


Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки


МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:
«ІНФОРМАЦІЙНА СИСТЕМА ПОСТАНОВКИ МЕДИЧНОГО ДІАГНОЗУ»
08-23.МКР.002.00.000 ПЗ

Виконав: студент 2-го курсу, групи КІ-20мз
спеціальності 123 – Комп'ютерна інженерія
(шифр і назва напрямку підготовки, спеціальності)


Свтухівський О. В.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. ОТ


Муращенко О.Г.
(прізвище та ініціали)

« » _____ 2022 р.

Опонент:

_____ Дудатьєв А. В.
(прізвище та ініціали)

« » _____ 2022 р.

Допущено до захисту

Завідувач кафедри ОТ

д.т.н., доц.

_____ Азаров О.Д.

« » _____ 2022 р.

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

Рівень вищої освіти II (магістерський)

Галузь знань 12 Інформаційні технології

Спеціальність 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ, д.т.н., проф.

Азаров О.Д.

«___» _____ 2022 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Євтухівський Олексій Васильович

1 Тема роботи «Інформаційна система постановки медичного діагнозу» керівник роботи Муращенко Олександр Геннадійович к.т.н., доц. каф. ОТ. затверджені наказом вищого навчального закладу від «24» березня 2022 року №65

2 Строк подання студентом проекту (роботи) — 16.05.2022 р.

3 Вихідні дані до роботи — база даних постановки мед.діагнозу, структура інформаційних систем, задачі інформаційно-експертних систем, вимоги до функціональних можливостей, структурні можливості та використання чат-ботів.



4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) — вступ, аналіз предметної області та постановка задачі, загальні відомості про інформаційні системи, технології для розробки системи медичного експерту на основі чат-бота, експериментальні дослідження системи, економічна частина.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): діаграма архітектури системи, діаграма використання, діаграма класів, діаграма стану та переходу, діаграма розгортання, діаграма

послідовності, візуальне «привітання» та початок роботи бота, зображення роботи чат- бота.

6 Консультанти розділів роботи приведені в таблиці 1.

Таблиця 1 — Консультанти

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Крупельницький Л.В., к.т.н доцент		
5	Руда Л. П., к.е.н., доцент		

7 Дата видачі завдання **01 жовтня 2021 року**.

8 Календарний план виконання роботи приведений в таблиці 2

Таблиця 2 — Календарний план

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Аналіз поставленої задачі	01.10.2021	Виконано
2	Огляд існуючих реалізацій	02.10.2021 – 14.11.2021	Виконано
3	Аналіз методів розробки інформаційних систем та їх використання	15.11.2021 – 05.12.2021	Виконано
4	Вивчення зв'язку емпіричних і числових систем	06.12.2021 – 02.01.2022	Виконано
5	Вивчення технологій для розробки системи на основі чат-бота	03.01.2022 – 06.02.2022	Виконано
6	Розрахунок перспективи продукту та його функціонал	07.02.2022 – 14.03.2022	Виконано
7	Проектування, розробка чат-бота та його тестування	15.03.2022 – 24.04.2022	Виконано
8	Розрахунок економічної доцільності системи та оформлення економічної частини в пояснювальній записці	25.04.2022 – 13.05.2022	Виконано
9	Перевірка якості виконання магістерської кваліфікаційної роботи та усунення недоліків	16.05.2022	Виконано


Магістрант


(підпис)

Євтухівський О.В.

(прізвище та ініціали)

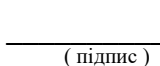
Керівник проекту (роботи)


(підпис)

Муращенко О.Г.

(прізвище та ініціали)

Опонент


(підпис)

Дудатьєв А.В.

(прізвище та ініціали)

« »

2022 р.

АНОТАЦІЯ

УДК 004.891.3

Євтухівський О.В. Інформаційна система постановки медичного діагнозу. Магістерська кваліфікаційна робота зі спеціальності 123 – Комп'ютерна інженерія. Вінниця: ВНТУ, 2022. 104 с.

На укр. мові. Бібліогр.: 18 назв; рис.: 21; табл. 14

В Магістерській кваліфікаційній роботі досліджено актуальність використання інформаційних систем та інформаційно-експертних систем для подальшого їх використання у створенні системи діагностування на базі чат-бота, платформи месенджера Telegram.

В ході дослідження отримані такі результати: визначені існуючі програмні засоби інформаційно-експертних систем, вивчена архітектура, методи конструювання, а також сфери використання даних систем. Розроблений план проєкту: визначено переліг робіт, терміни.

Побудований чат-бот, який взятий за основу інформаційно-експертної системи, допоможе в рекомендованих цілях для користувача виявити чи спрогнозувати ймовірний діагноз з налаштованої алгоритмічної бази симптоматики.

В економічній частині роботи прораховані витрати на розробку системи та перспективність розробки.

Ключові слова: інформаційна система; інформаційно-експертна система; чат-бот; clips; telegram.

ABSTRACT

UDC 004.891.3

Yevtukhivskyi OV. Information system for medical diagnosis. Master's thesis in specialty 123 - Computer Engineering. Vinnytsia: VNTU, 2022. 104 p.

In Ukrainian language. Bibliogr.: 18 titles; fig .: 21; table 14.

The Master's qualification work investigates the relevance of the use of information systems and information-expert systems for their further use in the established diagnostic systems based on the chatbot, the Telegram messenger platform.

The following results were obtained during the research: the existing software tools of information-expert systems were determined, the architecture, design methods, as well as the spheres of use of these systems were studied. The project plan is developed: the list of works, terms are defined.



The built chatbot, which is based on the information-expert system, will help in the recommended purposes for the user to identify or predict a probable diagnosis from the configured algorithmic base of symptoms.

In the economic part of the work, the costs of system development and development prospects are calculated.

Keywords: information system; information and expert system; chatbot; clips; telegram.

ЗМІСТ

ВСТУП	9
1 АНАЛІЗ, ЗАГАЛЬНІ ВІДОМОСТІ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	11
1.1 Сучасні проблеми.....	11
1.2 Огляд існуючих реалізацій.....	12
1.3 Аналіз методів розробки інформаційних систем.....	13
1.4 Підходи до обробки інформації.....	16
1.4.1 Побудова дерева рішень	15
1.4.2 Статистична обробка даних.....	17
1.4.3 Використання елементів штучного інтелекту.....	18
1.5 Основи роботи інформаційних систем	21
1.6 Зв'язок емпіричних і числових систем	27
1.6.1 Методи вимірювання ступеня впливу об'єктів.....	27
1.6.2 Метод ранжирування.....	28
1.6.3 Метод парних порівнянь.....	30
1.6.4 Метод безпосередньої оцінки.....	31
1.7 Характеристика та режими роботи групи експертів	33
1.8 Обробка експертних оцінок	35
1.9 Байесова мережі довіри	36
1.10 Діаграми впливу	42
2 ТЕХНОЛОГІЇ ДЛЯ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ МЕДИЧНОГО ЕКСПЕРТУ НА ОСНОВІ ЧАТ-БОТА	45
2.1 Перспектива та функції продукту	45
2.2 Класи та функції користувача.....	46
2.3 Обмеження проектування та провадження	46

