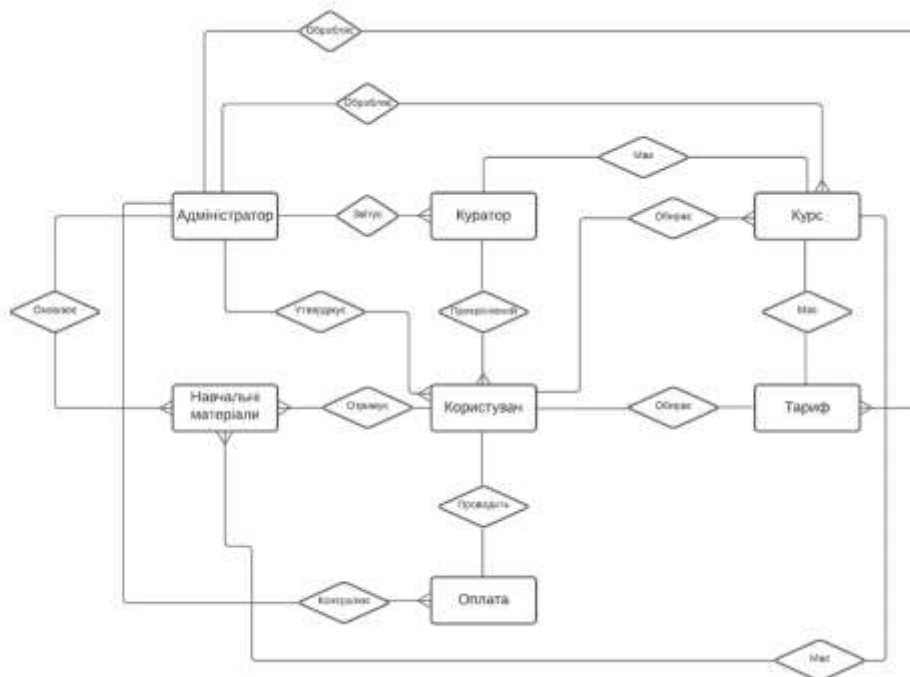


Рисунок 2.1 – ER-діаграма автоматизованого онлайн-навчання

Реляційна модель даних є основною моделлю даних, яка широко використовується в усьому світі для зберігання та обробки даних [19].

Реляційна модель має такі складові:

1. Таблиці. У реляційній моделі даних зв'язки зберігаються у форматі таблиць. Цей формат зберігає відношення між сутностями. Таблиця містить рядки та стовпці, де рядки це – записи, а стовпці – це атрибути.

2. Кортеж. Окремий рядок таблиці, який містить один запис для цього відношення, називається кортежем.

3. Екземпляр відношення. Кінцевий набір кортежів у системі реляційної бази даних представляє екземпляр відношення. Екземпляри відношення не мають повторюваних кортежів.

4. Схема зв'язку. Описує назву зв'язку (ім'я таблиці), атрибути та їхні імена.

5. Ключ відношення. Кожен рядок має один або декілька атрибутів, відомих як ключ відношення, які можуть унікально ідентифікувати рядок у відношення (таблиці).

Кожне відношення має певні умови, які мають виконуватися, щоб воно було дійсним. Ключовими умовами є:

- Назва відношення є унікальною;
- Кожна клітинка відношення містить рівно одне атомарне (єдине) значення;
- Кожен атрибут має унікальну назву;
- Кортежі не можуть повторюватися;
- Порядок кортежу може мати різну послідовність [20].

До переваг реляційної бази можна такі факти:

1. Простота моделі. Вона не вимагає жодних складних запитів, оскільки не має обробки чи структурування запитів, тому для обробки даних достатньо простих запитів SQL;

2. Простота використання. Користувачі можуть легко отримати необхідну інформацію за лічені секунди.

3. Точність. Ключовою особливістю реляційних баз даних є те, що вони чітко визначені та добре організовані, тому дані не дублюються. Реляційні бази даних мають точність завдяки своїй структурі без дублювання даних.

4. Цілісність даних. Бази даних також широко використовуються для забезпечення цілісності даних, оскільки вони забезпечують узгодженість у всіх таблицях. Цілісність даних забезпечує такі функції, як точність і простота використання.

5. Нормалізація. Оскільки дані стають дедалі складнішими, зростає потреба в ефективних способах їх зберігання. Нормалізація — це метод, який розбиває інформацію на керовані фрагменти, щоб зменшити розмір сховища. Дані можна розбити на різні рівні, будь-який рівень потребує підготовки перед переходом на інший рівень нормалізації даних.

До недоліків реляційної моделі можна віднести:

- не кожна предметна галузь може бути представлена у вигляді таблиці;
- у процесі проектування реляційної моделі з'являється багато таблиць, що
- база даних займає багато призводить до труднощів розуміння структури даних; зовнішньої пам'яті;
- відносно низька швидкість доступу до даних [21].

Фрагмент реляційної моделі даних для інформаційної технології автоматизації онлайн-навчання представлено на рис. 2.2.

	A	B	C	D	E	F	G	H	I	J
1	ПІБ куратора	Email	назва курсу	рівень навч.	загальний тест	ціна тарифу	особливості тарифу	ПІБ куратора	спеціалізація	відео-лекція
2	Воспезенко Василь	vita@gmail.com	Курси C++	Початковий	https://mainfest	1500	Підтримка 1:1 від коуч	Андрій Роман	C++ developer	https://lection
3	Діана Віта	vita@gmail.com	Веб-дизайн від 0	Середній	https://mainfest	4536	Курс з англійської мов	Андрій Лягуш	Web-designer	https://lection
4	Діана Яна	yana@gmail.com	Front End	Достатній	https://mainfest	1500	Допомога з оформленням	Криштолюк	Front End dev	https://lection
5	Дмитренко Ірина	irina@gmail.com	Java	Початковий	https://mainfest	2000	Допомога з оформленням	Дарина Панкун	Java developer	https://lection
6	Дудар Костянтин	kostya@gmail.co	QA	Середній	https://mainfest	9000	Підтримка 1:1 від коуч	Dobovskiy	QA	https://lection
7	Іванов Іван	ivanoivanovich@gmail	Курси C++	Достатній	https://mainfest	15000	Підготовка до співбес	Сіларіка	C++ developer	https://lection
8	Іванчук Ілля	ilya@gmail.com	Front End	Початковий	https://mainfest	25000	Курс з англійської мов	Makarova	Front End dev	https://lection
9	Кірило Кірило	kiril@gmail.com	Веб-дизайн від 0	Середній	https://mainfest	1670	Підготовка до співбес	Lesia Khizniak	Web-designer	https://lection
10	Кірило Володимир	tova@gmail.com	Front End	Достатній	https://mainfest	1500	Підготовка до співбес	Dokomova	Front End dev	https://lection
11	Мікошайчук Ми	mykola@gmail.co	Java	Початковий	https://mainfest	4536	Підтримка 1:1 від коуч	Андрій Роман	Java developer	https://lection
12	Олександрова Ол	oleksandr@gmail	Front End	Середній	https://mainfest	1500	Унікальні вакансії від	Андрій Лягуш	QA	https://lection
13	Олександрівко Олександр	oleksandr@gmail.co	Веб-дизайн від 0	Достатній	https://mainfest	2000	Курс з англійської мов	Криштолюк	Web-designer	https://lection
14	Олександрівко Ольга	volga@gmail.com	Front End	Початковий	https://mainfest	9000	Підтримка 1:1 від коуч	Дарина Панкун	Front End dev	https://lection
15	Пашенко Олександр	asha@gmail.com	QA	Середній	https://mainfest	15000	Підготовка до співбес	Dobovskiy	QA	https://lection
16	Петров Петро	pe-petro@gmail.com	Front End	Достатній	https://mainfest	25000	Курс з англійської мов	Сіларіка	Front End dev	https://lection
17	Сергієнко Серій	sergey@gmail.co	Java	Початковий	https://mainfest	1670	Унікальні вакансії від	Makarova	Java developer	https://lection
18	Сидоренко Юлія	yulia@gmail.com	Front End	Середній	https://mainfest	4536	Підтримка 1:1 від коуч	Lesia Khizniak	Front End dev	https://lection

Рисунок 2.2 – Загальний вигляд фрагменту реляційної моделі даних для інформаційної технології автоматизації онлайн-навчання

Для збереження даних онлайн-курсів було обрано базу даних MongoDB.

MongoDB – це документно-орієнтована база даних, яка використовується для зберігання великих обсягів даних. Замість використання таблиць і рядків, як у традиційних реляційних базах даних, MongoDB використовує колекції та документи. Документи складаються з пар ключ-значення, які є основною одиницею даних у MongoDB. Колекції містять набори документів і функції, що є еквівалентом таблиць реляційної бази даних.

Кожна база даних у MongoDB містить колекції, які, у свою чергу, містять документи. Кожен документ може мати різну кількість полів. Розмір і зміст кожного документа можуть відрізнятися один від одного. Структура документа

більше відповідає тому, як розробники будують свої класи та об'єкти на відповідних мовах програмування. Класи не є рядками та стовпцями, а мають чітку структуру з парами ключ-значення. Для рядків (або документів, як вони називаються в MongoDB) не потрібно заздалегідь визначати схему.

Модель даних, доступна в MongoDB, дозволяє легше представляти ієрархічні зв'язки, зберігати масиви та інші складніші структури.

Середовища MongoDB дуже масштабовані. Компанії в усьому світі визначили кластери, деякі з них працюють із 100+ вузлами з приблизно мільйонами документів у базі даних

MongoDB має такі переваги:

1. Рівні продуктивності. MongoDB зберігає більшість даних в оперативній пам'яті. Це забезпечує більш швидку роботу під час виконання запитів. Він збирає дані безпосередньо з оперативної пам'яті, ніж із жорсткого диска, і результати повертаються швидше;

2. Висока швидкість і вища доступність. Атрибути дозволяють збільшити доступність даних. Також легко отримати доступ до документів за допомогою індексації. MongoDB працює в 100 разів швидше, ніж інші реляційні бази даних, і забезпечує високу продуктивність;

3. Простота. MongoDB пропонує простий синтаксис запиту;

4. Гнучкість. Схема MongoDB не є попередньо визначеною. Це означає, що він має динамічну схематичну архітектуру, яка працює з неструктурованими даними та сховищем.

5. Масштабованість. MongoDB використовує шардинг під час обробки великих наборів даних. Шардинг – це процес поділу даних із великого набору та розподілу їх на декілька серверів. Якщо виникає проблема, коли сервер не може обробити дані через їх розмір, він автоматично розподіляє їх далі, не призупиняючи діяльність [22].

Зважаючи на описані вище переваги та недоліки було обрано реляційну структуру бази даних, діалогову мову програмування SQL та базу даних MongoDB для збереження даних інформаційної технології автоматизації онлайн-навчання.

### 2.3 Математична модель прийняття рішень для діалогу агента з користувачем

Діалог людини з чат-ботом можна розглядати як процес прийняття рішень, адже він має природню ієрархію: процес верхнього рівня прямо впливає на процес нижнього, а також кожен процес нижнього рівня підшукує примітивні дії для завершення своєї підзадачі, чим викликає наступний процес.

Описаний вище ієрархічний процес прийняття рішень можна представити у вигляді такої математичної структури, як Марковські процеси вирішування (Markov decision process, MDP) для генерації нових відповідей на запити користувачів.

Марковські процеси прийняття рішення забезпечують математичну систему для моделювання ухвалення рішень у ситуаціях, в яких наслідки є частково випадковими, а частково контрольованими ухвалювачем рішення [23].

Для роботи над текстом для чат-бота потрібно використовувати розширення MDP, у якому агент обиратиме лише примітивну дію на кожному кроці. Якщо розглядати кожен варіант як дію, то агент діалогу через взаємодію зі своїм середовищем через послідовність дискретних кроків, здійснюватиме навігацію до MDP. На кожному кроці агент спостерігатиме за поточним станом і вибиратиме дію відповідно до стратегії.

Навчання з підкріпленням (RL) – це парадигма навчання, де інтелектуальний агент вчиться робити оптимальні рішення, взаємодіючи з невідомим оточенням. У порівнянні з навчанням без вчителя, особливий виклик у RL – це навчання без заздалегідь відомих правильних результатів [24].

Взаємодія агент-середовище часто моделюється як дискретномарковський процес прийняття рішень, або MDP, описаний п'ятіркою:

$$M = (S, A, P, R, \gamma),$$

де  $S$  – це нескінченний набір станів, в яких може перебувати середовище;

$A$  – це нескінченний набір дій, які агент може прийняти в стані;

$P(s' | s, a)$  дає ймовірність переходу середовища в новий стан  $s'$  після дії  $a$ , що приймається в стані  $s$ ;

$R(s, a)$  – середня винагорода, що негайно отримується агентом після виконання дії  $a$  в стані  $s$ ;

$\gamma \in (0, 1)$  – коефіцієнт дисконтування.

Процес може бути записаний як траєкторія  $(s_1, a_1, r_1, \dots)$ , яка генерується на етапі  $t = 1, 2, \dots$  наступним чином:

1. Агент спостерігає за поточним станом середовища  $s_t \in S$  і приймає дію при  $a_t \in A$ ;

2. Перехід навколишнього середовища до наступного стану  $s_{t+1}$ , розподіленого за ймовірністю переходу  $P(\cdot | s_t, a_t)$ ;

3. Пов'язана з переходом безпосередня винагорода  $r \in R$ , середнє значення якої  $R(s, a)$ .

Опускаючи індекси, кожен крок призводить до появи кортежу  $(s, a, r, s')$ , що називається переходом. Метою агента RL є максимізація довгострокової винагороди шляхом прийняття оптимальних дій. Стратегія його вибору дій, позначена  $\pi$ , може бути детермінованою або стохастичною. У будь-якому випадку використовуємо  $\pi(s)$  для позначення вибору дії, тобто слідуємо  $\pi$  у стані  $s$ . З огляду на стратегію  $\pi$ , значення стану  $s$  є середньою дисконтованою довгостроковою винагородою від цієї стратегії:

$$V^\pi(s) := \mathbb{E}[r_1 + \gamma r_2 + \gamma^2 r_3 + \dots | s_1 = s, a_i \sim \pi(s_i), \forall i \geq 1].$$

Ми зацікавлені в оптимізації стратегії, щоб  $V^\pi$  максимізувався для всіх стратегій. Позначимо через  $\pi^*$  оптимальну стратегію, а  $V^*$  відповідне їй значення функції (також відоме як оптимальне значення функції). У багатьох випадках більш зручно для знаходження значення винагороди використовувати іншу функцію, що називається Q-функцією:

$$Q^\pi(s, a) := \mathbb{E}[r_1 + \gamma r_2 + \gamma^2 r_3 + \dots | s_1 = s, a_i = a, a_i \sim \pi(s_i), \forall i > 1],$$

яка вимірює середню дисконтовану довгострокову винагороду, вибираючи спочатку  $a$  у стані  $s$ , а потім наступну стратегію  $\pi$ . Оптимальна Q-функція, що відповідає оптимальній стратегії, позначається  $Q^*$ .

Опишемо два популярних класи алгоритмів на прикладі Q-learning та Policy Gradient відповідно.

Перше сімейство базується на спостереженні, що оптимальна стратегія може бути відразу знайдена, якщо доступна оптимальна Q-функція. Зокрема, оптимальну стратегію можна визначити так:

$$\pi^*(s) = \arg \max_a Q^*(s, a).$$

Тому, велике сімейство алгоритмів RL фокусується на вивченні  $Q^*(s, a)$  і колективно називаються методами на основі значень функції.

Інше сімейство алгоритмів (Policy Gradient) намагається оптимізувати стратегію безпосередньо, без необхідності вивчати Q-функцію. Тут сама стратегія безпосередньо параметризована по  $\theta$ , а  $\pi(s; \theta)$  часто є розподілом над діями. При будь-яких  $\theta$ , стратегія оцінюється за середньою довгостроковою винагородою, що потрапляє в траєкторію довжини:

$$H, \tau = (s_1, a_1, r_1, \dots, s_H, a_H, r_H).$$

Якщо можна оцінити градієнт  $\Delta_\theta J$ :

$$J(\theta) := \mathbb{E} \left[ \sum_{t=1}^H \gamma^{t-1} r_t | a_t \sim \pi(st; \theta) \right],$$

по траєкторіям, то можна застосувати метод SGD, щоб максимізувати  $J$ :

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta).$$

Отже, ми готові моделювати розмови між агентом системи та користувачем як проблему RL. Тут діалогова система є агентом RL, а користувачем є середовище. На кожному кроці діалогу, агент стежить за станом діалогу, на основі інформації, що була виявила до цього в розмові, а потім робить дію; дія може бути відповіддю користувачеві у вигляді дій, діалогу або внутрішньої операції, такої як пошук в базі даних або виклик API [24].

Далі користувач відповідає наступною реплікою, яка буде використовуватися агентом для оновлення стану внутрішнього діалогу на наступному кроці. Цей процес є взаємодією агент-середовище.

## **2.4 Висновок до розділу 2**

У другому розділі подано поняття NLP як галузі штучного інтелекту та обґрунтовано використання методів Deep Learning, Text Embedding та Question-Answering для обробки природньої мови. Наведено їх основні переваги та приведено пояснення щодо їх використання.

Розроблено математичну модель прийняття рішень для діалогу агента з користувачем на основі математичної структури MDP. Модель відповідає за генерацію нових відповідей на запити користувачів. Розглянуто діалог як процес прийняття рішень, який можна представити у вигляді Марковських процесів для моделювання розмови між агентом системи та користувачем.

Розроблено ER-модель бази даних онлайн-курсів, яка описує об'єкти та зв'язки предметної галузі «автоматизація онлайн-навчання». Для формування інформаційної бази навчальної організації було обрано реляційну модель даних, яка має перевагу за рахунок своєї простоти, гарної організації, збереження цілісності інформації та виключенням дублювання даних. Для опису предметної галузі вирішено обрати сутності користувач, куратор, курс, тариф, адміністратор, навчальні матеріали та оплата. Для управління організованою базою даних було обрано мову для управління організованою базою даних SQL, для збереження даних онлайн-курсів – базу даних MongoDB.



### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АВТОМАТИЗАЦІЇ ОНЛАЙН-НАВЧАННЯ

#### 3.1 Обґрунтування вибору інструментів для реалізації інформаційної технології автоматизації онлайн-навчання

Для реалізації інформаційної технології автоматизації онлайн-навчання було обрано клієнт-серверний тип архітектури. Архітектура «клієнт-сервер» представлена, як мережева інфраструктура, сервери якої виступають у ролі постачальників тих чи інших сервісів (послуг), а клієнтські комп'ютери є їх споживачами. У класичному поданні, клієнт-серверна архітектура має у мережі сервер і декілька підключених до нього клієнтів. Сервери таких систем, в основному, виступають як постачальники послуг з використання бази даних. Така архітектурна модель називається дворівневою (two-tier architecture) і показана на рис. 3.1.



Рисунок 3.1 – Схема подання «клієнт-серверної» архітектури

До переваг такої архітектури варто віднести такі аспекти, як:

- підтримка багатокористувацької роботи;

- цілковита гарантія цілісності даних;
  - наявність необхідних механізмів для керування правами доступу до ресурсів сервера;
  - розподіл функцій між різними вузлами мережі.

Серед недоліків можна виділити такі:

- якщо сервер виходить з ладу, це призводить до непрацездатності всієї системи;
- для роботи з цією архітектурою потрібно мати в штаті персонал високого технічного рівня;
- устаткування є помірно дорогим.

Така архітектура, як «клієнт-сервер» є переважаючою концепцією у розробці розподілених мережних застосунків і вимагає взаємодії та обміну даними між ними. Вона містить такі основні компоненти:

- набір серверів, які передають дані чи інші послуги модулям, які звертаються до них;
- набір клієнтів, які безпосередньо використовують сервіси, які надають сервери;
- мережа, за допомогою якої забезпечується взаємодія між клієнтами та серверами.

Сервери та клієнти є самостійними, автономними та незалежними один від одного. Немає жорсткої прив'язки клієнтів до серверів. Доволі звичною є ситуація, коли один сервер одночасно обробляє запити від декількох різних клієнтів. Так само клієнт може звертатися до різних серверів. Клієнти можуть як мати можливість прослідкувати які сервери доступні, так і не мати жодного уявлення про існування інших [25].

На старті реалізації будь-якої програми, варто зробити акцент на тому, що програмне забезпечення містить багато систем, які мають підсистеми, а ті в свою чергу теж мають підсистеми. Проектування такої великої та складної системи виявляється громіздким завданням, оскільки помилка в одній може зробити все

програмне забезпечення марним. Тому є доречним фрагментування таких систем на модулі, які називаються програмними.

Програмний модуль – це логічний піднабір функціональних можливостей, наданих програмним забезпеченням, який ліцензується окремо від інших програмних модулів. Програмне забезпечення може складатися з одного чи декількох програмних модулів. Модуль складається з одного блоку коду, який може бути викликаний таким же чином, як викликається процедура, функція або метод.

Кожен модуль є унікальним і стосується компонентів програмного забезпечення, які можна легко розв'язати та працювати з ними, не заважаючи на інші компоненти програмного забезпечення. Використання таких модулів називається модульною структурою для розробки.

