

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДІВ
ФОРМУВАННЯ АВТОМАТИЧНИХ ВІДПОВІДЕЙ НА ЗАПИТАННЯ
RESEARCH AND SOFTWARE IMPLEMENTATION OF METHODS FOR
AUTOMATIC GENERATION OF RESPONSES TO QUESTIONS

Виконав: студент(ка) 2 курсу, групи КНТ-129м
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Батяєв А.А.

(прізвище та ініціали)

Керівник Зайко Т.А.

(прізвище та ініціали)

Рецензент Скрупський С.Ю.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти магістр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 2020 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Батяєва Артура Алановича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація методів
формування автоматичних відповідей на запитання
Research and Software Implementation of Methods for Automatic Generation of
Responses to Questions

керівник проєкту (роботи) Зайко Тетяна Анатоліївна, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “3” листопада 2020 року № 301

2. Строк подання студентом проєкту (роботи) грудень 2020 року
3. Вихідні дані до проєкту (роботи) рекомендована література, технічне
завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження.
2. Матеріали і методи. 3. Опис програми. 4. Розробка програми. 5. Експлуатація
та тестування програми. 6. Економіко-організаційна частина, 7. Охорона праці
та безпека в надзвичайних ситуаціях.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-5 Основна частина	Зайко Т.А., доцент		
Економіко-організаційна частина	Пожуєва Т.О., професор		
Охорона праці та безпека в надзвичайних ситуаціях	Коробко О.В., ст. викладач		
Нормоконтролер	Дейнега Л.Ю., ст. викладач		

7. Дата видачі завдання “1” вересня 2020 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2–3 тижні	Розділ 1
3	Розробка методу.	4–5 тижні	Розділ 2
4	Розробка архітектури програми.	6 тиждень	Розділ 3
5	Розробка програми.	7–8 тижні	Розділ 4
6	Тестування та відлагодження програмного забезпечення.	9 тиждень	Розділ 5
7	Розробка організаційно-економічної частини.	10 тиждень	Розділ 6
8	Розробка заходів з охорони праці та цивільної безпеки.	11 тиждень	Розділ 7
9	Оформлення пояснювальної записки та документів до неї.	12-13 тиждень	Додатки
10	Нормоконтроль та рецензування.	14–15 тижні	
11	Захист роботи.	16 тиждень	

Студент(ка)

(підпис) Батяєв А.А.
(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис) Зайко Т.А.
(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
107 с., 23 рис., 7 табл., 3 дод., 36 джерел.

АВТОМАТИЧНА КОМУНІКАЦІЯ, ЗАДАЧА КЛАСИФІКАЦІЇ,
МЕТОД ОПОРНИХ ВЕКТОРІВ, РНР.

Об'єкт дослідження – процес формування автоматичних відповідей на запитання. Предмет дослідження – методи формування автоматичних відповідей на запитання.

Мета роботи – розроблення програмного забезпечення формування автоматичних відповідей на запитання.

Матеріали, методи та технічні засоби: мова програмування РНР, система керування базами даних MySQL, метод опорних векторів.

Результати. Розроблено вебзастосунок, який дозволяє виконувати формування автоматичних відповідей на запитання користувача в галузі продажу книжок, реалізовувати керування даним процесом, заповнювати даними для навчання модель, організовувати підтримку менеджера для користувача за необхідності.

Висновки. В роботі запропоновано метод формування автоматичних відповідей на запитання, що полягає в початковій класифікації запитань за категоріями, подальшій уточнюючій класифікації за конкретними схемами відповідей на запитання на основі використання методу опорних векторів, кодуванні запитання на основі словника предметної області та формуванні відповіді на основі виконання запиту до бази даних, що дозволяє відповідати на запитання за предметною областю без участі менеджерів компанії.

Галузь використання – автоматизовані помічники, консультанти на вебсайтах з продажу товарів.

ABSTRACT

Explanatory note to the diploma qualifying work of the master: 107 pages, 23 figures, 7 tables, 3 appendixes, 36 sources.

AUTOMATIC COMMUNICATION, CLASSIFICATION PROBLEM, SUPPORT VECTOR MACHINE, PHP.

The object of research is the process of forming automatic answers to questions. The subject of research is methods to support communication with the user.

The purpose of the work is to develop software for generating automatic answers to questions.

Materials, methods and technical means: PHP programming language, MySQL database management system, support vector machine.

Results. Web application was developed. The application allows users to perform the formation of automatic answers to user questions in the field of book sales, to implement management of this process, to fill in the model for training data, to organize the support of the user manager if necessary.

Conclusions. The paper proposes a method of forming automatic answers to questions, which consists in the initial classification of questions by categories, further refinement classification by specific schemes of answers to questions based on the method of support vector machine, coding questions based on the dictionary of the subject area and forming answers based on queries to database, which allows to answer questions on the subject area without participation of company managers.

Fields of usage are automated assistants, consultants on websites for the sale of goods.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз проблеми та постановка завдань дослідження	11
1.1 Методи оброблення природної мови	11
1.2 Системи «запитання-відповідь»	15
1.3 Висновки за розділом 1	16
2 Матеріали і методи.....	18
2.1 Задача класифікації даних.....	18
2.2 Метод опорних векторів.....	20
2.3 Метод формування автоматичних відповідей на запитання	22
2.4 Висновки за розділом 2	27
3 Опис програми.....	28
3.1 Визначення вимог до програмного забезпечення.....	28
3.2 Вибір мови програмування	30
3.3 Проєктування бази даних	33
3.4 Висновки за розділом 3	37
4 Розробка програми	38
4.1 Алгоритм функціонування програми.....	38
4.2 Опис розроблених модулів.....	39
4.3 Висновки за розділом 4	42
5 Експлуатація та тестування програми	43
5.1 Призначення програми	43
5.2 Умови виконання програми	43
5.3 Виконання програми.....	43
6 Економіко-організаційна частина.....	50
6.1 Планування розробки програмного продукту.....	50
6.2 Визначення витрат на розробку програми	52
6.2.1 Розрахунок основної заробітної плати.....	52

6.2.2 Відрахування на соціальне страхування.....	53
6.2.3 Визначення витрат на матеріали	54
6.2.4 Витрати на спеціальне устаткування	54
6.2.5 Накладні витрати.....	57
6.2.6 Плановий прибуток.....	58
6.3 Розрахунок капітальних вкладень	58
6.4 Розрахунок експлуатаційних витрат	59
6.5 Визначення економічної ефективності програмного продукту	60
6.6 Техніко-економічні показники	61
7 Охорона праці та безпека в надзвичайних ситуаціях.....	62
7.1 Аналіз потенційних небезпек.....	62
7.2 Заходи по забезпеченню безпеки	62
7.3 Заходи по забезпеченню промислової санітарії та гігієни праці	63
Висновки	71
Перелік джерел посилання	72
Додаток А Технічне завдання	76
Додаток Б Текст програми	81
Додаток В Слайди презентації.....	100

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

- ВДТ – візуальний дисплейний термінал;
ЕОМ – електронно-обчислювальна машина;
ПК – персональний комп'ютер;
ПУЕ – правила улаштування електроустановок.

ВСТУП

Все, що людина висловлює (усно або письмово), містить величезну кількість інформації: тема, яку ми обираємо, наш тон, підбір слів, все додає якийсь тип інформації, яку можна інтерпретувати та витягнути з неї цінність, теоретично можна зрозуміти і навіть передбачити поведінку людини, використовуючи цю інформацію, але є проблема: одна людина може генерувати сотні або тисячі слів у декларації, кожне речення з відповідною складністю, якщо потрібно масштабувати та аналізувати декілька сотень, тисяч або мільйони людей чи декларацій у певній географії, тоді ситуація некерована [1].

У комерційній діяльності аналіз природної мови є одним з важливих практичних завдань. Він стосується різних проблем, що охоплюють, наприклад, виділення оцінок або сенсу з коментарів користувачів, створення відповідей на запитання користувачів, прогнозування результатів роботи з товарами на основі текстових описів.

Дану роботу присвячено розгляду проблеми формування автоматичних відповідей на запитання користувача. Створення такої системи дозволяє автоматизувати процес взаємодії, швидше витягати з бази даних потрібні відповіді, ніж це зробить людина вручну, та підтримувати процес взаємодії з користувачем. По суті це проблема створення користувацького інтерфейсу природною мовою.

Створення такої системи формування автоматичних відповідей на запитання є актуальним практичним завданням, що на даний момент повністю не розв'язано, а у випадку різного роду комерційних систем невеликого масштабу потребує певного розвитку. Тому саме цю проблему обрано для розгляду в дипломній кваліфікаційній роботі магістра.

Об'єкт дослідження – процес формування автоматичних відповідей на запитання.

Предмет дослідження – методи формування автоматичних відповідей на запитання.

Мета роботи – розроблення програмного забезпечення формування автоматичних відповідей на запитання.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

1.1 Методи оброблення природної мови

Дані, отримані з розмов, декларацій або навіть твітів, є прикладами неструктурованих даних, неструктуровані дані не вписуються в традиційну структуру рядків і стовпців реляційних баз даних і представляють переважну більшість даних, наявних у реальному світі, тим не менше, завдяки досягненню таких дисциплін, як машинне навчання, щодо цієї теми відбувається велика революція, у наш час мова вже не про спроби інтерпретувати текст чи промову на основі його ключових слів, а про розуміння значення цих слів (когнітивний спосіб), таким чином можна виявити такі фігури мови, як іронія, або навіть провести аналіз настроїв [1]-[3].

Оброблення природної мови характеризується наступними властивостями:

– великі обсяги текстових даних: опрацювання природної мови допомагає комп'ютерам спілкуватися з людьми своєю мовою та масштабує інші мовні завдання, наприклад, оброблення природної мови дозволяє комп'ютерам читати текст, чути мову, інтерпретувати її, вимірювати настрої та визначати, які частини важливі, сучасні машини можуть аналізувати більше мовних даних, ніж люди, без втоми та послідовно, неупереджено, беручи до уваги приголомшливий обсяг неструктурованих даних, що генеруються щодня, – від медичних записів до соціальних мереж, автоматизація має вирішальне значення для повного ефективного аналізу текстових та мовних даних;

– структурування неструктурованого джерела даних: мова людини надзвичайно складна і різноманітна, людина виражає себе нескінченно, як усно, так і письмово, мало того, що існує сотні мов та діалектів, але всередині кожної мови є унікальний набір правил граматики та синтаксису, термінів та сленгу, коли людина пише, вона часто пишемо неправильно або скорочує

слова, або пропускає розділові знаки, коли люди говорять, вони мають регіональні акценти [1].

Основні завдання, які часто використовуються у можливостях оброблення природної мови вищого рівня:

- категоризація вмісту: короткий зміст документа на мовній основі, включаючи пошук та індексацію, попередження про вміст та виявлення дублікатів;

- розкриття теми та моделювання: необхідно точно фіксувати значення та теми в текстових колекціях та застосовувати до тексту вдосконалену аналітику, наприклад, оптимізацію та прогнозування;

- контекстне вилучення: автоматично отримувати структуровану інформацію з текстових джерел;

- аналіз настрою: визначення настрою або суб'єктивних думок у великій кількості тексту, включаючи аналіз думок;

- перетворення мови в текст і перетворення тексту в мову: перетворення голосових команд у письмовий текст і навпаки;

- узагальнення документів: автоматична генерація конспекту великих текстів;

- машинний переклад: автоматичний переклад тексту чи мови з однієї мови на іншу [4].

Синтаксис стосується розташування слів у реченні таким чином, що вони мають граматичний сенс, у обробці природної мови синтаксичний аналіз використовується для оцінки того, як природна мова узгоджується з граматичними правилами, комп'ютерні алгоритми використовуються для застосування граматичних правил до групи слів та виведення з них значення, ось декілька прийомів синтаксису, які можна використовувати:

- лематизація передбачає зменшення різних форм слова в єдину форму для зручного аналізу;

- морфологічна сегментація передбачає поділ слів на окремі одиниці, які називаються морфемами;

- сегментація слів передбачає поділ великого шматка безперервного тексту на окремі одиниці;
- позначення частини мови включає визначення частини мови для кожного слова;
- розбір передбачає проведення граматичного аналізу поданого речення;
- порушення речення передбачає розмічення меж речень на великому фрагменті тексту;
- стермінг передбачає вирізання складених слів до їх кореневої форми [5].

Семантика стосується значення, яке передається текстом: семантичний аналіз – один із складних аспектів обробки природної мови, який ще не до кінця вирішений, він передбачає застосування комп'ютерних алгоритмів для розуміння значення та інтерпретації слів та структури структури речень, зокрема існують наступні методи семантичного аналізу:

- розпізнавання іменованих сутностей: воно включає визначення частин тексту, які можна ідентифікувати та класифікувати за попередньо заданими групами, приклади таких груп включають імена людей та назви місць;
- неоднозначність сенсу слова: воно передбачає надання слову значення на основі контексту;
- генерація природної мови: передбачає використання баз даних для виведення семантичних намірів та перетворення їх на людську мову [5]-[7].

Модель «Bag of Words» – це загальновживана модель, яка дозволяє підрахувати всі слова в фрагменті тексту: в основному це створює матрицю входження для речення або документа, не враховуючи граматику та порядок слів, потім ці частотні слова або випадки використовуються як ознаки для підготовки класифікатора, цей підхід може мати декілька мінусів (таких як відсутність семантичного значення та контексту, а деякі слова (наприклад, "the" або "a") додають шуму до аналізу, для вирішення цієї проблеми одним

із підходів є масштабування частоти слів на основі того, як часто вони з'являються у всіх текстах (а не лише в тому, аналіз якого виконується), щоб оцінки такі часті слова, як „the”, які також часто зустрічаються в інших текстах, такий підхід до оцінювання називається TF-IDF і покращує обсяг слів за вагою, завдяки TF-IDF часті терміни в тексті «винагороджуються», але вони також отримують «покарання», якщо ці терміни часто зустрічаються в інших текстах, навпаки, цей метод виділяє та “винагороджує” унікальні або рідкісні терміни, враховуючи всі тексти, тим не менше, цей підхід досі не має контексту та семантики [1].

Токенізація – це процес сегментування тексту на речення та слова, по суті, це завдання розрізати текст на частини, які називаються лексемами, і в той же час викинути певні символи, такі як пунктуація: токенізація може також усунути пунктуацію, полегшуючи шлях до правильної сегментації слів, але також викликаючи можливі ускладнення, процес токенізації може бути особливо проблематичним при роботі з біомедичними текстовими доменами, які містять безліч дефісів, дужок та інших розділових знаків [1], [8].

Вилучення стоп-слів включає позбавлення від загальновживаних займенників та прийменників, таких як "and", "the" або "to" англійською мовою, у цьому процесі деякі найпоширеніші слова, які мають мале або взагалі не мають значення для цілі оброблення природної мови, відфільтровуються та виключаються з тексту, що підлягає обробці, таким чином видаляючи поширені та часті терміни, які не є інформативними щодо відповідного тексту, стоп-слова можна безпечно ігнорувати, виконуючи пошук у заздалегідь визначеному списку ключових слів, звільняючи простір бази даних та покращуючи час обробки, універсального списку зупинних слів немає, їх можна попередньо вибрати або побудувати з нуля, потенційний підхід полягає в тому, щоб почати з прийняття заздалегідь визначених стоп-слів і додати слова до списку пізніше, при цьому вилучення слів може знищити відповідну інформацію та змінити контекст у даному реченні.

(наприклад, якщо виконується аналіз настроїв, можна відкинути наш алгоритм, якщо видалити стоп-слово типу "не", за цих умов можна обрати мінімальний список слів і додати додаткові терміни залежно від конкретної мети) [1].

1.2 Системи «запитання-відповідь»

При пошуку інформації та обробці природних мов відповідь на запитання – це завдання автоматичного надання відповіді на питання, задане людиною природною мовою: можна виділити три основні окремі підзадачі, а саме: аналіз запитань, пошук документів та витягання відповідей, більшість систем відповідей на запитання відповідають цим трьом підзавданням, однак вони можуть відрізнятися тим, як вони реалізують кожну підзадачу.

Даний підхід стосується оброблення природних мов для взаємодії з системами «запитання-відповідь» з боку користувачів, при цьому існує багато типів питань, зокрема:

- фактоїдні запитання – це ті, що задаються головним чином про іменовану сутність, використовуючи, наприклад, слова: коли, де, скільки, хто та що, які задають відповідно факти про дату/час, місце, людину та організацію;

- запитання щодо визначення терміна чи поняття: запитання, в яких використовуються слова чому чи як, – це інший тип, на який важко відповісти, і дуже мало спроб відповісти на цей тип запитань [9].

Система відповідей на запитання допомагає ефективно знайти інформацію: пошукові системи використовуються для пошуку відповідних документів, коли реалізується пошук інформації в Інтернеті, однак, оскільки вони показують документи, необхідно прочитати документи та вирішити, чи містять вони потрібну інформацію, таким чином, комерційні пошукові системи мають функцію відповіді на запитання, щоб можна було ефективно знаходити інформацію, системи відповідей на запитання мають декілька

парадигм, але зазвичай використовуються дві основні парадигми: заснована на знаннях та відповідь на запитання на основі інформації:

- коли люди відповідають на питання, вони спочатку намагаються відповісти на це своїми знаннями, якщо вони не можуть відповісти на запитання, вони шукають відповідь в Інтернеті або в книгах, системи відповідей на запитання подібні до людських істот;

- перший етап: обробка запитань перетворює запитання на пошуковий запит, зазвичай видаляються слова зупинки та слова з характеристиками певної частини мови (власні імена (PROPN), цифри (NUM), дієслова (VERB), іменники (NOUN) та прикметники (ADJ));

- другий етап (витягання документа): під час пошуку документа отримуються відповідні документи за допомогою сформованого запиту, при подальшій обробці, оскільки відповідь буде витягнуто з цих документів, потрібно якомога більше шукати документи, які можуть містити відповідь;

- третій етап: під час пошуку уривків документи поділяються на менші одиниці (уривки), такі як речення та абзаци, і вибираються уривки, які, ймовірно, містять відповідь. Якщо документ короткий, це не потрібно, але якщо він довгий, вибір проходу ефективний, оскільки ми не знаємо, якій частині документа відповідає запит. Як правило, подальший процес вилучення відповідей займає багато часу, тому вибір уривків також має перевагу прискорення роботи всієї системи.

- четвертий етап: етап витягання відповідей витягує відповіді з уривків, модель виводить відповіді, зміщені з оцінками, потім реалізується класифікація відповіді на основі оцінки і представляється відповідь з найвищим балом користувачеві як остаточна відповідь [10].

1.3 Висновки за розділом 1

Виконано огляд методів оброблення природної мови та основних проблем і понять, які виникають в даному процесі. Розглянуто принципи

функціонування систем «запитання-відповідь».

Завдання дипломної кваліфікаційної роботи магістра:

- дослідження методів формування автоматичних відповідей на запитання;
- проєктування програмного забезпечення;
- розробка програмного забезпечення;
- тестування та відлагодження програми;
- економічне обґрунтування доцільності розробки.

2 МАТЕРІАЛИ І МЕТОДИ

Під час формування автоматизованих відповідей на запитання першим і визначальним етапом є етап аналізу запитання, як було розглянуто в розділі 1. Аналіз запитання передбачає виділення відповідної схеми роботи з даним запитанням. Відповідно співставлення деяких вхідних даних зі схемою є за своєю сутністю задачею класифікації даних. Тобто в процесі потрібно визначити шаблон реагування на запитання користувача в залежності від тексту такого запитання.