					08-23.МКР.002.00.000 ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата	Інформаційна система постановки медичного діагнозу	Літ.	Арк.	Аркушів
Розроб.		Євтухівський О.В.				6	102	
Перевір.		Муращенко О.Г.						
Реценз.		Дудатьєв А.В.						
Н. Контр.		Швець С.І.						
Затверд.		Азаров О.Д.			ВНТУ, гр. КІ-20м			

2.4	Вимоги до зовнішнього інтерфейсу	47
2.5	Інтерфейси програмного забезпечення.....	48
2.5.1	Інтерфейси програмного забезпечення на стороні клієнта.....	47
2.5.2	Інтерфейс ІСЕТ на стороні сервера.....	47
2.6	Особливості системи.....	49
2.6.1	Чат-бот.....	48
2.6.2	Стимули послідовностей відповіді.....	48
2.7	Інтерфейс чату та функціональні вимоги.....	50
2.7.1	Базові функції.....	50
2.7.2	Ризики	50
2.7.3	Стимули послідовності відповідей	51
2.8	Інші нефункціональні вимоги.....	52
2.8.1	Вимоги до продуктивності.....	51
2.8.2	Вимоги до безпеки.....	52
2.9	Моделювання проєктних процесів.....	53
3	АНАЛІЗ ТА ПРОЄКТУВАННЯ ЧАТ-БОТА.....	55
3.1	Діаграми архітектури інформаційної системи.....	55
3.1.1	Діаграми використання.....	55
3.1.2	Діаграми класів.....	56
3.1.3	Діаграми стану та переходу.....	57
3.1.4	Діаграми розгортання.....	58
3.2	Схема бази даних	60
3.3	Сервера.....	62
3.4	Диспетчер.....	63
3.5	Ядро	64
3.6	Дані запитань та відповідей	65
3.7	Інформаційно-експертна система.....	66
3.8	Лікар SkyNet	68
4	ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ СИСТЕМИ	71
4.1	Тестування графічного інтерфейсу	71

4.2 Тестування роботи серверу	72
5 ЕКОНОМІЧНА ЧАСТИНА	76
5.1 Проведення технологічного аудиту інформаційної системи	76
5.2 Розрахунок витрат на здійснення науково-дослідної роботи.....	81
5.3 Розрахунок економічної ефективності науково-технічної розробки за її впровадження безпосередньо розробником (замовником).....	87
ВИСНОВКИ	91
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	92
ДОДАТОК А Технічне завдання	94
ДОДАТОК Б Алгоритм розробки інформаційно-експертної системи.....	98
ДОДАТОК В Лістинг чат-бота	100
ДОДАТОК Г Структура схема ядра системи.....	102
ДОДАТОК Г Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень.....	104

ВСТУП

В сучасному світі інформаційні технології стали не лише як окремий кластер технологічного розвитку, але і прийшли на допомогу напрямкам, які вже були століттями у використанні. Один із таких є медичний сектор. Використавши переваги інформаційних технологій, а саме у нашому випадку інформаційної системи для удосконалення, спрощення і оптимізації процесів терапевтичної самодіагностики.

В часи реформування медичної системи нашої країни, **актуальність** питання оптимізації і розвантаження систем медичних закладів засобами інформаційних технологій є як ніяк актуальним.

Також використавши дані про те що в наш час майже кожен громадянин має при собі смартфон і доступ до інтернету, це дає нам простір для корисного використання як можливістю встановлення мобільних додатків інформаційних систем для спрощення і покращення життя.

Слід зауважити також і те, що сегмент мобільних додатків медичного розділу не тільки не є популярним серед громадян, але і те що інформаційних систем постановки медичного діагнозу в мобільних площадках App Store та Play Market в українському сегменті відсутні.

Об'єкт дослідження — алгоритмічний процес постановки медичного діагнозу з використанням вхідних симптоматичних алгоритмах збору даних користувача.

Предмет дослідження — інформаційна система клієнта з побудованим зв'язком серверу бази даних, яка надає відповідні дані (можливі діагнози) клієнту, при певному алгоритму вказаних клієнтом симптомів в додатку.

Метою і завданням магістерської роботи є розробка інформаційної системи, яка має допомогти користувачу у рекомендованому форматі проінформувати про можливі проблеми зі здоров'ям згідно вказаних користувачем симптомів.

Для досягнення мети необхідно вирішити такі задачі:

- збір, аналіз та узагальнення інформації, щодо можливих діагнозів з відповідними до них симптомами;
- вибір технології розробки та мови програмування інформаційної системи;
- створення серверної бази даних та подальшої прив'язки до клієнтської системи;
- розробка функціональних можливостей інформаційної системи.

Наукова новизна полягає в тому, що розроблена інформаційна система постановки медичного діагнозу буде використовуватись на новій платформі, різновиду штучного інтелекту – «чат-бот», на платформі месенджера Telegram, яка має переваги над відомими раніше медичними інформаційно-експертними системами.

Практичне значення отриманих результатів в тому, що кінцева інформація яка буде надана клієнту буде мати суто рекомендований характер і ні в якому разі не має бути визначеним як медичне заключення.

Інформація буде корисна для можливого виявлення хвороби та можливого заощадження часу лікаря, для подальшої консультації з ним.

Публікація — Євтухівський О.В., Муращенко О.Г., Крупельницький Л.В. Інформаційна експертна система постановки медичного діагнозу [1].

Апробація роботи виконана під час доповіді на LI науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії в секції обчислювальної техніки.

1 АНАЛІЗ, ЗАГАЛЬНІ ВІДОМОСТІ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Сучасні проблеми

Метою цього проекту є розроблення інформаційної системи постановки можливого медичного діагнозу на основі чат-боту. Лікарі кожного дня бачать десятки пацієнтів на день, їхня робота є безумовно важкою та в надзвичайних ситуаціях їм доводиться працювати понаднормово. Досить часто лікарі витрачають свій дорогоцінний час на збір особистої інформації про пацієнта та їх медичну історію щодо попередніх та теперішніх симптом.

Мета проекту — посилення першої взаємодії між лікарем та пацієнтом, коли лікар задає певні запитання аби зафіксувати історію хвороб пацієнта.

Щоб позбавити лікаря від зайвих клопотів та заощадити його дорогоцінний час, можна отримати основну історію хвороб та симптомів у пацієнта за допомогою спілкування через доброзичливий чат-бот.

Даний проєкт має на меті діяти лише як проміжний механізм між лікарем та пацієнтом. Завдяки його роботі, пацієнт має можливість спілкуватися із системою, що дозволяє повідомляти про свою особисту та медичну історію ботові, за яким комп'ютер здійснює облік. Після цієї взаємодії та отримання даних система намагатиметься встановити діагноз та повідомить про це лікаря, надіславши електронного листа.