Модулі полегшують роботу при розробці, дозволяючи зосередитися на одній області програмного забезпечення, не порушуючи потік інших. Модулі об'єднуються за допомогою інтерфейсів. Запровадження модульності дозволило розробникам повторно використовувати попередньо написані коди з новими програмами [26].

В інформаційній технології автоматизації онлайн-навчання подано такі модулі: як модуль інтерфейсу, модуль збору інформації, навчальний модуль, модуль роботи з базою даних, модуль координування. Загальна структура програмного модуля інформаційній технології автоматизації онлайн-навчання подана на рис. 3.2.

Користувач взаємодіє із системою через модуль Interface Module. Він пов'язаний з такими модулями як Information collection Module, Education Module, Coordination Module та Database Module. Основна задача модуля – коректне відображення навчальних матеріалів та запитів на екрані користувача та демонстрація питань і загальної інформації.

Модуль «Information collection Module» взаємодіє з Education Module, Database Module та Interface Module. Модуль виступає одним з головних у

програмному додатку. Його основна задача – збір даних про користувача та передача їх до бази даних при реєстрації на курс.

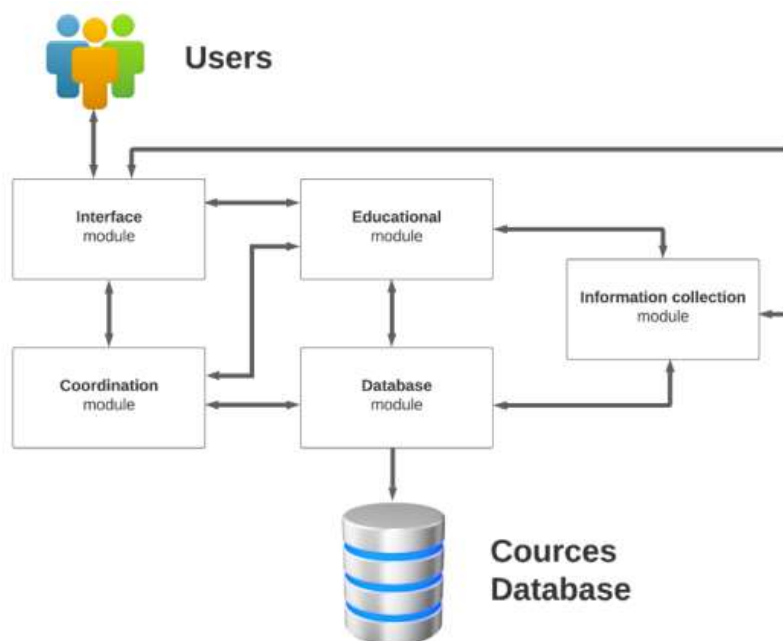


Рисунок 3.2 – Загальна структурна схема функціонування системи

Education Module – це наступний головний модуль програми. Він має односторонній зв'язок з модулем Information collection Module та Coordination Module і двосторонній із Interface Module та Database Module. Його зона відповідальності це весь процес навчання: забезпечення користувачів навчальними матеріалами, тестуваннями, забезпечення конекту з кураторами, просування за темами, пошук необхідних матеріалів.

Модуль Database Module взаємодіє з модулями Information collection Module, Education Module, Coordination Module та Interface Module і забезпечує безпосередній зв'язок з базою даних навчальної організації, де зберігаються уся інформація про авторизованих користувачів, навчання, клієнтські результати.

Coordination Module відповідає за роботу гілки адміністратора. Основна задача модуля – забезпечення оновлення навчальних матеріалів, комунікацію з кураторами та модерацію відгуків адміністратором навчальної організації.

База даних навчальної організації має містити усі навчальні матеріали і внесені користувачем при реєстрації дані та забезпечити швидкий, автоматичний імпорт/експорт необхідної інформації.

Для реалізації інформаційної технології можна обрати будь-яку мову програмування, для якої існують доступні реалізації API. З метою вибору потрібного варіанту, проаналізуємо найчастіше використовувані мови C++, C# та Python для реалізації чат-ботів.

Мова програмування C++ є високорівневою компільованою мовою програмування загального призначення, яка включає в себе статичну типізацію та використовується для створення різноманітних додатків. Зараз C++ фігурує як найпопулярніша і найпоширеніша мова. Вона підтримує такі парадигми програмування: процедурне, узагальнене, функціональне. Найбільша увага приділяється підтримці об'єктно-орієнтованого програмування.

У C++ введено класи, що відповідають за забезпечення трьох найважливіших властивостей ООП, а саме інкапсуляцію, успадкування і поліморфізм. Методами класу називаються функції, які застосовуються до екземплярів класу [27].

Ключові моменти, які варто виділити, аналізуючи мову C++ це:

1. Швидкодія. Швидкість роботи програм на C++ практично не поступається програмам на C, хоча програмісти отримали в свої руки нові можливості і нові засоби.

2. Масштабованість. На мові C++ розробляють програми для найрізноманітніших платформ і систем.

3. Можливість роботи на низькому рівні з пам'яттю, адресами, портами. (Що, при необережному використанні, може легко перетворитися на недолік.)

4. Можливість створення узагальнених алгоритмів для різних типів даних, їхня спеціалізація і обчислення на етапі компіляції, з використанням шаблонів.

5. Підтримуються різні стилі та технології програмування, включаючи традиційне директивне програмування, ООП, узагальнене програмування, метапрограмування (шаблони, макроси) [28].

C# – це мова програмування загального призначення з мультипарадигмальним підходом, який охоплює декілька дисциплін програмування, таких як статичний тип, імперативне, декларативне, функціональне, об'єктно-орієнтоване та компонентно-орієнтоване програмування. Саме такий підхід дозволяє C# бути настільки універсальним, що ви можете використовувати його для багатьох різних проектів. C# є сучасною мовою програмування, яка дозволяє інженерам створювати програми, що працюють в екосистемі .NET.

Як мова програмування загального призначення, C# можна використовувати для розробки будь-яких систем від мобільних і десктопних програм до корпоративного програмного забезпечення та хмарних платформ.

До переваг цієї мови програмування можна віднести такі:

1. Швидший час розробки. C# має декілька функцій, які дозволяють розробникам кодувати швидше ніж з іншими мовами. Деякі з цих функцій включають статично типізовану та легку для читання мову, синтаксис, який виглядає як розширена версія Java і величезну бібліотеку, наповнену функціональністю високого рівня;

2. Висока масштабованість. Статичний характер кодування C# перетворює всі його програми на надійні продукти, які можна легко налаштувати та змінити. Це означає, що інженери можуть швидко вносити зміни та надбудовувати будь-яку програму C#, щоб розширити її функціональність і підтримати більше користувачів;

3. Об'єктно-орієнтований. C# охопив об'єктно-орієнтоване програмування таким чином, що це може бути мовою, яка краще його використовує. Насправді об'єктно-орієнтованість дозволяє C# бути високоефективною і надзвичайно гнучкою, усе це робить розробку легшою та менш ресурсоємною;

4. Легке навчання. Як мову високого рівня, C# дуже легко вивчити та зрозуміти. І це без урахування багатьох вбудованих функцій, які дуже прості у використанні. Більше того, будь-який інженер, який уже знає C++ або Java,

почуватиметься комфортно, коли вперше скористається C#, оскільки ці мови мають багато однакових функцій і загальний підхід до програмування

5. Велика спільнота. C# є однією з найпоширеніших мов у світі, а це означає, що є багато розробників C#, готових допомогти вам із нею. Крім того, будучи продуктом Microsoft, C# має підтримку технічного гіганта, що означає допомогу експертів, додаткові ресурси та часті оновлення [29].

Python – це інтерпретована об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Його високорівневі вбудовані структури даних у поєднанні з динамічною типізацією та динамічним зв'язуванням роблять його дуже привабливим для швидкої розробки додатків, а також для використання як мови сценаріїв або з'єднувальної мови для з'єднання існуючих компонентів. Розглянемо переваги мови Python:

1. Простота. Python – це мова, яку легко читати та розуміти для розробників, які ніколи не писали на ній код. Таким чином, коли виникає проблема, розробник може зосередитися на ній і отримати допомогу від інших у спільноті, не турбуючись про складність мови.

2. Сумісність з різними платформами. Це одна з головних проблем, з якою зазвичай стикаються розробники, коли вони використовують інші мови. Добре підтримувані платформи на Python включають Linux, Windows, FreeBSD, macOS.

3. Об'єктно-орієнтований. Python підтримує об'єктно-орієнтоване програмування та є процедурно-орієнтованим. Об'єктно-орієнтоване програмування, у тому сенсі, використовує об'єкти, які базуються на даних і функціональності. Функція «Процедурно-орієнтована» пропонує застосовувати фрагменти коду для повторного використання.

4. Великі бібліотеки та фреймворки. Наявність великої бібліотеки з керуванням пам'яттю та порожнім дизайном допомагає підвищити продуктивність розробника Python. В результаті розробники можуть керувати базами даних, документацією, веб-браузерами; виконувати модульне тестування та багато інших функцій. Крім того, Python можна використовувати для розробки багатьох завдань,

таких як розробка веб-додатків і програм для настільних комп'ютерів, розробка складних систем обчислення, систем керування життєзабезпеченням [30].

Взявши до уваги вищесказане, для реалізації інформаційної технології автоматизації онлайн-навчання було обрано мову програмування C#.

В ході розробки програмного продукту було використано такі бібліотеки:

- System.Collections.Generic. Містить інтерфейси та класи, які визначають загальні колекції та дозволяють користувачам створювати строго типізовані колекції, що забезпечують кращу безпеку типів та продуктивність, ніж незагальні строго типізовані колекції [31].
- System.Linq. Це набір технологій, заснованих на інтеграції можливостей запитів безпосередньо в мову C#. Бібліотека дає можливість писати запити LINQ на C# для баз даних SQL Server, документів XML, наборів даних ADO.NET і будь-якої колекції об'єктів, яка підтримує IEnumerable або загальний інтерфейс IEnumerable<T> . Підтримка LINQ також надається третім сторонам для багатьох веб-служб та інших реалізацій баз даних [32].
- System.Text. Містить класи, які представляють кодування символів ASCII і Unicode; абстрактні базові класи для перетворення блоків символів у/та з блоків байтів; і допоміжний клас, який маніпулює та форматує об'єкти String без створення проміжних екземплярів String [33] .
- System.Threading.Tasks. Надає типи, які спрощують роботу з написання паралельного та асинхронного коду. Основними типами є Task , який представляє асинхронну операцію, яку можна зачекати та скасувати, і Task<TResult>, який є завданням, яке може повертати значення. Клас TaskFactory надає статичні методи для створення та запуску завдань, а клас TaskScheduler надає інфраструктуру планування потоків за замовчуванням [34].

У наведеному нижче фрагменті коду наведено приклад використання бібліотек System.Collections.Generic, System.Linq, System.Text, System.Threading.Tasks для проведення реєстрації користувача.



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

class ProgramBot
{
static void Main(string[] args)
{
    ChatBot cb = new ChatBot(
        new BotRule(
            Name: "name",
            Weight: 15,
            MessagePattern: new Regex("(what is your name) (now )?(.*)",
RegexOptions.IgnoreCase),
            Process: delegate(Match match, ChatSessionInterface session)
        {
            session.SessionStorage.Values["name"] = match.Groups[3].Value;
            return "My name is " + session.SessionStorage.Values["name"];
        }
    ),
        new BotRule(
            Name: "lastname",
            Weight: 15,
            MessagePattern: new Regex("(what is your last name)",
RegexOptions.IgnoreCase),
            Process: delegate(Match match, ChatSessionInterface session) {
                if (!session.SessionStorage.Values.ContainsKey("name"))
                {
                    return "I don't want to be named";
                }
                if (match.Value.ToLower() == "let's get to know each other")
                {
                    return "i am " + session.SessionStorage.Values["name"];
                }
                return "My name is " + session.SessionStorage.Values["name"];
            }
        )
    );
}
}

```

```

    }
),
    new BotRule(
        Name: "course",
        Weight: 10,
        MessagePattern: new Regex("which course you want to take?(.*)",
RegexOptions.IgnoreCase),
        Process: delegate(Match match, ChatSessionInterface session) {
            session.SessionStorage.Values["UserName"] = match.Groups[2].Value;
            return "Hi " + session.SessionStorage.Values["CourseName"];
        }
),
    new BotRule(
        Name: "tariff",
        Weight: 20,
        MessagePattern: new Regex("(which tariff you chose for yourself?",
RegexOptions.IgnoreCase),
        Process: delegate(Match match, ChatSessionInterface session) {
            if (!session.SessionStorage.Values.ContainsKey("tariff "))
            {
                return " Sorry, read the rates again";
            }
            return " Your chosen tariff " + session.SessionStorage.Values["tariff "];
        }
    ),
    cb.talkWith(new ConsoleChatSession());
}
}

```

Також використано спеціальні бібліотеки для розробки чат-ботів Telegram.Bot, Viber.Rest.API та Fbchat-sharp,

Telegram.Bot – це додатковий ресурс для C#, який покращує його можливості при створенні чат-бота для месенджера Telegram. Telegram.Bot є одною з найповніших бібліотек, доступних для взаємодії з API Telegram Bot.

Вона містить усі доступні методи в Bot API 5.0. Крім того, бібліотека має повний набір класів з усіма використовуваними об'єктами.

Розглянемо приклад використання бібліотеки у реалізованому програмному модулі.

Метод `getUpdates()` відповідає за оновлення бота, тобто обробляє повідомлення, які йому надходять та інформацію про відправника. Метод `from_chat_id()` володіє інформацією про ідентифікатор чату-відправника та відповідає за надсилання відповіді чат-ботом. Метод `sendMessage()` відповідає за безпосередню відправку повідомлення [35].

```
using System.Text.RegularExpressions;
using Telegram.Bot;
using Telegram.Bot.Types;
using Telegram.Bot.Types.Enums;
using Telegram.Bot.Types.InlineQueryResults;
var bot = new TelegramBotClient(token);
var me = await bot.GetMeAsync();
using var cts = new CancellationTokenSource();
bot.StartReceiving(HandleUpdateAsync, PollingErrorHandler, null, cts.Token);
Console.WriteLine($"Start listening for @ {me.Username}");
Console.ReadLine();
cts.Cancel();
Task PollingErrorHandler(ITelegramBotClient bot, Exception ex, CancellationToken ct)
{
    Console.WriteLine($"Exception while polling for updates: {ex}");
    return Task.CompletedTask;
}
async Task HandleUpdateAsync(ITelegramBotClient bot, Update update, CancellationToken ct)
{
    try
    {
        await (update.Type switch
        {
            UpdateType.InlineQuery => BotOnInlineQueryReceived(bot, update.InlineQuery!),
```

```

UpdateType.ChosenInlineResult          =>          BotOnChosenInlineResultReceived(bot,
update.ChosenInlineResult!),
    _ => Task.CompletedTask
}
);
}

```

Viber.Rest.API – унікальний секретний ідентифікатор облікового запису. Він використовується для автентифікації запиту в API Viber і для запобігання неавторизованим особам надсилати запити від імені бота. Кожен запит API має містити заголовок HTTP, X-Viber-Auth-Token, що містить маркер автентифікації облікового запису. Маркер автентифікації генерується під час створення бота, і його можуть переглядати адміністратори облікового запису на екрані «редагувати інформацію» свого бота або на панелі адміністратора Viber [36].

Fbchat-sharp – це клієнтська бібліотека Facebook Messenger для C#. Це є потужна та ефективна бібліотека для взаємодії з Facebook Messenger. Бібліотека відрізняється використанням звичайного облікового запису Facebook. Ключова відмінність від інших бібліотек месенджерів C# полягає в тому, що fbchat-sharp не використовує API чат-бота. Бібліотека може читати та надсилати повідомлення чату будь-якому з контактів користувача.

Fbchat-sharp наразі підтримує такі функції:

- надсилання багатьох типів повідомлень із файлами, наклейками, згадками тощо;
- отримання всіх повідомлень, ланцюжків і зображень у ланцюжках;
- пошук повідомлень і тем;
- створення груп, встановлення групових емодзі, зміна ніків, створення опитувань тощо;
- прослуховування, реагування на повідомлення та інші події в режимі реального часу [37].

У наведеному нижче фрагменті коду наведено приклад використання бібліотеки Fbchat-sharp:

```

using System.Collections.Generic;

using System.Net;