2.1 Задача класифікації даних

Класифікація даних загалом визначається як процес організації даних за відповідними категоріями, щоб вони могли бути використані та захищені більш ефективно, на базовому рівні процес класифікації полегшує пошук та отримання даних:

а) класифікація даних має особливе значення, коли мова йде про управління ризиками, дотримання та безпеку даних;

б) класифікація даних передбачає тегування даних, щоб зробити їх легкими для пошуку та відстеження, це також усуває багаторазове копіювання даних, що може зменшити витрати на зберігання та резервне копіювання, пришвидшуючи процес пошуку;

в) класифікація даних значно покращилась з часом: сьогодні ця технологія використовується для різних цілей, часто на підтримку ініціатив із захисту даних, але дані можуть бути класифіковані з ряду причин, включаючи легкість доступу, підтримання відповідності нормативним актам та досягнення різних інших ділових чи особистих цілей, у деяких випадках класифікація даних є нормативною вимогою, оскільки дані мають бути доступними для пошуку протягом визначених часових рамок, для цілей безпеки даних класифікація даних є корисною тактикою, яка полегшує

належні реакції захисту на основі типи даних, що отримуються, передаються або копіюються;

г) класифікація даних часто включає множину тегів та міток, що визначають тип даних, їх конфіденційність та цілісність, доступність може також враховуватися в процесі класифікації даних, рівень чутливості даних часто класифікується на основі різного рівня важливості або конфіденційності, що в подальшому відповідає заходам безпеки, введеним для захисту кожного рівня класифікації;

д) існує три основні типи класифікації даних, які вважаються галузевими стандартами:

- 1) класифікація на основі вмісту перевіряє та інтерпретує файли, що шукають конфіденційну інформацію;

- 2) класифікація, що базується на контексті, розглядає застосування, місцезнаходження та інші параметри серед інших змінних як непрямі показники конфіденційної інформації;

- 3) класифікація на основі користувачів залежить від вибору кожного документа вручну кінцевим користувачем: класифікація, заснована на користувачах, покладається на знання та розсуд користувачів при створенні, редагуванні, перегляді чи розповсюдженні, щоб позначити важливі документи;

е) підходи на основі вмісту, контексту та користувача можуть бути як правильними, так і неправильними залежно від бізнес-потреб та типу даних [11].

Виконання класифікаційного завдання правильно могло б раніше дозволити виявляти та діагностувати несправності та призводило б до швидких рішень щодо контролю, основними методами, які використовуються для розв'язання даної проблеми, є наступні:

– метод опорних векторів – це метод, що базується на відстані, він став одним із найкорисніших методів вирішення проблем машинного навчання з хорошими показниками узагальнення;

– дерева рішень: використання дерев рішень ефективно та підходить для великих тренувальних наборів, крім того, алгоритми генерації дерев рішень не вимагають додаткової інформації, крім тієї, що вже міститься в навчальних даних, дерева рішень, як класифікатор на основі правил, використовуються для класифікації;

– гаусівський наївний Байєс: незважаючи на спрощення припущень, що лежать в основі даного рішення, він може бути конкурентоспроможним з набагато складнішими алгоритмами;

– метод k -найближчих сусідів – це метод, що базується на відстані, і є лінійно контрольованою технікою класифікації, метод працює над положенням точки даних щодо інших точок даних, враховуючи k -найближчі шаблони в навчальних даних до тестового шаблону [12].

2.2 Метод опорних векторів

Метод опорних векторів – це контрольована модель машинного навчання, яка використовує алгоритми класифікації для класифікаційних задач двох груп: після передачі моделі SVM наборів позначених навчальних даних для кожної категорії, модель може класифікувати нові дані [13].

Метод опорних векторів визначається наступними характеристиками:

– швидкий і надійний алгоритм класифікації, який дуже добре працює з обмеженою кількістю даних для аналізу;

– головне завдання методу опорного вектора – знайти гіперплощину в N -мірному просторі (N – кількість ознак), яка чітко класифікує точки даних;

– для розділення двох класів точок даних існує множина можливих гіперплощин, які можна вибрати, мета – знайти площину, яка має максимальну відстань між точками даних обох класів, максимізація відстані поля забезпечує певне підкріплення, щоб майбутні точки даних можна було класифікувати з більшою впевненістю;

– гіперплощини – це межі прийняття рішень, які допомагають класифікувати точки даних: точки даних, що потрапляють по обидві сторони гіперплощини, можна віднести до різних класів, крім того, розмір гіперплощини залежить від кількості ознак: якщо кількість вхідних функцій дорівнює 2, тоді гіперплан – це просто лінія, якщо кількість вхідних ознак дорівнює 3, тоді гіперплощина стає двовимірною площиною;

– опорні вектори – це точки даних, які знаходяться ближче до гіперплощини і впливають на положення та орієнтацію гіперплощини: використовуючи ці вектори підтримки, максимізується запас класифікатора, видалення векторів підтримки змінить положення гіперплощини, це точки, які допомагають будувати метод опорних векторів;

– в логістичній регресії береться вихід лінійної функції і стискається значення в межах $[0,1]$ за допомогою сигмовидної функції: якщо зведене значення більше порогового значення (0,5), присвоюється мітка 1, інакше присвоюється мітка 0, у свою чергу в методі опорних векторів береться вихід лінійної функції, і якщо цей вихід більший за 1, ситуація ідентифікується з одним класом, а якщо результат -1, то ідентифікується з іншим класом [14].

Виходячи з описаної логіки, найпростіший варіант методу опорних векторів не підтримує багатокласову класифікацію, він підтримує двійкову класифікацію та розділяє точки даних на два класи, для багатокласової класифікації використовується той самий принцип після розбиття задачі багатокласової класифікації на множинні двійкові класифікаційні задачі:

– ідея полягає в тому, щоб зіставити точки даних з багатовимірним простором, щоб отримати взаємне лінійне розділення між кожними двома класами, це називається індивідуальним підходом, який розбиває багатокласову задачу на множинні двійкові класифікаційні проблеми;

– інший підхід, який можна застосувати, – це «один проти всіх», у цьому підході розбивка встановлюється на двійковий класифікатор для кожного класу;

- одинарний метод опорних векторів виконує двійкову класифікацію і може розрізняти два класи, щоб, згідно з двома підходами до розбиття, класифікувати точки даних із набору даних N класів;
- у підході «один проти всіх» класифікатор може використовувати N опорних векторів, кожен опорний вектор передбачав би членство в одному з N класів;
- у підході «один на один» класифікатор може використовувати $N \cdot (0.5 \cdot N - 0.5)$ опорних векторів;
- у підході «один на один» потрібна гіперплощина, щоб розділяти кожні два класи, нехтуючи точками третього класу, це означає, що поділ враховує лише бали двох класів у поточному розділенні;
- у підході «один проти всіх» потрібна гіперплощина, щоб відразу відокремити клас від усіх інших, це означає, що поділ враховує всі моменти, поділяючи їх на дві групи; група для балів за клас i група для всіх інших балів [15].

2.3 Метод формування автоматичних відповідей на запитання

Запропонований метод формування автоматичних відповідей на запитання ґрунтується на тому, що в ньому має виконуватися зведення запитання користувача до запиту до бази даних. Тоді за результатами виконання запиту можна за вибраними даними сформулювати відповідь. Це засіб автоматизації роботи в обраній предметній області. Відповідно в основі роботи методу лежить база даних, в якій зберігаються дані, необхідні для роботи в даній предметній області.

Звичайно, що для цього спочатку має бути певним чином оброблено саме запитання початкове. Окрім того зрозуміло, що потрібно якимось чином звести запитання, поставлене користувачем, до форми запитання, представленої в шаблоні. Це задача класифікації, адже в результаті має бути визначено, який саме шаблон відповідає поставленому запитанню.

Для формування автоматичних відповідей було спочатку сформовано групу стандартних запитань, що стосуються обраної предметній області. Для кожного запиту має бути визначено схему відповіді.

Схема відповіді передбачає собою наступні складові:

- саме запитання, виражене у загальному вигляді, тобто з нього можуть бути вилучені зайві, неінформативні для виділення його, частини;
- запит до бази даних, який має бути виконано в результаті;
- шаблон відповіді на запитання.

Така структура схеми дуже важлива, оскільки вона концентрує в собі і запитання, і подальшу думку, і відповідь.

Важливо, що не всі запитання і відповіді будуть вкладатися в таку схему, тому що існують запитання, на які можна дати статичну відповідь. Така структура моделює динамічну відповідь. При цьому її можна використати і для статичного варіанту, зберігаючи просто в позиції запиту до бази даних порожнє значення.

Окрім того ідея запропонованого методу полягає в двоступеневому розбитті запитань на групи, в результаті чого має бути виділено в кінці спосіб формування відповіді на запитання. Оскільки запитань, до яких має бути зведено процес розбору, може бути достатньо багато (тільки в цьому варіанті система зможе працювати достатньо якісно), то для того щоб не перенасичувати кількість класів, вирішено було спочатку виконати попереднє оброблення, на якому має бути вирішено, до якої саме категорії належить запитання.

Відповідно всі запитання мають бути поділені на окремі категорії. Кожна з цих категорій має бути заповнена набором конкретних запитань.

Для реалізації даної ідеї має використовуватися процес двоетапної класифікації. Тобто спочатку класом є категорія, а потім конкретне запитання, тобто конкретна схема відповіді.

Далі буде описано етапи, які загалом складають метод формування автоматичних відповідей на запитання.

На першому етапі має бути сформовано словник предметної області. До нього мають входити тільки ті слова, які є важливими для запитань предметної області. Це основні терміни, основні сутності, які впливають безпосередньо на розбір запитання в тому числі.

На другому етапі має бути визначено вибірку даних. Це ті запитання, для яких було співставлено категорію і схему відповіді. Категорія є міткою класу, що буде використовуватися на першому етапі класифікації, а схема відповіді є міткою класу, який має бути отримано в результаті для заданого запитання, тобто це результат другого етапу самої класифікації. Тобто для кожного запитання мають бути представлені конкретні приклади таких запитань, що мають один і той же сенс.

Всю вибірку даних має бути розділено на категорії. Відповідно підвибірок даних має бути стільки, скільки категорій передбачено.

На другому етапі має бути реалізовано навчання класифікації категорій на основі методу опорних векторів. Метод опорних векторів обрано зокрема і через те, що кількість даних в цій задачі може бути достатньо обмеженою, тому застосування нейронних мереж може бути ускладненим, особливо на перших етапах роботи такої системи, але можливо і в подальшому, якщо кількість користувачів, що будуть звертатися до системи, буде невеликою.

На третьому етапі має бути реалізовано навчання класифікації запитань всередині кожної конкретної категорії на основі методу опорних векторів. Кількість моделей на цьому рівні має відповідати кількості категорій.

Тобто на другому і третьому етапах має бути сформовано набір моделей, які будуть використовуватися для двоетапної класифікації. Загальна кількість моделей буде дорівнювати кількості категорій плюс 1.

З наступного етапу відбувається робота безпосередньо зі сформульованим запитанням користувача.

На четвертому етапі користувач формує власне запитання.

На п'ятому етапі реалізується оброблення даного запитання. Для цього кожне слово перетворюється до своєї початкової форми. Саме в ній представлені слова в словнику предметної області. Далі відбувається співставлення кожного слова в реченні зі словником, в результаті має бути сформовано код запитання.

Довжина вектора, що буде представляти код запитання дорівнює кількості слів у словнику.

Код запитання формується наступним чином:

- для кожного слова виконується його пошук в словнику;
- якщо слово знайдено, то встановлюється у відповідні позиції одиниця;
- якщо слово не знайдено, то встановлюється нуль.

Отримана послідовність нулів та одиниць, кількість яких дорівнює кількості слів у словнику, представляє результуючий код запитання.

На шостому етапі визначається за допомогою використання методу опорних векторів категорія, до якої належить запитання. Тоді в якості моделі за методом опорних векторів на наступний етап визначається модель, що відповідає визначеній категорії.

На сьомому етапі виконується класифікація і визначення конкретної схеми відповіді на запитання, в результаті чого виконується (за наявності) запит до бази даних, отримуються результуючі дані, підставляються в шаблон і на виході отримується конкретна текстова відповідь на запитання користувача.

Саме в даній семиетапній процедурі і полягає метод формування автоматичних відповідей на запитання, запропонований в роботі. Даний метод може застосовуватися для конкретної предметної області, адже потребує створення словника, в якому кількість слів не може бути безмежною, адже вона визначає розмірність вектору в результаті.

Дану сукупність етапів було схематично представлено у вигляді схеми формування автоматичних відповідей на запитання (рис. 2.1).

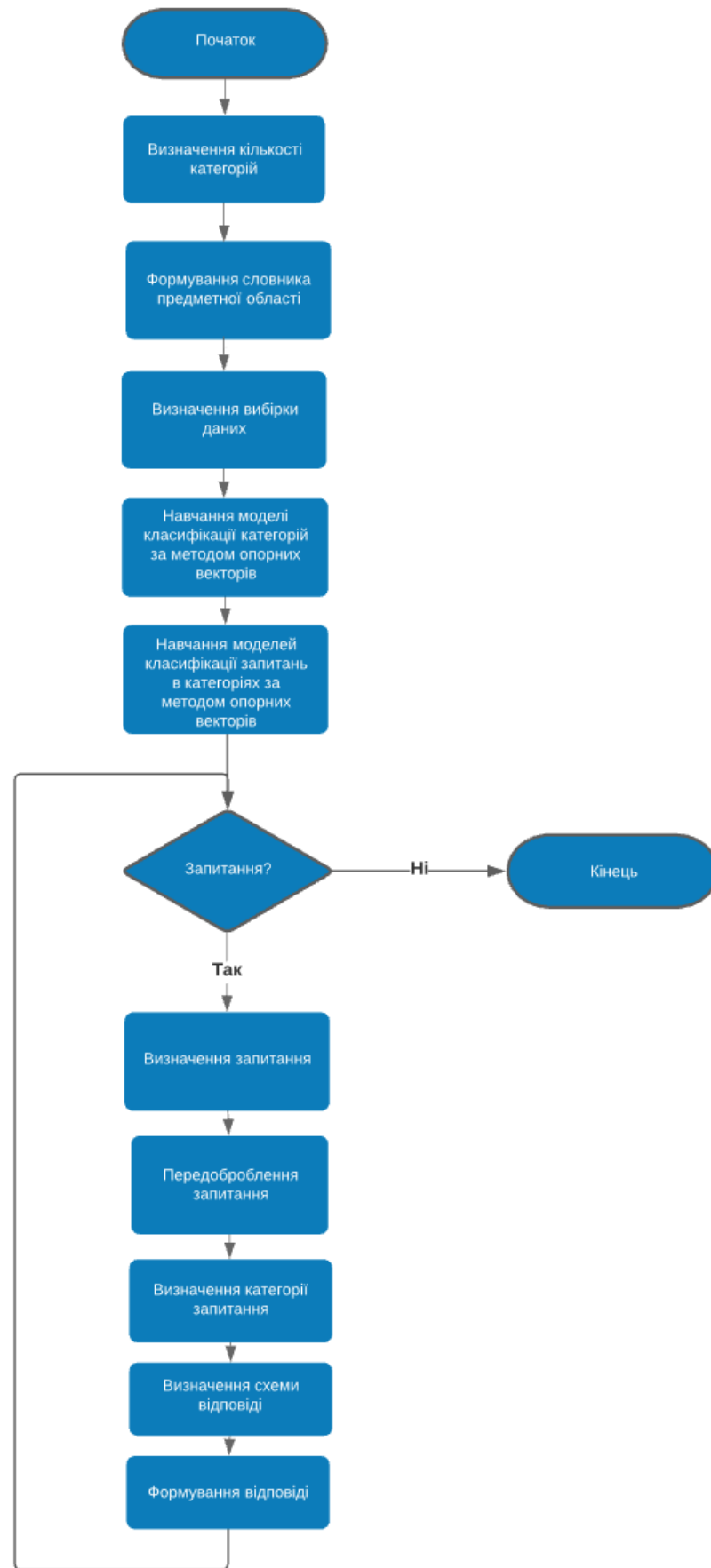


Рисунок 2.1 – Схема методу формування автоматичних відповідей на запитання

2.4 Висновки за розділом 2

У другому розділі отримано такі результати:

- проаналізовано задачу формування автоматичних відповідей на запитання як задачу класифікації;
- проаналізовано метод опорних векторів і особливості його застосування для задачі класифікації;
- представлено власний метод формування автоматичних відповідей на запитання;
- наведено схему роботи методу.

Наукова новизна роботи полягає в тому, що в ній запропоновано метод формування автоматичних відповідей на запитання, що полягає в початковій класифікації запитань за категоріями, подальшій уточнюючій класифікації за конкретними схемами відповідей на запитання на основі використання методу опорних векторів, кодуванні запитання та формуванні відповіді на основі виконання запиту до бази даних, що дозволяє відповідати на запитання за предметною областю без участі менеджерів компанії.

3 ОПИС ПРОГРАМИ

3.1 Визначення вимог до програмного забезпечення

Програма головним чином направлена на організацію автоматичних відповідей на запитання користувача, тому підтримка цієї функції є основою функціонування програми.

В якості основних вимог до програми, що розробляється, було виділено такі функціональні вимоги:

- пересилання повідомлень користувача на сервер і збереження їх в базі даних;
- формування автоматичних відповідей на запитання користувачів;
- відображення автоматичних відповідей користувачу;
- додавання схем розбору відповідей на запитання;
- змінювання схем розбору відповідей на запитання;
- додавання прикладів для обраної схеми розбору;
- ініціалізація дій за запитанням;
- перегляд історії автоматичних відповідей;
- коригування результатів класифікації відповідей;
- додавання категорій відповідей;
- виконання навчання;
- виконання повторного навчання;
- авторизація користувачів;
- реєстрація користувачів.

Лише частина з цих функцій має бути доступною звичайному користувачу, яким по суті є клієнт сервісу. Переважна частина даної функціональності має бути доступною адміністратору.

Користувач має взаємодіяти з програмою (рис. 3.1) шляхом виконання наступних функцій:

- відправлення повідомлень;
- відображення автоматичних відповідей;

- підтвердження дій у відповідь на запитання;
- авторизація користувачів;
- реєстрація користувачів.

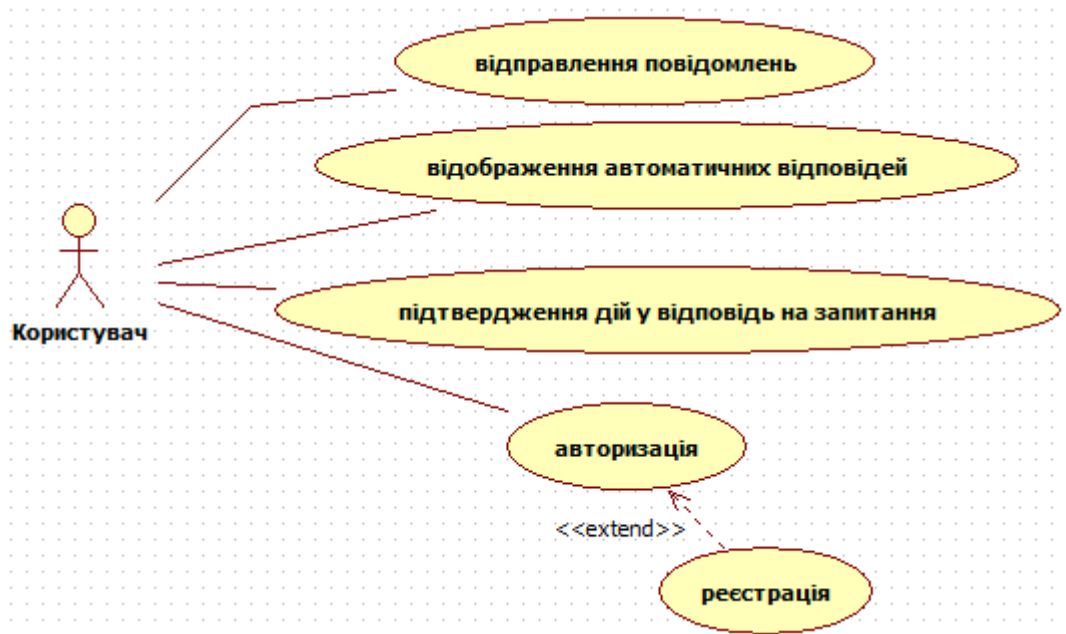


Рисунок 3.1 – Діаграма прецедентів користувача

Адміністратор має взаємодіяти з програмою (рис. 3.2) шляхом виконання наступних функцій:

- додавання схем розбору відповідей на запитання;
- змінювання схем розбору відповідей на запитання;
- додавання прикладів для обраної схеми розбору;
- перегляд історії автоматичних відповідей;
- коригування результатів класифікації;
- додавання категорій відповідей;
- виконання навчання;
- виконання повторного навчання;
- авторизація адміністраторів.