Інформаційна система [2] — це комп'ютерна система, що моделює здатність людини приймати рішення. Як зазначалось вище, зазвичай більшість лікарів перевантажені кількістю пацієнтів, яким вони надають свої послуги. Тому пропонується нове вирішення даної проблеми завдяки створенню медичного експерта на основі інформаційної системи, яка дозволяє вести комунікацію з пацієнтом, що дає можливість відстежувати його історію хвороби.

Для цього в якості інтерфейсу для пацієнтів використовується додаток Telegram для обміну повідомленнями. Telegram-bot — це широко

використовуваний, міжплатформний, відкритий і високобезпечний чат-додаток, який дає змогу ботам вести комунікацію з користувачами. Відбувається даний зв'язок за допомогою RESTful API, з яким може взаємодіяти будь-який чат-бот.

1.2 Огляд існуючих реалізацій

Експертні інформаційні системи на основі чат-ботів, існують вже 30 років [2]. Характерним прикладом є відомий чат-бот “Eliza”. Чат-боти на основі онтології, — це постійні зусилля, створені на моделях семантичних мереж і NLP (Natural Language Processing) [3] для підтримки взаємодії між користувачем та машиною. Система працює з використанням 3 модулів — система на основі знань, інтерпретація питань і Natural Language Generation. Онтологія — це формальне найменування й визначення властивостей, типів та зв'язків між сутностями.

Розробка системи Temporal Reasoning є вагомим подією у створенні експертно-інформаційних систем на базі чату. Тимчасові міркування є важливими, оскільки кінцева діагностика залежить від послідовності симптомів. Тимчасові міркування й логіка — це система правил для міркування та уявлення щодо кваліфікованих суджень заснованих на часі, наприклад, я ЗАВЖДИ голодний. У статті [3] досліджено тимчасові зв'язки між симптомами на основі теорії нечітких множин.

Інша оглядова стаття розкриває цікаві підходи моделювання процесу прийняття медичних рішень [4]. У ній проаналізовано AI та інформаційні системи, щоб ознайомити читачів із цією сферою та запропонувати шляхи, за допомогою яких дослідження в кінцевому результаті будуть використовуватись для подальшого медичного моніторингу. Стаття тлумачить, що інформаційною системою експертного типу є система, що містить і може використовувати спеціалізовані знання. Вона застосовує ці знання для припущень користувачів, які можуть не мати повних доступних знань.

Відмінність між інформаційними системами експертного типу і системами, заснованих на знаннях, полягає в тому, що інформаційні системи здатні пояснити користувачами свою поведінку. У пояснювальній записці також аналізуються різні методи, впроваджені для реалізації медичної інформаційної системи експертного типу.

1.3 Аналіз методів розробки інформаційних систем

Розробка нових методів діагностики та розширення існуючих можливостей методів була і досі залишається актуальним завданням у галузі медицини [6]. Поява нових діагностичних та лікувальних технологій базуються на використанні методів штучного інтелекту для обробки та інтерпретації даних з можливістю накопичення, зберігання і повторного використання медичних даних. Інформаційні системи експертного типу є одним із найефективніших інструментів у цій галузі. Вони дають змогу автоматизувати процес прийняття рішень під час обстеження пацієнтів, при цьому підвищуючи рівень навичок та знань користувача до рівня досвідченого експерта. Тому інформаційні системи повинні мати велику інформаційну ємність, бути стійкими до перешкод, володіти можливістю гнучкої постановки задач, бути причетними до різних областей біології та медицини й не потребувати тривалого часу для розробки.

Постійний розвиток комп'ютерних технологій розширює потенціал подібних систем, у зв'язку з чим потрібно постійно оновлювати та поповнювати знання у даній галузі.

Інформаційні системи експертного типу впродовж тривалого часу використовуються в медицині для діагностування захворювань. Однак сфера застосування кожної системи обмежена, оскільки початкова увага була зосереджена на розробці додатків для цих інформаційних систем, які спочатку не були призначені для використання в цих сферах, складних і часто взагалі неможливих, у тому числі тому, що вони обмежені вбудованими засобами виведення.

Рішення завдань розроблення інформаційних систем експертного типу можна розділити на дві категорії:

- задачі, що вирішуються за безумовними алгоритмами;
- задачі з прийому рішень на основі досвіду та знань.

Тому можна виокремити 2 шляхи рішення завдань — логічний та інтуїтивний. Перший метод базується на використанні набору формальних правил, другий — накопиченого досвіду. При вирішенні завдання за допомогою логічного способу завдання розбивається на окремі підзадачі, які у свою чергу, розбиваються ще на декілька простих функцій з відомим алгоритмом розрахунків. Знання послідовності кожної простої функції дозволяє вирішувати завдання найскладнішого рівня, поєднуючи елементарні функції в належному алгоритмі. Застосовуючи даний метод для рішення інтуїтивних задач програмісти можуть натрапити на неможливість відокремити послідовність елементарних функцій чи поділити завдання на такі функції.

Існують традиційні системи прийняття рішень, засновані на явних правилах висновку, які зазвичай створюються групою фахівців, у числі яких переважно програмісти, математики та експертні фахівці із завдань. Можливості налаштування цих систем для кінцевих користувачів часто є недостатніми.

Самонавчальні медичні інформаційні системи експертного типу (ICET) прийняття рішень повинні відповідати наступним вимогам [6]:

- індивідуалізація (особистий досвід і знання експертів, характеристики лікувально-діагностичних методик, адаптація на традиції клінічних шкіл, медико-біологічні набори даних, геосоціальні особливості області застосування);
- динамічний розвиток (накопичення досвіду в процесі експлуатації системи, наслідуючи зміни в пунктах, які вказані у попередній вимозі);
- можливість пере налаштування при швидкій переміні умов (наприклад, при перенесенні в інші регіони);

— здатність екстраполювати результат — протилежна вимога індивідуальності, яка означає, що система не повинна різко втрачати якість роботи при переміні умов;

— можливість кінцевого користувача створення з нуля (експерт повинен вміти винаходити абсолютно нову ICET та просто й швидко сконструювати її);

— «нечіткий» характер результату, тобто рішення, прийняті системою, не повинні бути остаточними, вони можуть бути імовірнісним або пропонувати вибір кількох варіантів і таким чином це дозволяє експерту критично оцінити рішення системи, а також не позбавляє його ініціативи у прийнятті кінцевого рішення;

— ICET є лише порадиником від фахівця та не претендує на цілковиту точність рішення, вона повинна накопичувати знання і досвід, значно пришвидшувати доступ до них, імітувати результат в міру зміни умов завдання, а фахівці завжди несуть відповідальність за рішення.

Більшість розроблених інформаційних систем задовольняють вище переліченим вимогам.