using System.Threading.Tasks;
using fbchat_sharp.API;
public class FBClient : MessengerClient
{ public FBClient()
  { On2FACodeCallback = get2FACode;
  }
  private async Task<string> get2FACode()
  { return null;
  }
  protected override async Task OnEvent(FB_Event ev)
  { switch (ev)
  { case FB_MessageEvent t1:
    Console.WriteLine(string.Format("Got new message from {0}: {1}", t1.author, t1.message)
);
    break;
  default:
    Console.WriteLine(string.Format("Something happened: {0}", ev.ToString())
);
    break; }
    await Task.Yield();
  }
  protected override async Task DeleteCookiesAsync()
  protected override async Task<Dictionary<string, List<Cookie>>>
ReadCookiesFromDiskAsync()
  { return null;
  }
  protected override async Task WriteCookiesToDiskAsync(Dictionary<string,
List<Cookie>> cookieJar)
  }
}

```

Ключовими рисами бібліотеки є потужність (немає обмеження API для чат-бота Facebook) та портативність (бібліотека розроблена як PCL з підтримкою .NET Standard 1.2)

Для вибору методу реалізації інформаційної технології для автоматизації навчання, проведемо аналіз характеристик найбільш використовуваних на сучасному ринку інструментів.

Для порівняльного аналізу інструментів обрано такі критерії:

1. Наявність підтримка української мови;
2. Ціна доступу до функціоналу платформи;
3. Кросплатформність;
4. Підтримка медіоконтенту;
5. Універсальність для будь-яких платформ та пристроїв;
6. Підтримка голосових повідомлень.

Результати порівняння наведено у табл. 3.1.

Таблиця 3.1 – Характеристика інструментів для створення чат-ботів

	Manychat	Google Dialogflow	Amazon Lex	IBM Watson	Botsify
Підтримка мови	+	+	-	-	+
Кросплатформність	-	+	-	+	+
Медіаконтент	+	+	+	-	+
Універсальність	+	+	-	-	+
Вартість доступу	10 ум. од./місяць	Умовно безкоштовно	17 ум. од./місяць	Близько 800 ум. од.	20 ум. од./місяць
Голосові повідомлення	-	+	+	+	-

З табл. 3.1 можна зробити висновок, що найкращими платформами для розробки інтелектуальних чат-ботів є Botsify та Google Dialogflow.

Botsify – це платформа для розробки чат-ботів машинного навчання, яка дозволяє створювати чат-ботів штучного інтелекту на веб-сайтах або у Facebook Messenger.

Платформа інтегрується з Alexa, WordPress, Shopify і Zapier. Крім того, існує також функція «Захоплення людиною», яка дозволяє легко налаштувати розмови та інтерфейс чату після спостереження за розмовами.

Інтерфейс користувача зрозумілий і відносно простий, а також має легкий спосіб перетягування та створення шаблонів. Платформа розроблена для нетехнічних користувачів, подібно до Chatfuel. Вона постачається з попередньо створеними шаблонами, розробленими для певних галузей і загальних поширених запитань.

Botsify проектує за допомогою «stories». Stories – це лінійна послідовність взаємодій, які формують розмову.

Потужною функцією є можливість зберігати відповіді у формі. Це чудовий спосіб збору інформації від користувачів. Відповіді форми можна легко експортувати у CSV через веб-інтерфейс [38].

Можливості конструктора:

1. Можливість інтеграції в месенджери використовуючи плагіни з використанням RSS-каналів або JSON API;
2. Використання шаблонного конструктора з підтримкою операції Drag n 'Drop;
3. Підбір подібних фраз та сутностей із запитів користувача, що досягається з використанням штучного інтелекту;
4. Робота із базою знань, з якої отримуються та зіставляються всі запити, призначені для користувача;
5. Безкоштовний тариф з об'ємом до 100 користувачів/міс;
6. Можливість машинного навчання [39].

Dialogflow – це платформа розробки чат-ботів, що належить Google. Він використовується для обробки природної мови за допомогою машинного навчання.

Dialogflow – це продукт на основі SaaS. Це означає, що він працює на інфраструктурі Google і може легко масштабуватися для мільйонів користувачів.

Dialogflow надає дві версії використання для користувачів – безкоштовна та платна з більш широким функціоналом.

Dialogflow містить функціональні та зручні інструменти. Платформа дає такі можливості:

1. Багатоканальна підтримка. Dialogflow підтримує інтеграцію в один клік для понад двадцяти платформ, включаючи Slack, Facebook Messenger, Twitter, Kik, Line тощо;

2. Вартість. Dialogflow можна використовувати безкоштовно у стандартній версії;

3. Багатомовна підтримка. Dialogflow підтримує понад 14 мов у всьому світі і має додаткову підтримку;

4. Досконалий NLP. Dialogflow має краще машинне навчання порівняно з конкурентами навіть із меншою кількістю навчальних даних;

5. Спілкування з користувачем на природній мові. Dialogflow генерує унікальну модель, яка вдало інтегрується з користувачем, що створює важливий ефект живої розмови;

6. Голосове управління, що включає розпізнавання мови. Dialogflow допомагає реалізувати функцію відповіді на голосові команди клієнта. Ця функція доступна в рамках одного виклику API, що поєднує розпізнавання мови з розумінням природної мови.

4. Підтримка сервісом підключення webhook, для конекту Dialogflow зі самим додатком, який, при його виклику, реалізує виконання бізнес-логіки, виклик зовнішнього API, звертається до сховищ даних. Отримуючи повідомлення на природній мові, Dialogflow порівнює запит з готовими шаблонами та обирає найбільш прийнятний [40].

Для кожної платформи можливо налаштувати такі алгоритми ведення діалогу, які в результаті приведуть до видозмінення одного і того ж запиту для кожної з них в контексті одного агента. Dialogflow підтримує такі платформи і мови



програмування, як iOS, Android, Cordova, JavaScript, Python, C #, Xamarin і Unity. Вагомою перевагою є наявність вбудованого редактора коду для кодування, тестування та реалізації різних дій безпосередньо в консолі Dialogflow.

Приймаючи до уваги усі переваги, наведені вище, для створення програмного модуля інформаційної технології автоматизації онлайн-навчання обрано платформу Google Dialogflow, оскільки: вона є універсальною для будь-яких платформ та пристроїв, що дозволить користувачам використання бота у зручному для себе месенджері чи браузері, з будь-якого пристрою; дає можливість створення застосунків з використанням модулів природної мови; функціоналу безкоштовної версії платформи повністю вистачає для розв'язання поставленої задачі, тому вважатимемо що сервіс є умовно безкоштовним.

### **3.2 Розробка алгоритму функціонування інформаційної технології автоматизації онлайн-навчання**

Алгоритм функціонування інформаційної технології автоматизації навчання можна розділити на алгоритм функціонування клієнтського модуля та алгоритм функціонування модуля адміністратора.

Графічне представлення алгоритму функціонування клієнтського модуля інформаційної технології автоматизації онлайн-навчання представлено на рис. 3.3.

Опишемо наведений алгоритм функціонування клієнтського модуля інформаційної технології автоматизації онлайн-навчання, який подано на рис. 3.3:

1. Користувач відкриває головну сторінку додатку.
2. Якщо користувач не зареєстрований, відбувається спілкування бота з клієнтом для ознайомлення з курсом, в іншому випадку – крок 10.
3. Чат-бот ознайомлює користувача з курсами, які надає компанія.
4. Чат-бот ознайомлює користувача з тарифами навчання.

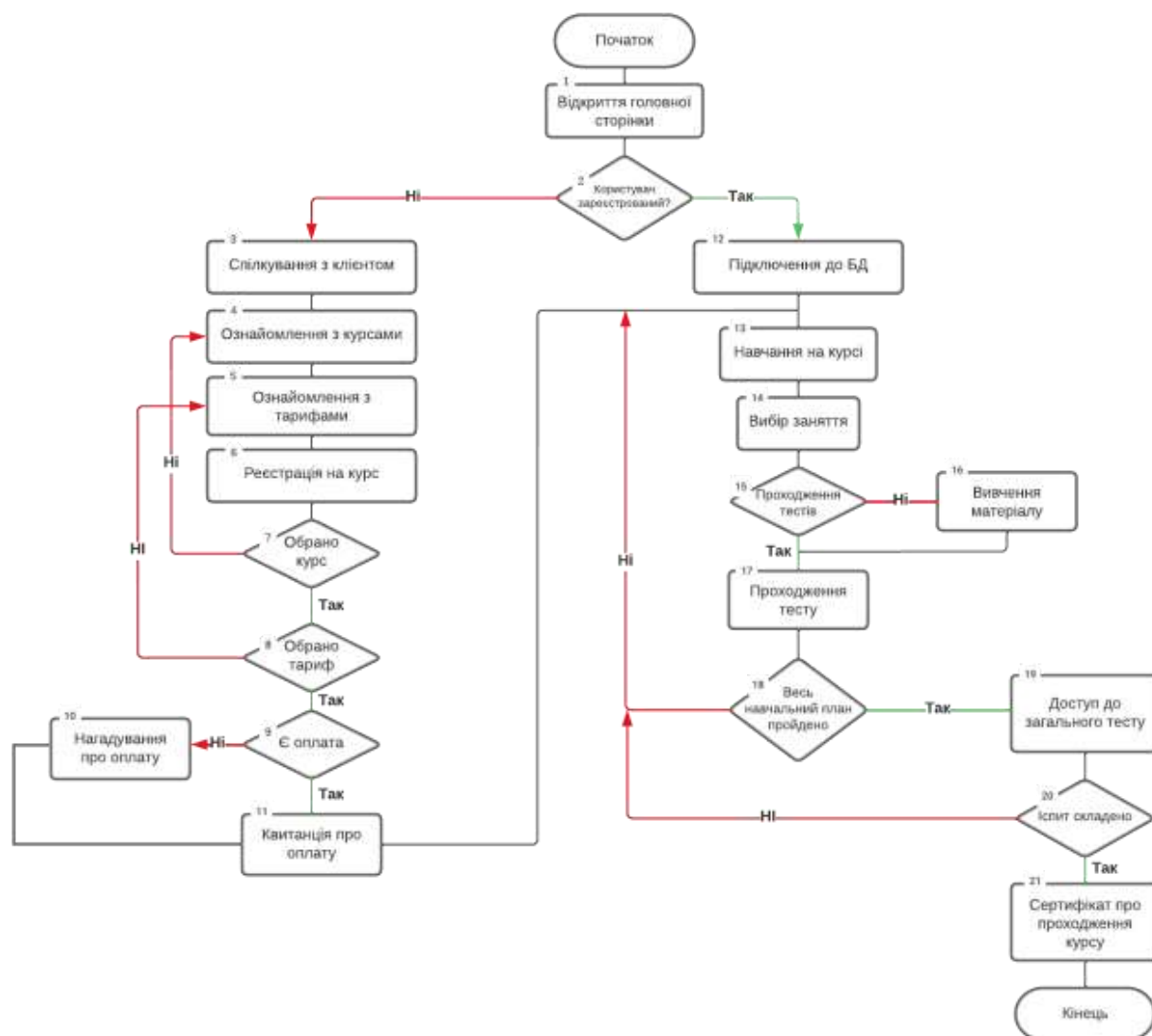


Рисунок 3.3 – Схема алгоритму функціонування клієнтського модуля інформаційної технології автоматизації онлайн-навчання

5. Проводиться реєстрація користувача на курс, усі дані, отримані на цьому етапі відправляються в базу даних.

6. Якщо користувачем обрано потрібний для нього курс – перехід до кроку 7, інакше – крок 3.

7. Якщо користувачем обрано потрібний для нього тариф – перехід до кроку 8, інакше – крок 4.

8. Якщо користувач провів оплату – перехід до кроку 9, інакше – чат-бот надсилає нагадування користувачу.

9. Користувач надсилає квитанцію про оплату навчання, щоб отримати доступ до матеріалів.

10. Користувач починає навчання та робить вибір заняття.

11. Якщо користувачем обрано вивчення матеріалу – відкривається база з заняттями та навчальними матеріалами, інакше – крок 13.

12. Користувач проходить навчання. Після вивчення теми користувач має пройти тестування по вивчених матеріалах.

13. Відкривається сторінка тестування.

14. Якщо користувачем ще не пройдено всю навчальну програму – перехід до кроку 10. Інакше – крок 11.

15. Користувач отримує доступ до загального іспиту.

16. Якщо іспит не складено, перехід до кроку 10. Інакше – крок 17.

17. Користувач отримує офіційний сертифікат про проходження курсу.

18. Закінчення діалогу користувача із системою.

Графічне представлення алгоритму функціонування модуля адміністратора інформаційної технології автоматизації онлайн-навчання представлено на рис. 3.4.

Опишемо наведений алгоритм функціонування модуля адміністратора інформаційної технології автоматизації онлайн-навчання, який подано на рис. 3.4:

1. Адміністратор відкриває головну сторінку додатку.  
2. Вибір адміністратора працювати з студентом чи самим курсом.  
3. Якщо адміністратор планує працювати з профілем студента – перехід до кроку 4, якщо планується робота над курсом – перехід до кроку 6.

4. Керування профілем студента.

5. Якщо адміністратор бажає завершити роботу у системі – перехід до кроку 19, інакше – крок 2.

6. Вибір адміністратора працювати над модернізацією відгуків чи з навчальними матеріалами.

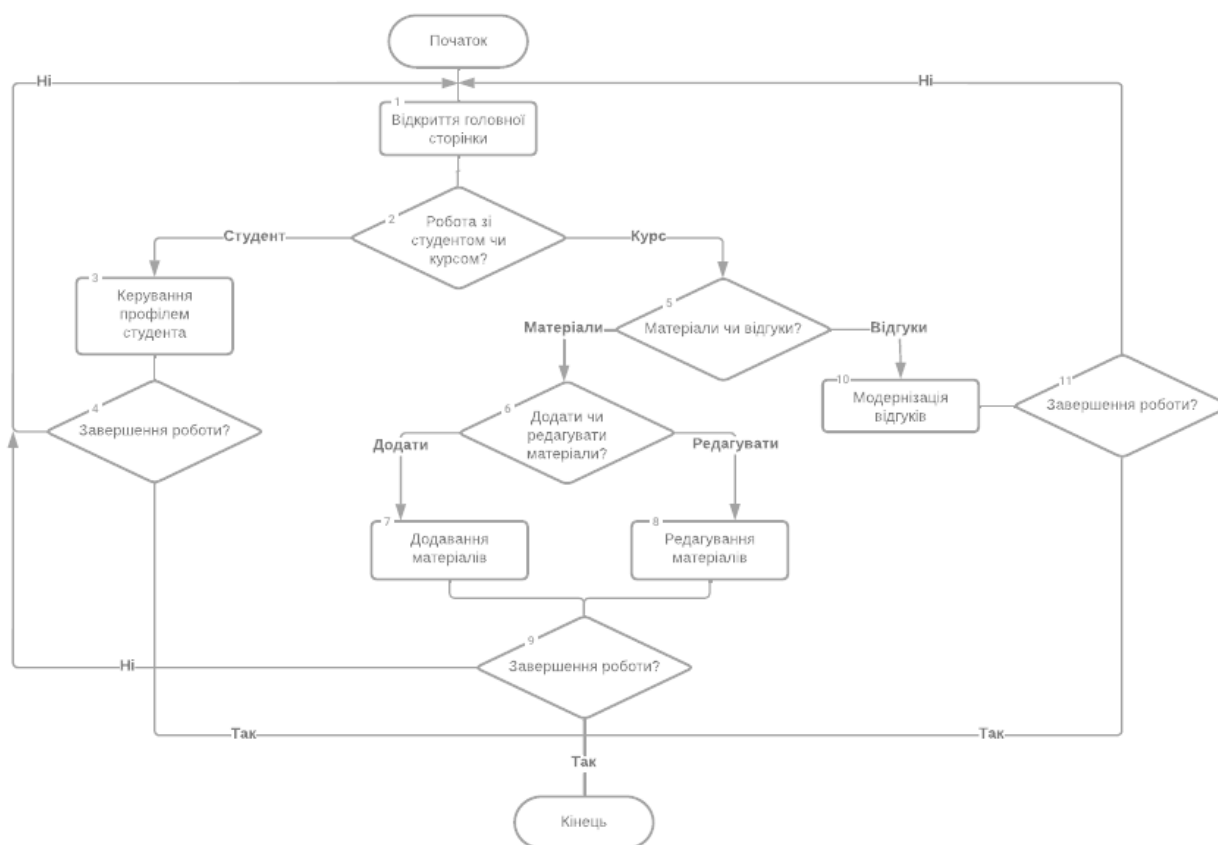


Рисунок 3.4 – Схема алгоритму функціонування модуля адміністратора інформаційної технології автоматизації онлайн-навчання

7. Якщо адміністратор планує працювати з відгуками – перехід до кроку 8, якщо планується робота з навчальними матеріалами – крок 10.
8. Модернізація відгуків.
9. Якщо адміністратор бажає завершити роботу у системі – перехід до кроку 19, інакше – крок 2.
10. Вибір адміністратора: редагувати чи додати матеріали.
11. Якщо адміністратор планує редагувати матеріали – перехід до кроку 12, якщо планується додавання матеріалів – крок 14.
12. Редагування матеріалів.
13. Якщо адміністратор бажає завершити роботу у системі – перехід до кроку 19, інакше – крок 2.
14. Додавання матеріалів.

15. Якщо адміністратор бажає завершити роботу у системі – перехід до кроку 19, інакше – крок 2.

16. Завершення діалогу адміністратора із системою.

### **3.3 Обґрунтування вибору інструменту для реалізації експертної системи для інформаційної технології автоматизації онлайн-навчання**

Одною з основних функцій інформаційної технології автоматизації онлайн-навчання є підбір правильного онлайн-курсу для користувача, в плані зацікавленості та рівня знань.

Для визначення рівня знань користувача чи правильного підбору курсу, чат-бот має отримати певний перелік інформаційних даних для виведення максимально правильного результату. Тому, вдалим рішенням буде інтеграція експертної системи у чат-бот, яка зможе провести своєрідне інтерв'ю з клієнтом, задавши йому потрібні для аналізу даних питання та визначить який саме курс прийнятний для конкретного користувача. Така функція є максимально корисною як для клієнтів, так і для самої організації адже на виході клієнт отримує максимально корисні і потрібні саме для нього знання, отримані на онлайн-навчанні, а організація отримує задоволених професійним рівнем компанії клієнтів, які в майбутньому знову звертатимуться за їхніми послугами та радитимуть іншим потенційним клієнтам.

Експертна система – це комп'ютерна програма (програмне забезпечення), яка використовує штучний інтелект (ШІ) для відтворення суджень людини з експертними знаннями в певній галузі. Експертна система намагається діяти як людина-експерт у певній предметній галузі. Дані в базі знань додаються людьми, які є експертами в певній галузі, а експертна система використовується не експертом для отримання інформації.

Будь-яка експертна система складається з таких компонентів, як:

1. База знань. Тут зберігаються знання (інформація). Вона створена на основі інформації, наданої людьми-експертами. Це збірка фактів та правил. Чим

краща якість інформації та розуміння проблем для кінцевого користувача, тим кращі результати.

2. Механізм висновків. Працює як пошукова система, перевіряючи базу знань на наявність інформації, яка відповідає запиту/пошуку користувача.

3. Інтерфейс користувача. Гарний користувацький інтерфейс дозволяє користувачам, які не є фахівцями, звертатися до експертної системи (ставити запитання) і отримувати поради (відповідь, яку легко зрозуміти) [41].

Комп'ютерна система «КАРКАС» реалізує основні інструментальні засоби, сервіси (middleware) для побудови баз знань предметної галузі за допомогою ієрархічної функціональної системи і таким чином полегшує побудову експертних та експертно-навчальних систем. Використання месенджерів як співрозмовника при роботі з «КАРКАС» дає більше можливостей оперативно консультуватися з експертною системою через смартфон, що, наприклад, важливо для прийняття ефективних рішень у різних предметних галузях. Для представлення знань розроблена ієрархічна функціональна система, яка генерується системою на основі правил продукцій і фреймів. Машина висновку використовує ієрархічну функціональну систему під час консультації з користувачем. Архітектура системи побудована за модульним принципом і тому має можливість приєднання інших додаткових модулів [42].

В якості інструменту для побудови експертних систем обрано мову програмування CLIPS.

CLIPS – це мова програмування на основі правил, що використовується для створення експертних систем та інших програм, де евристичні рішення легше реалізувати та підтримувати, ніж алгоритмічні рішення. CLIPS є багатопарадигмальною мовою та підтримує програмування на основі правил, об'єктно-орієнтоване та процедурне програмування.

Оскільки у CLIPS відсутній власний графічний інтерфейс користувача, інтегрування в програму – єдиний спосіб використання його можливості.

Програма, написана в CLIPS, може складатися з правил, фактів і об'єктів. Механізм логічного висновку вирішує, які правила мають бути виконані та коли.

Експертна система на основі правил, написана на CLIPS, є програмою, керованою даними, де факти та, за бажанням, об'єкти є даними, які стимулюють виконання за допомогою механізму логічного висновку.

Стандартне використання полягає у створенні або редагуванні бази знань за допомогою будь-якого стандартного текстового редактора, збереженні бази знань як один або декілька текстових файлів, виході з редактора та виконанні CLIPS, а потім завантаженні бази знань у CLIPS.

Основна ідея полягає в представленні знань у вигляді такої форми:

Правило1:

ЯКЩО

(виконуються умови1)

ТОДІ

(виконати дії1)

Правило2:

ЯКЩО

(виконуються умови2)

ТОДІ

(виконати дії2) ...

Основними перевагами мови програмування CLIPS є:

1. Це високорівневий інтерпретатор правил (оболонка);
2. Синтаксис подібний до LISP;
3. Факти та база правил схожі на Prolog;
4. Вищий рівень порівняно з LISP або Prolog;
5. Працює на UNIX, Linux, DOS, Windows, MacOS;
6. Загальнодоступне та добре задокументоване програмне забезпечення;
7. Включає об'єктно-орієнтовані конструкції [43].

Зважаючи на описані вище переваги та недоліки для реалізації бази знань обрано систему «Каркас», а для реалізації самої експертної системи мову програмування CLIPS.

### 3.4 Реалізація програмного модуля інформаційної технології автоматизації онлайн-навчання

Для зарахування користувача на курс, потрібно зареєструвати його в системі. Реєстрація – це співбесіда чат-бота з користувачем за схемою «питання-відповідь», усі отримані від користувача дані заносяться в базу даних. Після реєстрації користувач починає навчання за обраною спеціалізацією та згідно з обраним тарифом.

Діаграма діяльності сервера процесу реєстрації користувача зображена на рис. 3.5.



Рисунок 3.5 – Діаграма діяльності сервера для проведення реєстрації

Зображений вище алгоритм (рис. 3.5) можна записати у вигляді коду:

```

private static async void BotOnCallbackQueryReceived(object sender,
CallbackQueryEventArgs callbackQueryEventArgs)
{
    var callbackQuery = callbackQueryEventArgs.CallbackQuery;
    await Bot.AnswerCallbackQueryAsync(

```



```

callbackQueryId: callbackQuery.Id,
text: $"Received {callbackQuery.Data}"
);
await Bot.SendTextMessageAsync(
chatId: callbackQuery.Message.Chat.Id,
text: $"Received {callbackQuery.Data}"
);
}

```

Під час вибору курсу та тарифу сервер має поставити уточнювальні запитання користувачу. У випадку, коли серверу потрібно уточнення, він ставить запитання та надсилає запити до бази даних. Відповідно до відповіді користувача сервер виводить отриманий результат у вигляді привітання з успішною реєстрацією на конкретний курс та дає доступ до навчальних матеріалів.

Діаграма діяльності сервера для вибору онлайн-курсу зображена на рис. 3.6.



Рисунок 3.6 – Діаграма діяльності сервера для вибору онлайн-курсу

Зображений вище алгоритм (рис. 3.6) можна записати у вигляді коду:  
if (msg.Text == /start)

```

{
    if(sql.State == ConnectionState.Closed) // Перевірка чи закрито підключення до sql
    {
        sql.Open(); //Открываем sql подключение
        SqlCommand command = new SqlCommand($"insert into %Назва_таблиці% (%Стовпець_1%,
        %Стовпець_2%) values ('{msg.Chat.Id}', '0')", sql); //Прописуємо додавання даних до даблиці
        await command.ExecuteNonQueryAsync(); //Виконання команди
        sql.Close();
    }
}

if (msg.Text == "Ім'я")
{ if(sql.State == ConnectionState.Closed) //Перевірка стану підключення
    {
        //Відправка повідомлення, щоб ввели ім'я
        sql.Open();
        SqlCommand command = new SqlCommand($"Update %Назва_таблиці% Set
        %Стовпець_2% = 'Ім'я' where %Стовпець_1% = '{msg.Chat.Id}'", sql);
        //Присвоєння значення ім'я
        await command.ExecuteNonQueryAsync();
        sql.Close();
    }
}

if(msg.Text == "Прізвище")
{
    if(sql.State == ConnectionState.Closed)
    {
        sql.Open();
        SqlCommand command = new SqlCommand($"Update %Назва_таблиці% Set
        %Стовпець_2% = 'Прізвище' where %Стовпець_1% = '{msg.Chat.Id}'", sql);
        await command.ExecuteNonQueryAsync();
        //Присвоєння значення Прізвище
        sql.Close();
    }
}

```

Важливою функцією реалізованої інформаційної технології є розподіл ролей для різних типів користувачів, що дозволяє використовувати чат-бота і в ролі адміністратора.

Користувач-адміністратор зможе керувати профілем студента, при необхідності редагувати досягнення, додавати матеріали, оголошення, акції та інший потрібний контент. Також адміністратор бачить весь список студентів.

Діаграма діяльності сервера для роботи адміністратора з навчальними матеріалами показано на рис. 3.7

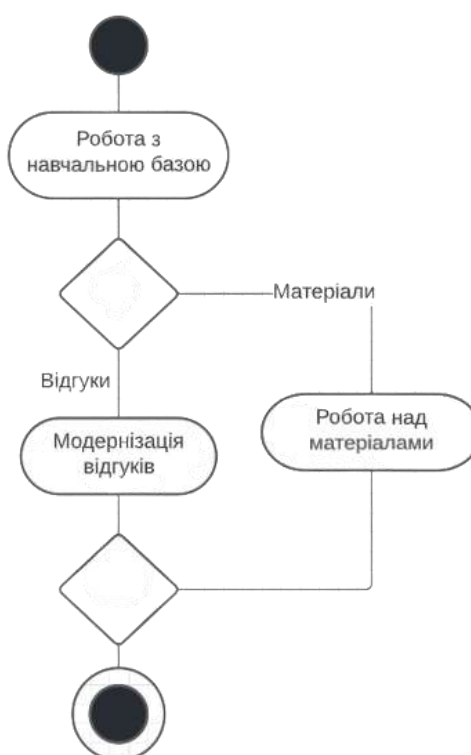


Рисунок 3.7 – Діаграма діяльності сервера для роботи адміністратора з навчальними матеріалами

Реалізувати розподіл ролей для нашої інформаційної технології можна таким чином:

```

{
  {&quot;id&quot;;,id},
  {&quot;first_name&quot;;, first_name},
  {&quot;username&quot;;, username},
  {&quot;photo_url&quot;;, photo_url},

```

```

{&quot;auth_date&quot;:, auth_date},
{&quot;hash&quot;:, hash}
});
// if the authorization was successful, create the user (if not exist) and sign in
if (auth == Authorization.Valid)
{
var user = await _userManager.FindByNameAsync($&quot;tg{id}&quot;);
if (null == user)
{
user = new ApplicationUser()
{
UserName = $&quot;tg{id}&quot;,
TelegramNativeId = long.Parse(id),
TelegramUserName = username,
FirstName = first_name,
PhotoUrl = photo_url
};
var result = await _userManager.CreateAsync(user);
if (!result.Succeeded)
{
ViewBag.ErrorTitle = &quot;Internal error&quot;;
ViewBag.ErrorMessage = $&quot;Failed to create user tg{id}&quot;;
return View(&quot;Error&quot;);
}
user = await _userManager.FindByNameAsync($&quot;tg{id}&quot;);
if (null == user)
{

```

У ході реалізації програмного модуля інформаційної технології автоматизації навчання застосовано такі класи як User, Admin, Processing та Product. Ці класи потрібні для підключення та інтеграції необхідного стеку технологій, обміну, конвертації, запису інформації і взаємодії чат-бота з користувачем. Підключення сторонніх сервісів дає можливість спростити розробку програми і створити потужний продукт на базі готових рішень.

На рис. 3.8 зображено UML-діаграма класів інформаційної технології автоматизації онлайн-навчання.

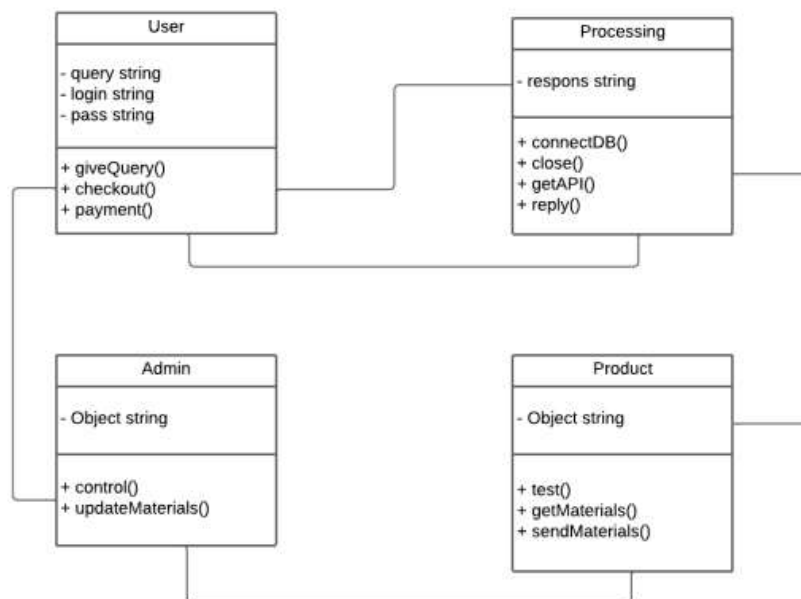


Рисунок 3.8 – Загальна UML-діаграма класів

Клас Processing та його метод connectDB() забезпечує можливість застосунку підключитися до сервера баз даних MongoDB. Метод close() служить для закриття підключень до БД. Метод getAPI() – дозволяє отримати

API ключ для інтеграції з сервісом від Google – DialogFlow. Метод reply() створений для обробки «відповідей» на запити користувача та запису необхідних даних з відповідей бота в базу даних. Клас User створений для взаємодії користувачів з програмою. Метод giveQuery() слугує для обробки запитів користувача (фраз людською мовою) та запису необхідних даних з відповідей бота в базу даних. Метод checkout() дозволяє реалізувати систему реєстрації/входу користувача в систему. Метод payment() дозволяє підв'язати платіжну систему та зробити просту систему оплати.

Клас Product містить в собі методи необхідні для виводу інформації з бази даних/сервера. Метод test() створений для реалізації попереднього тестування користувачів, що проблематично зробити, через сервіс DialogFlow.

Призначення `getMaterials()` керування доступом до матеріалів курсів. `sendMaterial()` – упорядковує інформацію, надіслану користувачами. Метод створений для взаємодії з кураторами.

### 3.5 Тестування інформаційної технології автоматизації онлайн-навчання та аналіз отриманих результатів

Щоб почати роботу з інформаційною технологією автоматизації онлайн навчання потрібно мати смартфон, комп'ютер чи планшет. Головною умовою є стабільне підключення до мобільного інтернету або мережі Wi-Fi.

Важливо, що для якісного відображення інтерфейсу інформаційної технології не є важливим який пристрій використовується користувачем. При розробці системи акцентувалася увага на обов'язковій зручності використання на будь-якому пристрої.

Інформаційна технологія автоматизації онлайн-навчання була протестована для утвердження коректності її роботи. Проведено 120 запусків додатку, протестовано його можливості та оцінено роботу.

На рис. 3.9 подано «Вікно початкової активності» – перший крок роботи.

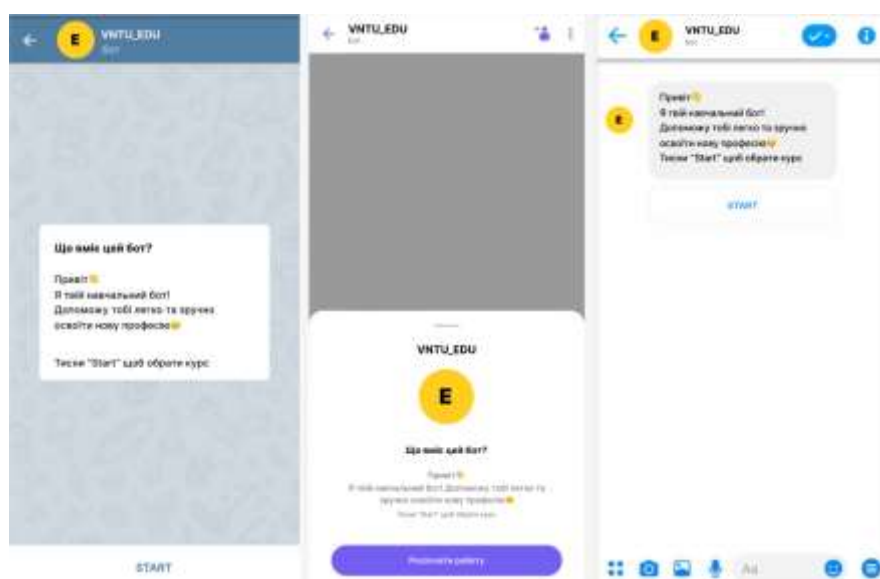


Рисунок 3.9 – Загальний вигляд інтерфейсного вікна початкової активності у месенджерах Telegram, Viber, Facebook Messenger

Першим кроком є запуск програмного модуля на зручній для користувача платформі. Після запуску відкривається вікно початкової активності в месенджері. Вікно початкової активності містить функцію /start для виклику бота.

Чат-бот знайомить користувача з системою, описує свої можливості та можливості користувача. Після знайомства користувач має можливість ознайомитися з онлайн-курсами, які надає навчальна організація та тарифами на навчання. Також користувач має можливість пройти опитування, якщо він не визначився точно яка спеціалізація йому підходить найбільше. Також користувачеві надається можливість пройти тестування на визначення існуючого рівня знань для компетентнішого підбору навчального курсу. Перед користувачем розміщені такі кнопки, як «Курс 1», «Курс 2», «Курс 3», «Курс 4», «Курс 5», «Тарифи», «Відгуки», «Тестування», «Підбір курсу» (рис. 3.10).

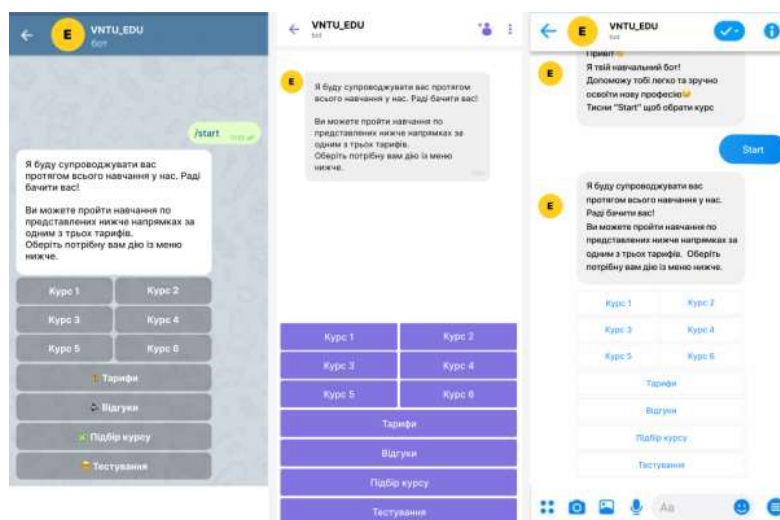


Рисунок 3.10 – Загальний вигляд інтерфейсного вікна знайомства з користувачем у месенджерах Telegram, Viber, Facebook Messenger

Коли користувачем обрано навчальний курс, він має натиснути відповідну кнопку з назвою курсу, після чого з'явиться необхідна ознайомлювальна інформація про навчання. Користувач має змогу ознайомитися з цією інформацією, повернутися до наявних курсів, переглянути тарифи і безпосередньо зареєструватися на курс. Відповідно, перед користувачем розміщено кнопку «Реєстрація», «До курсів», «Тарифи» (рис. 3.11).

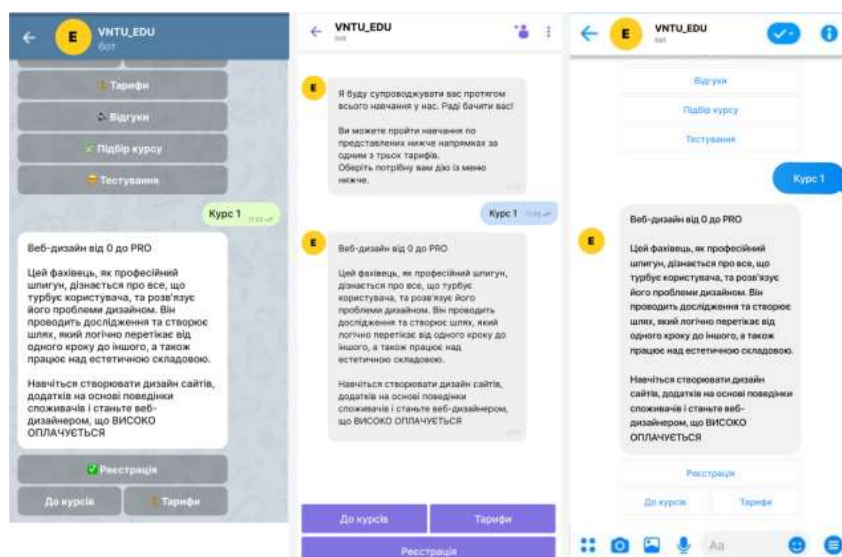


Рисунок 3.11 – Загальний вигляд інтерфейсного вікна ознайомлення з курсом у месенджерах Telegram, Viber, Facebook Messenger

Коли користувач запускає функцію реєстрації на курс, чат-бот починає проведення своєї рідної співбесіди, ставлячи користувачеві запитання, необхідні для заповнення реєстраційних даних, які потім заносяться у базу даних навчальної організації. Це такі дані, як «Прізвище», «Ім'я», «Електронна пошта», «Обраний тариф навчання».

На рис. 3.12 показано процес реєстрації.

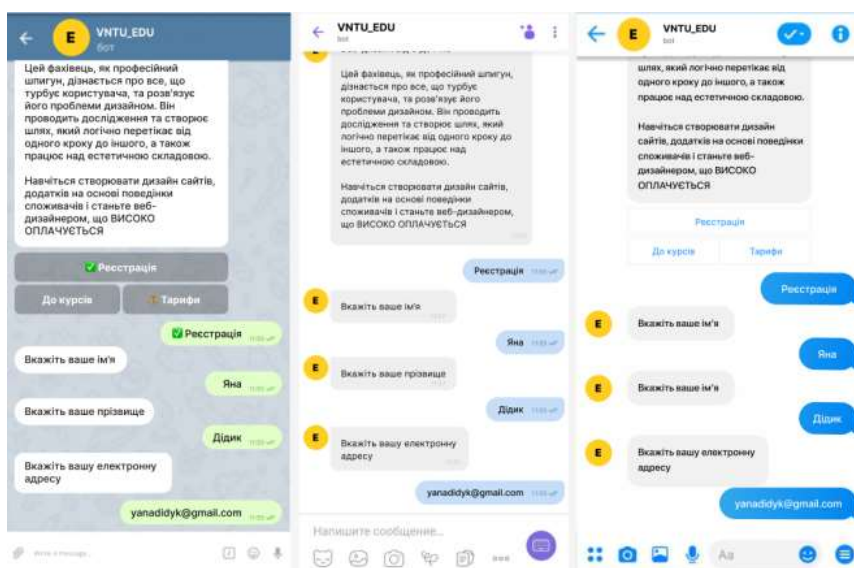


Рисунок 3.12 – Загальний вигляд інтерфейсного вікна реєстрації на курс у месенджерах Telegram, Viber, Facebook Messenger



Коли процес реєстрації завершено, користувач отримує на вказану ним раніше електронну скриньку форму для оплати згідно з обраним тарифом. Користувач має оплатити навчання та надіслати документ про оплату чат-боту. Той у свою чергу відправляє квитанцію адміністратору для перевірки.

Після вдалої оплати користувач отримує доступ до своїх навчальних матеріалів.

На рис. 3.13 показано приклад факту відправки користувачем документу про оплату за навчання.

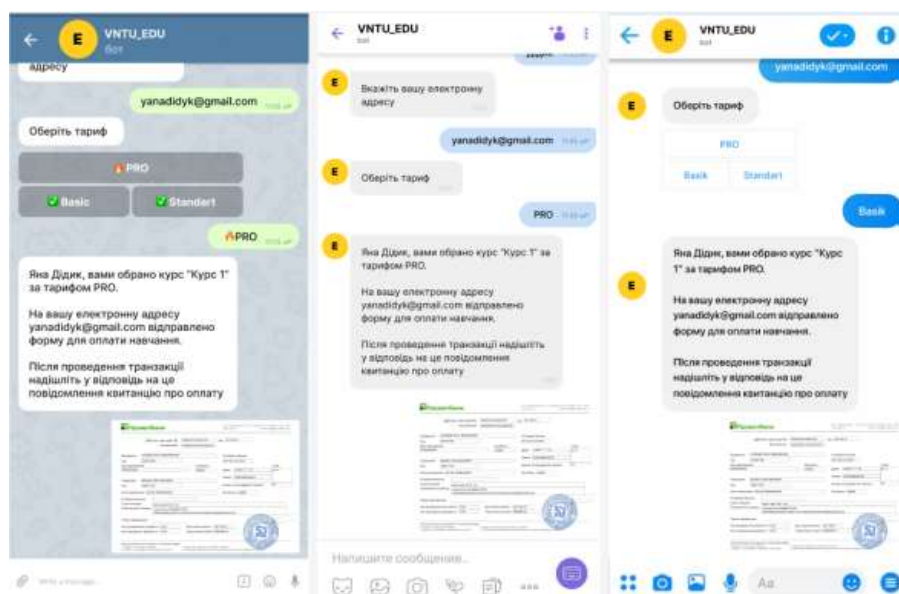


Рисунок 3.13 – Загальний вигляд інтерфейсного вікна підтвердження оплати навчання у месенджерах Telegram, Viber, Facebook Messenger

Користувач поступово отримує та освоює матеріали по обраному курсу. Туди входять відео-лекції, текстові файли з літературою, текстові файл з домашнім завданням та посилання на тести. По проходженні матеріалу теми користувач надсилає виконане домашнє завдання чат-боту, натиснувши відповідно кнопку «Надіслати домашнє завдання», яке потім перенаправляється куратору для перевірки. З часом бот повідомляє результати та надсилає перелік помилок, на які варто звернути увагу. Якщо у користувача виникають питання по зауваженнях, він може отримати консультацію куратора, відповідно натиснувши кнопку

«Консультація куратора». Якщо питання відсутню – користувач переходить на наступну тему (рис. 3.14).

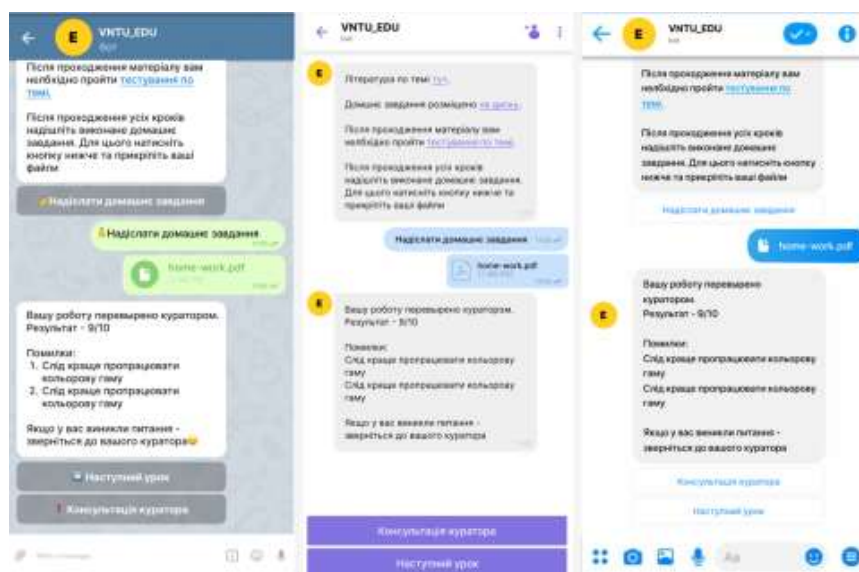


Рисунок 3.14 – Загальний вигляд інтерфейсного вікна процесу навчання у месенджерах Telegram, Viber, Facebook Messenger

Після успішного проходження усієї навчальної програми, чат-бот перенаправляє користувача до екзаменаційного тестування. Для цього потрібно натиснути кнопку «Екзаменаційне тестування». У разі вдалого проходження тестування, користувач отримує сертифікат про проходження курсу та має можливість залишити відгук про навчання (рис. 3.15).

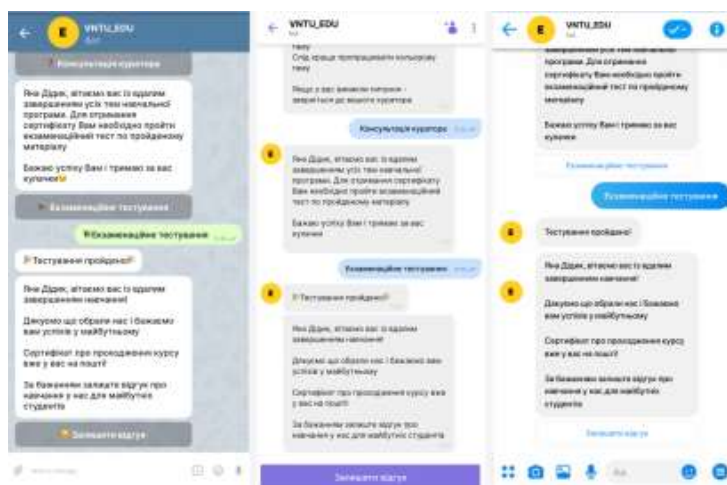


Рисунок 3.15 – Завершення процесу навчання у месенджерах Telegram, Viber, Facebook Messenger

Підсумуємо результати тестування та порівняємо розроблений програмний модуль з існуючими системами-аналогами (табл. 3.2).

Таблиця 3.2 – Порівняльний аналіз систем-аналогів

Характеристика	vntubot	HomeOnlineEnglishBot	ProgAcademyBot	InMind
Вибір курсу для навчання	+	-	+	-
Реєстрація користувача на курс	+	-	-	-
Вибір тарифу та проведення оплати	+	-	+	-
Навчання	+	+	+	+
Тестування	+	+	+	+
Розподіл ролей	+	-	-	-
Мултіленг	+	+	-	-
Підтримка різних меседжерів	+	-	-	-
Підтримка голосових повідомлень	+	-	-	-

У результаті порівняльної характеристики розробленого програмного модуля «vntubot» з існуючими системами-аналогами («HomeOnlineEnglishBot», «ProgAcademyBot», «InMind»), було виявлено такі головні переваги:

- можливість реєстрації користувача в базі навчального курсу;
- можливість проведення оплати навчання безпосередньо в системі чат-бота;
- можливість розподілу ролей (гілка студента та гілка адміністратора);
- підтримка голосових повідомлень;
- підтримка різних месенджерів, при чому, навчання не прив'язане до одного месенджера, користувач в будь-який момент може продовжити навчання у будь-якому із запропонованих вище обмінників, так боти у кожному месенджері оперують даними однієї бази.

У табл. 3.3 наведено порівняння результатів роботи розробленої програми з програмами-аналогами ProgAcademyBot і HomeOnlineEnglishBot. При поданні матеріалів курсів з використанням розробленої інформаційної технології більша кількість студентів проходить курс до кінця.

Таблиця 3.3 – Порівняльний аналіз достовірності роботи алгоритмів програмних додатків

Програмний додаток	Кількість завершених навчальних програм, %
Розроблений додаток	75,7%
ProgAcademyBot	73,4%
HomeOnlineEnglishBot	73,1%

Порівнюючи результати роботи розробленого програмного продукту з програмами-аналогами ProgAcademyBot та HomeOnlineEnglishBot, за критерієм «більша кількість завершених програм», розроблений програмний модуль інформаційної технології автоматизації онлайн-навчання забезпечує більшу кількість пройдених до кінця курсів, яка становить 75,7% від усіх слухачів на початковому етапі вивчення, ніж програми-аналоги, а саме 73,4% для ProgAcademyBot та 73,1% для HomeOnlineEnglishBot.

Проаналізувавши можливості та переваги розробленого програмного модуля інформаційної технології автоматизації онлайн-навчання, можна зробити висновок, що така інформаційна технологія є ефективним засобом для автоматизації онлайн-навчання. З її допомогою можна організувати процес набору, реєстрації на онлайн-курси та безпосереднього навчання студентів, при цьому не маючи прив'язки до дня чи періоду доби, графіку викладачів, особистих обставин студентів. Реалізація цієї інформаційної технології зможе вдосконалити онлайн-навчання в навчальних організаціях, надавши можливість користувачам детально ознайомитися з програмою того чи іншого онлайн-курсу, зареєструватися,

проводити оплату, проходити тестування, переглядати лекції та отримувати матеріали в будь-який зручний для нього час у форматі 24/7.

### **3.6 Висновок до розділу 3**

У третьому розділі обґрунтовано вибір засобів реалізації інформаційної технології автоматизації онлайн-навчання. Для реалізації інформаційної технології автоматизації онлайн-навчання було обрано клієнт-серверний тип архітектури, оскільки така архітектура підтримує багатокористувацьку роботу, гарантує цілісність даних, має механізми керування правами доступу до ресурсів сервера та можливість розподілу функцій між вузлами мережі.

Проаналізовано найчастіше використовувані для розробки чат-ботів мови програмування: C++, C# та Python. Визначено їх особливості, переваги та недоліки, а також доцільність у використанні для поставленого завдання. В результаті аналізу для реалізації програмного модуля для інформаційної технології було обрано мову C#. Переваги мови програмування C#: повна об'єктно-орієнтованість мови; потужність з можливостями спадкування й універсалізації; потужність бібліотек каркасів, що забезпечує зручність реалізації будь-яких типів додатків на C#, даючи змогу легко будувати Web-служби, просто зберігати та використовувати інформацію з бази даних.

Проаналізовано такі інструменти для розробки чат-ботів, як ManyChat, Google Dialogflow, Amazon Lex, IBM Watson, Botsify. Для реалізації інформаційної технології автоматизації онлайн-навчання обрано сервіс Google Dialogflow, оскільки він відзначився універсальністю для різних платформ та пристроїв, можливістю створення застосунків з використанням модулів природної мови, безкоштовним використанням сервісу.

Побудовано архітектуру інформаційної технології автоматизації онлайн-навчання, спроектовано алгоритм функціонування клієнтського модуля та алгоритм функціонування модуля адміністратора.

Описано процес реалізації реєстрації користувача та вибору курсу для гілки користувача та роботи над навчальними матеріалами для гілки адміністратора та наведені їх діаграми діяльності сервера. Також продемонстровано розподіл ролей для інформаційної технології автоматизації навчання.

В ході досліджень чат-бот показав себе як якісний продукт, що відповідає усім поставленим вимогам.

Проведено тестування реалізованого програмного модуля інформаційної технології автоматизації онлайн-навчання, яке підтвердило правильність його функціонування.

Порівнюючи результати роботи розробленого програмного продукту з програмами-аналогами ProgAcademyBot та HomeOnlineEnglishBot, за критерієм «більша кількість завершених програм», розробка забезпечує більшу кількість пройдених до кінця курсів, яка становить 75,5% від усіх слухачів на початковому етапі вивчення, ніж програми-аналоги, а саме 73,4% для ProgAcademyBot та 73,1% для HomeOnlineEnglishBot. Розроблений програмний продукт показав гарні результати в порівнянні з аналогами.

Тестування інформаційної технології показало повну відповідність до поставлених задач.

## 4 ЕКОНОМІЧНА ЧАСТИНА

### 4.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нового програмного продукту інформаційної технології автоматизації онлайн-навчання. Дана робота допомагає організувати такі процеси, як набір та реєстрація на онлайн-курси та безпосереднього онлайн-навчання студентів, допомога з вибором потрібного онлайн-курсу та визначення існуючого рівня знань потенційного студента, при цьому не маючи прив'язки до дня чи періоду доби, графіку викладачів, особистих обставин студентів. Особливістю програми є те, що дана технологія використовує інтеграцію експертної системи для вдалого підбору курсу та наявність гілок адміністратора та користувача.

Аналогом може бути HomeOnlineEnglishBot за ціною 10000 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 4.1.

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах

Продовження таблиці 4.1

Бали (за 5-ти бальною шкалою)					
Кри- терій	0	1	2	3	4
Ринкові переваги					
2	Багато аналогів на малому ринку	Ринкові п Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно до-рівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі влас-тливості продук-ту значно гірші, ніж в аналогів	Технічні та споживчі влас-тливості продук-ту трохи гірші, ніж в аналогів	Технічні та споживчі влас-тливості продук-ту на рівні аналогів	Технічні та споживчі влас-тливості продук-ту трохи кращі, ніж в аналогів	Технічні та споживчі вла-стивості продук-ту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позити-вної динаміки	Ринок малий, але має пози-тивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ри-нок з позитивною динамікою
7	Активна конкуренція великих ком-паній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з ко-мерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне не-значне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так із комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресур-си. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування



## Продовження таблиці 4.1

Бали (за 5-ти бальною шкалою)					
Практик на здійсненність					
Кри-терій	0	1	2	3	4
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промислового комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в табл. 4.2.

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	4	4	4
Наявність аналогів на ринку	2	2	2
Цінова політика	4	4	4

Продовження таблиці 4.2

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	3	2	2
Ринок збуту	4	3	4
Конкурентоспроможність	3	4	2
Фахівці з технічної і комерційної реалізації	2	2	2
Фінансування	4	2	3
Матеріально-технічна база	4	2	3
Термін реалізації ідеї	3	3	4
Супровідна документація	2	3	4
Сума	39	34	38
Середньоарифметична сума балів	$(39+34+38) / 3 = 37$		

За даними табл. 4.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в табл. 4.3.

Таблиця 4.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 - 10	Низький
11 - 20	Ниже середнього
21 - 30	Середній
31 - 40	Вище середнього
41 - 48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є вище середнього, що дана технологія допомагає організувати такі процеси, як набір та реєстрація на онлайн-курси та безпосереднього онлайн-навчання студентів, допомога з вибором потрібного онлайн-курсу та визначення існуючого рівня знань потенційного студента, при цьому не маючи прив'язки до дня чи періоду доби, графіку викладачів, особистих обставин студентів [44].

#### 4.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t$$

де  $M$  – місячний посадовий оклад конкретного розробника (дослідника), грн.;

$T_p$  – число робочих днів в місяці, 22 днів;

$t$  – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до табл. 4.4.

Таблиця 4.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	30000	1363,64	48	65454,545
Інженер	27000	1227,27	48	58909,091
Всього				124363,64

Оскільки в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

Додаткова заробітна плата прийнято розраховувати як 15 % від основної заробітної плати розробників та робітників:

$$Z_d = Z_o * \frac{15\%}{100\%};$$

$$Z_d = 124363,64 * \frac{15\%}{100\%} = 18654,55 \text{ (грн.)}.$$

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$H_z = (Z_o + Z_d) * \frac{22\%}{100\%};$$

$$H_z = (124363,64 + 18654,55) * \frac{22\%}{100\%} = 31464,00 \text{ (грн.)}.$$

Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді амортизація обладнання, що використовувалась для розробки розраховується за формулою:

$$A = \frac{Ц}{T_B} * \frac{t_{\text{вик}}}{12} \text{ [грн.]},$$

де  $Ц$  – балансова вартість обладнання, грн.;

$T$  – термін корисного використання обладнання згідно податкового законодавства, років;

$t_{\text{вик}}$  – термін використання під час розробки, місяців.

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова

вартість якого становить 32600 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 2,18 міс.

$$A_{\text{обл}} = \frac{32600}{2} * \frac{2,18}{12} = 2963,64 \text{ (грн).}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до табл. 4.5. Для розрахунку амортизації нематеріальних ресурсів використовується формула:

$$A_{\text{н.р.}} = C_{\text{н.р.}} * N_a * \frac{t_{\text{вик}}}{12}.$$

Але, так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю,  $B_{\text{нем.ак.}} = 2400$  грн. (Microsoft Visual Studio).

Таблиця 4.5 – Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та компютерна периферія	32600	2	2,18	2963,636
Офісне обладнення	28000	4	2,18	1272,727