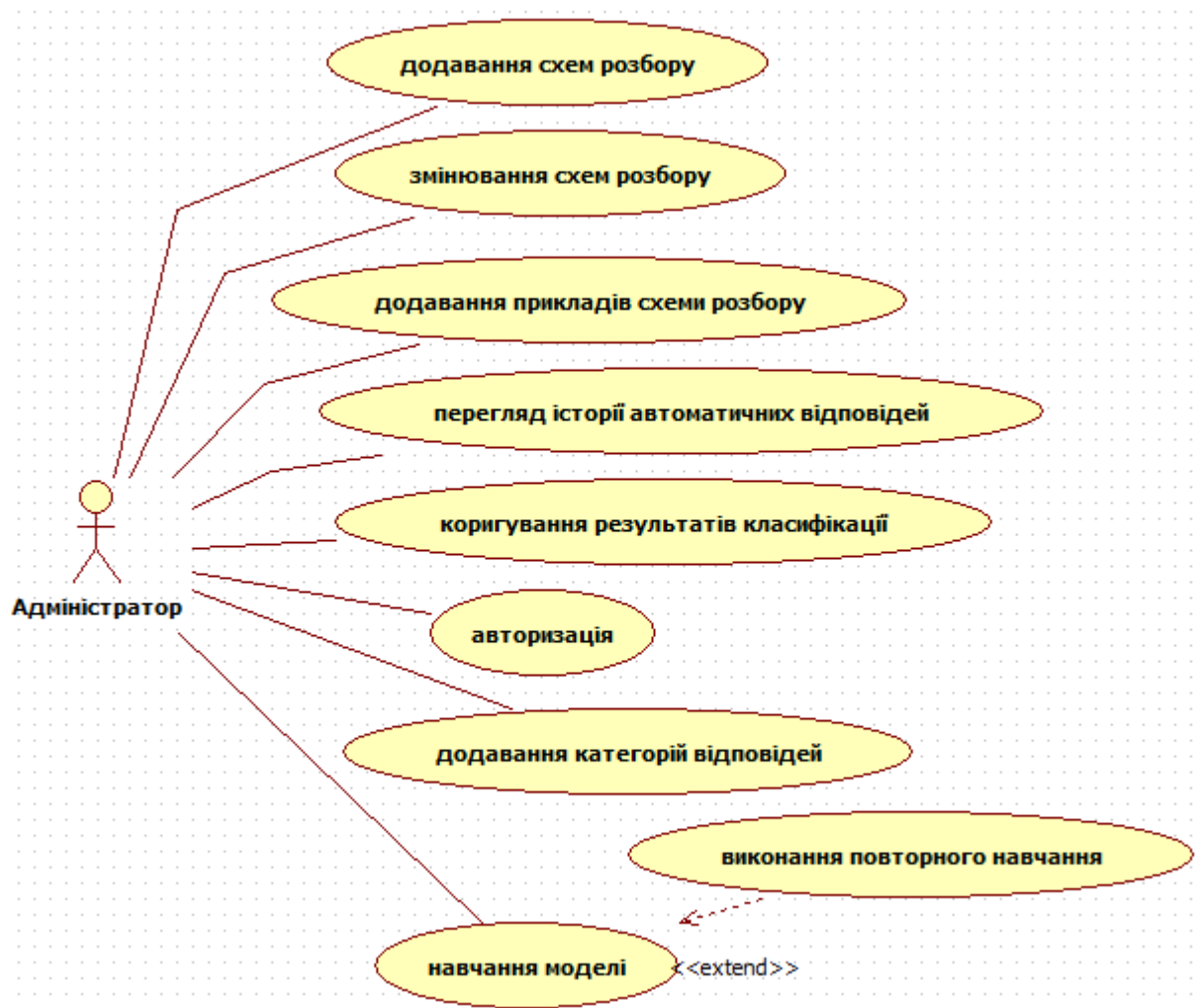


Рисунок 3.2 – Діаграма прецедентів адміністратора

3.2 Вибір мови програмування

Для підтримки взаємодії між сервером та клієнтом, необхідність чого диктується по суті самою назвою теми і відповідно проблеми, що розглядається, зручно використовувати як раз вебрішення. До того ж вебзастосунок буде зручним для користувачів рішень, адже не потребує встановлення, буде мати потенційну можливість зручного налаштування під всі типи пристроїв. Тобто в результаті даної логіки саме вебзастосунок було обрано в якості способу побудови рішення, що буде створюватися для формування автоматичних відповідей на запитання.

Загальновідомим фактом є, що найпопулярнішою мовою веброзробки є мова програмування PHP. Щодо цього потрібно виділити, що мова програмування PHP має наступні переваги:

- достатня документація: доступно багато навчальних посібників, посібників та інших довідкових матеріалів, що полегшує розробку та може надати допомогу у важких ситуаціях, до того ж сам процес вивчення мови PHP загалом і вивчення її особливостей (як в даному випадку) є порівняно простим;

- покращена швидкість завантаження: використання PHP призводить до швидшого завантаження сторінок вебсайтів порівняно з багатьма іншими технологіями веброзробки: наприклад, в даний час для більшості сценаріїв використання PHP приблизно втричі швидше, ніж Python, у свою чергу, менший час завантаження є важливим фактором рейтингу пошукової оптимізації, який допомагає надалі просувати вебсайт, забезпечуючи конкурентні переваги, більш висока швидкість завантаження забезпечує задоволення клієнтів і, в поєднанні з іншими перевагами, допомагає побудувати та зберегти клієнтську базу;

- широкий вибір баз даних: PHP дозволяє підключатися майже до будь-якого типу бази даних, найпоширенішим вибором є MySQL, головним чином тому, що вона безкоштовна, ефективна і популярна серед розробників, іншими надійними варіантами систем управління базами даних, сумісними з PHP, є mSQL, MS-SQL, SQLite, PostgreSQL та інші, крім того, PHP можна однаково добре використовувати з Elasticsearch, Redis, MongoDB та іншими нереляційними базами даних, таким чином, розробники не обмежуються використанням певної бази даних і можуть вибрати найбільш оптимальну для майбутнього програму, беручи до уваги всі відповідні фактори;

- недороге програмне забезпечення з відкритим кодом: PHP – це безкоштовна технологія, яка забезпечує значну економію бюджету розробки, крім того, більшість засобів розробки, які зазвичай використовуються в поєднанні з PHP, є програмним забезпеченням з відкритим кодом і можуть

використовуватися безкоштовно; таким чином, вони додатково знижують вартість проекту, більше того, існує дуже багато фреймворків, таких як Laravel та CodeIgniter, а також різні CMS, такі як Wordpress та Drupal, які розширюють функціональність PHP та роблять процес розробки простішим та ефективнішим;

- дешевші послуги хостингу: найбільш поширеним сценарієм роботи вебсайту PHP є стек LAMP – це означає, що вебсайт працює на веб-сервері Apache HTTP, розгорнутому в системі Linux, і використовує MySQL як базу даних, – всі ці компоненти безкоштовні, а стек добре перевірений, що передбачає скорочення необхідного часу та коштів на розробку;

- відмінна сумісність з HTML: PHP пропонує вбудоване програмування HTML, що є причиною неймовірної синергії між цими двома технологіями, у більшості випадків PHP-скрипт не втручається в HTML-код вебсторінки, а замість цього заповнює його, залишаючись всередині меж, визначених тегами `<?php ?>`, і навпаки, якщо створено цілу сторінку на PHP, можна інтегрувати HTML-код за допомогою сценарію, у цьому випадку відмінністю від традиційної HTML-сторінки буде перевернутий порядок відкриття тегів: теги PHP передують тегам HTML;

- хороша гнучкість: гнучкість дозволяє PHP ефективно поєднуватись з багатьма іншими мовами програмування, щоб програмний продукт міг використовувати найефективнішу технологію для кожної конкретної функції, більше того, PHP – це міжплатформна мова, що означає, що розробники можуть використовувати будь-яку основну операційну систему – Windows, Linux, macOS – для виконання кодування, така гнучкість значно полегшує процес розробки, роблячи його швидшим і дешевшим;

- сумісність із хмарними сервісами: багато сучасних продуктів, як правило, використовують хмарні обчислювальні рішення, такі як Amazon Web Services, для різних цілей, застосунки, написані на PHP, підтримуються різними хмарними сервісами, такими як AWS Lambda, наприклад, таким чином, PHP-програму можна розгорнути на хмарному сервері та досягти

чудової масштабованості та інших корисних ефектів, більше того, область хмарних обчислень не монополізована іншими мовами кодування, тому PHP зайняв своє місце в таких реалізаціях [16]-[18].

Альтернативою для веброзробки є мова програмування Java. Результати порівняння даних мов програмування було в підсумку зведено до табл. 3.1.

Таблиця 3.1 – Порівняння мов програмування

Критерій	PHP	Java
Призначення	Загальне, веброзробка	Загальне
Система типів	Динамічна	Статична
Безкоштовність	Так	Так
Швидкість завантаження	Дещо менше	Середня

3.3 Проєктування бази даних

В якості системи для управління базою даних було обрано MySQL, що відзначається такими особливостями:

- простота у використанні: можна створювати та взаємодіяти з MySQL, використовуючи лише декілька простих операторів SQL;
- безпека: MySQL представлений надійним рівнем захисту даних, який захищає конфіденційні дані від зловмисників, крім того, паролі шифруються;
- підтримка клієнт-серверної архітектури: MySQL слідує за роботою архітектури клієнт-сервер, існує сервер баз даних і багато клієнтів (прикладних програм), які взаємодіють із сервером, тобто вони можуть запитувати дані, зберігати зміни тощо;

– безкоштовне завантаження: MySQL можна використовувати безкоштовно, завантажуючи його з офіційного вебсайту MySQL без будь-яких витрат;

– масштабованість: MySQL підтримує багатопоточність, що робить його легко масштабованим рішенням, може обробляти майже будь-який обсяг даних, до 50 мільйонів рядків і більше, розмір файлу за замовчуванням становить близько 4 Гб, однак можна збільшити це число до теоретичної межі 8 Тб даних;

– швидкість: MySQL вважається однією з найшвидших мов баз даних;

– висока гнучкість: MySQL підтримує велику кількість вбудованих програм, що робить MySQL дуже гнучким;

– сумісна з багатьма операційними системами: MySQL сумісна для роботи в багатьох операційних системах;

– дозволяє відкочувати транзакції, фіксувати та відновлювати збої;

– ефективність роботи з пам'яттю;

– висока працездатність: MySQL є швидшим, надійнішим та дешевшим завдяки своїй унікальній архітектурі механізму зберігання, це забезпечує дуже високі результати в порівнянні з іншими базами даних, не втрачаючи при цьому суттєвої функціональності програмного забезпечення;

– висока продуктивність: MySQL використовує тригери, збережені процедури та подання, які дозволяють розробнику забезпечити більшу продуктивність;

– незалежна платформа: може завантажувати, встановлювати та виконувати більшість доступних операційних систем;

– розбиття – ця функція покращує продуктивність та забезпечує швидке управління великою базою даних [19].

Для роботи за описаною функціональністю з програмою необхідно зберігати такі основні сутності:

– Account – обліковий запис;

– Dialog – діалог;

- Message – повідомлення;
- Category – категорія;
- AnswerScheme – схема автоматичної відповіді;
- Example – приклад запитання;
- MessageAnswer – автоматична відповідь.

Для таблиці Account визначено такі стовпці:

- Id – ідентифікаційний номер;
- Name – ім'я;
- Password – пароль;
- Email – пошта;
- CreatedDate – дата створення;
- Type – тип облікового запису (користувач, адміністратор, автоматизатор).

Для таблиці Dialog визначено такі стовпці:

- Id – ідентифікаційний номер;
- AccountId1 – номер першого облікового запису;
- AccountId2 – номер другого облікового запису.

Для таблиці Message визначено такі стовпці:

- Id – ідентифікаційний номер;
- DialogId – номер діалогу;
- Author – автор повідомлення (перший або другий користувач з діалогу);
- Information – текст повідомлення;
- When – дата і час створення повідомлення.

Для таблиці MessageAnswer визначено такі стовпці:

- Id – ідентифікаційний номер;
- MessageId – номер повідомлення;
- AnswerSchemeId – номер схеми відповіді;
- Status – стан (коректна або некоректна відповідь).

Для таблиці Category визначено такі стовпці:

– Id – ідентифікаційний номер;

– Title – назва категорії.

Зв'язки між цими таблицями представлені на рис. 3.3.

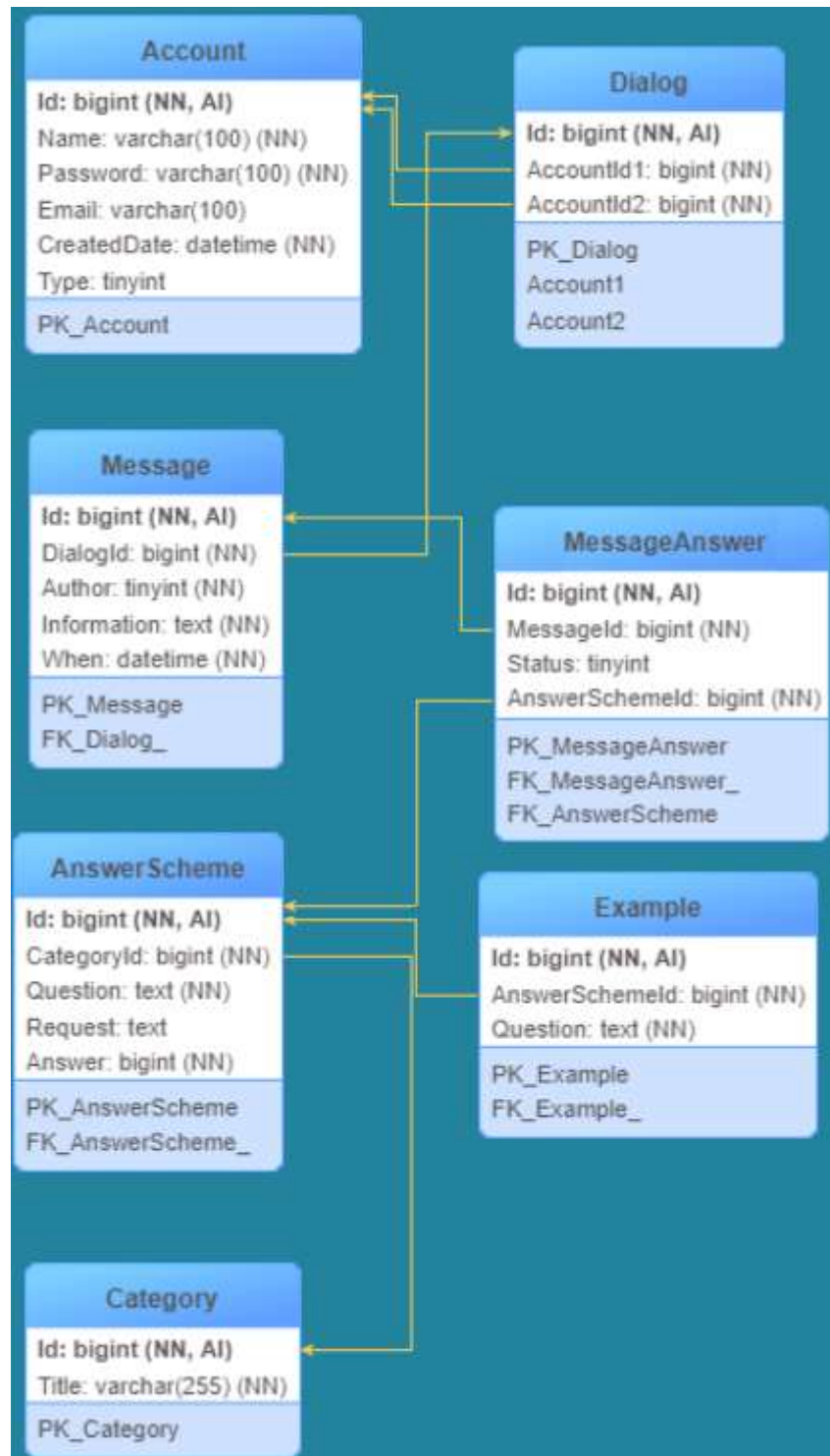


Рисунок 3.3 – Структура бази даних

Для таблиці AnswerScheme визначено такі стовпці:

- Id – ідентифікаційний номер;
- CategoryId – категорія запитання;
- Question – текст запитання;
- Request – SQL-запит;
- Answer – схема відповіді на запитання.

Для таблиці Example визначено такі стовпці:

- Id – ідентифікаційний номер;
- AnswerSchemeId – номер схеми відповіді;
- Question – запитання.

3.4 Висновки за розділом 3

У третьому розділі отримано такі результати:

- функціональні вимоги сформульовано;
- побудовано діаграми прецедентів для користувача та адміністратора;
- обрано мову програмування PHP для реалізації;
- обрано систему MySQL для реалізації бази даних;
- визначено структуру бази даних.

4 РОЗРОБКА ПРОГРАМИ

4.1 Алгоритм функціонування програми

Програмою підтримується роботи неавторизованих користувачів, адміністраторів та авторизованих користувачів.

Спочатку користувач є неавторизованим, йому доступний тільки набір стандартних дій, що включають перегляд даних, підтримка яких в предметній області виконується, консультування (відповідно користувач не зможе виконати замовлення) та можливість авторизації. Користувач сам обирає, яку саме функцію потрібно виконати.

Після того як користувач авторизувався, обирається тип його облікового запису.

Якщо це звичайний користувач, то йому доступний набір варіантів роботи з даними, але вже з підтримкою замовлення, консультування, з якого можна також виконати замовлення книги.

Якщо це адміністратор, то йому надається можливість широкої функціональності, що включає засоби перегляду історії комунікації в автоматичному режимі, керування цим спілкуванням, відповідей на запитання користувачів, безпосередньо наповнення даними та керування замовленнями.

Окрім того адміністратор може створювати схем відповідей, редагувати їх та наповнювати прикладами, після чого ініціювати навчання або повторне навчання. Вся ця функціональність доступна тільки адміністратору і дозволяє повноцінно керувати моделлю.

В будь-який момент користувач може вийти з власного облікового запису або завершити роботу з програмою.

На рис. 4.1 представлена відповідна створена схема функціонування програми.

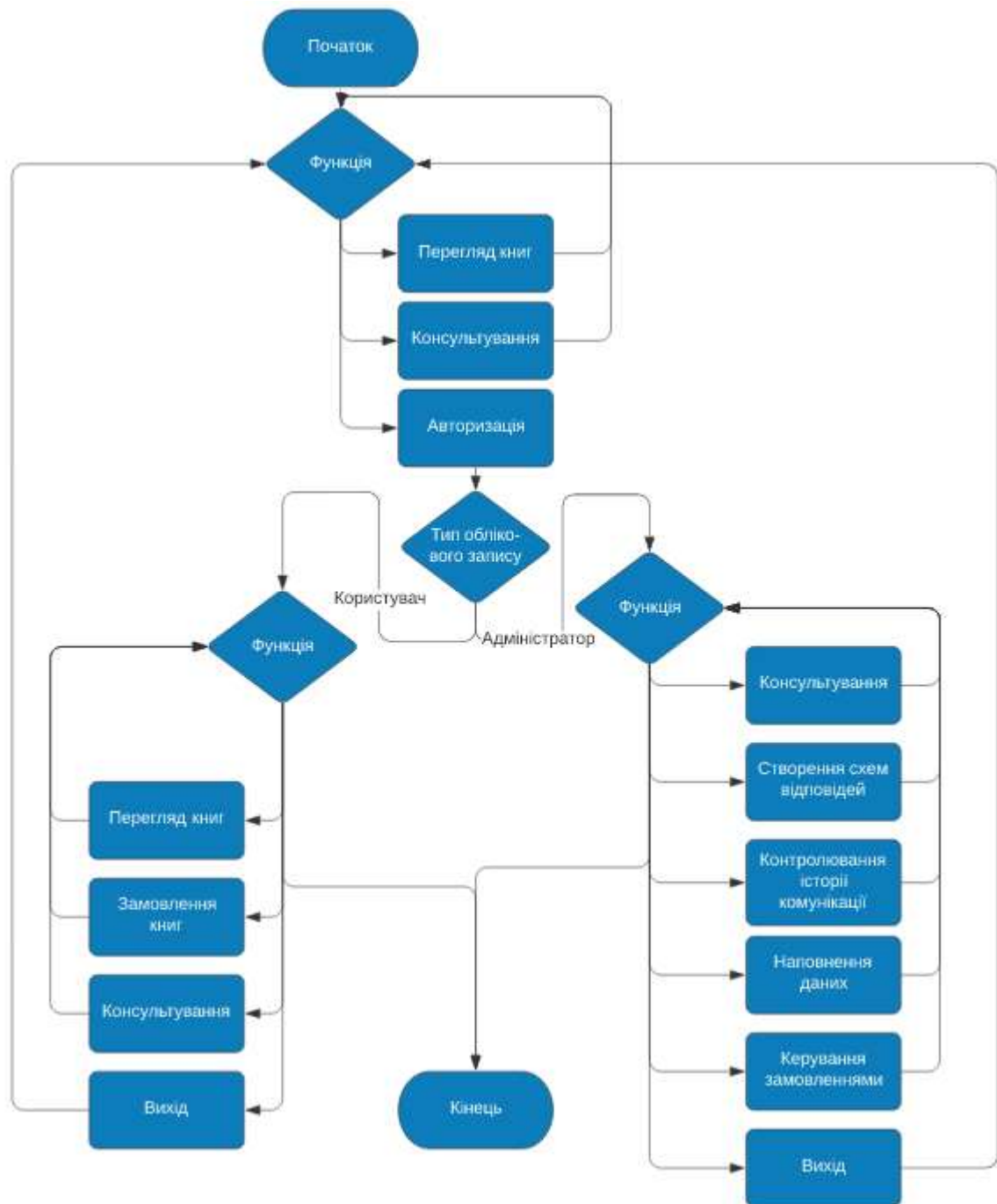


Рисунок 4.1 – Схема функціонування програми

4.2 Опис розроблених модулів

Інтерфейс вебзастосунку розроблено з використанням безкоштовного HTML-шаблону [20].