Перелічимо найбільш відомі приклади існуючих інформаційних систем експертного типу [6]:

— PUFF — інформаційні система експертного типу, яка проводить діагностування захворювань легень на основі легеневих функціональних тестів;

— AI / RHEUM — діагностика захворювань сполучних тканин;

— CADUCEOS — діагностика внутрішніх захворювань загального профілю;

— SPE — проводить діагностування станів при запальних процесах;

— BLUE FOX — діагностика депресивного стану;

— PIP — діагностика ниркових захворювань;

— MYCIN — діагностика інфекційних захворювань;

— GUIDON — навчальна система діагностики інфекційних захворювань;

— ONCOCIN — лікування хворих на рак хіміотерапією та спостереження за ними;

— МОДІС-2 — діагностика симптоматичної гіпертонії;

— CASNET/GLACOMA — здійснює діагностику й лікування захворювань очей, що пов'язані з глаукомою;

— ABEL — діагностика електролітних й кислотних захворювань.

Кожна з вищенаведених систем має вузьку сферу застосування і може використовуватися в ній лише за призначенням.

1.4 Підходи до обробки інформації

Медичні інформаційні системи засновані на різноманітних методах обробки інформації. Зазвичай використовуються наступні підходи [12]:

— побудова дерева рішень;

— статистична обробка даних;

— використання елементів штучного інтелекту.

1.4.1 Побудова дерева рішень

При застосуванні даного методу в програмі фіксується протокол послідовності питань, які задає лікар для вирішення діагностичної проблеми. Такі протоколи будуються у вигляді дерева. Тобто кожна вершина дерева — це конкретне запитання, поставлене пацієнту, а розгалуження з вершини відповідають альтернативним відповідям на питання та призводять до нових запитань. Програма здійснює перехід від питання до питання до того часу, поки не буде знайдено рішення або вичерпано можливі переходи. Проте існують певні недоліки такого підходу:

— при побудові подібних дерев для вирішення складних діагностичних завдань кількість вершин і розгалужень стає настільки великою, що аналіз логічного дерева виявляється занадто складним;

— найменші зміни, які вносяться до логіки програми, призводять до необхідності перебудови дерева заново і перепрограмування всієї задачі.

Водночас такий підхід надзвичайно зручний, оскільки дозволяє уявити в програмі логіку складання лікарем послідовності запитань при вирішенні діагностичного завдання в клінічних умовах. Даний підхід дає можливість моделювати процес прийняття рішення лікарем при встановленні діагнозу.

Оскільки все більше і більше досліджень проводяться в різних галузях, база знань (КБ) продовжує поступово зростати, а в деяких випадках експоненціальна. Використовувати таку велику базу знань і робити висновки рішень, засновані на цих даних, досить складно. Тому це вимагає багато зусиль та вивчення. Рішення, що прийняті на основі наявних знань, складаються з логічних тверджень. Система, що досліджується, використовує наявні знання в конкретних галузях і допомагає користувачам у прийнятті рішень на їх основі. Система спілкуватиметься з користувачами через соціальну платформу (Чат бот у Telegram) і надаватиме рішення на основі накопичених знань.

Знання будуть структурованими, а рішення будуть прийматися на основі цієї структури з урахуванням того, яке рішення чи інформація потрібна користувачеві. Наша система є модульною і може використовуватись у будь-якій сфері. Ми обрали сферу медицини, так як вона має велике коло переваг для суспільства. Галузь медицини має неабияку базу знань про хвороби, симптоми, стани, ліки та сезонні умови. Дана система використовуватиме медичну сферу як свою базу знань і буде намагатись змоделювати першу взаємодію пацієнта з лікарем.

Така система не може замінити і не замінить лікарів, але полегшить діагностику та зробить перший візит пацієнта до лікаря більш продуктивним. Ми розробили систему з метою покращення медичної документації історій хвороб та симптомів, отримання досвіду та надання пацієнтам можливості дистанційно почати попередню діагностику.

1.4.2 Статистична обробка даних

Даний підхід полягає в застосуванні методів математичної статистики. Він заснований на обробці великої кількості зібраної інформації по захворюваннях, що вимагає машинної діагностики. Оброблену інформацію можна використовувати різними способами.

Деякі медичні інформаційно-експортні системи засновані на застосуванні теорії розпізнавання образів. При цьому потрібно мати багато конкретних історій захворювань з відомими діагнозами. Вони аналізуються для визначення статистичних «типових» образів кожної картини захворювання.

Найбільш характерні ознаки захворювання визначаються на основі інформації, зібраної та представленої за допомогою способу статистичної обробки. «Типові» картини хвороб використовуються під час аналізу історій конкретного пацієнта для того, щоб визначити наскільки «схожий» розглянутий випадок на «типовий».

При оцінці «відстані» між картинами, що порівнюються програма генерує рішення щодо діагнозу. Ряд способів комп'ютерної діагностики побудований на використанні теореми Байеса. Даний підхід дає змогу визначити ймовірність захворювання у людини. Він базується на встановленні частоти появи ознак при захворюванні. Так, припустимо, що D_i — одне з n захворювань, яке підлягає діагностиці, а E — ознаки, що виявлені у хворого. Тоді якщо $P(D_i)$ — ймовірність появи i -го захворювання, то ймовірність i -го захворювання за умови наявності ознак E буде

$$P(D_i/E) = \frac{P(D_i)P(E/D_i)}{\sum_{j=1}^n P(D_j)P(E/D_j)} \quad (1)$$

Безперечною перевагою статистичного методу є те, що він використовуються для об'єктивізації наявної інформації про захворювання. Проте даний метод має також суттєві недоліки.

По-перше, це відсутність інформації, необхідної для побудови статистичних моделей. Для накопичення, зберігання та обробки інформації потрібні спеціальні засоби обчислювальної техніки та необхідно створити базу даних захворювань. Обсяг систематизованої інформації в деяких галузях медицини все ще залишається мізерним. Та й сама побудова статистичних моделей — непросте завдання.

Використання теореми Байеса [14] має певні обмеження, оскільки ґрунтується на застосуванні деяких припущень. Перш за все, передбачається, що кожна хвороба має свій унікальний набір симптомів, тоді як на практиці часто трапляється ситуація, коли різні захворювання мають однакові симптоми. Інше припущення застосовується при визначенні умовних ймовірностей появи ознак при захворюванні й базується на незалежності симптомів між собою. Щоправда, так залежності є і вони створюють певні обмеження при використанні теореми.

Наступним недоліком цього методу є внесення нової інформації в моделі, що спричиняє значні труднощі, так як веде до зміни цієї ж моделі і перерахунку попередніх ймовірностей.

Ще одним із недоліків можна виокремити те, що отриманий результат не має нічого спільного зі звичним процесом прийняття рішення лікарем при огляді пацієнта, оскільки даний результат є наслідком здійснення математичних операцій.

1.4.3 Використання елементів штучного інтелекту

Обробка даних, заснована на використанні елементів штучного інтелекту [14]. У таких системах відбувається спроба змодельовати здатність людини розглянути предметну область та зробити висновки з відхиленням найменш перспективних шляхів пошуку. Для цього використовується набір

правил, який задається попередньо і є знаннями експертів в певній проблемній сфері. Саме якістю знань визначається «компетентність» проблемно-орієнтованої інформаційної системи. Формалізовані певним чином знання щодо проблемної області, зберігаються окремо від інших програм. Це дає можливість видозмінювати роботу системи штучного інтелекту, змінюючи набір знань при цьому не зачіпаючи інших складових системи. Сукупність цих правил називається базою знань, а сукупність даних про пацієнтів — базою даних.