Приміщення	1100000	20	2,18	10000,000
Всього				14236,36

Тарифи на електроенергію для побутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників

(енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V * P * \Phi * K_n,$$

де  $V$  – вартість 1 кВт-години електроенергії для 1 класу підприємства,  $V = 6,2$  грн./кВт;

$P$  – встановлена потужність обладнання, кВт.  $P = 0,4$  кВт;

$\Phi$  – фактична кількість годин роботи обладнання, годин;

$K_n$  – коефіцієнт використання потужності,  $K_n = 0,9$ ,

$$V_e = 0,9 * 0,4 * 8 * 48 * 2,01 = 277,8624 \text{ (грн.)}.$$

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$I_B = (Z_o * Z_p) * \frac{H_{iB}}{100\%},$$

де  $H_{iB}$  – норма нарахування за статтею «Інші витрати»,

$$I_B = 124363,64 * \frac{100\%}{100\%} = 124363,6 \text{ (грн.)}.$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з

набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{\text{нзв}} = (З_о * З_р) * \frac{H_{\text{ів}}}{100\%},$$

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

$$H_{\text{нзв}} = 124363,64 * \frac{135\%}{100\%} = 167891 \text{ (грн.)}.$$

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$\begin{aligned} B_{\text{заг}} &= 124363,64 + 18654,55 + 31464,00 + 14236,36 + 2400 + \\ &+ 277,86 + 124363,6 + 167891 = 483650,95 \text{ (грн.)}. \end{aligned}$$

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

$$ЗВ = \frac{B_{\text{заг}}}{\eta} \text{ (грн)},$$

де  $\eta$  – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи [45].

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то  $\eta=0,1$ ; технічного проектування, то  $\eta=0,2$ ; розробки конструкторської документації, то  $\eta=0,3$ ; розробки технологій, то  $\eta=0,4$ ; розробки дослідного зразка, то  $\eta=0,5$ ; розробки промислового зразка, то  $\eta=0,7$ ; впровадження, то  $\eta=0,9$ . Оберемо  $\eta = 0,5$ , так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = \frac{483650,95}{0,5} = 967302 \text{ (грн)}.$$

### **4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором**

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

- вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;
- зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);
- кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;
- визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).



Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

#### **4.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем**

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

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

де  $\pm\Delta\Pi_0$  – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

$N$  – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

$\Pi_0$  – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки,  $\Pi_0 = \Pi_0 \pm \Delta\Pi_0$ ;

$\Pi_0$  – вартість програмного продукту у році до впровадження результатів розробки;

$\Delta N$  – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

$\lambda$  – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт  $\lambda = 0,8333$ ;

$\rho$  – коефіцієнт, який враховує рентабельність продукту;

$\vartheta$  – ставка податку на прибуток, у 2022 році  $\vartheta = 18\%$ .

Припустимо, що при прогнозованій ціні 3800 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 300 грн. Кількість одиниць

реалізованої продукції також збільшиться: протягом першого року – на 3000 шт., протягом другого року – на 2000 шт., протягом третього року на 1500 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0 * 300 + (3800 + 300) * 3000) * 0,8333 * 0,49 * (1 - 0,18) = 3817099,847 \text{ (грн.)},$$

$$\Delta\Pi_2 = (0 * 300 + (3800 + 300) * (3000 + 2000)) * 0,8333 * 0,49 * (1 - 0,18) = 6864083,059 \text{ (грн.)},$$

$$\Delta\Pi_3 = (0 * 300 + (3800 + 300) * (3000 + 2000 + 1500)) * 0,8333 * 0,49 * (1 - 0,18) = 8923307,976 \text{ (грн.)}.$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 19604490,88 грн.

#### **4.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності**

Розраховуємо приведену вартість збільшення всіх чистих прибутків  $\Pi\Pi$ , що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$\Pi\Pi = \sum_1^T \frac{\Delta\Pi_i}{(1 + \tau)^t}$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

$T$  – період часу, протягом якого виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

$\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau = 0,05 \dots 0,15$ ;

$t$  – період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

$$\begin{aligned} \text{ПП} &= \left( \frac{3817099,847}{(1 + 0,1) * 1} \right) + \left( \frac{6864083,059}{(1 + 0,1) * 2} \right) + \left( \frac{8923307,976}{(1 + 0,1) * 3} \right) = \\ &= 3470090,77 + 5672795,92 + 6704213,36 = 15847100,04 \text{ (грн.)}. \end{aligned}$$

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

$$PV = k_{\text{інв}} * ЗВ,$$

де  $k_{\text{інв}}$  – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай  $k_{\text{інв}}=2\dots5$ , але може бути і більшим;

$ЗВ$  – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 967302 = 1934603,81 \text{ (грн.)}.$$

Тоді абсолютний економічний ефект  $E_{\text{абс}}$  або чистий приведений дохід ( $NPV$ , *Net Present Value*) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$\begin{aligned} E_{\text{абс}} &= \text{ПП} - PV, \\ E_{\text{абс}} &= 15847100,04 - 1934603,81 = 13912496,23 \text{ (грн.)}. \end{aligned}$$

Оскільки  $E_{\text{абс}} > 0$  то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності ( $IRR$ , *Internal Rate of Return*) вкладених інвестицій та порівняти її з так званою

бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій  $E_B$ . Для цього використаємо формулу:

$$E_B = \sqrt[T_{ж}] \left( 1 + \frac{E_{абс}}{PV} \right) - 1,$$

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

$$E_B = \sqrt[3] \left( 1 + \frac{13912496,23}{1934603,81} \right) - 1 = 1.016.$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f,$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні  $d = (0,09...0,14)$ ;

$f$  – показник, що характеризує ризикованість вкладень; зазвичай, величина  $f = (0,05...0,5)$  [46].

$$\tau_{min} = 0.14 + 0.05 = 0.19.$$

Оскільки  $E_B > \tau_{min}$ , то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_B},$$

$$T_{ок} = 1 / 1,016 = 0,98 \text{ (р.)}.$$

Оскільки  $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,98 роки, то фінансування даної наукової розробки є доцільним.

#### **4.4 Висновок до розділу 4**

Економічна частина роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 967 302 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні цієї розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,98 роки.

## ВИСНОВКИ

У ході виконання магістерської кваліфікаційної роботи реалізовано інформаційну технологію автоматизації онлайн-навчання.

Проведено аналіз технічного рівня чат-ботів для онлайн-курсів та обґрунтовано доцільність розробки інформаційної технології автоматизації онлайн-навчання. Розглянуті системи-аналоги є корисними та доволі актуальними, проте не в повній мірі задовольняють потреби користувачів, відчутно обмежуючи їх в процесі навчання, в порівнянні зі звичайними аудиторними курсами.

На основі аналізу існуючих моделей та методів, які застосовуються для вирішення поставленої задачі було обґрунтовано використання в інформаційній технології математичної моделі прийняття рішень для діалогу агента з користувачем на основі математичної структури MDP. Математична структура MDP відповідає за генерацію нових відповідей на запити користувачів. Розглянуто діалог як процес прийняття рішень, який можна представити у вигляді Марковських процесів для моделювання розмови між агентом системи та користувачем.

Було розроблено структуру інформаційної технології автоматизації онлайн-навчання та спроектовано схему алгоритму роботи системи. Розглянуто структурну організацію системи навчальних онлайн-курсів.

Здійснено проектування та розробку інформаційної технології автоматизації онлайн-навчання.

Обґрунтовано вибір мови програмування та інструментів для розробки. Реалізацію інформаційної технології проведено на об'єктно-орієнтованій мові програмування C# з використанням сервісу DialogFlow.

Здійснено програмну реалізацію інформаційної технології автоматизації онлайн-навчання, детально описано класи та функції, які використовуються в інформаційній технології.

Проведено тестування розробленої програми та підтверджено її працездатність. Порівняно з прототипом розроблена інформаційна технологія

забезпечує більшу кількість пройдених до кінця курсів, а саме 75,5% від усіх слухачів, коли ProgAcademyBot має результат 73,4% та 73,1% – HomeOnlineEnglishBot, що означає доведення поставленої мети.

Економічно обґрунтовано доцільність розробки інформаційної технології автоматизації онлайн-навчання. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат, результатом стала сума 967 302 гривень. Шляхом аналізу розрахунків було доведено що розроблена інформаційна технологія автоматизації онлайн-навчання є дешевшою і висококонкурентоспроможнішою за аналог HomeOnlineEnglishBot. Період окупності інформаційної технології автоматизації онлайн-навчання складе близько 0,98 роки.

Завдання, які були поставлені для реалізації інформаційної технології автоматизації онлайн-навчання, виконані в повному обсязі.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дідик Я. В., Крилик Л. В. Розробка інтелектуального чат-бота інтегрованого з месенджером telegram для онлайн курсів // *Матеріали I науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2021) : збірник доповідей* [Електронний ресурс]. Вінниця: ВНТУ, 2021. С. 541–542.

URL: [https://conferences.vntu.edu.ua/public/files/1/vntu\\_2021\\_netpub.pdf](https://conferences.vntu.edu.ua/public/files/1/vntu_2021_netpub.pdf). (дата звернення: 18.10.22).

2. Дідик Я. В., Крилик Л. В. Удосконалення функціоналу інтелектуального чат-бота для автоматизації онлайн-навчання // *Матеріали II науково-технічної конференції підрозділів Вінницького національного технічного університету (НТКП ВНТУ–2022) : збірник доповідей* [Електронний ресурс]. Вінниця: ВНТУ, 2022. С. 735 – 737 (PDF, 2830 с.).

URL: [https://conferences.vntu.edu.ua/public/files/1/vntu\\_2022\\_netpub.pdf](https://conferences.vntu.edu.ua/public/files/1/vntu_2022_netpub.pdf). (дата звернення: 18.10.22).

3. Комп'ютерна програма «Інформаційна технологія автоматизації онлайн-навчання з допомогою інтелектуального чат-бота» : свідоцтво про реєстрацію авторського права на твір №114838, видане ДП «Український інститут інтелектуальної власності» (Укрпатент) від 16.09.2022 р. / Крилик Л. В., Дідик Я. В.

4. Ушакова І. О. Підходи до створення інтелектуальних чат-ботів. *Системи обробки інформації*. 2019. № 2(157). С. 76–83.

5. Яровий А. А., Кудрявцев Д. С. Інтелектуальний чат-бот для задачі розпізнавання природної мови [Електронний ресурс]. URL: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki2019/paper/view/7290>. (дата звернення: 15.10.22).

6. Facebook Messenger [Електронний ресурс]. URL: [https://uk.info.org/wiki/Facebook\\_Messenger](https://uk.info.org/wiki/Facebook_Messenger). (дата звернення: 15.10.22).



7. Работа в програмі Viber («Вайбер») [Електронний ресурс]. URL: <https://azbukainterneta.ru/schoolbook/extended/module7/part5/>. (дата звернення: 15.10.22).

8. Що таке Ватсап: загальний опис месенджера WhatsApp [Електронний ресурс]. URL: <http://smartandyoung.com.ua/shho-take-vatsap-zagalnij-opis-mesendzhera>. (дата звернення: 15.10.22).

9. Що таке Skype? [Електронний ресурс]. URL: [:https://www.skype.com/uk/about/](https://www.skype.com/uk/about/). (дата звернення: 15.10.22).

10. Telegram [Електронний ресурс]. URL: <https://play.google.com/store/apps/details?id=org.telegram.messenger>. (дата звернення: 15.10.22).

11. Instagram [Електронний ресурс]. URL: [:https://sites.google.com/site/djddjfdjfhdfh23/internet-servisi-dla-socmerezah](https://sites.google.com/site/djddjfdjfhdfh23/internet-servisi-dla-socmerezah). (дата звернення: 15.10.22).

12. Чат-боти у навчанні: 7 ідей для їх використання [Електронний ресурс]. URL: <https://learnlifelong.net/chat-boty-u-navchanni-7-idej-dlya-yih-vukory/>. (дата звернення: 20.10.22).

13. What is an Expert System and What is Used For? [Електронний ресурс]. URL: <https://expertisefinder.com/expert-systems/>. (дата звернення: 20.10.22).

14. Інтегрування месенджерів з експертними системами [Електронний ресурс]. URL: <https://expertisefinder.com/expert-systems/>. (дата звернення: 21.10.22).

15. What is Natural Language Processing? [Електронний ресурс]. URL: <https://www.ibm.com/cloud/learn/natural-language-processing>. (дата звернення: 21.10.22).

16. What is Deep Learning and How does it work? [Електронний ресурс]. URL: <https://towardsdatascience.com/what-is-deep-learning-and-how-does-it-work-2ce44bb692ac>. (дата звернення: 25.10.22).

17. Dynamic memory networks [Електронний ресурс]. URL: <https://medium.datadriveninvestor.com/dynamic-memory-networks-43b68da77481>. (дата звернення: 25.10.22).

18. SQL Introduction networks [Електронний ресурс]. URL: [https://www.w3schools.com/sql/sql\\_intro.asp](https://www.w3schools.com/sql/sql_intro.asp). (дата звернення: 15.10.22).

19. Relational Database: What It Is and Why It's Important [Електронний ресурс]. URL: <https://www.coursera.org/articles/relational-database>. (дата звернення: 29.10.22).

20. Relational Database: What It Is and Why It's Important [Електронний ресурс]. URL: <https://www.coursera.org/articles/relational-database>. (дата звернення: 29.10.22).

21. Introduction to Relational Database and SQL [Електронний ресурс]. URL: <https://www.googleadservices.com/pagead/aclk>. (дата звернення: 29.10.22).

22. What is MongoDB - Working and Features [Електронний ресурс]. URL: <https://www.guru99.com/what-is-mongodb.html>. (дата звернення: 29.10.22).

23. Навчання з підкріпленням у машинному навчанні [Електронний ресурс]. URL: <https://evergreens.com.ua/ua/articles/reinforcement-learning.html>. (дата звернення: 29.10.22).

24. Understanding Markov Decision Process (MDP) [Електронний ресурс]. URL: <https://towardsdatascience.com/understanding-the-markov-decision-process-mdp-8f838510f150>. (дата звернення: 29.10.22).

25. Моделі і методи проектування інформаційних систем [Електронний ресурс]. URL: [https://elearning.sumdu.edu.ua/free\\_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20160217112601/170352/index.html](https://elearning.sumdu.edu.ua/free_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20160217112601/170352/index.html). (дата звернення: 01.11.22).

26. What is a module in software, hardware and programming? [Електронний ресурс]. URL: <https://www.techtarget.com/whatis/definition/module>. (дата звернення: 01.11.22).

27. Query By Example (QBE) [Електронний ресурс]. URL: <https://www.techopedia.com/definition/5459/query-by-example-qbe>. (дата звернення: 02.11.22).

28. C++ Introduction [Электронный ресурс]. URL: <https://socscan.ru/uk/holodilniki/yazyk-programmirovaniya-c-si-laby-po-informatike-ege-primery-na-yazyke.html>. (дата звернення: 02.11.22).
29. A tour of C# - Overview [Электронный ресурс]. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>. (дата звернення: 05.11.22).
30. What is Python? Executive Summary [Электронный ресурс]. URL: <https://www.python.org/doc/essays/blurb/>. (дата звернення: 05.11.22).
31. System.Collections.Generic Namespace [Электронный ресурс]. URL: <https://learn.microsoft.com/en-us/dotnet/api/system.collections.generic?> (дата звернення: 07.11.22).
32. System.Linq Namespace [Электронный ресурс]. URL: <https://learn.microsoft.com/en-us/dotnet/api/system.linq?view=net-7.0>. (дата звернення: 07.11.22).
33. System.Text Namespace [Электронный ресурс]. URL: [https://sscli.github.io/sscli\\_ref\\_20021101/System.Text/System.Text.html](https://sscli.github.io/sscli_ref_20021101/System.Text/System.Text.html). (дата звернення: 07.11.22).
34. System.Threading.Tasks 4.3.0 [Электронный ресурс]. URL: <https://www.nuget.org/packages/System.Threading.Tasks/>. (дата звернення: 08.11.22).
35. Telegram bot and the methods [Электронный ресурс]. URL: <https://developers.viber.com/docs/api/rest-bot-api/>. (дата звернення: 08.11.22).
36. Viber REST API | Viber Developers Hub [Электронный ресурс]. URL: <https://stackoverflow.com/questions/34087808/telegram-bot-and-the-method-getupdates>. (дата звернення: 10.11.22).
37. FBChat-Sharp: Facebook Messenger client library for C# [Электронный ресурс]. URL: <https://github.com/gave92/fbchat-sharp>. (дата звернення: 10.11.22).
38. Botsify | A Fully Automated Chatbot Platform To Build AI-Chatbot [Электронный ресурс]. URL: <https://botsify.com/>. (дата звернення: 13.11.22).
39. Botsify Create Intelligent Chatbots For Your Website [Электронный ресурс]. URL: <https://www.garyfox.co/startup-resources/botsify/>. (дата звернення: 13.11.22).

40. Dialogflow Documentation | Google Cloud [Електронний ресурс]. URL: <https://cloud.google.com/dialogflow/docs>. (дата звернення: 15.11.22).

41. What is an Expert System? [Електронний ресурс]. URL: <https://expertisefinder.com/expert-systems/> (дата звернення: 16.11.22).

42. КАРКАС: інструментарій для створення бази знань [Електронний ресурс]. URL: <https://www.it-karkas.com.ua>. (дата звернення: 17.11.22).

43. Introduction to the CLIPS Programming Language [Електронний ресурс]. URL: <https://perugini.cps.udayton.edu/> (дата звернення: 18.11.22).

44. Технологічний аудит [Електронний ресурс]. URL: <https://studfile.net/preview/7336428/>. (дата звернення: 25.11.22).

45. Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської) та конструкторсько-технологічної роботи [Електронний ресурс]. URL: <https://web.posibnyky.vntu.edu.ua>. (дата звернення: 25.11.22).

46. Ефективність наукових досліджень [Електронний ресурс]. URL: <https://studfile.net/preview/6872280/page:25/>. (дата звернення: 27.11.22).

## Результат перевірки на плагіат в онлайн-системі UNICHECK



Имя пользователя:  
Озеранський В.С. КН

ID проверки:  
1013269621

Дата проверки:  
11.12.2022 18:09:30 EET

Тип проверки:  
Doc vs Internet + Library

Дата отчета:  
11.12.2022 18:12:48 EET

ID пользователя:  
62038

Название файла: 122МКР-ДідикЯВ2022

Количество страниц: 73 Количество слов: 13510 Количество символов: 105014 Размер файла: 1.02 MB ID файла: 1013028393

### 8.14% Совпадения

Наибольшее совпадение: 3.52% с Интернет-источником (<https://conf.vntu.edu.ua/Index.php/all-fksa/all-fksa-2022/pape..>)

4.49% Источники из Интернета 5 ..... Страница 75

3.92% Источники из Библиотеки 128 ..... Страница 75

### 0% Цитат

Исключение цитат выключено

Исключение списка библиографических ссылок выключено

### 21.7% Исключений

Некоторые источники исключены автоматически (фильтры исключения: количество найденных слов меньш...

2.02% Исключений из Интернета 84 ..... Страница 76

21.3% Исключенного текста из Библиотеки 225 ..... Страница 76

### Модификации

Обнаружены модификации текста. Подробная информация доступна в онлайн-отчете.

Замененные символы 7

## Лістинг програми