Клас DataBaseProcessor забезпечує виконання запитів до бази даних. Для надання відповідних засобів було розроблено наступні методи класу:

- SaveAccount(\$name, \$password, \$email, \$createddate, \$type) – збереження даних акаунта в базі даних;
- GetDialogsWithManager(\$num) – метод отримання переліку діалогів, які велися користувачем з заданим номером з менеджерами;
- GetDialogsWithBot(\$num) – метод отримання переліку діалогів, які велися користувачем з заданим номером з програмою;
- UpdateAccountEmail(\$name, \$email) – метод оновлення електронної поштової адреси користувача;
- ChangeAccountType(\$name, \$type) – метод зміни типу акаунта;
- UpdateAccountPassword(\$name, \$pass) – метод зміни паролю акаунта;
- GetAccountByEmail(\$email) – метод пошуку акаунту за електронною поштою користувача;
- GetCategory(\$num) – метод отримання даних категорії за її номером;
- GetAnswerScheme(\$num) – метод отримання даних схеми відповіді на запитання за номером;
- GetMessageAnswer(\$num) – метод отримання автоматизованої відповіді на запитання за номером;
- GetAccountByName(\$name) – метод отримання акаунту за логіном;
- GetAccountType(\$name) – метод визначення типу облікового запису;
- LoadAccounts() – метод завантаження переліку всіх акаунтів;
- IfAccount(\$name) – перевірка існування акаунта з заданим логіном;
- SaveMessage(\$dialog, \$author, \$information, \$when) – метод збереження нового повідомлення в базі даних;
- SaveMessageNow(\$dialog, \$author, \$information) – метод збереження нового повідомлення в базі даних з поточним часом;
- AccountCheck(\$name, \$pass) – метод авторизації акаунта, в результаті повертається його номер;
- LoadDialogMessages(\$id) – метод завантаження повідомлень діалогу з заданим номером;

- LoadDialogMessage(\$id) – метод завантаження повних даних повідомлення з заданим номером;
- LoadMessageAnswers() – метод завантаження всіх автоматичних відповідей на запитання;
- LoadFullMessageAnswers() – метод завантаження повних даних всіх автоматичних відповідей на запитання;
- LoadMessages(\$num) – метод завантаження всіх повідомлень з діалогу з заданим номером;
- LoadUserMessages(\$num) – метод завантаження повідомлень користувача з заданим номером;
- UpdateMessage(\$id, \$author, \$information, \$when) – метод редагування повідомлення;
- ChangeMessageText(\$id, \$information) – метод оновлення тексту повідомлення;
- UpdateDialog(\$id, \$acc1, \$acc2) – метод редагування даних діалогу;
- UpdateMessageAnswer(\$id, \$message, \$status, \$as) – метод редагування автоматичного повідомлення;
- UpdateAnswerScheme(\$id, \$categ, \$question, \$request, \$answer) – метод редагування схеми відповіді на повідомлення;
- UpdateExample(\$id, \$as, \$question) – метод редагування прикладу;
- UpdateCategory(\$id, \$title) – метод редагування даних категорії;
- LoadExample(\$num) – метод отримання повних даних прикладу з заданим номером;
- LoadExamples() – метод завантаження всіх прикладів;
- LoadExamplesOnAnswerScheme(\$num) – метод завантаження всіх прикладів, що відповідають схемі з заданим номером;
- LoadCategories(\$num) – метод завантаження всіх категорій;
- LoadCategoryAnswerScheme(\$num) – метод завантаження всіх схем відповідей на запитання з заданої категорії;

- LoadCategoryAnswerSchemeOnName(\$name) – метод завантаження даних схем відповідей на запитання за назвою її категорії;
- LoadExamplesOnAnswerScheme(\$num) – метод завантаження прикладів за схемю відповідей на запитання за номером;
- SaveExample(\$as, \$question) – метод збереження нового прикладу;
- SaveCategory(\$title) – метод збереження нової категорії;
- SaveDialog(\$acc1, \$acc2) – метод збереження новго діалогу;
- SaveMessageAnswer(\$mess, \$status, \$as) – метод збереження нової автоматичної відповіді;
- SaveAnswerScheme(\$cid, \$question, \$request, \$answer) – метод збереження нової схеми відповідей;
- LoadDialog(\$num) – метод завантаження даних діалогу за номером;
- LoadAnswerSchemes() – метод завантаження схем відповідей.

Дані методи використовуються в php-скриптах для отримання відповідних значень. Зокрема було розроблено скрипти:

- SendMessage.php – відправлення повідомлення;
- LoadBotMessages.php і LoadManagerMessages.php – завантаження повідомлень у спілкуванні користувача з програмою і з менеджером;
- SaveAnswerScheme.php – створення нової схеми відповіді на запитання.

4.3 Висновки за розділом 4

Розроблено програмне забезпечення формування автоматичних відповідей на запитання у вигляді вебзастосунку мовою програмування PHP.

Для демонстрації практичного застосування розробленого рішення було його інтегровано до складу вебзастосунку замовлення книг. Консультування користувачів виконається саме за даною тематикою. Описано основні програмні рішення, які було прийнято під час реалізації програми.

5 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМИ

5.1 Призначення програми

Програмне забезпечення представлене вебзастосунком, який дозволяє виконувати формування автоматичних відповідей на запитання користувача в галузі продажу книжок, реалізовувати керування даним процесом шляхом ініціалізації адміністратором через графічний інтерфейс процесів навчання та повторного навчання, заповнювати даними для навчання модель, аналізувати відповіді на запитання користувача, організовувати підтримку менеджера для користувача за необхідності.

5.2 Умови виконання програми

Для роботи сервера мають забезпечуватися наступні мінімальні вимоги:

- 2-ядерний процесор з частотою не менше 2 ГГц;
- вільний простір на жорсткому диску – 100 Гб;
- обсяг оперативного запам'ятовуючого пристрою – не менше 4 Гб.

5.3 Виконання програми

Для роботи з програмою потрібно запустити на сервері вебсервер Apache, запустити інтерпретатор мови програмування PHP та впенитися в доступ до системи керування базою даних MySQL.

Після того, як ці дії виконані, користувачі зі своїх клієнтських комп'ютерів можуть через браузер звертатися до сервера та отримувати у відповідь вебсторінку застосунку.

Для прикладу демонстрації можливостей було інтегровано розроблений програмний засіб до вебзастосунку замовлення книг.

На рис. 5.1 продемонстровано основну сторінку авторизованого користувача.

Авторизованому користувачу доступні можливості, що стосуються:

- перегляду доступних книг та їх замовлень з пункту меню «Книги»;
- перегляду власних замовлень з пункту меню «Замовлення»;
- виконання консультацій програмою або менеджером з пункту меню «Підтримка».



Рисунок 5.1 – Робота з вебзастосунком замовлення книг

Приклад взаємодії користувача з програмою з пункту меню «Підтримка» наведено на рис. 5.2.

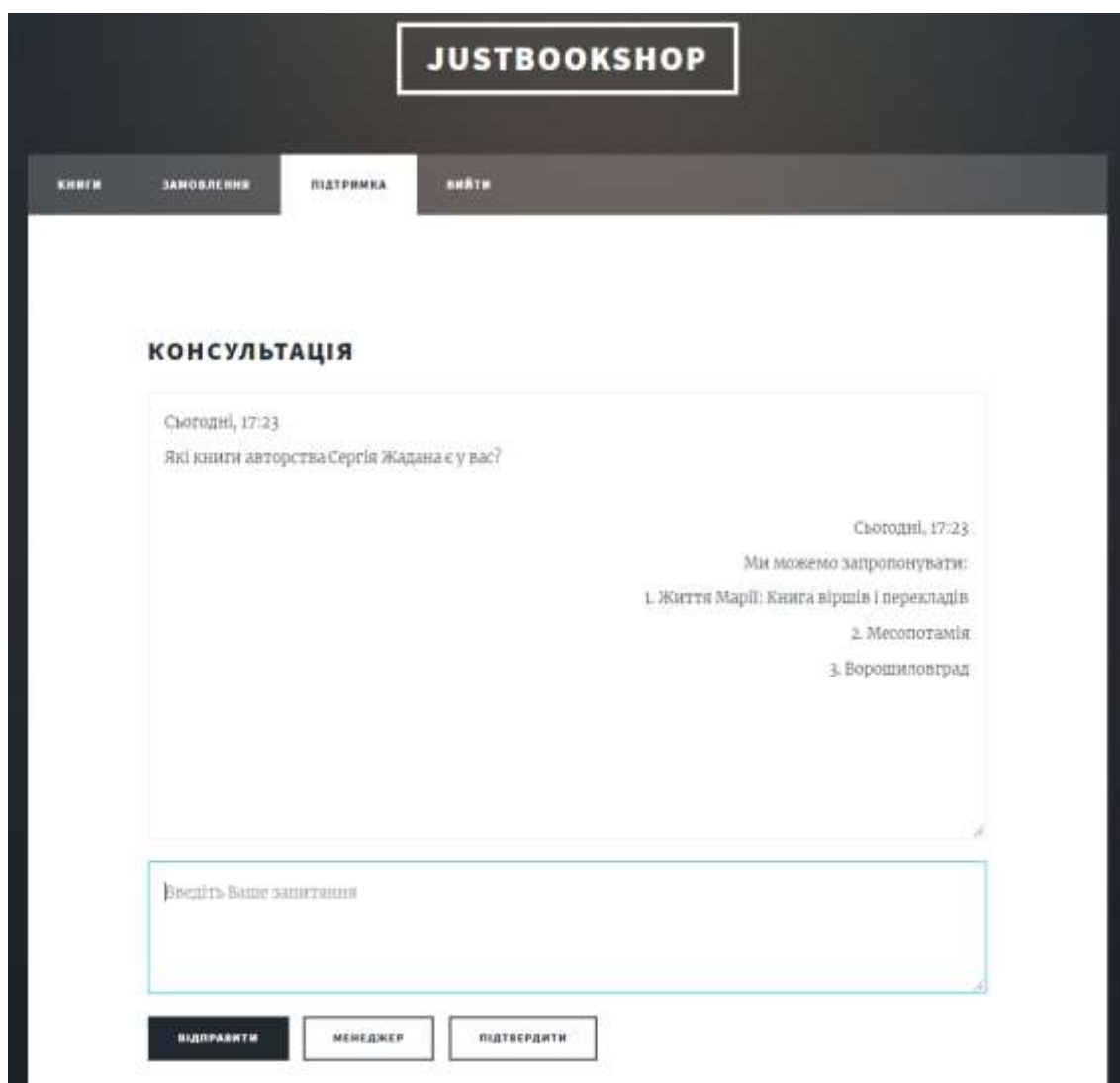


Рисунок 5.2 – Автоматизована консультація для користувача

З даної сторінки користувач може отримувати відповіді на свої запитання. Для цього він може створювати власне повідомлення, відправляти його та отримувати у великому полі відповідь на нього. Для відправки повідомлення призначена однойменна кнопка.

Кнопка «Менеджер» дозволяє користувачу за необхідності перенаправити власні повідомлення людині (менеджеру) замість програми.

Окрім того користувачу доступна кнопка «Підтвердити». Вона дозволяє виконувати дії з реалізації внесення даних в базу даних. Зокрема оформлення замовлень (рис. 5.3). У такому випадку буде запитано користувача про параметри замовлення, сформовано замовлення і додано результат до бази даних.

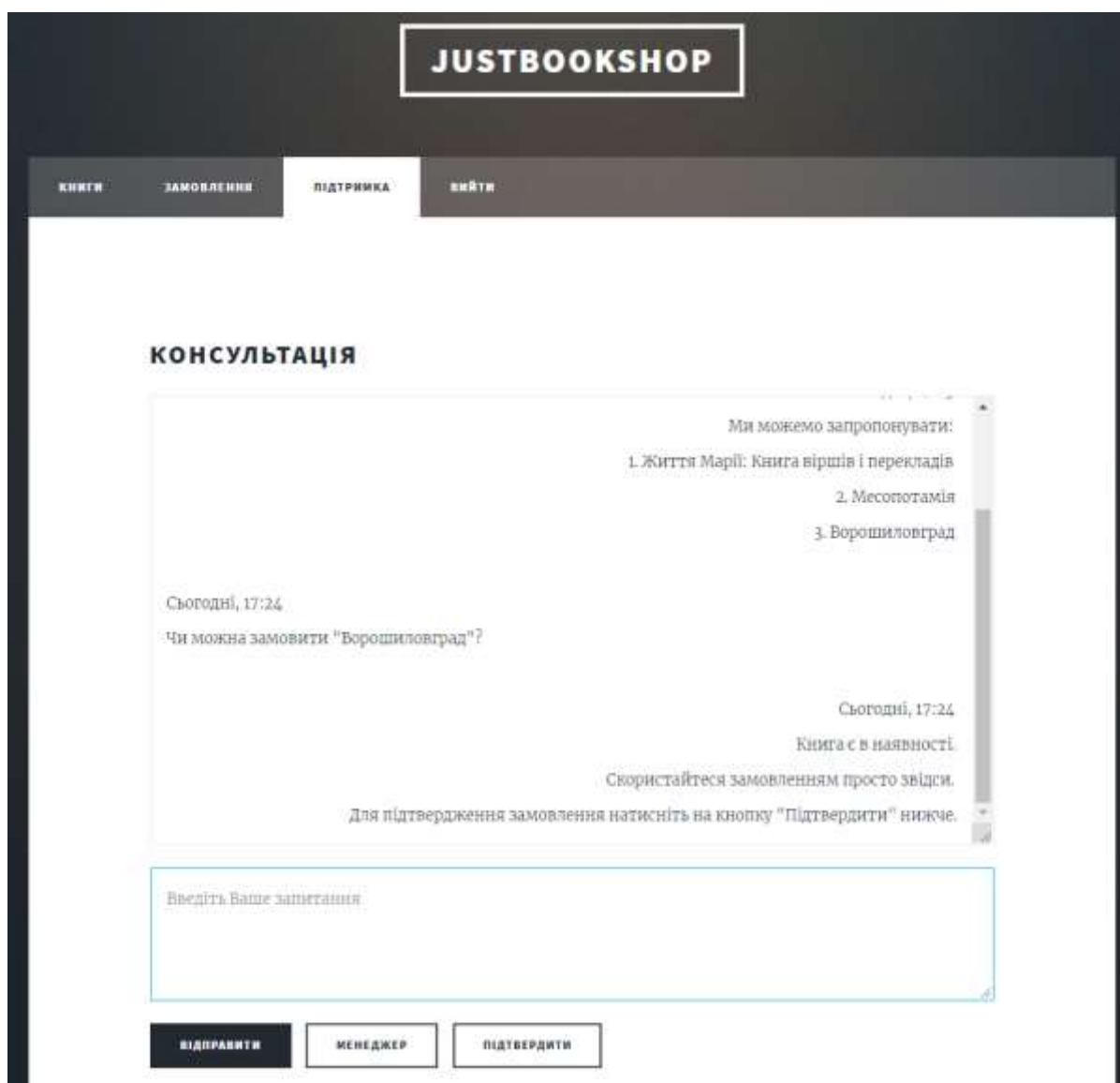


Рисунок 5.3 – Замовлення книги через консультацію

Адміністратору для роботи доступні наступні засоби:

- підтримки в режимі менеджера взаємодії з користувачем з пункту меню «Підтримка»;
- змінювання доступних схем відповідей з пункту меню «Схеми»;
- перегляду історії розібраних повідомлень і сформованих відповідей з пункту меню «Історія»;
- додавання і редагування даних про книги з пункту меню «Книги»;
- роботи над замовленнями користувачів з пункту меню «Замовлення».

На рис. 5.4 продемонстрована сторінка, з якої адміністратор може переглянути розбір запитань, який було виконано автоматично. Тобто це робота з пунктом меню «Історія». Дана вкладка необхідна для коригування дій програми та подальшого навчання на нових отриманих даних.

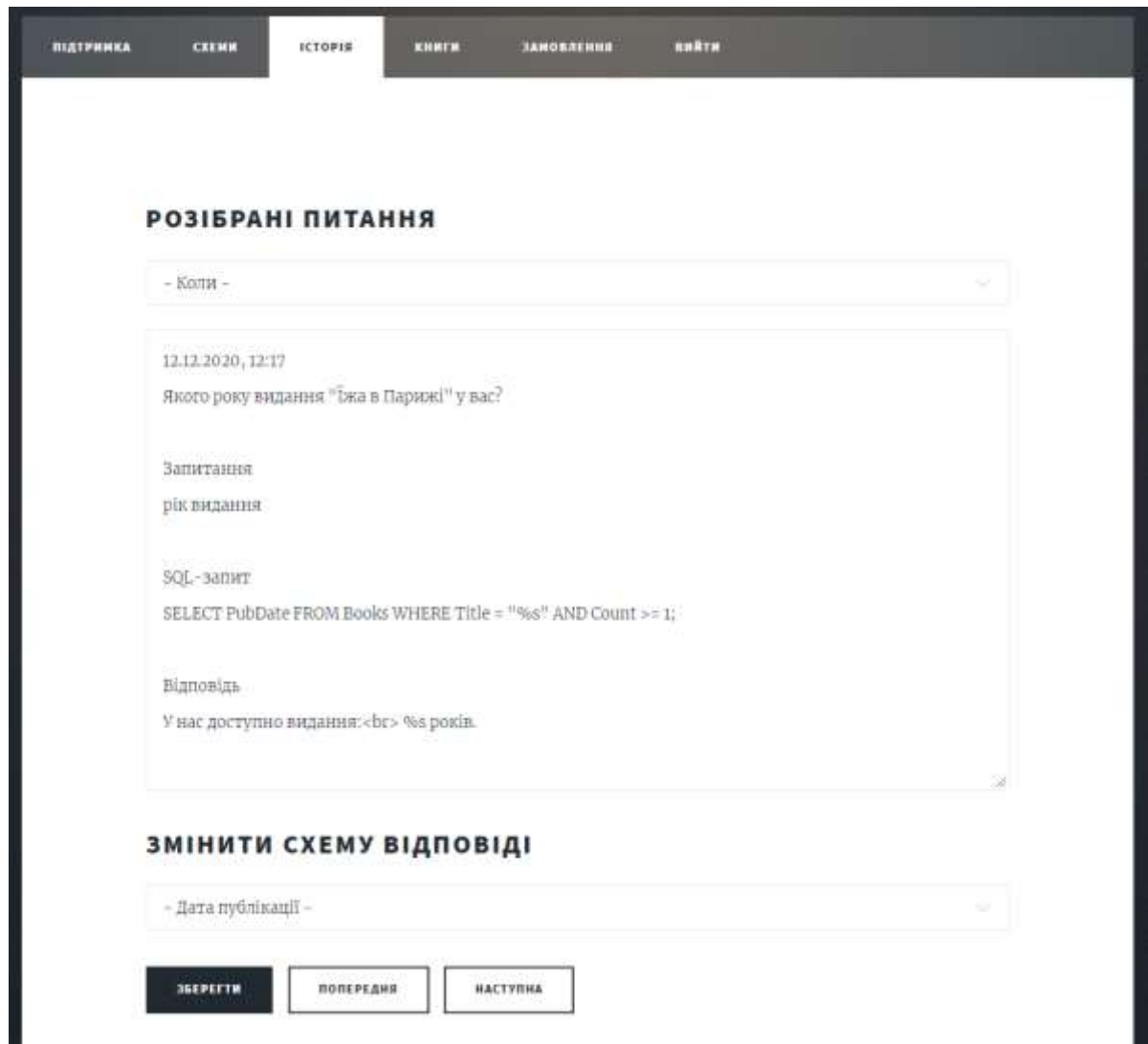


Рисунок 5.4 – Перегляд розборів запитань

Звідси адміністратору доступна можливість переглянути, яку схему було виділено за сформованим запитанням. Окрім того за необхідності він може змінити схему розбору і зберегти її за допомогою відповідної кнопки. Тоді даний приклад буде розглядатися під час навчання моделі.