Виділяють два основних підходи до організації баз знань, які виявляються в різній структурі відповідних систем:

- опис зв'язків між описаними об'єктами;
- представлення обробки знань у вигляді процедур.

Одним з найвідоміших способів демонстрації знань є продукційний. Такий підхід дозволяє чітко розділити дані, операції та управління. Існує глобальна база даних (не сплутувати з базою даних у традиційному розумінні), над якою виконуються деякі операції, описані набором правил продукції. Цей процес керується відповідно до деяких глобальних політик управління.

Глобальна база даних — це структура даних, які аналізуються та трансформуються системами штучного інтелекту. Тип бази даних залежить від вирішення завдань і може бути як простим (наприклад, структура списку), так і складним.

Правила продукції складаються із двох частин: умова (ліва частина) і дія (права частина). Перша частина визначає застосовність правила до бази даних. Друга полягає у зміні інформації в глобальній базі даних.

Система управління обирає, яке з певних правил використовувати, а також визначає коли зупинити подальші дії шляхом перевірки термінальних умов в базі даних.

Пошук вирішення відбувається на основі двох основних методів. Перший підхід — «від даних до мети», суть якого полягає в послідовній зміні бази даних усіма правилами, що можуть бути застосованими, і в пошуку після

зміни нових застосованих правил. Пошук припиняється при появі твердження, що є рішенням, або при здійсненні умови завершення пошуку. Другий метод — «від мети до даних», який зводиться до перевірки певних тверджень, що можуть виступати як найбільш ймовірні рішення. При цьому здійснюється пошук правих частин правил для того, щоб виявити шукане твердження і перевірка лівих частин відповідних правил на базі даних - при підтвердженні : або гіпотеза є дійсною, або ліва частина буде розглядатись як нова гіпотеза.

1.5 Основи роботи інформаційних систем

У роботі ICET [6] розрізняють два основних режими: режим здобуття знань і режим вирішення задач (режим консультації або режим використання). У режимі придбання знань спілкування з ICET здійснюється експертом за допомогою інженера знань.

Експерти, використовуючи компонент придбання знань, описують проблемну область у вигляді набору фактів і правил. Тобто, «наповнює» ICET знаннями, що дозволить їй самостійно вирішувати задачі у проблемних областях. У традиційних методах програмування даному режиму відповідають наступні етапи: алгоритмізація, програмування та налагодження, які виконує програміст. Тому на відміну від традиційного підходу, у випадку ICET розробка програм буде здійсненна не програмістом, а експертом, який не займається програмуванням.

У режимі консультацій спілкування з ICET здійснюється кінцевим користувачем, який зацікавлений в отриманні результату та /або способі його отримання. Слід зазначити, що в залежності від призначення ICET користувачі можуть:

- бути фахівцями, і в такому випадку вони звертають до ICET для того, щоб прискорити отримання результату, покладаючи на ICET рутинну роботу;
- не бути фахівцями у даній галузі, і в такому випадку вони звертаються до ICET за результатом, який не може отримати сам.

Варто зазначити, що ICET на відміну від традиційних програм при рішенні завдання не лише виконують запропоновану алгоритмом послідовність дій, а й сама попередньо формує її.

Добре побудована й налаштована ICET здатна самонавчатися на вирішенні завдань, автоматично збільшуючи свою базу даних результатами одержаних висновків та рішень.

Особливостями ICET, які відрізняють їх від традиційних програм, є те, що вони повинні володіти [12]:

1) компетентністю;

— досягати прийняття рішень на рівні експертів (тобто мати такий же рівень професіоналізму, як і експерт-людина);

— мати адекватну продуктивність (тобто можливість лише поступового зниження якості роботи в міру наближення до меж діапазону компетентності або допустимої надійності даних);

— бути вмілою (тобто швидко та ефективно використовувати знання, уникаючи зайвих обчислень);

2) можливостями для символічних міркувань;

— демонструвати знання в символічному вигляді;

— вміти переформулювати символічні знання, тобто на жаргоні штучного інтелекту символ — це рядок знаків, пов'язаних із змістом поняття, вони об'єднують, щоб відобразити зв'язки між ними, а зв'язки, представлені в ICET називаються символічними структурами;

3) глибиною;

— вміти застосовувати складні правила (тобто використовувати їх велику кількість чи складні конструкції правил);

— працювати в предметних областях, які містить складні завдання;

4) самосвідомістю;

— проводити дослідження своїх міркувань (тобто перевіряти їх правильність);

— вміти пояснити свої дії.

Існує ще одна важлива відмінність ІСЕТ. Якщо традиційні програми розробляються так, щоб кожного разу сформулювався вірний результат, то ІСЕТ розроблені з тим, щоб вести себе як експерти. Як правило, вони дають правильні відповіді, але іноді, як і люди, можуть помилятися.

Традиційні програми для вирішення складних завдань, також здатні допустити помилку. Проте їх дуже важко виправити, оскільки алгоритми, які закладені в їх основу, явно в них не сформульовані. Отже, знайти і виправити помилки нелегко. ІСЕТ подібні людям і мають потенційну можливість навчатися на своїх помилках.

Технологія розробки ІСЕТ, включає в себе шість етапів (рис.1.1): ідентифікація, концептуалізація, формалізація, виконання, тестування, дослідна експлуатація. Розглянемо більш детально послідовність дій, які потрібно виконати на кожному з цих етапів.

Етап ідентифікації передбачає виконання наступних дій:

- визначення задач та цілей розробки;
- визначення експертів і тип користувачів.



Рисунок 1.1 — Алгоритм розробки інформаційних систем експертного типу

Етап концептуалізації передбачає:

- проведення змістовного аналізу предметної області;

- виокремлення основних понять та їх взаємозв'язків;
- визначення методів вирішення задач.

На етапі формалізації:

- обираються програмні засоби розробки ICET;
- формалізуються основні поняття;
- визначаються методи подання всіх видів знань.

Під час етапу виконання (найважливішого та трудомісткого) експерт наповнює базу даних, при чому процес набуття знань поділяється на:

- "витяг" знання з експертів;
- організацію знань, яка забезпечить ефективну роботу ICET;
- подання знань у вигляді, що буде зрозумілим для ICET.

Набуття знань здійснюється інженерами по знаннях на основі діяльності експертів.

Під час п'ятого етапу тестування експерт разом з інженером по знаннях за допомогою діалогових та пояснювальних засобів здійснюють перевірку компетентності ICET. Процес тестування триватиме до моменту, коли експерт не вирішить, що система отримала необхідний рівень компетентності.

На шостому етапі дослідної експлуатації здійснюється перевірка придатності ICET для кінцевого користувача. За результатами даного етапу можлива значна модернізація ICET.