```

using System;
using System.IO;
using System.Linq;
using System.Threading.Tasks;
using Telegram.Bot.Args;
using Telegram.Bot.Types;
using Telegram.Bot.Types.Enums;
using Telegram.Bot.Types.InlineQueryResults;
using Telegram.Bot.Types.InputFiles;
using Telegram.Bot.Types.ReplyMarkups;
using ApiAiSDK;
using ApiAiSDK.Model;
namespace Telegram.Bot.Examples.Echo
{
public static class Program
{
private static TelegramBotClient Bot;
private static ApiAi ApiAi;
public static async Task Main()
{
Bot = new TelegramBotClient(""1888512854:AAHK-2a_EHObF44wOzu69PFsgwBbtY4ccRE");
AIConfiguration config = new AIConfiguration(""", SupportedLanguage.Russian);
var me = await Bot.GetMeAsync();
Console.Title = me.Username;
Bot.OnMessage += BotOnMessageReceived;
Bot.OnMessageEdited += BotOnMessageReceived;
Bot.OnCallbackQuery += BotOnCallbackQueryReceived;
Bot.OnInlineQuery += BotOnInlineQueryReceived;
Bot.OnInlineResultChosen += BotOnChosenInlineResultReceived;
Bot.OnReceiveError += BotOnReceiveError;
Bot.StartReceiving(Array.Empty<UpdateType>());
Console.WriteLine(""Start listening for @ {me.Username}");
Console.ReadLine();
Bot.StopReceiving();
}
private static async void BotOnMessageReceived(object sender, MessageEventArgs
messageEventArgs)
{
var message = messageEventArgs.Message;
if (message == null || message.Type != MessageType.Text)
return;

62

switch (message.Text.Split(&#39; &#39;).First())
{
// Send inline keyboard
case &quot;/inline&quot;:

```

```

await SendInlineKeyboard(message);
break;
// send custom keyboard
case "/keyboard"::
await SendReplyKeyboard(message);
break;
// send a photo
case "/photo"::
await SendDocument(message);
break;
// request location or contact
case "/request"::
await RequestContactAndLocation(message);
break;
default:
await Usage(message);
break;
}
// Send inline keyboard
// You can process responses in BotOnCallbackQueryReceived handler
static async Task SendInlineKeyboard(Message message)
{
await Bot.SendChatActionAsync(message.Chat.Id, ChatAction.Typing);
// Simulate longer running task
await Task.Delay(500);
var inlineKeyboard = new InlineKeyboardMarkup(new[]
{
// first row
new []
{
InlineKeyboardButton.WithCallbackData("1.1", "11"),
InlineKeyboardButton.WithCallbackData("1.2", "12"),
},
// second row
new []
{
InlineKeyboardButton.WithCallbackData("2.1", "21"),
InlineKeyboardButton.WithCallbackData("2.2", "22"),
}
});
await Bot.SendTextMessageAsync(
chatId: message.Chat.Id,
text: "Choose",
replyMarkup: inlineKeyboard
);
}
static async Task SendReplyKeyboard(Message message)
{
var replyKeyboardMarkup = new ReplyKeyboardMarkup(
new KeyboardButton[][]
{
new KeyboardButton[] { "1.1", "1.2" },
new KeyboardButton[] { "2.1", "2.2" },
}
);
}
}

```

63

```

    },
    resizeKeyboard: true
  );
  await Bot.SendTextMessageAsync(
    chatId: message.Chat.Id,
    text: "Choose",
    replyMarkup: replyKeyboardMarkup
  );
}
static async Task SendDocument(Message message)
{
  await Bot.SendChatActionAsync(message.Chat.Id, ChatAction.UploadPhoto);
  const string filePath = @"Files\tux.png";
  using var fileStream = new FileStream(filePath, FileMode.Open,
    FileAccess.Read, FileShare.Read);
  var fileName = filePath.Split(Path.DirectorySeparatorChar).Last();
  await Bot.SendPhotoAsync(
    chatId: message.Chat.Id,
    photo: new InputOnlineFile(fileStream, fileName),
    caption: "Nice Picture"
  );
}
static async Task RequestContactAndLocation(Message message)
{
  var RequestReplyKeyboard = new ReplyKeyboardMarkup(new[]
  {
    KeyboardButton.WithRequestLocation("Location"),
    KeyboardButton.WithRequestContact("Contact"),
  });
  await Bot.SendTextMessageAsync(
    chatId: message.Chat.Id,
    text: "Who or Where are you?",
    replyMarkup: RequestReplyKeyboard
  );
}
static async Task Usage(Message message)
{
  const string usage = "Usage:\n" +
    "inline - send inline keyboard\n" +
    "keyboard - send custom keyboard\n" +
    "photo - send a photo\n" +
    "request - request location or contact";
  await Bot.SendTextMessageAsync(
    chatId: message.Chat.Id,
    text: usage,
    replyMarkup: new ReplyKeyboardRemove()
  );
}
// Process Inline Keyboard callback data
private static async void BotOnCallbackQueryReceived(object sender,
  CallbackQueryEventArgs callbackQueryEventArgs)

```



```

{
var callbackQuery = callbackQueryEventArgs.CallbackQuery;
await Bot.AnswerCallbackQueryAsync(
callbackQueryId: callbackQuery.Id,
text: $"Received {callbackQuery.Data}"
);

64

await Bot.SendTextMessageAsync(
chatId: callbackQuery.Message.Chat.Id,
text: $"Received {callbackQuery.Data}"
);
}
#region Inline Mode
private static async void BotOnInlineQueryReceived(object sender,
InlineQueryEventArgs inlineQueryEventArgs)
{
Console.WriteLine($"Received inline query from:
{inlineQueryEventArgs.InlineQuery.From.Id}");
InlineQueryResultBase[] results = {
// displayed result
new InlineQueryResultArticle(
id: "3",
title: "TgBots",
inputMessageContent: new InputTextMessageContent(
"hello")
)
);
};
await Bot.AnswerInlineQueryAsync(
inlineQueryId: inlineQueryEventArgs.InlineQuery.Id,
results: results,
isPersonal: true,
cacheTime: 0
);
}
private static void BotOnChosenInlineResultReceived(object sender,
ChosenInlineResultEventArgs chosenInlineResultEventArgs)
{
Console.WriteLine($"Received inline result:
{chosenInlineResultEventArgs.ChosenInlineResult.ResultId}");
}
#endregion
private static void BotOnReceiveError(object sender, ReceiveErrorEventArgs
receiveErrorEventArgs)
{
Console.WriteLine($"Received error: {0} — {1}",
receiveErrorEventArgs.ApiRequestException.ErrorCode,
receiveErrorEventArgs.ApiRequestException.Message
);
}
}
}
}

```

```

/ speech api component/
using
System;

using System.IO;
using Vosk;

public class VoskDemo
{
public static void DemoBytes(Model model)
{
// Demo byte buffer
VoskRecognizer rec = new VoskRecognizer(model, 16000.0f);
rec.SetMaxAlternatives(0);
rec.SetWords(true);
using(Stream source = File.OpenRead(&quot;test.wav&quot;)) {
byte[] buffer = new byte[4096];
int bytesRead;
while((bytesRead = source.Read(buffer, 0, buffer.Length)) &gt; 0) {
if (rec.AcceptWaveform(buffer, bytesRead)) {
Console.WriteLine(rec.Result());
} else {
Console.WriteLine(rec.PartialResult());
}
}
}
Console.WriteLine(rec.FinalResult());
}

public static void DemoFloats(Model model)
{
// Demo float array
VoskRecognizer rec = new VoskRecognizer(model, 16000.0f);
using(Stream source = File.OpenRead(&quot;test.wav&quot;)) {
byte[] buffer = new byte[4096];
int bytesRead;
while((bytesRead = source.Read(buffer, 0, buffer.Length)) &gt; 0) {
float[] fbuffer = new float[bytesRead / 2];
for (int i = 0, n = 0; i &lt; fbuffer.Length; i++, n+=2) {
fbuffer[i] = BitConverter.ToInt16(buffer, n);
}
if (rec.AcceptWaveform(fbuffer, fbuffer.Length)) {
Console.WriteLine(rec.Result());
} else {

2

Console.WriteLine(rec.PartialResult());
}
}
}
Console.WriteLine(rec.FinalResult());
}

```

```

public static void DemoSpeaker(Model model)
{
// Output speakers
SpkModel spkModel = new SpkModel(&quot;model-spk&quot;);
VoskRecognizer rec = new VoskRecognizer(model, 16000.0f);
rec.SetSpkModel(spkModel);

using(Stream source = File.OpenRead(&quot;test.wav&quot;)) {
byte[] buffer = new byte[4096];
int bytesRead;
while((bytesRead = source.Read(buffer, 0, buffer.Length)) &gt; 0) {
if (rec.AcceptWaveform(buffer, bytesRead)) {
Console.WriteLine(rec.Result());
} else {
Console.WriteLine(rec.PartialResult());
}
}
Console.WriteLine(rec.FinalResult());
}

public static void Main()
{
Vosk.Vosk.SetLogLevel(0);

Model model = new Model(&quot;model&quot;);
DemoBytes(model);
DemoFloats(model);
DemoSpeaker(model);
}

/ Multilingual component/
public Task StartAsync(IDialogContext context)
{
context.PostAsync($&quot;Hi!&quot;);
PromptDialog.Choice(context,
HandleLang,
new[] { &quot;English&quot;, &quot;Ukrainian&quot;, &quot;Russian&quot;, &quot;Chinese&quot;;&quot;,
&quot;German&quot;},

3

&quot;Please select a language:&quot;,
&quot;I didn't get that&quot;);
return Task.CompletedTask;
}
public async Task HandleLang(IDialogContext context, IAwaitable<string> result)
{
string lang = await result as string;
switch (lang)
{
case &quot;English&quot;:
Strings.Culture = new CultureInfo(&quot;en-US&quot;);

```

```

break;
case "Ukrainian ":
Strings.Culture = new CultureInfo("es-UA");
break;
case "Russian":

Strings.Culture = new CultureInfo("es-RU");
break;
case "Chinese":

Strings.Culture = new CultureInfo("es-CH");
break;
case "German":

Strings.Culture = new CultureInfo("es-DE");
break;
default:
break;
}
... do whatever here
}
/ roles component/
public class AccountController : Controller
{
private readonly IConfiguration _config;
private readonly SignInManager<ApplicationUser> _signInManager;
private readonly UserManager<ApplicationUser> _userManager;
public AccountController(IConfiguration config,
SignInManager<ApplicationUser> signInManager,
UserManager<ApplicationUser> userManager)
{
_config = config;
_signInManager = signInManager;
_userManager = userManager;
}
[HttpGet]
public async Task<IActionResult> Logout()
{
await _signInManager.SignOutAsync();
return RedirectToAction("index", "home");
}
[HttpGet]
public async Task<IActionResult> Login()
{
await Task.CompletedTask;
var botname = _config["BOT_NAME"];
var fqdn = _config["FQDN"];
ViewBag.WidgetEmbedCode = WidgetEmbedCodeGenerator.GenerateRedirectEmbedCode(
_config["BOT_NAME"],
$"https://{fqdn}{Url.Action("tcallback", "account")}",
ButtonStyle.Large,
true,
true);
return View();
}

```

4

```

}
[HttpGet]
public async Task<IActionResult> Tgcallback(
string id,
string first_name,
string username,
string photo_url,
string auth_date,
string hash)
{
var token = _config[&quot;BOT_TOKEN&quot;];
var loginWidget = new LoginWidget(token);
var auth = loginWidget.CheckAuthorization(new SortedDictionary<string, string>()
{
{&quot;id&quot;,id},
{&quot;first_name&quot;, first_name},
{&quot;username&quot;, username},
{&quot;photo_url&quot;, photo_url},
{&quot;auth_date&quot;, auth_date},
{&quot;hash&quot;, hash}
});
if (auth == Authorization.Valid)
{
var user = await _userManager.FindByNameAsync(&quot;tg{id}&quot;);
if (null == user)
{
user = new ApplicationUser()
{
UserName = &quot;tg{id}&quot;,
TelegramNativeId = long.Parse(id),
TelegramUserName = username,
FirstName = first_name,
PhotoUrl = photo_url
};
var result = await _userManager.CreateAsync(user);
if (!result.Succeeded)
{
ViewBag.ErrorTitle = &quot;Internal error&quot;;
ViewBag.ErrorMessage = &quot;Failed to create user tg{id}&quot;;
return View(&quot;Error&quot;);
}
user = await _userManager.FindByNameAsync(&quot;tg{id}&quot;);
if (null == user)
{
ViewBag.ErrorTitle = &quot;Internal error&quot;;
ViewBag.ErrorMessage = &quot;Failed to create user tg{id}&quot;;
return View(&quot;Error&quot;);
}
}
await _signInManager.SignInAsync(user, true);
}
}

```

```
return RedirectToAction("&quot;index&quot;, &quot;home&quot;);
}
}
```

```
/ calendar events component/
using Google.Apis.Auth.OAuth2;
using Google.Apis.Calendar.v3;
using Google.Apis.Calendar.v3.Data;
using Google.Apis.Services;
using Google.Apis.Util.Store;
using System;
using System.Collections.Generic;
```

5