На рис. 5.5 продемонстровано роботу адміністратора над схемами відповідей.

ПІДТРИМКА
СХЕМИ
ІСТОРІЯ
КНИГИ
ЗАМОВЛЕННЯ
ВИЙТИ

КАТЕГОРІЯ

- Вартість -

СХЕМА

НАЗВА

Вартість книги

ЗАПИТАННЯ

скільки коштує книга

SQL-ЗАПИТ

SELECT Price FROM Books WHERE Title = "%s";

ВІДПОВІДЬ

Ви можете придбати цю книгу в нас за %s грн.

ЗМІНИТИ

ДОДАТИ

ПРИКЛАДИ

ПОПЕРЕДНЯ

НАСТУПНА

НАВЧИТИ

Рисунок 5.5 – Визначення схеми відповіді

Звідси адміністратор може змінити існуючу схему відповіді. Для цього йому потрібно обрати з випадаючого переліку іншу схему, а потім натиснути кнопку «Змінити», щоб зберегти внесені зміни. Окрім того адміністратор може звідси ініціалізувати відкриття аналогічної сторінки для додавання нової схеми за допомогою використання кнопки «Додати», розташованої внизу. Також доступна можливість виконати первинне навчання моделі або повторне навчання за допомогою кнопки «Навчити».

Доступна також можливість переглянути приклади конкретних запитань за наведеною схемою, для цього призначена кнопка «Приклади». Зі

сторінки, що при цьому відкриється, можна як редагувати створені приклади за необхідності, так і додавати нові. Тобто цей інтерфейс дозволяє створювати по суті вибірку даних, що буде застосовуватися під час навчання моделі. Окрім того під час навчання будуть використовуватися дані коригування автоматичних відповідей на запитання, які були раніше виконані програмою, але змінені менеджером в подальшому. Все це основа для самовдосконалення.

Таким чином адміністратору доступні всі можливості керування процесом навчання моделі, виконаними фактами розбору запитань і автоматичного формування відповідей. Це головне призначення програми зі сторони адміністратора.

Для користувача головне призначення програми – можливість отримання консультацій за предметною областю з переключенням між менеджером та програмним засобом. Для переключення на сторінці «Підтримка» користувача застосовується кнопка, що має назву «Менеджер» під час стандартної роботи і дозволяє спілкуватися з програмою, а при натисканні призводить до відправлення повідомлень менеджеру, а власну назву змінює на «Бот».

Можливість роботи з книгами в даному випадку – це додаткова можливість, щоб продемонструвати інтеграцію розробленого програмного забезпечення в один із сервісів. А автоматичний розбір запитань і формування відповідей – це основне призначення розробленого програмного забезпечення. Вже під час інтеграції з сервісом можна вирішити, для чого саме такі автоматизовані дії будуть виконуватися.

Таким чином, програма може застосовуватися для підтримки роботи користувачів щодо формування автоматичних відповідей на запитання, при цьому організовуючи як безпосередньо виконання такої підтримки, так і налаштування її, охоплюючи зокрема процес створення вибірки даних, навчання моделі.

6 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА

6.1 Планування розробки програмного продукту

В роботі проведено дослідження та програмно реалізовано методи формування автоматичних відповідей на запитання.

Програмне забезпечення призначено для виконання консультацій клієнтів під час надання послуг шляхом відповіді на їх запитання природними мовами. На підприємстві до впровадження даного програмного забезпечення функції, автоматизація яких виконувалася за допомогою програмного забезпечення, виконувались консультантами. Для обслуговування клієнтів було передбачено 5 консультантів, які займалися підтримкою клієнтів. Впровадження даного програмного забезпечення повинно скоротити штат співробітників підприємства на 2 людини, оптимізувавши процес консультування клієнтів.

Весь комплекс розробки програмного продукту можна підрозділити на етапи, на кожному етапі необхідно вказати трудомісткість, виконавців і тривалість робіт [21]. Отримана інформація зведена в табл. 6.1.

Тривалість робіт визначають за формулою (6.1):

$$T_{\text{ц}} = \frac{Q}{R}, \quad (6.1)$$

де $T_{\text{ц}}$ – тривалість циклу, днів;

Q – трудомісткість, людино-дн.;

R – кількість виконавців, чол. [21].

Виходячи із завдання (час на розробку – 3 міс.) програміст працює 3 місяці із заробітною платою 12000 грн/міс., а консультант працює 1 місяць із зарплатою 10000 грн/міс.

Таблиця 6.1 – Характеристика робіт з розробки програми

Найменування роботи	Трудомісткість		Виконавці		Тривалість, днів
	людино-днів	% до підсумку	спеціальність	кількість	
1. Розробка та затвердження технічного завдання	10	12,2	Програміст, консультант	2	5
2. Концептуальне проєктування програмного забезпечення	10	12,2	Програміст, консультант	2	5
3. Розробка схеми алгоритму функціонування системи	10	12,2	Програміст, консультант	2	5
4. Проєктування загальної структури програмного забезпечення	10	12,2	Програміст	2	5
5. Проєктування схеми бази даних	5	6,1	Програміст, консультант	1	5
6. Проєктування додатків	19	23,2	Програміст	1	19
7. Розробка користувацького інтерфейсу	10	12,2	Програміст	1	10
8. Тестування системи	2	2,4	Програміст, консультант	1	2
9. Розробка експлуатаційної документації	2	2,4	Програміст	1	2
10. Впровадження	4	4,9	Програміст	1	4
Разом	82	100			62

У табл. 6.2 наведений зведений стрічковий графік планування розробки програмного продукту.

Таблиця 6.2 – Зведений графік планування розробки програмного продукту

Найменування роботи	Календарний період, дні													
	7.09	14.09	21.09	28.09	5.10	12.10	19.10	26.10	2.11	9.11	16.11	23.11	30.11	
1. Розробка та затвердження технічного завдання														
2. Концептуальне проєктування програмного забезпечення														
3. Розробка схеми алгоритму функціонування системи														
4. Проєктування загальної структури програмного забезпечення														
5. Проєктування схеми бази даних														
6. Проєктування застосунку														
7. Розробка користувацького інтерфейсу														
8. Тестування системи														
9. Розробка експлуатаційної документації														
10. Впровадження														

6.2 Визначення витрат на розробку програми

6.2.1 Розрахунок основної заробітної плати

Розрахунок основної заробітної плати провадиться за формулою (6.2):

$$ЗП_{\text{осн}} = \sum_{i=1}^n ЗП_{\text{міс}_i} \cdot T_{p_i}, \quad (6.2)$$

де $ЗП_{\text{міс}_i}$ – місячний оклад i -го працівника;

T_{p_i} – час, затрачуваний на розробку даної програми i -м працівником [21].

Отже:

$$ЗП_{\text{осн}} = (12000 \cdot 3) + (10000) \cdot 1 = 46000 \text{ грн.}$$

Результати розрахунків зведемо в табл. 6.3.

Таблиця 6.3 – Розрахунок основної заробітної плати

Посада виконавця	Чисельність, чол.	Місячний оклад, грн.	Кількість місяців роботи	Премії та доплати, грн.	Середня місячна заробітня плата, грн	Сума зарплати, грн.
Програміст	1	12000	3	—	12000	36000
Консультант	1	10000	1	—	10000	10000
Разом						46000

Додаткова заробітна плата визначається в розмірі 10% від основної зарплати штатних працівників за формулою (6.3):

$$ЗП_{\text{дод}} = ЗП_{\text{осн}} \cdot 0,1 \text{ [21]}. \quad (6.3)$$

Виходить:

$$ЗП_{\text{дод}} = 46000 \cdot 0,1 = 4600 \text{ грн.}$$

6.2.2 Відрахування на соціальне страхування

Відрахування на єдиний соціальний внесок 22 % за формулою (6.4):

$$B = (3\Pi_{\text{осн}} + 3\Pi_{\text{доп}}) \cdot 0,22 \text{ [21]}, \quad (6.4)$$

$$B = (46000 + 4600) \cdot 0,22 = 11132 \text{ грн.}$$

6.2.3 Визначення витрат на матеріали

Для розробки програми знадобилися наступні витратні матеріали:

- компакт-диск (1 шт) – 5 грн;
- папір – 500 грн.

Витрати на матеріали розраховують за формулою (6.5):

$$M = \sum_{i=1}^n (\Pi_i \cdot N_i \cdot (1 + K_{\text{т.з.}}) - \Pi_{\text{із}} \cdot N_{\text{із}}), \quad (6.5)$$

де M – витрати на матеріали, покупні напівфабрикати та комплектуючі;

$K_{\text{т.з.}}$ – коефіцієнт, що враховує транспортно-заготівельні витрати;

Π_i – ціна i -го найменування матеріалу, напівфабрикату й комплектуючого;

N_i – потреба в i -ому матеріалі, напівфабрикаті й комплектуючому;

$\Pi_{\text{із}}$ – ціна зворотних відходів i -го найменування матеріалу;

$N_{\text{із}}$ – кількість зворотних відходів i -го найменування;

n – число найменувань матеріалів, напівфабрикатів, комплектуючих [21].

Виконаємо відповідні розрахунки:

$$\Pi_{\text{із}} = 0, N_{\text{із}} = 0, K_{\text{т.з.}} = 0,05,$$

$$M = 505 \cdot (1 + 0,05) = 530,25 \text{ грн.}$$

6.2.4 Витрати на спеціальне устаткування

Амортизаційні відрахування визначають за формулою (6.6):

$$A = \Phi_6 \cdot \frac{H_a}{100}, \quad (6.6)$$

де Φ_6 – балансова вартість обчислювальної техніки, грн.;

H_a – норма амортизаційних відрахувань на повне відновлення обчислювальної техніки, % [21].

Виходить:

$$\Phi_6 = 7200 \cdot 1,05 = 7560 \text{ грн.}, H_a = 25\%,$$

$$A = 7560 \cdot 0,25 = 1890 \text{ грн.}$$

Витрати на електроенергію визначають за формулою (6.7):

$$C_e = N_n \cdot \Phi_{\text{еф}} \cdot K_{\text{зч}} \cdot K_{\text{зп}} \cdot C_e, \quad (6.7)$$

де N_n – номінальна потужність електронно-обчислювальної машини (ЕОМ), кВт;

$\Phi_{\text{еф}}$ – річний ефективний фонд часу роботи ЕОМ, машино-год.;

$K_{\text{зч}}$ – середній коефіцієнт завантаження за часом;

$K_{\text{зп}}$ – коефіцієнт завантаження за потужністю;

C_e – ціна однієї кВт·год. електроенергії, грн./(кВт·год.) [21].

Виходить:

$$C_e = 0,2 \cdot 1800 \cdot 0,75 \cdot 0,7 \cdot 2,11 = 400,30 \text{ грн.}$$

Витрати на допоміжні матеріали (спирт, серветки для протирання) становлять 49,70 грн.

Отже, витрати статті «Експлуатація устаткування» становлять:

$$E_{об} = 400,30 + 49,70 = 450 \text{ грн.}$$

Витрати за статтею «Заробітна плата обслуговуючих робітників і відрахування на єдиний соціальний внесок» обчислюються за формулою (6.8):

$$ЗП_{обсл} = \Phi ЗП_p \cdot (1 + K_{відрах}) \cdot \frac{t_{обсл}}{\Phi_{эф.обсл}}, \quad (6.8)$$

де $\Phi ЗП_p$ – річний фонд заробітної плати обслуговуючих робітників;

$K_{відрах}$ – коефіцієнт, що враховує відрахування на соціальне страхування;

$t_{обсл}$ – час протягом року, необхідний на технічне обслуговування ЕОМ;

$\Phi_{эф.обсл}$ – річний ефективний фонд часу роботи обслуговуючого персоналу [21].

Машину обслуговує один співробітник з основною заробітною платою 4800 грн./міс. (10% – премія):

$$\Phi ЗП_p = (4800 + 4800 \cdot 0,1) \cdot 12 = 63360 \text{ грн.},$$

$$ЗП_{обсл} = 63360 \cdot (1 + 0,22) \cdot 18 / 1750 = 795,08 \text{ грн.}$$

Зведемо отримані дані в табл. 6.4.

Витрати на оплату машинного часу ЕОМ для налагодження даної програми визначаються за формулою (6.9):

$$C_{мо} = P_{екс} \cdot t_{мо}, \quad (6.9)$$

де $C_{мо}$ – витрати на оплату машинного часу, грн.;

$P_{екс}$ – експлуатаційні витрати на одну годину машинного часу цієї цифрової ЕОМ, грн./ машино-год.;

$t_{\text{мо}}$ – машинний час цифрової ЕОМ для написання та налагодження даного програмного продукту, машино-год. [21].

Таблиця 6.4 – Кошторис витрат на утримання та експлуатацію устаткування

Найменування статей витрат	Сума, грн.
Амортизація устаткування	1890,00
Експлуатація устаткування (крім витрат на поточний ремонт)	450,99
Заробітна плата основних і додаткова обслуговуючих робітників з відрахуваннями на єдиний соціальний внесок	795,08
Поточний ремонт устаткування	226,8
Інші витрати	100,89
Разом	3463,76

У результаті експлуатаційні витрати на одну годину машинного часу і відповідні витрати рівні:

$$P_{\text{екс}} = 3463,76 / 1800 = 1,92 \text{ грн.},$$

$$t_{\text{мо}} = 8 \cdot 62 = 496 \text{ годин},$$

$$C_{\text{мо}} = 1,92 \cdot 496 = 954,46 \text{ грн.}$$

6.2.5 Накладні витрати

Накладні витрати становлять 40% основної заробітної плати виробничого персоналу, що дорівнює:

$$H_{\text{в}} = 46000 \cdot 0,4 = 18400 \text{ грн.}$$

6.2.6 Плановий прибуток

Результати визначення витрат зведені в табл. 6.5.

Таблиця 6.5 – Калькуляція кошторисної вартості робіт з розробки програми

Статті витрат	Сума, грн.	Питома вага до підсумку, %
Основна заробітна плата	46000,00	56,36
Додаткова заробітна плата	4600,00	5,64
Відрахування на соціальне страхування та інше	11132,00	13,64
Матеріали та комплектуючі вироби	530,25	0,65
Витрати на оплату машинного часу	954,46	1,17
Накладні витрати	18400,00	22,54
Разом	81616,71	100,00

6.3 Розрахунок капітальних вкладень

Розмір капітальних вкладень споживача програми K , пов'язаних з її розробкою та впровадженням, визначають за формулою (6.10):

$$K = K_{\text{пв}} + K_{\text{вф}}, \quad (6.10)$$

де $K_{\text{пв}}$ – передвиробничі витрати, грн.;

$K_{\text{вф}}$ – капітальні вкладення у виробничі фонди, необхідні для впровадження програми, грн. [21].

Передвиробничі витрати визначають за формулою (6.11):

$$K_{\text{ип}} = \frac{S_{\text{рп}}}{n} + K_{\text{ин.пв}}, \quad (6.11)$$

де $S_{\text{рп}}$ – сумарні витрати на розробку або придбання програмного продукту;

$K_{\text{ин.пв}}$ – інші передвиробничі витрати [21].

Виконаємо розрахунки:

$$K_{\text{ип}} = 81616,71/1 + 0 = 81616,71 \text{ грн.}$$

Питомі капітальні вкладення у виробничі фонди, що доводяться на частку даного програмного продукту, визначають за формулою (6.12):

$$K_{\text{пф}} = \frac{K_{\text{ктз}} \cdot T_{\text{ктз}}}{\Phi_{\text{еф.ктз}}}, \quad (6.12)$$

де $K_{\text{ктз}}$ – витрати на придбання, установку, монтаж і налагодження комплексу технічних засобів ($3 \cdot 7560$ грн), грн.;

$T_{\text{ктз}}$ – машинний час комплексу технічних засобів, потрібний даному споживачеві програми для завдань роботи;

$\Phi_{\text{еф.ктз}}$ – річний ефективний фонд часу роботи комплексу технічних засобів, машино-год./рік [21].

Отже, виходить:

$$K_{\text{пф}} = 3 \cdot 7560 \cdot 1800 / 1800 = 30000 \text{ грн.,}$$

$$K = 81616,71 + 22680 = 104296,71 \text{ грн.}$$

6.4 Розрахунок експлуатаційних витрат

Витрати, пов'язані з експлуатацією програми впродовж року, визначаємо за формулою (6.13):

$$I = P_{\text{екс.КТЗ}} \cdot T_{\text{КТЗ}} + \frac{S_{\text{рп}}}{n \cdot T_{\text{с}}}, \quad (6.13)$$

де $P_{\text{екс.КТЗ}}$ – експлуатаційні витрати;

n – кількість споживачів даної програми;

$T_{\text{с}}$ – термін служби даної програми до її морального зношування [21].

Отже:

$$I = 1,92 \cdot 3 \cdot 1800 + 81616,71 / (1 \cdot 5) = 26691,34 \text{ грн.}$$

6.5 Визначення економічної ефективності програмного продукту

Річна економія системи E і коефіцієнт ефективності капітальних вкладень E_{ϕ} розраховуються за формулами (6.14)-(6.15):

$$E = Ezn - I, \quad (6.14)$$

$$E_{\phi} = \frac{E}{K} [21]. \quad (6.15)$$

Тоді:

$$E = 12 \cdot 5000 \cdot 1,1 \cdot 1,22 \cdot 2 - 26691,34 = 134348,64 \text{ грн.}$$

$$E_{\phi} = 134348,64 / 104296,71 = 1,29.$$

Строк окупності капітальних вкладень $T_{\text{ок}}$ розраховується за формулою (6.16):

$$T_{\text{ок}} = \frac{K}{E} [21]. \quad (6.16)$$

Тоді:

$$T_{\text{ок}} = 104296,71 / 134348,64 = 0,78 \text{ роки.}$$

6.6 Техніко-економічні показники

Відповідно до розрахунків, у результаті впровадження програмного забезпечення вдається одержати річну економію в розмірі 134348,64 грн. Строк окупності даної системи становить 0,78 роки, що менше нормативного, який складає для соціальної сфери 6,67 року. Таким чином, впровадження програмного забезпечення економічно доцільно.

Техніко-економічні показники зведені в табл. 6.6.

Таблиця 6.6 – Техніко-економічні показники

Показники	Одиниця виміру	Варіанти	
		базовий	новий
Витрати на розробку програми	грн.		81616,71
Капітальні вкладення	грн.		104296,71
Експлуатаційні витрати	грн.	161040,00	26691,34
Річна економія	грн.		134348,64
Строк окупності програми	років		0,78
Тривалість розрахунків		більше	менше
Повнота контролю		гірше	краще
Вірогідність розрахункової інформації		менше	більше

7 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

7.1 Аналіз потенційних небезпек

У дипломній кваліфікаційній роботі магістра проводиться дослідження, розробка та програмна реалізація методів формування автоматичних відповідей на запитання.