Процес створення ICET не зводиться до суворої хронології цих етапів, оскільки під час розробки часто доводиться повертатися на попередні етапи та переглядати прийняті там рішення.

Нам також потрібно визначити знання експертів. Ефективність на початкових етапах розробки ICET (ідентифікація та концептуалізація) значною мірою залежить від успішного формування авторитетної експертної групи та отримання від них якісних знань, які становлять основу будь-якого ICET.

Суть процесу виявлення знань полягає в тому, щоб організувати інтуїтивно-логічний аналіз проблемної області експертами та кількісно оцінити їхні судження. На даному етапі експерти:

— формують об'єкти та поняття предметної області (рішення, цілі, альтернативні ситуації тощо);

— здійснюють вимірювання характеристик (ймовірність звернення подій, коефіцієнти важливості цілей, сильні сторони рішень тощо).

Експертна оцінка — це процедура вимірювання, яку можна визначити як процес порівняння об'єктів за обраними характеристиками (показниками). У це визначення входить три поняття: об'єкт, показник (характеристика) і процедура порівняння. Об'єктом може бути предмет, явище, рішення. Як ознаки порівняння використовуються: фізичні, просторово-часові, психічні та інші характеристики об'єктів. До процесу порівняння включають:

— визначення причинно-наслідкових зв'язків між об'єктами;

— встановлення на скільки одні об'єкти впливають на інші. Це вимагає здійснити порівняння об'єктів, які визначають будь-який результат, за рівнем їх впливу на нього. Запровадження конкретних показників порівняння дає змогу експертам встановити зв'язки між об'єктами, до прикладу, "краще", "більше", "більш ніж", "гірше", "однакові" тощо.

Для формального опису багатьох об'єктів та зв'язків між ними вводиться поняття емпіричної системи зі зв'язками:

$$M = \langle O; R \rangle, \quad (2)$$

де $O = (O_1, O_2 \dots O_n)$ — набір об'єктів для предметної області;

$R = (R_1, R_2 \dots R_n)$ — набір зв'язків між ними.

Запис $O_i R_k O_j$ означає, що об'єкт O_i знаходиться у відношенні R_k до об'єкта O_j . Такий зв'язок називається бінарним (двомісний) оскільки пов'язує два об'єкта. Зв'язки можуть бути потрійними тощо. У загальному випадку k -

місцевими, по числу об'єктів, яких вони пов'язують. Основними властивостями відносин є:

- відношення R буде рефлексивним, якщо $O_i R_k O_j$ — це правда;
- відношення R буде симетричним, якщо з $O_i R_k$ слідує $R_k O_j$;
- відношення R буде транзитивним, якщо з $O_i R O_j$ і $O_i R O_k$ слідує $O_i R O_k$

Відношення, що мають властивості рефлексивності, симетричності та транзитивності має назву відношення еквівалентності.

$$O_i \sim O_j \quad (3)$$

При експертній оцінці, окрім відношення еквівалентності використовується також відношення порядку. Таке відношення може означати : "більш ніж", "раніше ніж", "сильніше ніж", "краще ніж" і т.д. Відношення порядку антирефлексивне, транзитивне і позначається:

$$O_i > O_j \quad (4)$$

для випадку, якщо O_i краще O_j .

Під час експертного оцінювання у процесі виявлення знань від експертів, поряд з емпіричною системою відношень використовується також числова система.

$$H = \langle N ; S \rangle, \quad (5)$$

де N — набір дійсних чисел;

S — множина відносин між цими числами.

Саме така система дає змогу не лише встановлювати причинно-наслідковий зв'язок між об'єктами предметної області, а також визначати їх взаємозв'язки і вплив.

1.6 Зв'язок емпіричних і числових систем

При здійсненні експертної оцінки предметної області важливим є здатність для емпіричної системи відношень побудувати числову систему з відношеннями, які описують вплив об'єктів та відносини між ними за допомогою чисел.

Для того, щоб числова система мала можливість зберігати властивості і відношення об'єктів, потрібно, щоб вона була ізоморфною емпіричною системою. Для того, щоб пояснити дане поняття ми спершу дамо визначення поняттю подібності двох систем. Дві системи з відносинами називаються подібними, якщо:

$$M = (O ; R_1, R_2 \dots R_k) \quad (6)$$

$$H = (N ; S_1, S_2 \dots S_m) \quad (7)$$

де, M — місцевість відносин є однаковою (наприклад R_i і S_i двомісні відносини);

N — число відносини, які задані на множині об'єктів і дійсних чисел, однакове, тобто $k = m$;

Визначивши поняття подібності, тепер ми можемо сформулювати визначення ізоморфної числової та емпіричної системи. Числова система з відносинами $H = (N ; S_1, S_2 \dots S_m)$ ізоморфна емпіричній системі $M = (O ; R_1, R_2 \dots R_k)$ за умови якщо дані системи є подібними та існує взаємно однозначне відображення (функція) f об'єктів на числову множину таке, що ставлення R_k між об'єктами має місце тоді і тільки тоді, коли має місце ставлення S_m між числами, які відображають об'єкти на числовій осі.

Наприклад, для випадку двомісних відносин $O_i R_k O_j$ це буде мати місце тоді і тільки тоді, коли має місце $r_i S_k r_j$, де r_i і r_j одержані відображенням об'єктів $r_i = f(O_i)$ і $r_j = f(O_j)$.

Проблема єдності визначає яка кількість способів може описати дану емпіричну систему різними ізоморфними числовими системами і як дані числові системи пов'язані між собою. Така проблема визначається як проблема визначення типу шкали. Шкала — це сукупність:

- емпіричної системи;
- числової системи;
- відображення, тобто $\langle M, H, f \rangle$.

Нехай $\langle M, H, f \rangle$ і $\langle M', H', g \rangle$ дві шкали, які мають різні відображення, тоді постає питання про взаємозв'язок числових значень, отриманих цими відображеннями. Наприклад $r_i = f(O_i)$, $r'_i = g(O_i)$. Зв'язок між числами r_i і r'_i відобразимо за допомогою функції φ : $r_i = \varphi(r'_i)$ або $f(O_i) = \varphi[g(O_i)]$.

Функція φ має назву допустимого перетворення шкали. Єдиність опису емпіричної системи числовими системами виражається у властивостях допустимого перетворення шкали, тобто у властивостях функції φ .

1.6.1 Методи вимірювання ступеня впливу об'єктів

Найбільш часто використовувані методи при експертній оцінці: безпосередня оцінка, парне порівняння та ранжування. При описі кожного з даних методів будемо вважати, що існує кінцеве число вимірюваних об'єктів і сформульований один або кілька ознак порівняння, за якими досліджується рівень впливу об'єктів на результат.

Отже, способи вимірювання будуть відрізнятися лише процедурою порівняння об'єктів. Ця процедура включатиме:

- вибір функції f , яка відобразатиме об'єкти емпіричної системи на числову систему;
- побудову відносин між об'єктами емпіричної системи;
- визначення шкали вимірювань.