```
using System.IO;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using Telegram.Bot;
namespace CalendarBot
{
class Program
{
private static readonly TelegramBotClient bot = new TelegramBotClient("&quot;Enter code here&quot;);
static string[] Scopes = { CalendarService.Scope.CalendarReadOnly };
static string ApplicationName = &quot;Google Calendar API .NET Quickstart&quot;;
static UserCredential credential;
static Events events;
static CalendarService service;
static void Main(string[] args)
{
using (var stream =
new FileStream("&quot;client_secret.json&quot;, FileMode.Open, FileAccess.Read))
{
string credPath = System.Environment.GetFolderPath(
System.Environment.SpecialFolder.Personal);
credPath = Path.Combine(credPath, &quot;.credentials/calendar-dotnet-quickstart.json&quot;);
credential = GoogleWebAuthorizationBroker.AuthorizeAsync(
GoogleClientSecrets.Load(stream).Secrets,
Scopes,
&quot;user&quot;,
Cancellation.Token.None,
new FileDataStore(credPath, true)).Result;
}
bot.OnMessage += bot_OnMessage;
bot.StartReceiving();
Console.ReadLine();
bot.StopReceiving();
}
private static void get_calendar()
{
service = new CalendarService(new BaseClientService.Initializer()
{
```

```

HttpClientInitializer = credential,
ApplicationName = ApplicationName,
});
}
private static void bot_OnMessage(object sender, Telegram.Bot.Args.MessageEventArgs e)
{
string text = e.Message.Text;
string report = null;
Boolean checkIfNothing = true;
Console.WriteLine(text);
if (e.Message.Type == Telegram.Bot.Types.Enums.MessageType.TextMessage &&
e.Message.Text.Equals(&quot;/getCal&quot;))
{

get_calendar();
EventsResource.ListRequest request = service.Events.List(&quot;primary&quot;);
request.TimeMin = DateTime.Now;
request.TimeMax = new DateTime(DateTime.Now.Year, DateTime.Now.Month,
DateTime.Now.Day, 23, 59, 59);

request.ShowDeleted = false;
request.SingleEvents = true;
request.MaxResults = 10;
request.OrderBy = EventsResource.ListRequest.OrderByEnum.StartTime;
events = request.Execute();

if (events.Items != null && events.Items.Count > 0)