При роботі постійно або періодично діють наступні небезпечні і шкідливі чинники:

– негативний вплив при роботі з відео дисплейними терміналами на функціональний стан людини який обумовлений електростатичними полями, які створюються моніторами з електронно-променевою трубкою, що може призвести до зниження імунітету, захворювань перефрідної нервової системи або бронхо-легеневих та шкіряних захворювань;

– можливість ураження електрострумом внаслідок небезпечного рівня напруги в електричному ланцюзі, тому у приміщені можуть виникнути аварійні ситуації: замикання, загоряння проводки, поразка електрострумом, що може привести до електротравм або летальних наслідків [22].

7.2 Заходи по забезпеченню безпеки

У розділі 4 Державних санітарних правилі норм 3.3.2.007-98 «Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» передбачені гігієнічні вимоги до організації устаткування робочих місць візуальних дисплейних терміналів (ВДТ) ЕОМ [22].

Велике значення надається ергономічним та естетичним вимогам, цього домоглися за допомогою колірною рішення приміщення й устаткування з урахуванням вимог ГОСТ 17.2.032-78 «Робоче місце при виконанні робіт сидячи» – у приміщеннях стіни пофарбовані фарбами пастельних тонів [23].

При хронічних скаргах працюючих з ВДТ ЕОМ і персональними ЕОМ на зорове втомлення, незважаючи на дотримання санітарно-гігієнічних вимог до режимів праці і відпочинку, а також застосування засобів локального захисту очей – допускаються індивідуальних підхід до обмеження часу робіт з ВДТ, зміни характеру праці, чергування з іншими видами діяльності, не пов'язаними з ВДТ [23]

Згідно правил улаштування електроустановок (ПУЕ) приміщення виробництва відноситься до категорії «Приміщення без підвищеної небезпеки» [24].

Відповідно до ПУЕ в електроустановках до 1000В мінімальне значення опору ізоляції силових і освітлювальних електропроводок на ділянці між суміжними запобіжниками або за останніми запобіжниками між будь-яким проведенням і землею становить не менш 0,5МОм [25].

Відповідно до ПУЕ захисне заземлення 4 Ом застосовується у всіх електроустановках змінного струму під напругою 220В і вище й постійного струму 440В і вище, у приміщеннях з підвищеною небезпекою заземлення використовується в електроустановках змінного струму напругою 42В та постійного струму 110В, захисному заземленню підлягають корпуси трансформаторів, оболонки кабелів, металеві огороження електроустановок відповідно до ГОСТ 12.1.030-81 «Електробезпека. Захисне заземлення. Занулення» [26].

В якості заходів захисту запропоновано:

- щоденно в приміщенні з персональними комп'ютерами (ПК) проводити вологе прибирання;
- установити нейтралізатори статичної електрики [27].

7.3 Заходи по забезпеченню промислової санітарії та гігієни праці

Оптимальні норми температури повітря в робочій зоні виробничих приміщень у холодний і перехідний періоди року при II категорії робіт згідно

Державних санітарних норм 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень – 20...23 °С, відносна вологість – 60...40%, швидкість руху повітря – 0.2 м/сек. У теплий період року температура – 22...25 °С, відносна вологість – 60...40 %, швидкість руху повітря – 0.2 м/сек [28].

Загальне освітлення варто виконувати у виді суцільних чи переривчастих ліній світильників, розташованих збоку від робочих місць.

Розрахуємо систему загального рівномірного штучного освітлення в адміністративному приміщенні. Приймаємо розміри приміщення довжина $A = 11$ м, ширина $B = 7$ м, висота $H = 5$ м. Висота робочої поверхні складає $h = 0,7$ м.

Коефіцієнти відбиття поверхонь приміщення: колір стелі, стін - сірий, незначне пиловиділення, $\rho_c = 50\%$; $\rho_{ст} = 30\%$; $\rho_n = 10\%$.

Розряд зорової роботи – III б.

Згідно Державних будівельних норм В.2.5-28-2006 «Природне і штучне освітлення. Норми проектування» рівень нормованого загального освітлення в приміщенні нормуються як $E_n = 400$ Лк.

Обираємо світильник типу ЛПП, призначений для приміщень висотою до 6 м. IP=65 – ступінь захисту світильника, $[L/h] = 1,4$ – числове значення коефіцієнта світильника.

Коефіцієнт запасу k_z враховує зниження рівня освітленості з часом в результаті забруднення та старіння ламп, світильників і поверхонь приміщення, приймається в залежності від виробничих умов. Приймаємо $k_z = 1,6$.

Коефіцієнт нерівномірності (мінімального) освітлення z (відношення середньої освітленості до мінімальної освітленості), для люмінесцентних ламп низького тиску, як правило дорівнює $z = 1,1$.

Розрахуємо кількість рядів світильників у приміщенні на основі формули (7.1):

$$N_p = \frac{B}{(H-h_p) \cdot [L/h]} = \frac{7}{(5-0,7) \cdot 1,4} = 1,16 \approx 2 \text{ ряди світильників.} \quad (7.1)$$

Визначимо максимально припустиму відстань між рядами світильників за формулою (7.2):

$$L_{\max} = \frac{B}{N_p}, \quad (7.2)$$

де B – ширина приміщення;

N_p – кількість рядів.

Розрахуємо:

$$L_{\max} = \frac{7}{2} = 3,5 \text{ м.}$$

Розраховуємо висоту підвісу світильника над робочою поверхнею за формулою (7.3):

$$h = \frac{L_{\max}}{[L/h]}, \quad (7.3)$$

$$h = \frac{3,5}{1,4} = 2,5 \text{ м.}$$

Знайдемо висоту звисання світильника від стелі за формулою (7.4):

$$h_3 = H - h_p - h, \quad (7.4)$$

де H – висота приміщення;

h_p – висота робочої поверхні;

h – висота підвісу світильника над робочою поверхнею.

Розрахуємо:

$$h_3 = 5 - 0,7 - 2,5 = 1,8 \text{ м.}$$

Чисельне значення індексу приміщення визначаємо за рівнянням (7.5):

$$i = \frac{AB}{h(A+B)}, \quad (7.5)$$

де A – довжина приміщення, м;

B – ширина приміщення, м;

h – висота розміщення світильників над робочою поверхнею, м.

Розрахуємо:

$$i = \frac{11 \cdot 7}{2,5 \cdot (11 + 7)} = 1,71.$$

Значення коефіцієнта використання світлового потоку η вибирається в залежності від виду джерела світла, типу обраного світильника, коефіцієнтів відбиття поверхонь приміщення (ρ_c , $\rho_{ст}$, $\rho_{п}$) та індексу приміщення. В нашому випадку $\eta = 54\%$.

Визначаємо сумарний світловий потік освітлювальної установки у даному виробничому приміщенні на основі формули (7.6):

$$\Phi_{\Sigma} = \frac{E_n \cdot S \cdot k_s \cdot z}{\eta}, \quad (7.6)$$

де Φ_{Σ} – розрахункове значення сумарного світлового потоку у приміщенні, лм.

Розрахуємо:

$$\Phi_{\Sigma} = \frac{400 \cdot 11 \cdot 7 \cdot 1,6 \cdot 1,1}{0,54} = 100385 \text{ лм.}$$

Визначаємо умовну загальну кількість світильників у приміщенні за формулою (7.7):

$$N^* = A \cdot \frac{B}{L_{\max}^2}, \quad (7.7)$$

$$N^* = \frac{11 \cdot 7}{3,5 \cdot 3,5} = 6,3 \approx 8 \text{ шт.}$$

Розрахуємо світловий потік умовного джерела світла за формулою (7.8):

$$\Phi_{\text{л}}^* = \frac{\Phi_{\Sigma}}{N_{\text{л}}}, \text{ лм}, \quad (7.8)$$

де Φ_{Σ} – розрахункове значення сумарного світлового потоку у приміщенні, лм;

$N_{\text{л}}$ – загальна кількість ламп у приміщенні, шт.

Розрахуємо загальна кількість ламп у приміщенні за формулою (7.9):

$$N_{\text{л}} = N^* \cdot n, \quad (7.9)$$

де n – кількість ламп у світильнику (в ЛПП світильнику знаходиться 2 лампи).

Розрахуємо:

$$N_{\text{л}} = 8 \cdot 2 = 16 \text{ шт.}$$

Тоді

$$\Phi_{\text{л}} = \frac{100385}{16} = 6274 \text{ лм.}$$

Обираємо лампу ЛБ із $P_{\text{л}}=40$ Вт, $\Phi_{\text{л}} = 5800$ лм за формулою (7.10):

$$m = \frac{\Phi_{\text{л}}^*}{\Phi_{\text{л}}}, \quad (7.10)$$

$$m = \frac{6274}{5800} = 1,081.$$

Визначаємо оптимальну (фактичну) кількість світильників у приміщенні за формулою (7.11):

$$N = N^* \cdot m, \quad (7.11)$$

$$N = 8 \cdot 1,081 \approx 8 \text{ шт.}$$

Фактична кількість ламп у приміщенні за формулою (7.12):

$$N_{\text{фл}} = N_{\text{ф}} \cdot n, \quad (7.12)$$

$$N_{\text{фл}} = 8 \cdot 2 = 16 \text{ шт.}$$

Визначаємо загальну розрахункову освітленість E_p у приміщенні, що створюється при застосуванні стандартних ламп, за формулою (7.13):

$$E_p = \frac{\Phi_{\text{л}} N_{\text{фл}} \eta}{S k_z z}, \quad (7.13)$$

$$E_p = \frac{5800 \cdot 16 \cdot 0,54}{11 \cdot 7 \cdot 1,6 \cdot 1,1} = 370 \text{ лк.}$$

Тобто виконується умова $E_p = (-10\% \dots +20\%) \cdot E_n$, лк.

Таким чином, на основі розрахунків для освітлення приміщення площею 77 м^2 встановлено 8 світильників, розташованих у 2 рядах по 4 світильники у кожному ряді, що дозволяє отримати освітленість на робочому місці 400лк відповідно до Державних будівельних норм В.2.5-28-2006 «Природне і штучне освітлення. Норми проектування».

7.4 Заходи по забезпеченню пожежної безпеки

Заходи з протипожежної безпеки розроблені для виробничого приміщення обладнаного ПК з ВДТ розроблені згідно вимог Нормативно-правового акту з пожежної безпеки А.01.001-2014 «Правила пожежної безпеки в Україні» [29].

Виходячи з аналізу речовин та матеріалів, які використовуються при роботі у приміщенні, обладнаному ПК з ВДТ:

- згідно ГОСТ 27331-87 «Пожежна техніка. Класифікація пожеж» [30] у приміщенні, обладнаному ПК з ВДТ можлива пожежа класу – А та Е;

- відповідно до вимог ДСТУ Б В.1.1-36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою» [31], воно належить до категорії «Д» з пожежної небезпеки – простір у приміщенні, у якому перебувають тверді горючі речовини та матеріали [30]-[32]

Згідно вимог Державних будівельних норм В.2.5-56:2014 «Системи протипожежного захисту», в приміщенні обладнаному ПК з ВДТ встановлена система пожежної й охоронної сигналізації «Фотон-ОФІС», яка забезпечує виявлення теплових і димових ознак пожежі і місця виникнення пожежі з точністю до місця розміщення датчика [33].

Оскільки приміщення (дослідницької лабораторії, конструкторського бюро, тощо) що обладнане ПК з ВДТ має площу 77 м², тому відповідно до вимог п. 3.8 розділу «Типові норми належності вогнегасників» ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників. Загальні технічні вимоги» для гасіння електроустановок, що знаходяться під напругою, передбачені вуглекислотні вогнегасники типу ВВК-3,5 у кількості 4 штуки (з розрахунку один вогнегасник с величиною заряду вогнегасної речовини 3 кг і більше, на 20 м² площі приміщення), відстань між вогнегасниками та місцями можливих загорянь не перевищує 10 м [34].

7.5 Заходи по забезпеченню безпеки у надзвичайних ситуаціях

Укриття населення в захисних спорудах — це комплекс заходів із завчасним будівництвом захисних споруд, а також пристосуванням наявних приміщень для захисту населення та підтримання їх у готовності до використання [35].

Укриття населення в захисних спорудах є надійним способом захисту від уражаючих факторів ядерної, хімічної, бактеріологічної, звичайної зброї, у разі аварій і деяких стихійних лих (ураганів, снігових заносів) [36].

Відповідно до вимог п. 3.8 розділу «Типові норми належності вогнегасників» ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників. Загальні технічні вимоги» для гасіння електроустановок, що знаходяться під напругою, передбачені вуглекислотні вогнегасники типу ВВК-3,5 у кількості 4 штуки.

ВИСНОВКИ

У дипломній кваліфікаційній роботі магістра розглянуто проблему формування автоматичних відповідей на запитання, що виникає під час створення програмних помічників для роботи з користувачами вебсайтів та віртуальних асистентів.

Проаналізовано метод опорних векторів і особливості його застосування для задачі класифікації. Представлено власний метод формування автоматичних відповідей на запитання, наведено схему роботи методу.

Сформульовано функціональні вимоги до програми, побудовано діаграми прецедентів для користувача та адміністратора. Обрано мову програмування PHP для реалізації програм та систему MySQL для реалізації бази даних. Визначено структуру бази даних.

Розроблено програмне забезпечення формування автоматичних відповідей на запитання. Програмне забезпечення представлене вебзастосунком, який дозволяє виконувати формування автоматичних відповідей на запитання користувача в галузі продажу книжок, реалізовувати керування даним процесом шляхом ініціалізації адміністратором через графічний інтерфейс процесів навчання та повторного навчання, заповнювати даними для навчання модель, аналізувати відповіді на запитання користувача, організовувати підтримку менеджера для користувача за необхідності.

Наукова новизна роботи полягає в тому, що в ній запропоновано метод формування автоматичних відповідей на запитання, що полягає в початковій класифікації запитань за категоріями, подальшій уточнюючій класифікації за конкретними схемами відповідей на запитання на основі використання методу опорних векторів, кодуванні запитання та формування відповіді на основі виконання запиту до бази даних, що дозволяє відповідати на запитання за предметною областю без участі менеджерів компанії.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Yse, D.L. Your Guide to Natural Language Processing (NLP) [Electorinic resource] : How machines process and understand human language / D.L. Yse. – Access mode : <https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1>.
2. Провотар, А.И. Особенности и проблемы виртуального общения с помощью чат-ботов [Текст] / А.И. Провотар, К.А. Ключко // Наукові праці ВНТУ. – 2013. – № 3. – С. 63-67.
3. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика [Текст] : учеб. пособие / Е. И. Большакова, Э. С. Клышинский, Д. В. Ландэ, А. А. Носков, О. В. Пескова, Е. В. Ягунова. – М. : МИЭМ, 2011. – 272 с.
4. What is Natiural Language Processing? [Electorinic resource]. – Access mode : https://www.sas.com/ru_ru/insights/analytics/what-is-natural-language-processing-nlp.html.
5. Garbade, M. J. A Simple Introduction to Natural Language Processing [Electorinic resource] / M. J. Garbade. – Access mode : <https://becominghuman.ai/a-simple-introduction-to-natural-language-processing-ea66a1747b32>.
6. Касевич, В.Б. Элементы общей лингвистики [Текст] / В. Б. Касевич. – М. : Наука, 1977. – 183 с.
7. Manning, Ch. D. Foundations of Statistical Natural Language Processing [Text] / Ch. D. Manning, H. Schütze. – Massachusetts : MIT Press, 1999. – 620 p.
8. Шевченко, Н. В. Основы лингвистики текста [Текст] : учебное пособие / Н. В. Шевченко. – М. : Приор-издат, 2003. – 160 с.
9. Question Answering Systems [Text] : Survey and Trends / Abdelghani Bouziane, Djelloul Bouchina, Noureddine Doumi and Mimoun Malki // Procedia Computer Science. – 2017. – № 73. – Pp. 366-375.

10. Building an application of question answering system from scratch [Electronic resource]. – Access mode : <https://towardsdatascience.com/building-an-application-of-question-answering-system-from-scratch-2dfc53f760aa>.

11. Groot, J. What is Data Classification? A Data Classification Definition [Electronic resource] / J. Groot. – Access mode : <https://digitalguardian.com/blog/what-data-classification-data-classification-definition>.

12. Optimal Features Selection for Designing a Fault Diagnosis System [Text] / Mohammadhamed Ardakani, Mahdieh Askarian, Ahmed Shokry, Gerard Escudero, Moises Graells, Antonio Espuna // Computer Aided Chemical Engineering. – 2016. – Volume 38. – Pp. 1111-1116.

13. An Introduction to Support Vector Machines (SVM) [Electronic resource]. – Access mode : <https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/>.

14. Gandhi, R. Support Vector Machine – Introduction to Machine Learning Algorithms [Electronic resource] / R. Gandhi. – Access mode : <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>.

15. Multiclass Classification Using Support Vector Machines [Electronic resource]. – Access mode : <https://www.baeldung.com/cs/svm-multiclass-classification>.

16. Roznovsky, A. Why Use PHP? Main Advanatges and Disadvantages [Electronic resource] / A. Roznovsky. – Access mode : <https://light-it.net/blog/why-use-php-main-advantages-and-disadvantages/>.

17. Зандстра, М. PHP. Объекты, шаблоны и методики программирования [Текст] / М. Зандстра. – СПб. : Диалектика, 2015. – 576 с.

18. Ларри, У. PHP и MySQL [Электрон. ресурс] : создание интернет-магазинов. – 2-е изд. – СПб. : Диалектика, 2015. – 544 с.

19. MySQL Features [Electronic resource]. – Access mode : <https://www.javatpoint.com/mysql-features>.

20. Massively | HTML5 UP [Electronic resource]. – Access mode : <https://html5up.net/massively>.

21. Методичні вказівки по економічному обґрунтуванню дипломних проєктів для студентів спеціальностей 8.05010301 «Програмне забезпечення автоматизованих систем» та 8.05010201 «Інформаційні технології проєктування» для усіх форм навчання [Текст] / В.В. Остапенко, Є.М. Сердюк. – Запоріжжя : ЗНТУ, 2015. – 22 с.

22. ДСанПіН 3.3.2.007-98 Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин [Текст]. – К. : МОЗ, 1998. – 10 с.

23. Правила улаштування електроустановок [Текст] : Нормативне виробничо-практичне видання : вид. 5-те, перероблене й доповнене : затв. М-вом енергетики та вугільної промисловості України 20.06.14 : введення в дію з 20.11.14. – Х. : Міненерговугілля України, 2014. – 793 с.

24. ГОСТ 17.2.032-78 Рабочее место при выполнении работ сидя. Система стандартов безопасности труда [Текст] : Сб. ГОСТов. – М. : ИПК Издательство стандартов, 2001. – 7 с.

25. ГОСТ 12.1.030-81 «Электробезопасность. Защитное заземление. Зануление» [Текст]. – М. : ИПК Издательство стандартов, 1981. – 8 с.

26. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень [Текст]. – Харків : Форт, 2000. – 24 с.

27. ДБН В.2.5-28:2018. Природне та штучне освітлення. Посібник [Текст] : проблеми природного і штучного освітлення. – К. : Етін, 2019. – 180 с.

28. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку [Текст]. – К. : МОЗ України, 1999. – 24 с.

29. ДБН В.1.1-7:2016 Пожежна безпека об'єктів будівництва. Загальні вимоги [Текст]. – Київ : Міністерство регіонального розвитку, будівництва та житлово-комунального господарства України, 2017. – 39 с.

30. СНиП 2.09.02-85 Производственные здания [Текст]. – М. :

Госстрой, 1991. – 17 с.

31. НПАОП 40.1-1.32-01 Правила будови електроустановок. Електрообладнання спеціальних установок [Текст]. – Харків : Індустрія, 2010. – 118 с.

32. Кодекс цивільного захисту України [Текст] : [прийнятий ВР України 02.10.2012 р. № 5403-VI] : офіц. текст : станом на 06.12.2012 р. / ВР України. – К. : Алерта, 2012. – 120 с.

33. Жидецький, В. Ц. Основи охорони праці [Текст] : підручник / В. Ц. Жидецький. – 5-те вид., доп. – К. : Знання, 2014. – 373 с.