Простежитимемо детальніше питання, які виникають при застосуванні кожного з даних методів вимірювань.

1.6.2 Метод ранжирування

Ранжирування — це процес впорядкування об'єктів за ступенем їх впливу на результат, що здійснюється експертами у процесі виявлення його знань [10]. На основі своїх знань та досвіду експерт має впорядкувати об'єкти залежно від їх переваг, керуючись одним або декількома ознаками порівняння. Залежно від типу відносин між об'єктами можливі різні варіанти впорядкування об'єктів.

Припустимо серед об'єктів немає еквівалентних за ступенем впливу на результат. Тоді у такому випадку між об'єктами існує відношення строгого порядку, яке має наступні властивості:

- несиметричність (якщо $O_i \succ O_j$, то $O_j \not\succ O_i$);
- транзитивність (якщо $O_i \succ O_j$, $O_j \succ O_k$, то $O_i \succ O_k$);
- зв'язність (для будь-яких двох об'єктів, або $O_i \succ O_j$, або $O_j \succ O_i$).

У результаті порівняння всіх об'єктів по відношенню строгого порядку експерт складає впорядковану послідовність:

$$O_1 \succ O_2 \succ \dots \succ O_n \quad (8)$$

де, O_1 — об'єкт від номером один є найкращим з усіх;

O_2 — об'єкт під номером два менш кращий ніж перший, але краще усіх інших, тощо.

Дана система з відношенням порядку $\langle O, \succ \rangle$ утворює серію. Для такої серії доведено існування числової системи елементами якої є числа, а відношення порядку f є ставлення "краще ніж" чи "більше ніж".

Це означає, що існує таке числове уявлення $f(O_i)$, що послідовності (8) відповідає послідовність чисел

$$f(O_1) \succ f(O_2) \succ \dots \succ f(O_n) \quad (9)$$

На практиці експертного ранжирування найчастіше використовується послідовність натуральних чисел

$$r_1 = f(O_1) = 1 \succ r_2 = f(O_2) = 2 \succ \dots \succ r_n = f(O_n) = n \quad (10)$$

Числа r_1, r_2, \dots, r_n називаються рангами. Найкращому присвоюється ранг 1, другому - ранг 2 і т.д. На практиці, серед об'єктів можуть бути і еквівалентні за рівнем їх впливу на результат. Наприклад, упорядкування може мати вигляд

$$O_1 \succ O_2 \succ O_3 \sim O_4 \sim O_5 \dots \succ O_{n-1} \sim O_n \quad (11)$$

У цій послідовності об'єкти O_3, O_4 і O_5 еквівалентні між собою, а O_{n-1} і O_n — між собою. Для еквівалентних об'єктів призначають однакові ранги, рівні середнім арифметичним значенням рангів, що приписуються однаковим об'єктам. Ці ранги називають пов'язаними рангами. Для прикладу упорядкування у разі $n = 10$ ранги об'єктів O_3, O_4 і O_5 будуть однаковими і рівними:

$$r_3 = r_4 = r_5 = \frac{(3+4+5)}{3} = 4 \quad (12)$$

$$r_9 = r_{10} = \frac{(9+10)}{2} = 9,5 \quad (13)$$

Як ми бачимо з наведеного прикладу пов'язані ранги можуть бути дробовими. Зручність застосування пов'язаних рангів полягає в тому, що сума рангів n -об'єктів дорівнює сумі натуральних чисел від 1 до n . При цьому будь-які комбінації пов'язаних рангів не змінюють цієї суми, що значно полегшує обробку результатів ранжирування під час групової експертної оцінки.

1.6.3 Метод парних порівнянь

Парне порівняння — це процедура встановлення переваги об'єктів при порівнянні всіх допустимих пар. На відміну від ранжирування, коли

відбувається впорядкування усіх об'єктів одразу, парне порівняння являє собою простішу задачу для експертів. При порівнянні кожної пари об'єктів можливі відносини або еквівалентності, або порядку. Парне порівняння є виміром у шкалі порядку.

У результаті порівняння кожної пари об'єктів O_j та O_i експерту необхідно привести до ладу цю пару, стверджуючи, що:

$$\text{або } O_i > O_j, \text{ або } O_j > O_i \text{ або } O_j \sim O_i .$$

Перехід від емпіричної системи до числової системи з відносинами здійснюється вибором такої функції f , що:

$$\text{якщо } O_i > O_j, \text{ то } f(O_i) > f(O_j) ,$$

$$\text{якщо } O_j > O_i, \text{ то } f(O_j) > f(O_i) .$$

І якщо об'єкти будуть еквівалентними, то безпосередньо можна припустити, що $f(O_j) = f(O_i)$. Найчастіше в практиці експертного оцінювання використовуються наступні числові уявлення (див. таб.1.1)

Таблиця 1.1 — Результати порівняння експертом всіх пар об'єктів

Емпірична система	Подання 1		Подання 2		Подання 3	
	$f(O_i)$	$f(O_j)$	$f(O_i)$	$f(O_j)$	$f(O_i)$	$f(O_j)$
$O_i > O_j$	2	0	1	-1	1	0
$O_i \sim O_j$	1	1	0	0	0,5	0,5

З даної таблиці (таб.1.2) бачимо, що об'єкт O_1 краще об'єктів O_2 , O_3 , O_5 і еквівалентний O_4 . Об'єкт O_2 краще O_3 , еквівалентний O_4 і менш кращий, ніж O_1 і O_5 . Порівняння цих об'єктів у всіх можливих парах не дає повного

упорядкування всіх об'єктів. У результаті цього виникає задача про ранжирування об'єктів на основі парного порівняння.

Таблиця 1.2 — Результати проведеного парного порівняння п'яти об'єктів за допомогою числового уявлення

O_i	O_1	O_2	O_3	O_4	O_5
O_1	1	2	2	1	2
O_2	0	1	2	1	0
O_3	0	0	1	0	1
O_4	1	1	2	1	0
O_5	0	2	1	2	1

1.6.4 Метод безпосередньої оцінки

Безпосередня оцінка — це процедура приписування об'єктам числових значень у шкалі інтервалів. Такі значення відповідають ступеню впливу того чи іншого об'єкта на спостережуваний результат.

У процесі виявлення знань експерту необхідно поставити у відповідності кожному об'єкту точку на безперервній числової осі, наприклад, на відрізку $[0,1]$. Безсумнівно очікувати, щоб еквівалентним по впливу об'єктів приписувалося б одне і теж число.

Вимірювання переваги в шкалі інтервалів можна здійснити з високим ступенем довіри лише за хорошої інформованості експерта щодо властивостей об'єктів та предметної області.

У ряді випадків, аби природно ослабити ці умови за рахунок зменшення точності вимірювання замість безперервної числової осі розглядають велику оцінку, у якій застосовуються 5, 10, 100 - бальні шкали.