6

{
checkIfNothing = false;
report += &quot;Upcoming events for today:&quot;;
for (int i = 0; i &lt;= events.Items.Count() - 1; i++)
{
var eventItem = events.Items[i];
string when = eventItem.Start.DateTime.ToString();
if (String.IsNullOrEmpty(when))
{
when = eventItem.Start.Date;
}
//bot.SendMessageAsync(e.Message.Chat.Id, eventItem.Summary + &quot; &quot; + when +
&quot; &quot; + eventItem.Location);
report += &quot;\r\n&quot; + eventItem.Summary + &quot; &quot; + when + &quot; &quot; + eventItem.Location;
}
}
//else
//bot.SendMessageAsync(e.Message.Chat.Id, &quot;No upcoming events found.&quot;);
if (checkIfNothing)
bot.SendMessageAsync(e.Message.Chat.Id, &quot;No upcoming events found.&quot;);
else
bot.SendMessageAsync(e.Message.Chat.Id, report);
}else{

```


bot.SendMessageAsync(e.Message.Chat.Id, &quot;Please enter the correct command.&quot;);

Додаток В (обов'язковий)


110

ІЛЮСТРАТИВНА ЧАСТИНА  
«ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АВТОМАТИЗАЦІЇ  
ОНЛАЙН-НАВЧАННЯ»

Виконала: студентка 2-го курсу, групи  
1КН-21м  
спеціальності 122 – «Комп'ютерні  
науки»

  
Дідик Я. В.  
(прізвище та ініціали)

Керівник: д.т.н., доц. каф. КН

  
Крилик Л. В.  
(прізвище та ініціали)  
« 15 » 12 2022 р.



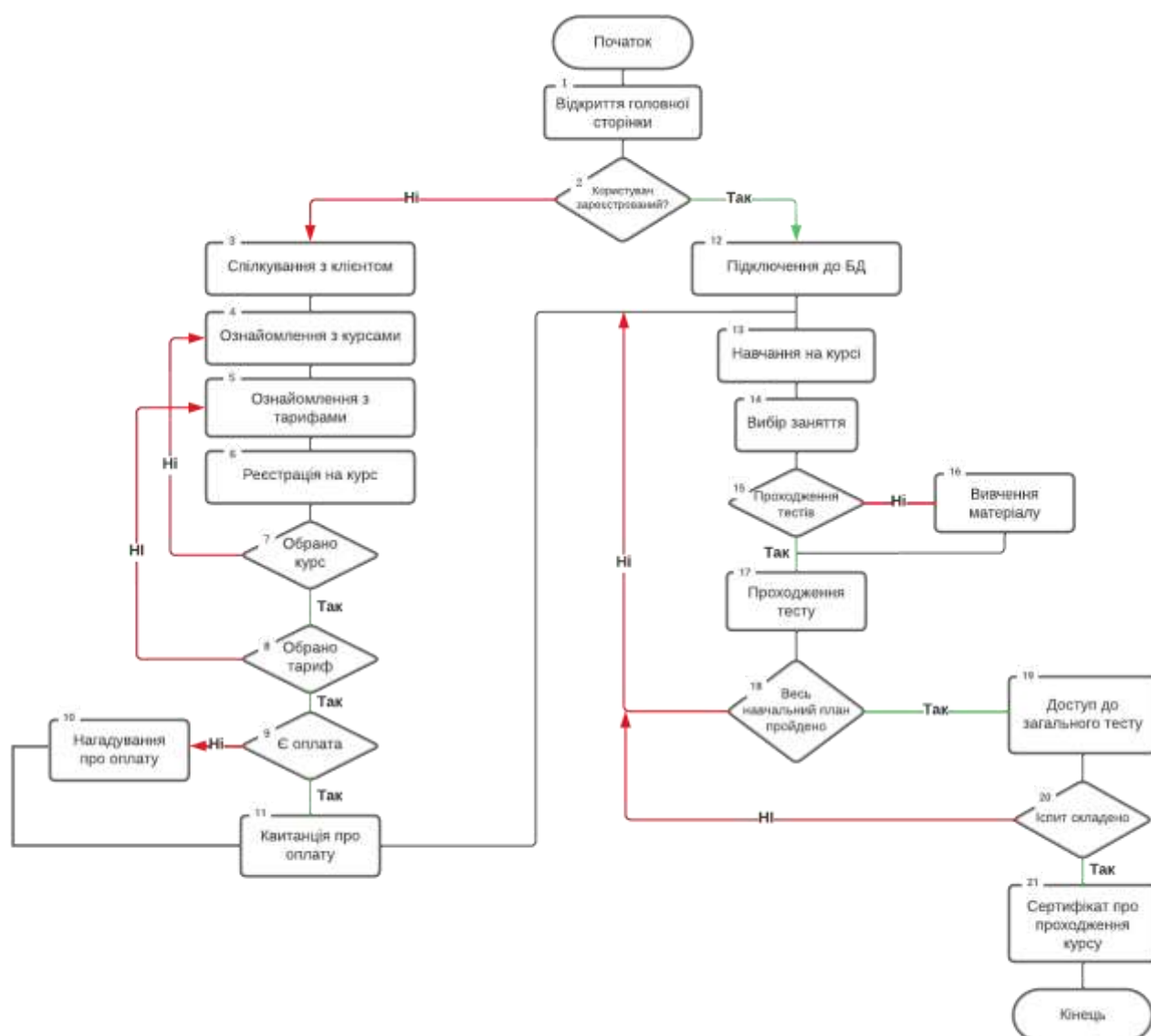


Рисунок В.1 – Схема алгоритму функціонування клієнтського модуля інформаційної технології автоматизації онлайн-навчання

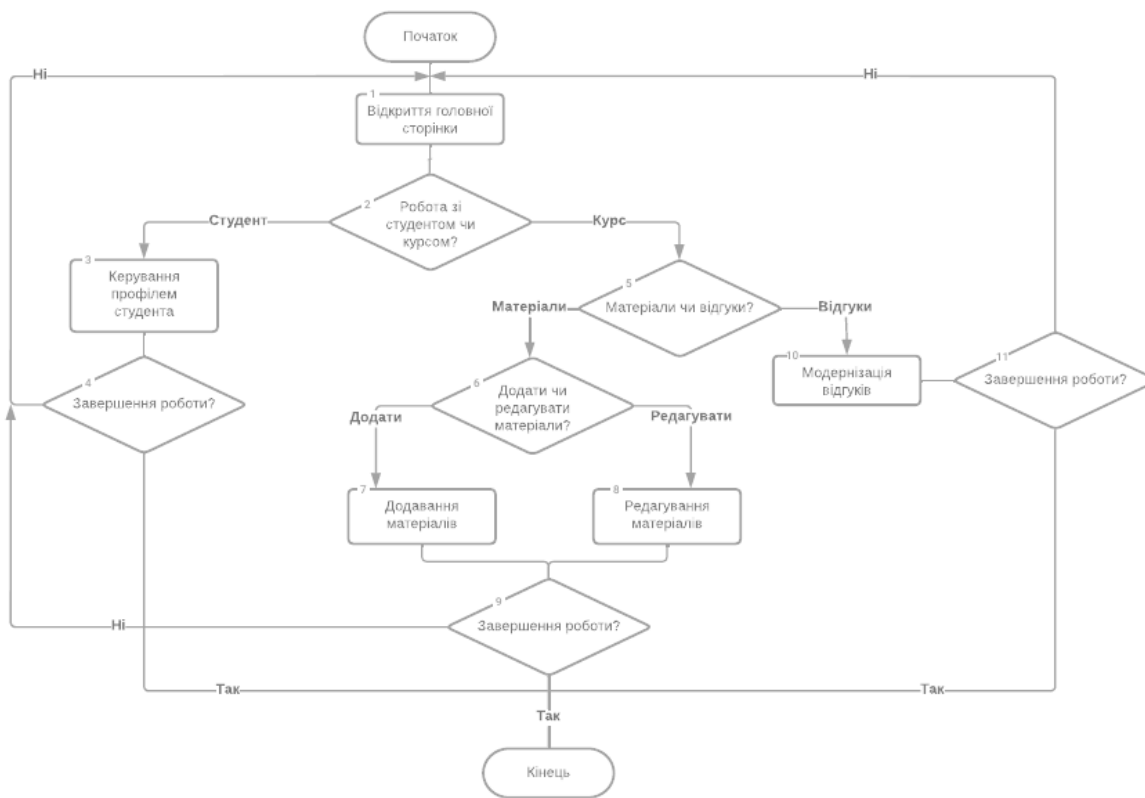


Рисунок В.2 – Схема алгоритму функціонування модуля адміністратора інформаційної технології автоматизації онлайн-навчання

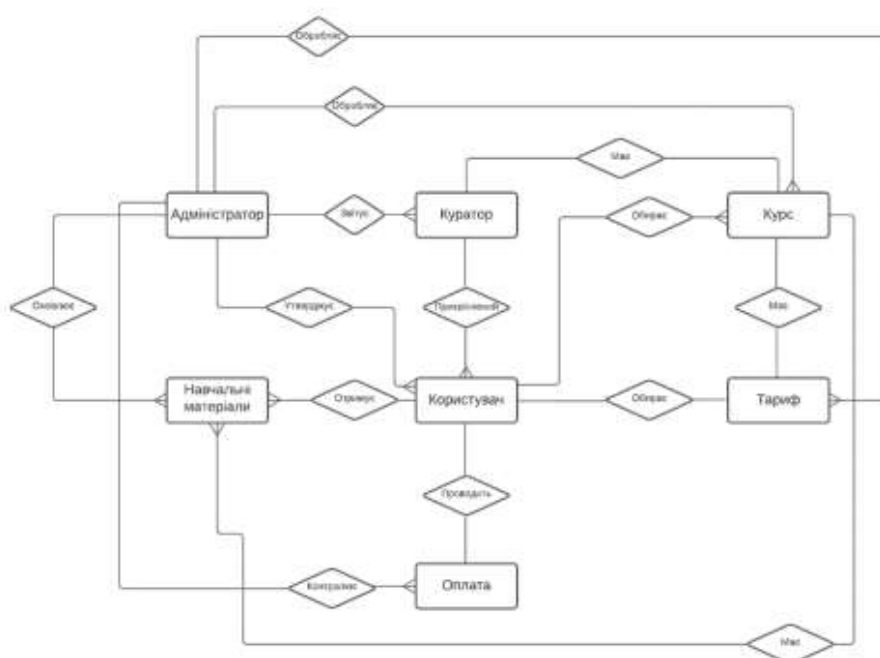


Рисунок В.3 – ER-діаграма автоматизованого онлайн-навчання

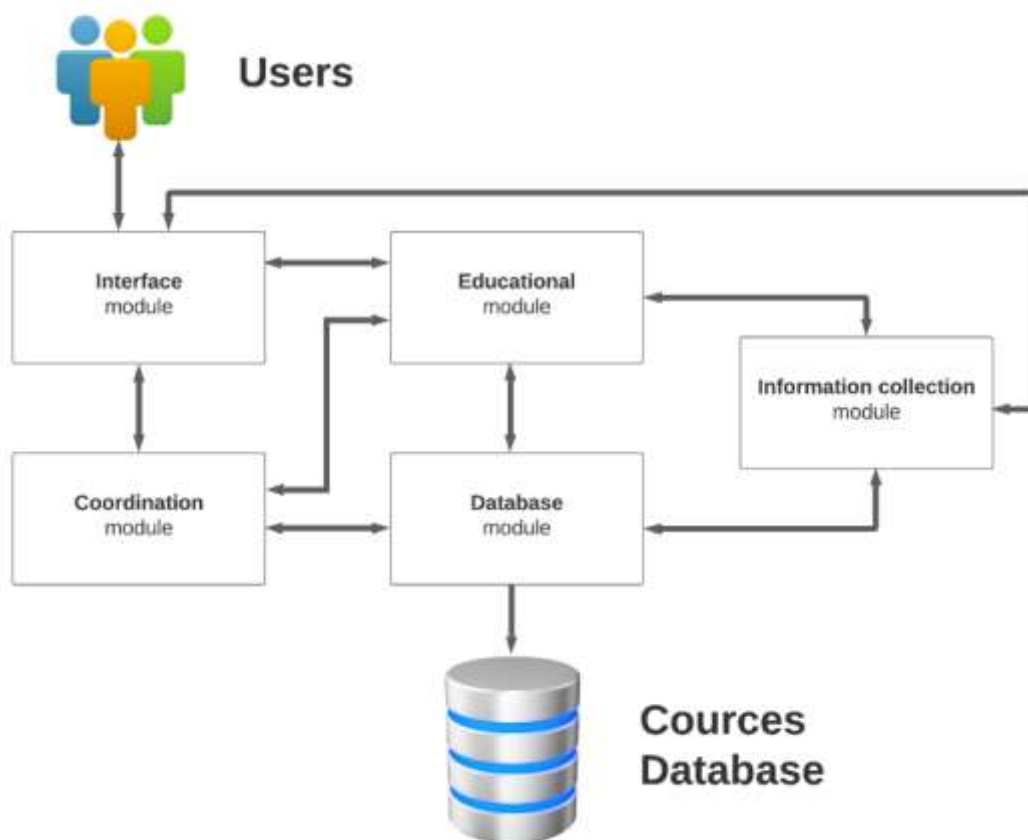


Рисунок В.4 – Загальна структурна схема функціонування системи



Рисунок В.5 – Діаграма діяльності сервера для проведення реєстрації



Рисунок В.6 – Діаграма діяльності сервера для вибору онлайн-курсу



Рисунок В.7 – Діаграма діяльності сервера для роботи адміністратора з

навчальними матеріалами

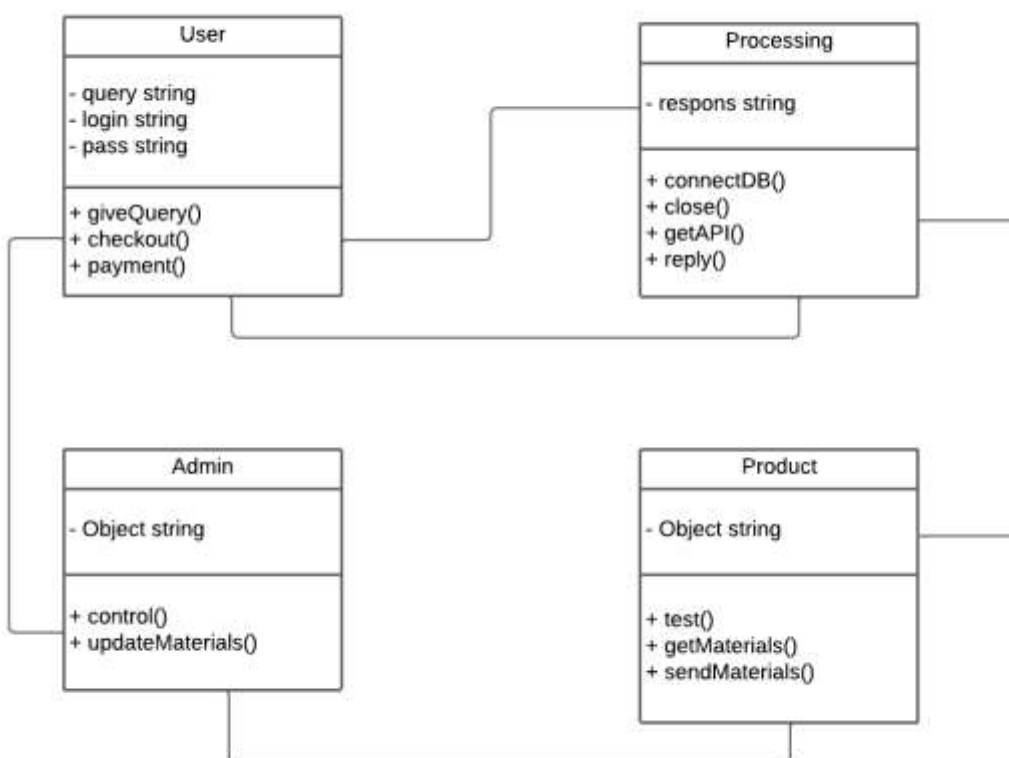


Рисунок В.8 – Загальна UML-діаграма класів

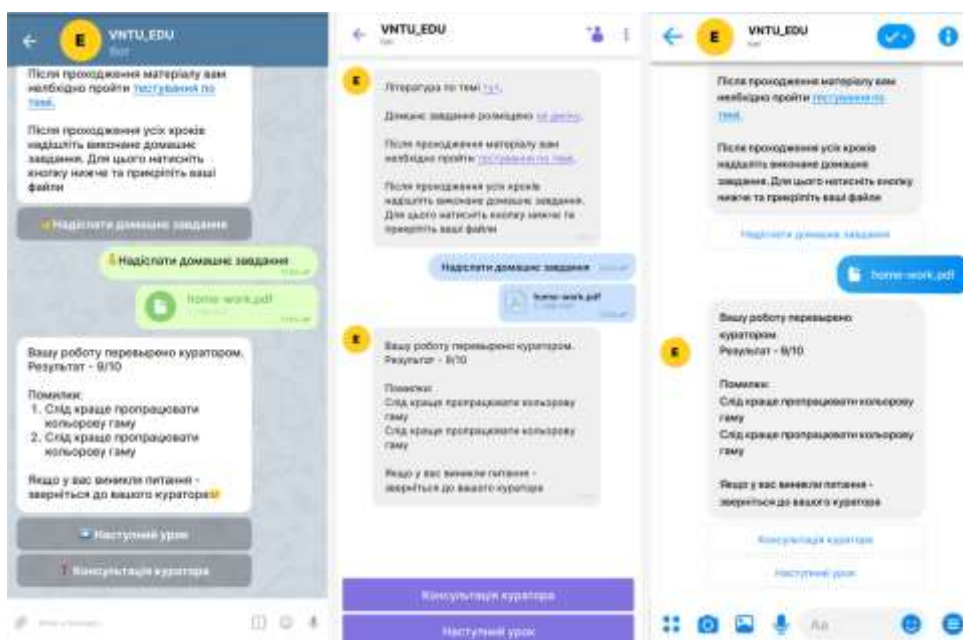


Рисунок В.9 – Загальний вигляд інтерфейсного вікна роботи програми

## Додаток Г (довідниковий)

### Інструкція користувача

Першим кроком є запуск програмного модуля на зручній для користувача платформі. Після запуску відкривається вікно початкової активності в месенджері. Вікно початкової активності містить функцію /start для виклику бота.

Чат-бот знайомить користувача з системою, описує свої можливості та можливості користувача. Після знайомства користувач має можливість ознайомитися з онлайн-курсами, які надає навчальна організація та тарифами на навчання. Також користувач має можливість пройти опитування, якщо він не визначився точно яка спеціалізація йому підходить найбільше. Також користувачеві надається можливість пройти тестування на визначення існуючого рівня знань для компетентнішого підбору навчального курсу. Перед користувачем розміщені такі кнопки, як «Курс 1», «Курс 2», «Курс 3», «Курс 4», «Курс 5», «Тарифи», «Відгуки», «Тестування», «Підбір курсу» (рис. Г.1).

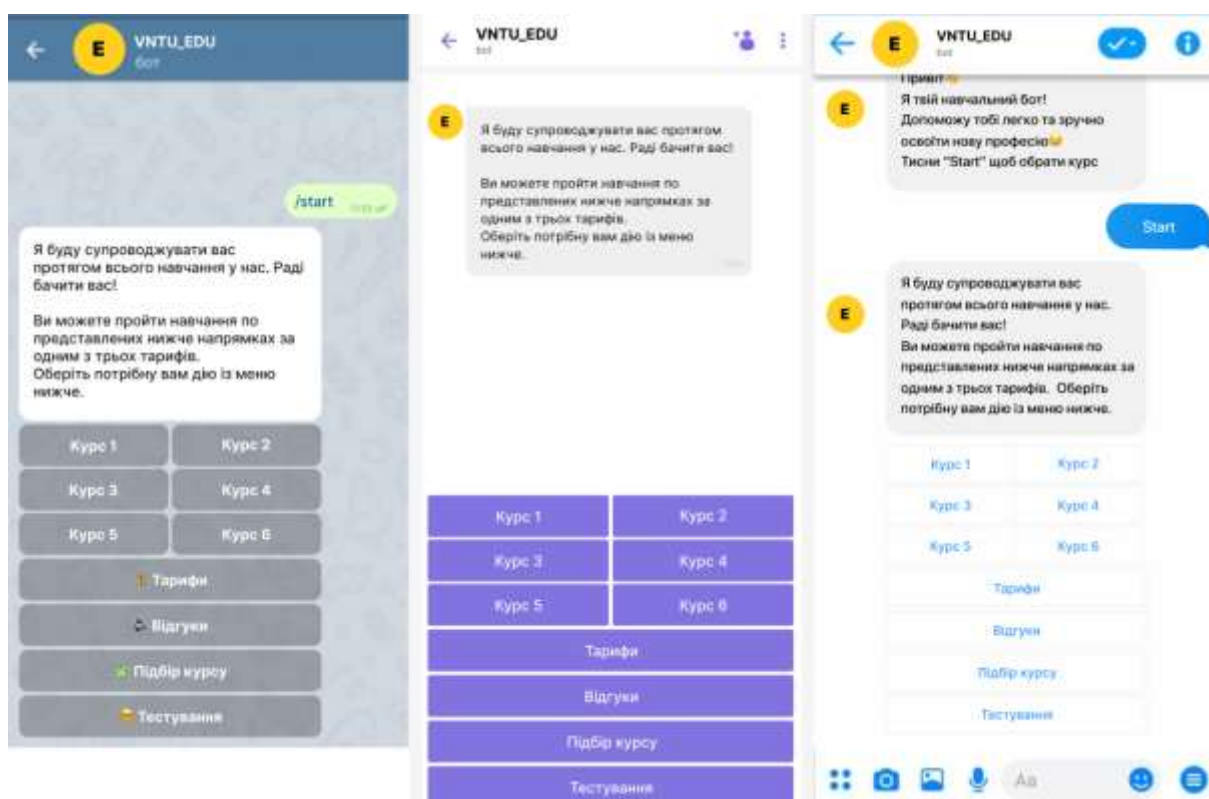


Рисунок Г.1 – Загальний вигляд інтерфейсного вікна знайомства з користувачем у месенджерах Telegram, Viber, Facebook Messenger

Коли користувачем обрано навчальний курс, він має натиснути відповідну кнопку з назвою курсу, після чого з'явиться необхідна ознайомлювальна інформація про навчання. Користувач має змогу ознайомитися з цією інформацією, повернутися до наявних курсів, переглянути тарифи і безпосередньо зареєструватися на курс. Відповідно, перед користувачем розміщено кнопку «Реєстрація», «До курсів», «Тарифи» (рис. Г.2).

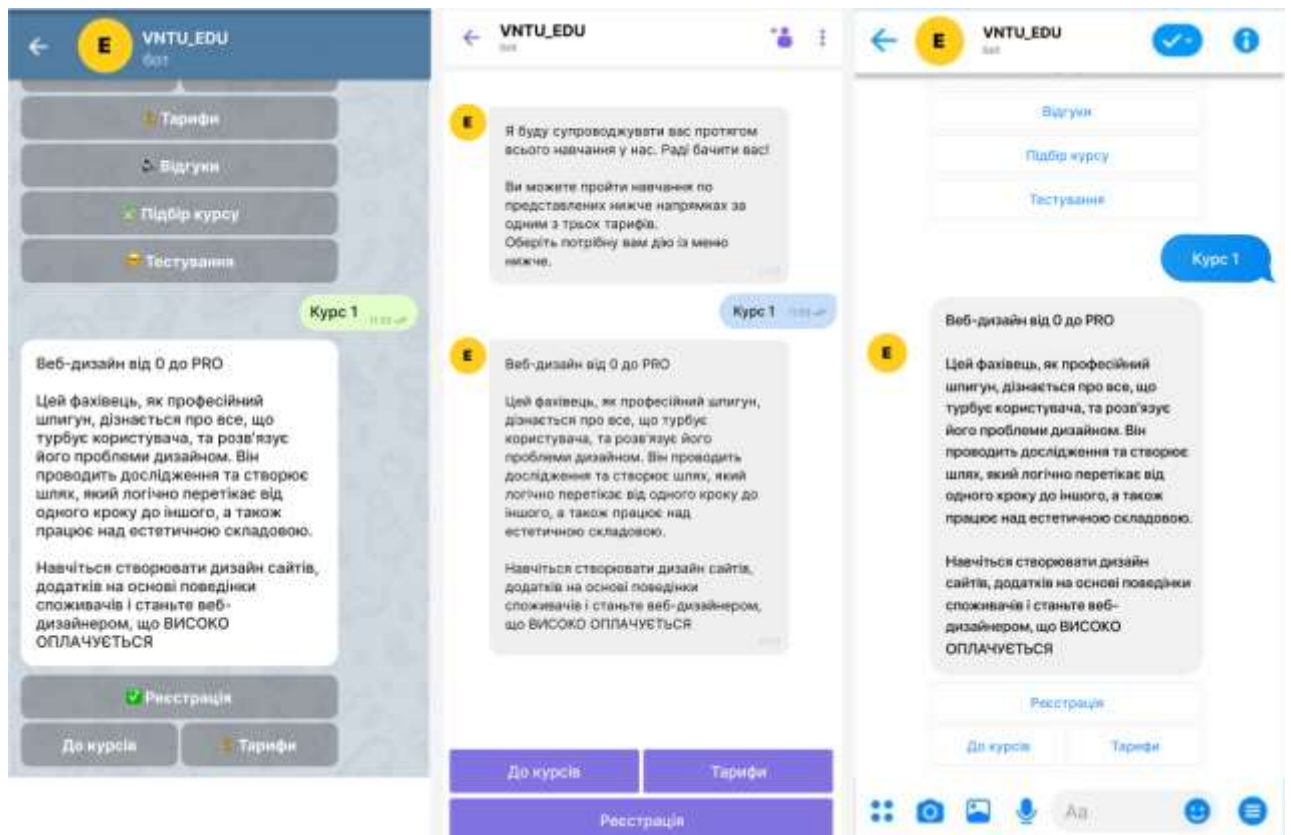


Рисунок Г.2 – Загальний вигляд інтерфейсного вікна ознайомлення з курсом у месенджерах Telegram, Viber, Facebook Messenger

Коли користувач запускає функцію реєстрації на курс, чат-бот починає проведення своєї спільної бесіди, ставлячи користувачеві запитання, необхідні для заповнення реєстраційних даних, які потім заносяться у базу даних навчальної організації. Це такі дані, як «Прізвище», «Ім'я», «Електронна пошта», «Обраний тариф навчання».

На рис. Г.3 показано процес реєстрації.

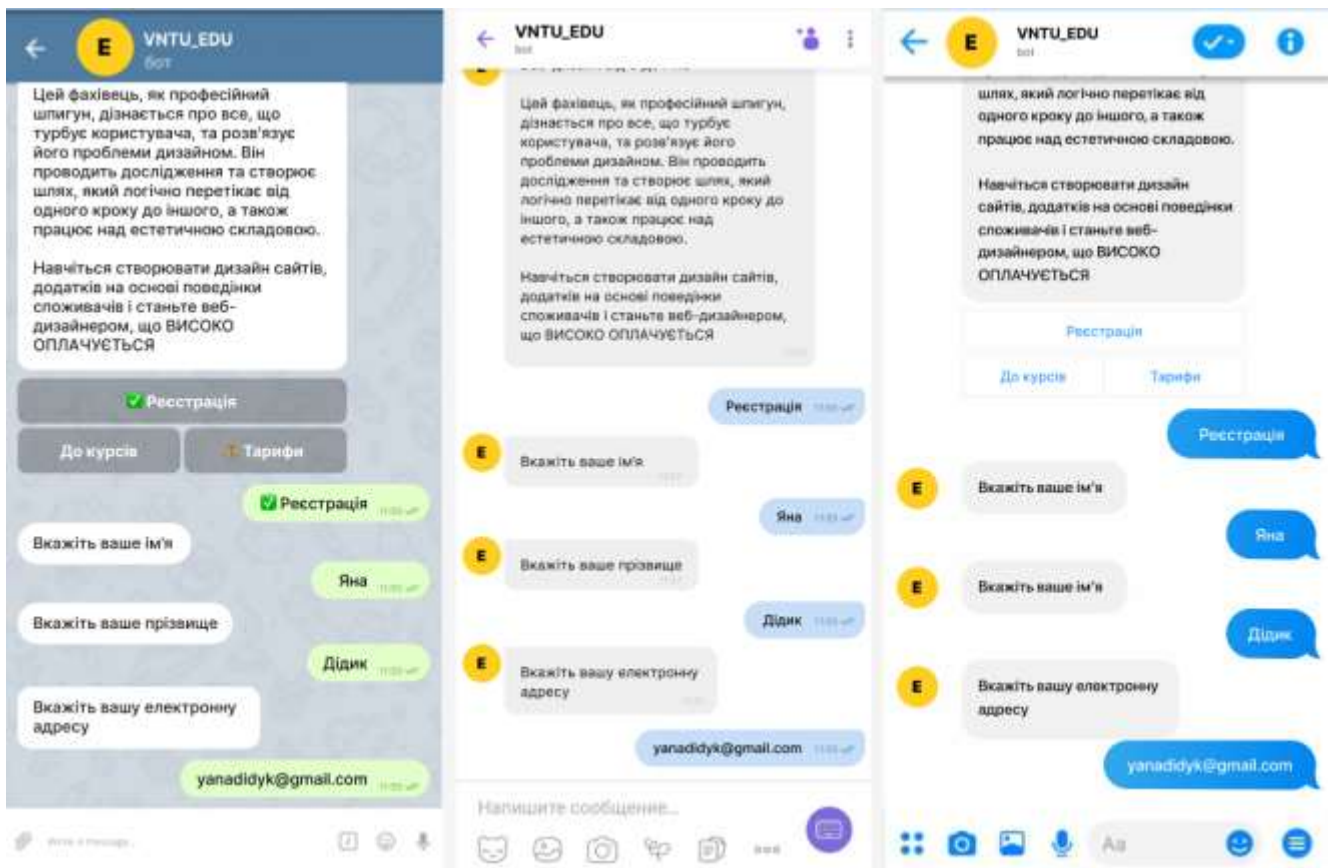


Рисунок Г.3 – Загальний вигляд інтерфейсного вікна реєстрації на курс у месенджерах Telegram, Viber, Facebook Messenger

Коли процес реєстрації завершено, користувач отримує на вказану ним раніше електронну скриньку форму для оплати згідно з обраним тарифом. Користувач має оплатити навчання та надіслати документ про оплату чат-боту. Той у свою чергу відправляє квитанцію адміністратору для перевірки.

Після вдалої оплати користувач отримує доступ до своїх навчальних матеріалів.

На рис. Г.4 показано приклад факту відправки користувачем документу про оплату за навчання.



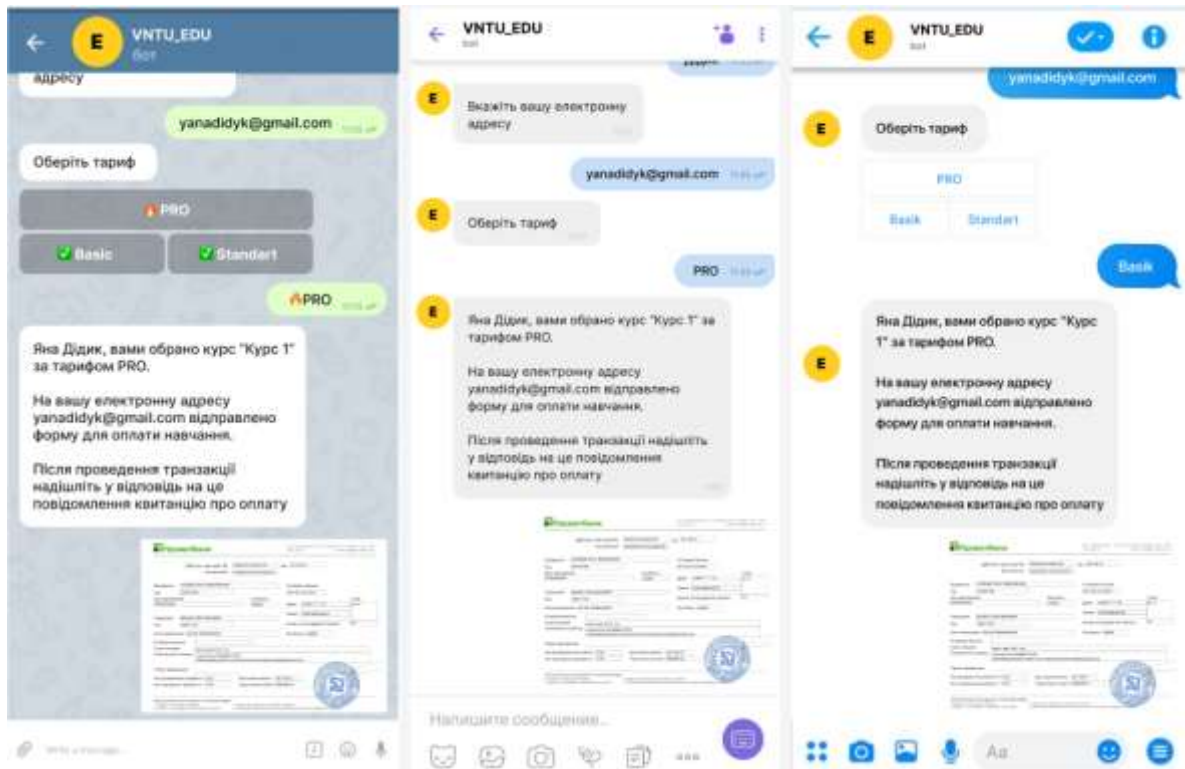


Рисунок Г.4 – Загальний вигляд інтерфейсного вікна підтвердження оплати навчання у месенджерах Telegram, Viber, Facebook Messenger

Користувач поступово отримує та освоює матеріали по обраному курсу. Туди входять відео-лекції, текстові файли з літературою, текстові файл з домашнім завданням та посилання на тести. По проходженні матеріалу теми користувач надсилає виконане домашнє завдання чат-боту, натиснувши відповідно кнопку «Надіслати домашнє завдання», яке потім перенаправляє куратору для перевірки. З часом бот повідомляє результати та надсилає перелік помилок, на які варто звернути увагу. Якщо у користувача виникають питання по зауваженнях, він може отримати консультацію куратора, відповідно натиснувши кнопку «Консультація куратора». Якщо питання відсутню – користувач переходить на наступну тему (рис. Г.5).

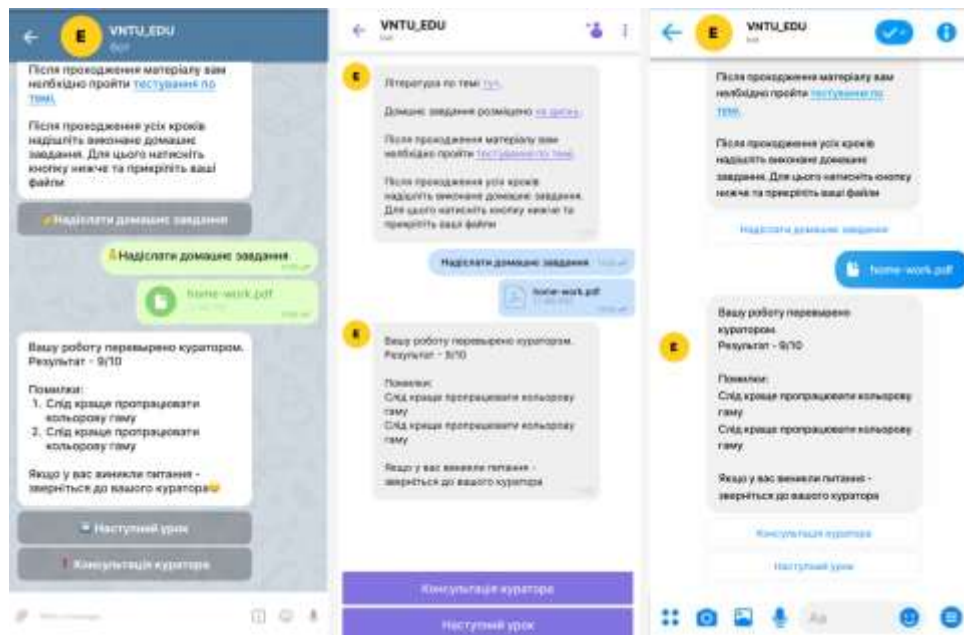


Рисунок Г.5 – Загальний вигляд інтерфейсного вікна процесу навчання у месенджерах Telegram, Viber, Facebook Messenger

Після успішного проходження усієї навчальної програми, чат-бот перенаправляє користувача до екзаменаційного тестування. Для цього потрібно натиснути кнопку «Екзаменаційне тестування». У разі вдалого проходження тестування, користувач отримує сертифікат про проходження курсу та має можливість залишити відгук про навчання (рис. Г.6).

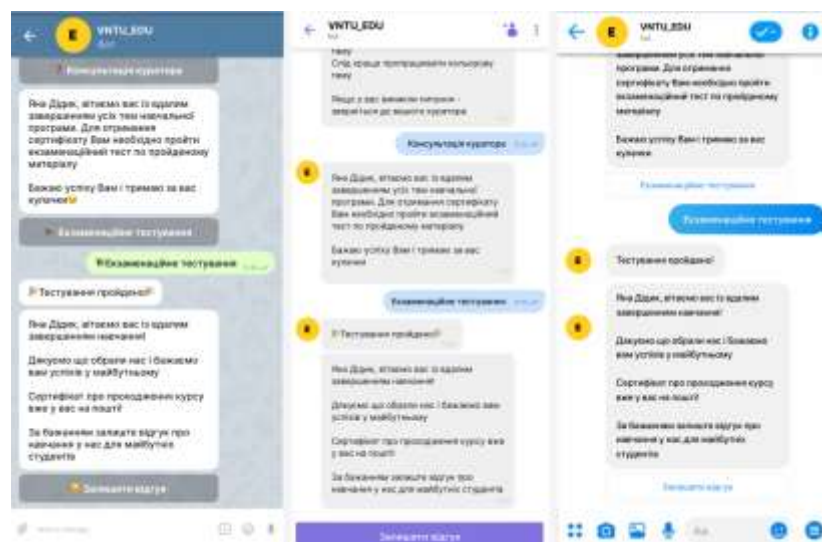



Рисунок Г.6 – Завершення навчання

## Довідка про участь в конкурсі студентських наукових робіт

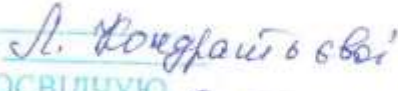

### Д О В І Д К А

Надана ДІДИК Яні Вікторівні, ст. гр. ІКН-21м в тому, що вона брала участь у Всеукраїнському конкурсі студентських наукових робіт в 2021/2022 н.р.

За підсумками І-го туру наукова робота «Програмний модуль інтелектуального чат-бота для онлайн-курсів» була рекомендована для участі у ІІ –му турі конкурсу і направлена до Центральноукраїнського НТУ у м. Кропивницький, який не проводився в зв'язку з введенням в країні воєнного стану (Наказ МОН України № 508 від 31 травня 2022).

Секретар оргкомітету  Людмила КОНДРАТЬЄВА



Підпис   
ПОСВІДЧУЮ  
Зак. канцелярії 

## Довідка про впровадження розробки

ТОВ "ATLAS.UA"  
Україна, м. Київ

1 грудня 2022 року

### Довідка про впровадження

Довідка дана студентці Вінницького національного технічного університету групи ІКН-21м Дідик Яні Вікторівні в тому, що результати магістерської кваліфікаційної роботи «Інформаційна технологія автоматизації онлайн-навчання» пройшли експериментальне випробування, відповідають замовленню, мають практичне значення і плануються до впровадження та подальшого використання.

Директор



Пашенко О.П.