34. Катренко, Л. А. Охорона праці. Курс лекцій. Практикум [Текст] : навчальний посібник / Л. А. Катренко, Ю. В. Кіт, І. П. Піскун. – 2-ге вид., стер. – Суми : ВТД «Університетська книга», 2007. – 103 с.

35. Основи охорони праці [Текст] : підручник / В. В. Березуцький, Т. С. Бондаренко, Г. Г. Валенко та ін.; за ред. проф. В. В. Березуцького. – Х. : Факт, 2005. – 327 с.

36. Шоботов, В. М. Цивільна оборона [Текст] : Навчальний посібник / В. М. Шоботов. – Вид. 2-ге, перероб. – К. : Центр навчальної літератури, 2006. – 438 с.

ДОДАТОК А
Технічне завдання

Вступ

Цінність роботи має бути виражена в створенні теоретичного базису розробки та програмній реалізації її, щоб в сукупності це дозволило розв'язати проблему формування автоматичних відповідей на запитання.

A.1 Підстави для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу магістра за темою “Дослідження та програмна реалізація методів формування автоматичних відповідей на запитання”, затверджене наказом № 301 від 3 листопада 2020 р. за Національним університетом «Запорізька політехніка».

A.2 Призначення розробки

Програма призначена для того, щоб виконувати формування автоматичних відповідей на запитання користувача в галузі продажу книжок, реалізовувати керування даним процесом шляхом ініціалізації адміністратором через графічний інтерфейс процесів навчання та повторного навчання, заповнювати даними для навчання модель, аналізувати відповіді на запитання користувача, організовувати підтримку менеджера для користувача за необхідності.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Функціональні характеристики, які мають виконуватися в програмі:

- пересилання повідомлень користувача на сервер і збереження їх в базі даних;

- формування автоматичних відповідей на запитання користувачів;
- відображення автоматичних відповідей користувачу;
- додавання схем розбору відповідей на запитання;
- змінювання схем розбору відповідей на запитання;
- додавання прикладів для обраної схеми розбору;
- ініціалізація дій за запитанням;
- перегляд історії автоматичних відповідей;
- коригування результатів класифікації відповідей;
- додавання категорій відповідей;
- виконання навчання;
- виконання повторного навчання;
- авторизація користувачів;
- реєстрація користувачів.

A.3.2 Вимоги до надійності

Надійність роботи програми має забезпечуватися шляхом використання трьох напрямків:

- підтримки надійної роботи сервера;
- забезпечення введення адміністратором коректних SQL-запитів;
- перевіркою інших введених користувачем і адміністратором даних.

A.3.3 Умови експлуатації

Для експлуатації програми від користувача ніяких умінь окрім вміння працювати з комп'ютером не вимагається. Він повинен знати адресу вебсервера.

Для адміністратора вимагається розуміння відповідної природної мови, вміння створювати SQL-запити (якщо схеми відповідей будуть

змінюватися), а також базові принципи роботи запропонованого в роботі методу.

А.3.4 Вимоги до складу та параметрів технічних засобів

Для роботи сервера мають забезпечуватися наступні мінімальні вимоги:

- 2-ядерний процесор з частотою не менше 2 ГГц;
- вільний простір на жорсткому диску – 100 Гб;
- обсяг оперативного запам'ятовуючого пристрою – не менше 4 Гб.

До клієнта окрім вимог щодо встановленого сучасного веббраузера не висувається.

А.3.5 Вимоги до транспортування та збереження

Збереження програми має виконуватися на сервері з постійним забезпеченням резервування вебзастосунку самого та бази даних зокрема. Окрім того остання версія програми має зберігатися на переносних носіях для забезпечення накопичення версій.

А.3.6 Вимоги до програмної документації

Програмний документ «Текст програми» повинен містити фрагмент етксту програми і включатися в додаток до пояснювальної записки.

А.3.7 Вимоги до маркування та упакування

Якщо програма зберігається на переносному носії, то він має обов'язково зберігатися в чохлі, на якому має бути зафіксовано назву роботи.

А.4 Порядок контролю та приймання

Робота має бути виконана в строк згідно календарного плану. Контроль відбувається шляхом зіставлення отриманих результатів з зазначеними в примітках календарного плану показниками та даним технічним завданням.

ДОДАТОК Б
Текст програми

Б.1 Текст файлу DataBaseObtainer.php

```
//клас з'єднання з базою даних
<?php

class DataBaseObtainer {

    function __construct() {

    }

    function __destruct() {

    }

    public function connect() {
        require_once 'settings.php';
        $mc = mysqli_connect(SERVADDR, SERVLOGIN, SERVPASS,
SERVDB);

        return $mc;
    }

    public function close() {
        mysqli_close();
    }
}
?>
```

Б.2 Текст файлу DataBaseProcessor.php

```
//клас виконання запитів до бази даних
<?php

class DataBaseProcessor {

    private $cdbase;

    //конструктор для створення з'єднання на основі класу
    DataBaseObtainer
    function __construct() {
        include_once './DataBaseObtainer.php';
        $this->cdbase = new DataBaseObtainer ();
    }
}
```

```

        $this->cdbase->connect();
    }

    function __destruct() {}

    //збереження даних акаунта в базі даних
    public function SaveAccount($name, $password, $email,
    $createddate, $type) {
        $v = mysqli_query($this->cdbase->connect(),
        "INSERT INTO Account(Name, Password, Email, CreatedDate,
    Type) "+
        "VALUES('$name', '$password', '$email', NOW(), '$type')");
        return v;
    }

    //отримання переліку діалогів, які велися користувачем з
    заданим номером з менеджером
    public function GetDialogsWithManager($num) {
        $v = mysqli_query($this->cdbase->connect(),
        "SELECT * FROM Dialog "+
        "INNER JOIN Account ON Dialog.AccountId2=Account.ID "+
        "WHERE Account.Type=2 ");
        return v;
    }

    //отримання переліку діалогів, які велися користувачем з
    заданим номером з програмою
    public function GetDialogsWithBot($num) {
        $v = mysqli_query($this->cdbase->connect(),
        "SELECT * FROM Dialog "+
        "INNER JOIN Account ON Dialog.AccountId2=Account.ID "+
        "WHERE Account.Type=3 ");
        return v;
    }

    //оновлення електронної поштової адреси користувача
    public function UpdateAccountEmail($name, $email) {
        $v = mysqli_query($this->cdbase->connect(),
        "UPDATE Account SET Email='$email' "+
        "WHERE Name = '$name'");
    }

```

```

        return v;
    }

    //зміна типу акаунта користувача
    public function ChangeAccountType($name, $type) {
        $v = mysqli_query($this->cdbase->connect(),
"UPDATE Account SET Type='$type' "+
"WHERE Name = '$name'");
        return v;
    }

    //зміна паролю акаунта
    public function UpdateAccountPassword($name, $pass) {
        $v = mysqli_query($this->cdbase->connect(),
"UPDATE Account SET Password='$pass' "+
"WHERE Name = '$name'");
        return v;
    }

    //пошук акаунту за електронною поштою користувача
    public function GetAccountByEmail($mail) {
        $v = mysqli_query($this->cdbase->connect(),
"SELECT * FROM Account "+
"WHERE Email = '$mail'");
        return $v;
    }

    //отримання даних категорії за її номером
    public function GetCategory($num) {
        $v = mysqli_query($this->cdbase->connect(),
"SELECT * FROM Category WHERE Id = '$num'");
        return $v;
    }

    //отримання даних схеми відповіді на запитання за номером
    public function GetAnswerScheme($num) {
        $v = mysqli_query($this->cdbase->connect(),
"SELECT * FROM AnswerScheme WHERE Id = '$num'");
        return $v;
    }

```

```
}
```

//отримання автоматизованої відповіді на запитання за номером

```
public function GetMessageAnswer($num) {
    $v = mysqli_query($this->cdbase->connect(),
"SELECT * FROM MessageAnswer WHERE Id = '$num'");
    return $v;
}
```

//отримання даних акаунту за логіном

```
public function GetAccountOnName($name) {
    $v = mysqli_query($this->cdbase->connect(),
"SELECT * FROM Account WHERE Name = '$name'");
    return $v;
}
```

//визначення типу облікового запису

```
public function GetAccountType($name) {
    $v = mysqli_query($this->cdbase->connect(),
"SELECT Type FROM Account WHERE Name = '$name'");
    return $v;
}
```

//завантаження переліку всіх акаунтів

```
public function LoadAccounts() {
    $v = mysqli_query($this->cdbase->connect(),
"SELECT * FROM Account ");
    return $v;
}
```

//перевірка існування акаунта з заданим логіном

```
public function IfAccount($name) {
    $v = mysqli_query($this->cdbase->connect(),
"SELECT * from Account WHERE Name = '$name'");
    $num = mysqli_num_rows($v);
    if ($num>0) {
        return true;
    } else {
```

```

        return false;
    }
}

//збереження нового повідомлення в базі даних
public function SaveMessage($dialog, $author,
$information, $when) {
    $v = mysqli_query($this->cdbase->connect(),
    "INSERT INTO Message(DialogId, Author, Information, When) "+
    "VALUES('$dialog', '$author', '$information', '$when')");
    return v;
}

//збереження нового повідомлення в базі даних з поточним
часом
public function SaveMessageNow($dialog, $author,
$information) {
    $v = mysqli_query($this->cdbase->connect(),
    "INSERT INTO Message (DialogId, Author, Information, When) "+
    "VALUES('$dialog', '$author', '$information', '$when',
NOW())");
    return v;
}

//завантаження всіх повідомлень з бази даних
public function LoadMessages() {
    $v = mysqli_query($this->cdbase->connect(),
    "SELECT * FROM Message ");
    return v;
}

//авторизація акаунта, в результаті повертається його
номер
public function AccountCheck($name, $pass) {
    $v = mysqli_query($this->cdbase->connect(),
    "SELECT Id from Account "+
    "WHERE Name = '$name' AND WHERE Password = '$pass'");
    $num = mysqli_num_rows($v);
    if ($num>0) {
        return v['Id'];
    }
}

```

```

    } else {
        return v['Id'];
    }
}

```

//завантаження повідомлень діалогу з заданим номером

```

public function LoadDialogMessages($id) {
    $v = mysqli_query($this->cdbase->connect(),
"SELECT * FROM Message WHERE DialogId = '$id' ");
    return $v;
}

```

номером

```

public function LoadDialogMessage($id) {
    $v = mysqli_query($this->cdbase->connect(),
"SELECT * FROM Message WHERE Id = '$id' ");
    return $v;
}

```

//завантаження всіх автоматичних відповідей на запитання

```

public function LoadMessageAnswers() {
    $v = mysqli_query($this->cdbase->connect(),
"SELECT * FROM MessageAnswer ");
    return $v;
}

```

на запитання

```

public function LoadFullMessageAnswers() {
    $v = mysqli_query($this->cdbase->connect(),
"SELECT MessageAnswer.Id, MessageAnswer.Status, "+
"MessageAnswer.AnswerSchemeId, Message.DialogId, "+
"Message.Information, Message.When FROM MessageAnswer "+
"INNER JOIN Message ON MessageAnswer.MessageId = Message.Id ");
    return $v;
}

```

//завантаження всіх повідомлень з діалогу з заданим номером

```

        public function LoadMessages($num) {
            $v = mysqli_query($this->cdbase->connect(),
"SELECT * FROM Message WHERE DialogId = $num");
            return v;
        }

        //завантаження повідомлень користувача з заданим номером
        public function LoadUserMessages($num) {
            $v = mysqli_query($this->cdbase->connect(),
"SELECT * FROM Message "+
"INNER JOIN Dialog ON Message.DialogId=Dialog.Id "+
"WHERE Dialog.AccountId1 = $num");
            return v;
        }

        //редагування повідомлення
        public function UpdateMessage($id, $author, $information,
$when) {
            $v = mysqli_query($this->cdbase->connect(),
"UPDATE Message "+
"SET Author='$author', Information='$information', "+
"When='$when' "+
"WHERE Id = '$id'");
            return v;
        }

        //оновлення тексту повідомлення
        public function ChangeMessageText($id, $information) {
            $v = mysqli_query($this->cdbase->connect(),"UPDATE
Message SET Information='$information' WHERE Id = '$id'");
            return v;
        }

        //редагування даних діалогу
        public function UpdateDialog($id, $acc1, $acc2) {
            $v = mysqli_query($this->cdbase->connect(),
"UPDATE Dialog "+
"SET AccountId1='$acc1', AccountId2='$acc2' "+
"WHERE Id = '$id'");
            return v;
        }

```

```

    }

    //редагування автоматичного повідомлення
    public function UpdateMessageAnswer($id, $message,
    $status, $as) {
        $v = mysqli_query($this->cdbase->connect(),
        "UPDATE MessageAnswer "+
        "SET MessageId='$message', Status='$status', "+
        "AnswerSchemeId='$as' "+
        "WHERE Id = '$id'");
        return v;
    }

    //редагування схеми відповіді на повідомлення
    public function UpdateAnswerScheme($id, $categ,
    $question, $request, $answer) {
        $v = mysqli_query($this->cdbase->connect(),
        "UPDATE AnswerScheme "+
        "SET CategoryId='$categ', Question='$question', "+
        "Request='$request', Answer='$answer' "+
        "WHERE Id = '$id'");
        return v;
    }

    //редагування даних прикладу
    public function UpdateExample($id, $as, $question) {
        $v = mysqli_query($this->cdbase->connect(),
        "UPDATE Example SET AnswerSchemeId='$as', "+
        "Question='$question' "+
        "WHERE Id = '$id'");
        return v;
    }

    //редагування даних категорії
    public function UpdateCategory($id, $title) {
        $v = mysqli_query($this->cdbase->connect(),
        "UPDATE Category "+
        "SET Title='$title' "+
        "WHERE Id = '$id'");
        return v;
    }

```

```

    }

    //отримання повних даних прикладу з заданим номером
    public function LoadExample($num) {
        $v = mysqli_query($this->cdbase->connect(),
        "SELECT * FROM Example WHERE Id = $num");
        return $v;
    }

    //завантаження всіх прикладів
    public function LoadExamples() {
        $v = mysqli_query($this->cdbase->connect(),
        "SELECT * FROM Example ");
        return $v;
    }

    //завантаження всіх прикладів, що відповідають схемі з
    заданим номером
    public function LoadExamplesOnAnswerScheme($num) {
        $v = mysqli_query($this->cdbase->connect(),
        "SELECT * FROM Example WHERE AnswerSchemeId = '$num'");
        return $v;
    }

    //завантаження всіх категорій
    public function LoadCategories($num) {
        $v = mysqli_query($this->cdbase->connect(),
        "SELECT * FROM Category ");
        return $v;
    }

    //завантаження всіх схем відповідей на запитання з
    заданої категорії
    public function LoadCategoryAnswerScheme($num) {
        $v = mysqli_query($this->cdbase->connect(),
        "SELECT * FROM AnswerScheme WHERE CategoryId = '$num'");
        return $v;
    }

```

```

        //завантаження даних схем відповідей на запитання за
        назвою її категорії
        public function LoadCategoryAnswerSchemeOnName($name) {
            $v = mysqli_query($this->cdbase->connect(),
            "SELECT * FROM AnswerScheme "+
            "INNER JOIN Category On Category.Id = AnswerScheme.CategoryId
"+
            "WHERE Category.Title = '$name'");
            return v;
        }

        //завантаження прикладів за схемою відповідей на
        запитання за номером
        public function LoadExamplesOnAnswerScheme($num) {
            $v = mysqli_query($this->cdbase->connect(),
            "SELECT * FROM Example WHERE AnswerSchemeId = '$num'");
            return v;
        }

        //збереження нового прикладу
        public function SaveExample($as, $question) {
            $v = mysqli_query($this->cdbase->connect(),
            "INSERT INTO Example(AnswerSchemeId, Question) "+
            "VALUES('$as', '$question')");
            return v;
        }

        //збереження нової категорії
        public function SaveCategory($title) {
            $v = mysqli_query($this->cdbase->connect(),
            "INSERT INTO Category(Title) VALUES('$title')");
            return v;
        }

        //збереження новго діалогу
        public function SaveDialog($acc1, $acc2) {
            $v = mysqli_query($this->cdbase->connect(),
            "INSERT INTO Dialog(AccountId1, AccountId2) "+
            "VALUES('$acc1', '$acc2')");
            return v;
        }

```

```

    }

    //збереження нової автоматичної відповіді
    public function SaveMessageAnswer($mess, $status, $as) {
        $v = mysqli_query($this->cdbase->connect(),
"INSERT INTO MessageAnswer(MessageId, Status, AnswerSchemeId)
"+
"VALUES('$mess', '$status', '$as')");
        return v;
    }

    //збереження нової схеми відповідей
    public function SaveAnswerScheme($cid, $question,
$request, $answer) {
        $v = mysqli_query($this->cdbase->connect(),
"INSERT INTO MessageAnswer(CategoryId, Question, Request,
Answer) "+
"VALUES('$cid', '$question', '$request', '$answer')");
        return v;
    }

    //завантаження даних діалогу за його номером
    public function LoadDialog($num) {
        $v = mysqli_query($this->cdbase->connect(),
"SELECT * FROM Dialog WHERE Id = '$num'");
        return v;
    }

    //завантаження схем відповідей на запитання
    public function LoadAnswerSchemes() {
        $v = mysqli_query($this->cdbase->connect(),
"SELECT * FROM AnswerSchemes ");
        return v;
    }

    }

    }

    ?>

```

Б.3 Текст файлу SignUp.php

```
//реєстрація акаунту
<?php
if (isset($_GET["name"]) && isset($_GET["password"]) &&
isset($_GET["email"]) && isset($_GET["createddate"]) &&
isset($_GET["type"]))
{
    $name = $_GET["name"];
    $password = $_GET["password"];
    $id = $_GET["acc_id"];
    $type = $_GET["type"];

    include_once './DataBaseProcessor.php';

    $cdbase = new DataBaseProcessor ();

    $acc=$cdbase->GetAccountOnName ($name);
    $v = mysqli_num_rows($acc);
    if($v==0){
        $sa = $cdbase ->SaveAccount($name, $password,
$email, $createddate, $type);
        if($sa!=0){
            echo "успішно";

        }else{

            echo "помилка";
        }
    }
    else {
        $sa =$cdbase ->IfAccount ($name);
        if($sa){
            echo "успішно ";
        }
        else {
            echo "помилка";
        }
    }
} else {
    echo "пропущено";
}
```

```
}
?>
```

Б.4 Текст файлу SendMessage.php

```
//відправлення повідомлення
<?php
    $name = $_GET["name"];
    $dialog = $_GET["dialogid"];
    $mtext = $_GET["text"];

    include_once './DataBaseProcessor.php';

    $cdbase = new DataBaseProcessor ();

    $acc=$cdbase ->GetAccountOnName ($name);
    $v = mysqli_num_rows($acc);
    if($v==0){
        echo "false";
    }
    else {
        echo "true";
        $res = LoadDialog($dialog);
        $acc1 = $res['AccountId1'];
        $acc2 = $res['AccountId2'];
        $accnum = 0;
        if($acc1 == $v['Id']){
            $accnum = 1;
        }
        else {
            if($acc2 == $v['Id']){
                $accnum = 2;
            }
        }
        $m=$cdbase->SaveMessageNow($dialog, $accnum,
$mtext);
    }
?>
```

Б.5 Текст файлу LoadMessages.php

```
//завантаження всіх повідомлень користувача
```

```

<?php
    $name = $_GET["name"];
    $mode = $_GET["mode"];

    include_once './DataBaseProcessor.php';

    $cdbname = new DataBaseProcessor ();

    $acc=$cdbname->GetAccountOnName ($name);
    $v = mysqli_fetch_row ($acc);

    $messages=$cdbname -> LoadUserMessages ($v['Id']);

?>

```

Б.6 Текст файлу LoadManagerMessages.php

//завантаження всіх повідомлень користувача у спілкуванні з менеджерами

```

<?php
    $name = $_GET["name"];
    $mode = $_GET["mode"];

    include_once './DataBaseProcessor.php';

    $cdbname = new DataBaseProcessor ();

    $acc=$cdbname->GetAccountOnName ($name);
    $v = mysqli_fetch_row ($acc);

    $dialogs=$cdbname -> GetDialogsWithManager ($v['Id']);
    $d = mysqli_fetch_row ($dialogs);

    messages=LoadDialogMessages($s['Id']);

?>

```

Б.7 Текст файлу LoadBotMessages.php

//завантаження повідомлень у спілкуванні користувача з програмою

```

<?php

```

```

$name = $_GET["name"];
$mode = $_GET["mode"];

include_once './DataBaseProcessor.php';

$cdbase = new DataBaseProcessor ();

$acc=$cdbase->GetAccountOnName ($name);
$v = mysqli_fetch_row ($acc);

$ddialogs=$cdbase -> GetDialogsWithBot ($v['Id']);
$d = mysqli_fetch_row ($dialogs);

messages=LoadDialogMessages($s['Id']);

?>

```

Б.8 Текст файлу LoadCategories.php

```

//завантаження переліку категорій
<?php

include_once './DataBaseProcessor.php';

$cdbase = new DataBaseProcessor ();

$categories=$cdbase -> LoadCategories ();

?>

```

Б.9 Текст файлу LoadAnswerSchemes.php

```

//завантаження схем відповідей на запитання
<?php

include_once './DataBaseProcessor.php';

$cdbase = new DataBaseProcessor ();

$categories=$cdbase -> LoadAnswerSchemes ();

?>

```

Б.10 Текст файлу LoadExamples.php

```
//завантаження прикладів відповідей на запитання
<?php
    $as = $_GET["answerscheme"];

    include_once './DataBaseProcessor.php';

    $cdbase = new DataBaseProcessor ();

    $examples=$cdbase -> LoadExamplesOnAnswerScheme ($as);

?>
```

Б.11 Текст файлу SaveExample.php

```
//збереження прикладу відповіді на запитання
<?php
    $as = $_GET["answerscheme"];
    $question = $_GET["question"];

    include_once './DataBaseProcessor.php';

    $cdbase = new DataBaseProcessor ();

    $ex=$cdbase -> SaveExample ($as, $question);
    $e = mysqli_num_rows($ex);
    if($e==0){
        echo "false";
    }
    else {
        echo "true";
    }

?>
```

Б.12 Текст файлу SaveAnswerScheme.php

```
//створення нової схеми відповіді на запитання
<?php
//розбір отриманих вхідних даних
    $cid = $_GET["category"];
```

```

$question = $_GET["question"];
$request = $_GET["request"];
$answer = $_GET["answer"];

include_once './DataBaseProcessor.php';

$cdbase = new DataBaseProcessor ();

//збереження схеми в базі даних
$ex=$cdbase -> SaveAnswerScheme ($cid, $question,
$request, $answer);

//перевірка отриманого результату
$e = mysqli_num_rows($ex);
if($e==0){
    echo "false";
}
else {
    echo "true";
}
?>

```

Б.13 Текст файлу UpdateAnswerScheme.php

```

//оновлення даних схеми відповіді на запитання
<?php
//розбір отриманих вхідних даних
$cid = $_GET["category"];
$question = $_GET["question"];
$request = $_GET["request"];
$answer = $_GET["answer"];

//підключення до бази даних
include_once './DataBaseProcessor.php';

$cdbase = new DataBaseProcessor ();

//збереження в базу даних
$ex=$cdbase -> UpdateAnswerScheme ($id, $categ,
$question, $request, $answer);

//перевірка отриманого результату

```

```
$e = mysqli_num_rows($ex);  
if($e==0){  
    echo "false";  
}  
else {  
    echo "true";  
}  
?>
```

ДОДАТОК В
Слайди презентації



Рисунок В.1 – Слайд 1

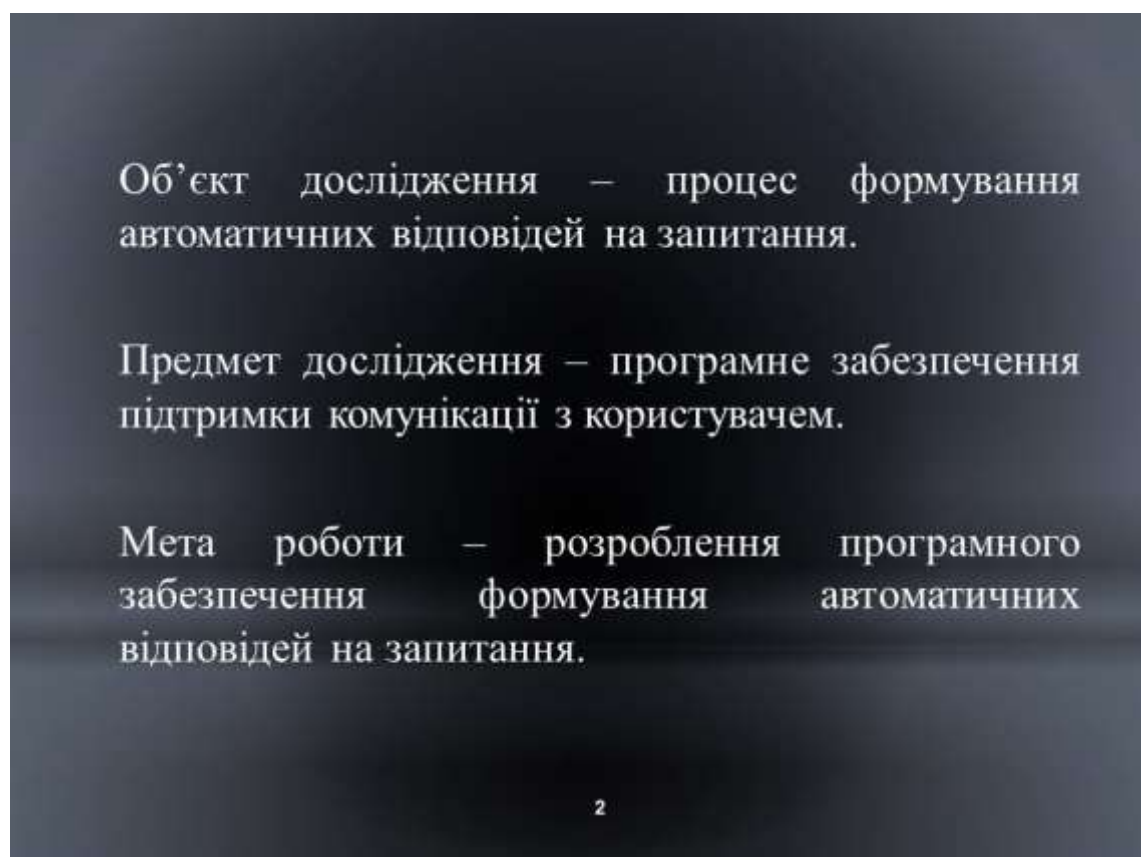


Рисунок В.2 – Слайд 2

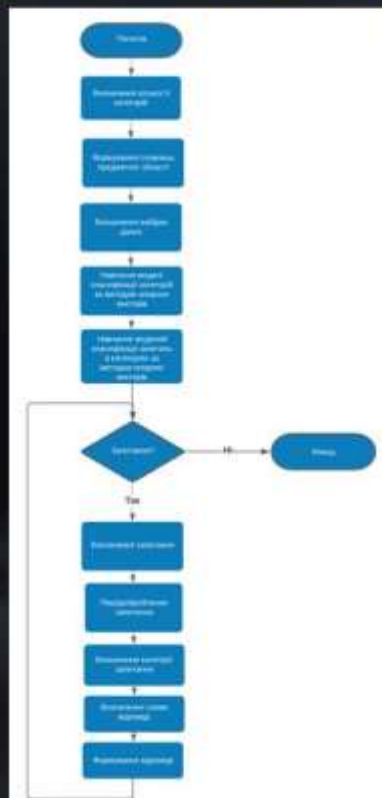
Завдання

- ✓ дослідження методів формування автоматичних відповідей на запитання;
- ✓ проєктування програмного забезпечення;
- ✓ розробка програмного забезпечення;
- ✓ тестування та відлагодження програми;
- ✓ економічне обґрунтування доцільності розробки.

3

Рисунок В.3 – Слайд 3

Метод формування автоматичних відповідей на запитання



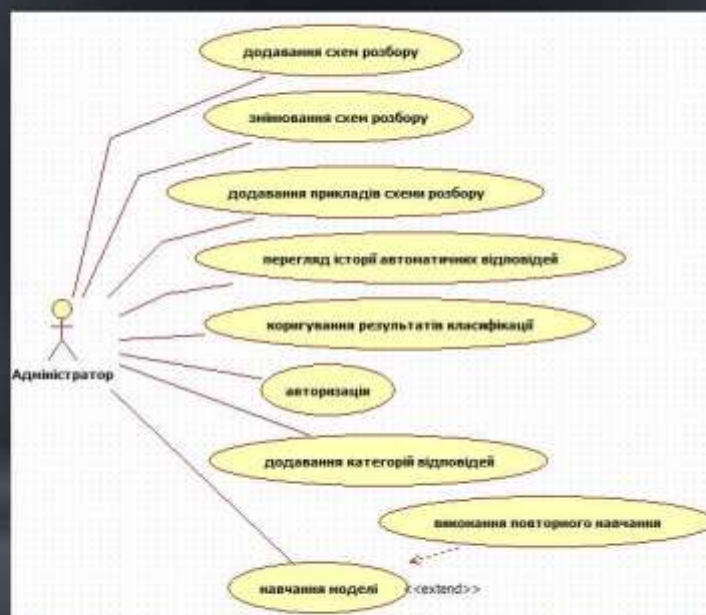
Діаграма прецедентів користувача



5

Рисунок В.5 – Слайд 5

Діаграма прецедентів адміністратора



6

Рисунок В.6 – Слайд 6

Порівняння і вибір мови програмування

Критерій	PHP	Java
Призначення	Загальне, веброзробка	Загальне
Система типів	Динамічна	Статична
Безкоштовність	Так	Так
Швидкість завантаження	Дещо менше	Середня

7

Рисунок В.7 – Слайд 7

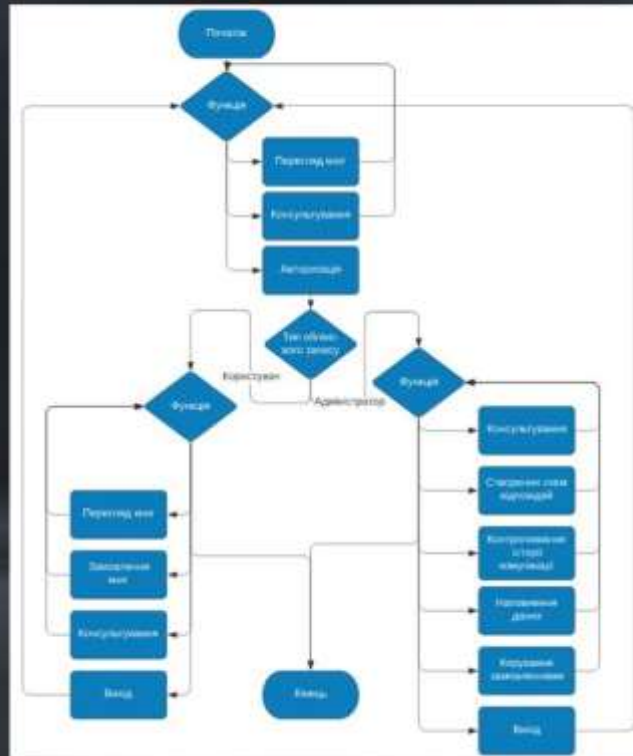
Структура бази даних



8

Рисунок В.8 – Слайд 8

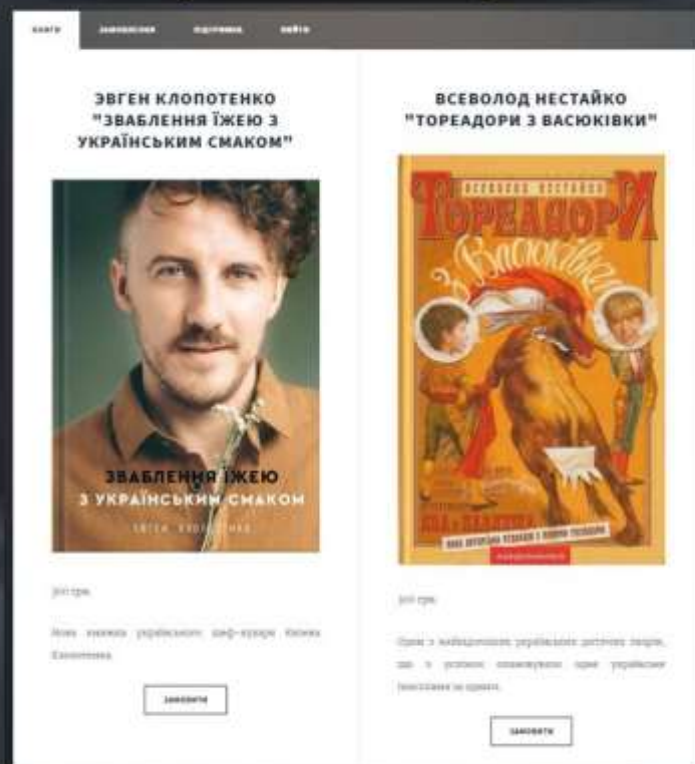
Схема функціонування програми



9

Рисунок В.9 – Слайд 9

Приклад інтеграції



10

Рисунок В.10 – Слайд 10

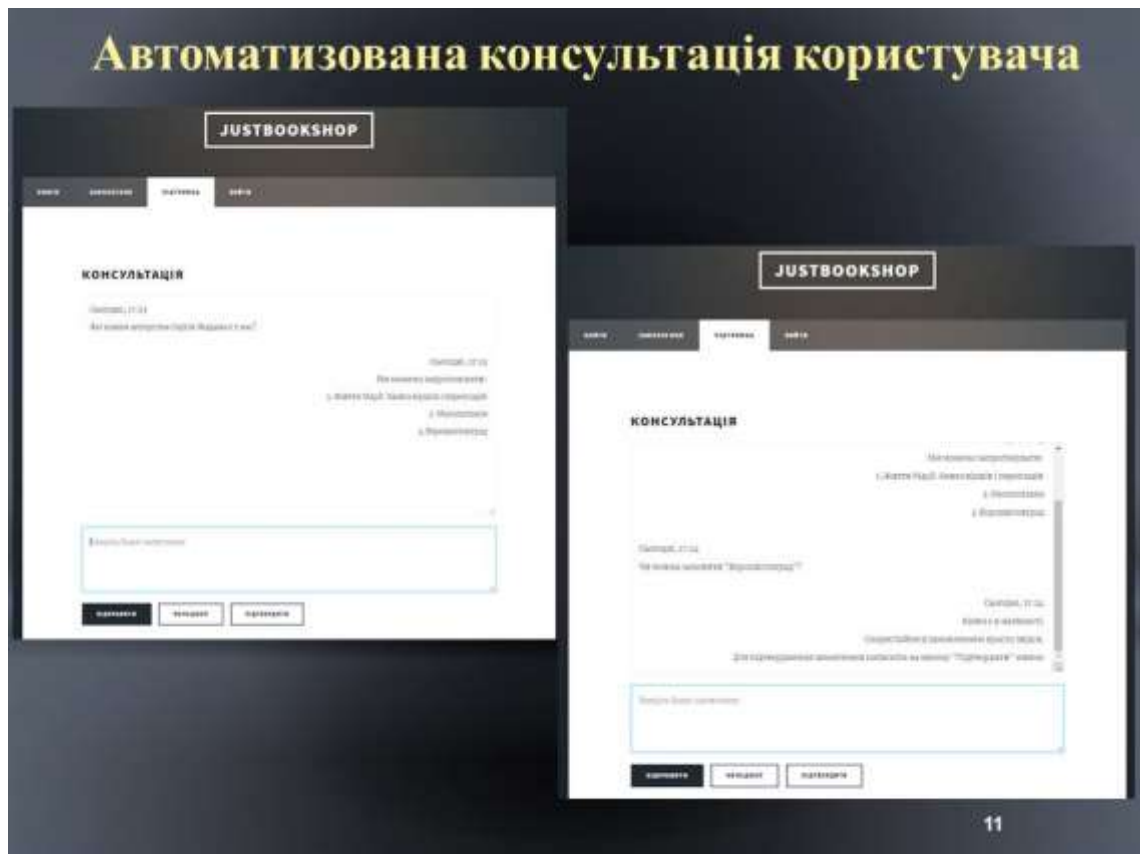


Рисунок В.11 – Слайд 11

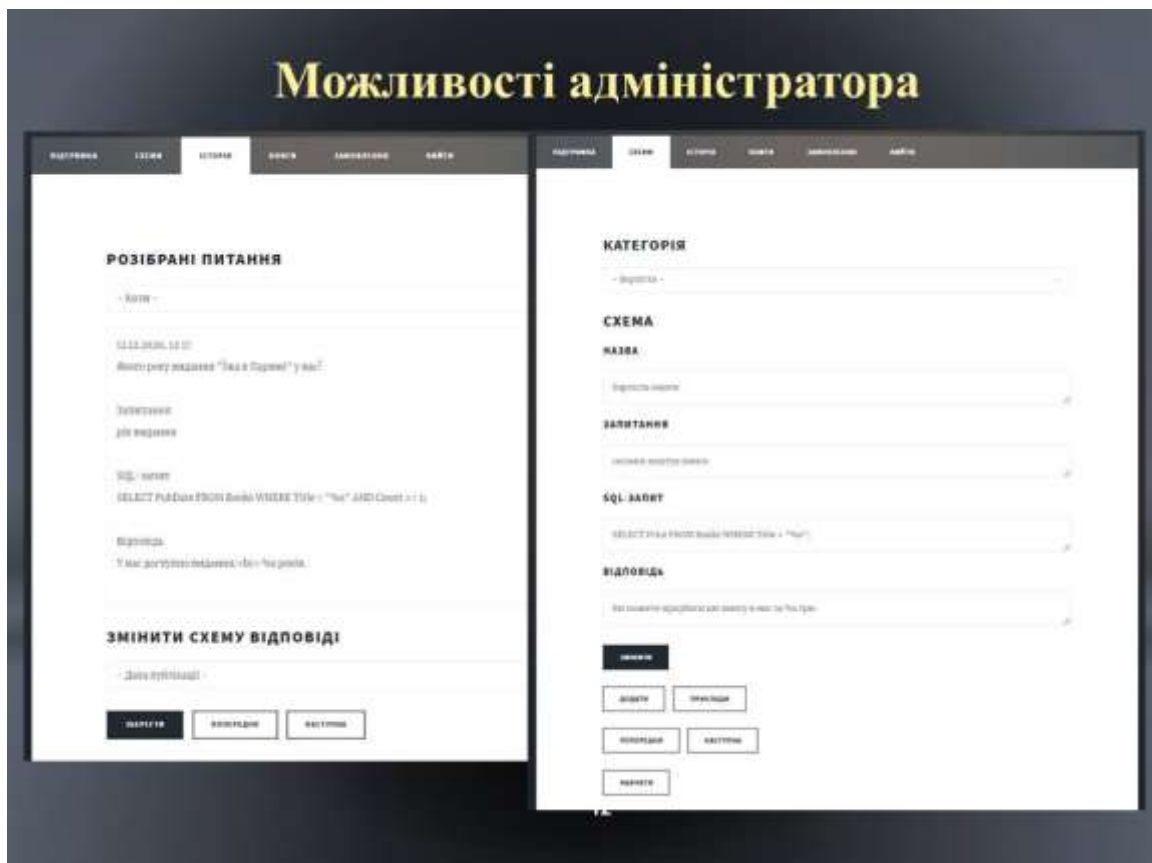


Рисунок В.12 – Слайд 12

Висновки

У дипломній кваліфікаційній роботі магістра розглянуто проблему формування автоматичних відповідей на запитання, що виникає під час створення програмних помічників для роботи з користувачами вебсайтів та віртуальних асистентів.

Наукова новизна роботи полягає в тому, що в ній запропоновано метод формування автоматичних відповідей на запитання, що полягає в початковій класифікації запитань за категоріями, подальшій уточнюючій класифікації за конкретними схемами відповідей на запитання на основі використання методу опорних векторів, кодуванні запитання та формування відповіді на основі виконання запиту до бази даних, що дозволяє відповідати на запитання за предметною областю без участі менеджерів компанії.

13

Рисунок В.13 – Слайд 13