Проте безпосередня оцінка не завжди має використовувати числові шкали. Зокрема, колір об'єкта неможливо уявити у вигляді будь-якого

числового значення, а перехід до значень частот спектра в багатьох випадках важкий для експерта.

При формуванні експертної групи на етапі виявлення знань необхідно враховувати наступні характеристики експертів :

- компетентність — рівень кваліфікації експерта у даній галузі знань;
- креативність — здатність до вирішення творчих завдань;
- конформізм — схильність до впливу авторитетів, за якого думка авторитету може пригнічувати осіб, які більш високою -компетентні;
- ставлення до експертизи (негативне чи пасивне);
- самокритичність та колективізм.

Розглянемо один із способів кількісного опису характеристик експерта, який побудований на розрахунку відносних коефіцієнтів компетентності за результатами висловлювання фахівців щодо складу групи експертів.

Зміст даної методики полягає в тому, що ряду фахівців пропонують висловити думку про особовий склад групи експертів. Якщо в цьому переліку з'являться особи, які не ввійшли в вихідний список, їм також пропонують назвати фахівців для участі в експертизі. Після декількох етапів буде отриманий повний список претендентів в групу.

За результатами такого опитування складається матриця, по рядках і стовпцях якої записуються прізвища експертів, а елементами таблиці є змінні $x_{ij} = (1, \text{якщо } j\text{-ий експерт назвав } i\text{-го}; 0, \text{якщо } j\text{-ий експерт не назвав } i\text{-го})$

При цьому експерт може включати себе або не включати до експертної групи (тобто $x_{ij} = 0$ або $x_{ij} = 1$). У даній таблиці можемо обчислити відносні коефіцієнти компетентності, використовуючи алгоритм рішення задач про лідера.

1.7 Характеристика та режими роботи групи експертів

Головними характеристиками експертної групи є достовірність експертизи та витрати, пов'язані з нею. Ці показники визначають кількість експертів у групі та якість їх роботи.

Разом з компетентністю, кожен з експертів може характеризуватися достовірністю його суджень. Ця ознака визначається на основі інформації про його минулий досвід участі у вирішенні проблем. Кількісно достовірність експерта оцінюють за формулою:

$$D_i = \frac{Nn_i}{N_i} \quad (i = 1, 2 \dots m) \quad (14)$$

де N_i — загальне число випадків участі i -ого експерта в експертизі,

Nn_i — число опитувань, коли експерт дав прийнятне практикою рішення.

Можна враховувати внесок кожного експерта в достовірність всієї групи. Відносна достовірність виражається формулою:

$$D_i^{om} = \frac{D_i}{\frac{1}{m} \sum_{i=1}^m D_i} \quad (15)$$

де m — число експертів в групі, а в знаменнику — середня достовірність експертної групи.

Для рішення проблем з високим ступенем інформаційного потенціалу знань, збільшення кількості експертів в групі призводить, як це впливає з теорії обробки спостережень, до одноманітного зростання достовірності експертизи.

Експериментальні дослідження підтверджують цю залежність (рис.1.2).

Опитування є основним етапом спільної роботи експертної групи. При приведенні колективної експертизи застосовуються наступні види опитування:

- дискусія;
- мозковий штурм;
- колективна генерація ідей;
- інтерв'ювання та анкетування.

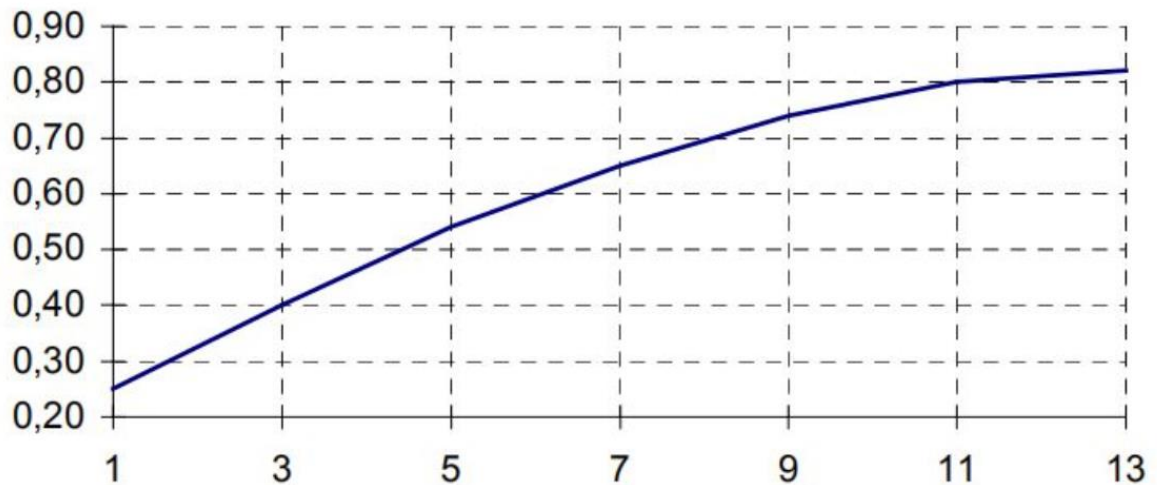


Рисунок 1.2 — Достовірність експертизи від кількості експертів

Результатом опитування є інформація, що відображає переваги експертів і змістовно підтверджує ці переваги. Наявність числових та змістовних висловлювань експертів призводить до необхідності використання як якісних, так і кількісних методів обробки результатів оцінки експертної групи.

1.8 Обробка експертних оцінок

У залежності від мети експертного оцінювання і методів обліку експертних оцінок виникають такі основні завдання:

- побудувати загальну оцінку понять і об'єктів на основі особистих оцінок експертів;
- побудувати узагальнене оцінювання на основі парного порівняння об'єктів кожним експертом;
- визначити узгодженість думок експертів;
- визначити відносну вагомість взаємозв'язку об'єктів;
- визначити залежності між ранжируванням;
- оцінити надійність обробки результатів.

При розв'язуванні багатьох задач недостатньо впорядкувати об'єкти по одному показнику чи їх групі. Потрібно також мати числові значення для кожного з об'єктів, що визначають його сильну сторону у порівнянні з іншими

об'єктами. Наявність такого оцінювання дасть змогу визначити загальну оцінку в цілому для всієї групи експертів.

Визначення узгодженості думок фахівців відбувається за допомогою обчислення числових заходів, що визначає ступінь близькості індивідуальних міркувань. Проведення аналізу значення заходів узгодження сприяє формуванню вірної думки про загальний рівень знань по вирішенню проблеми та виявлення думок експертів.

Обробка експертних оцінок дає змогу розкрити пов'язані ознаки порівняння та виконати групування за ступенем зв'язку. Наприклад, якщо показниками порівняння є різні цілі, а об'єктами порівняння є засоби досягнення цих цілей, то встановлення взаємозв'язку між ранжируванням, що впорядковують засоби з точки зору досягнення цілей, дає змогу змістовно дати відповідь на питання: "яким чином досягнення однієї цілі при даних засобах сприяє досягненню іншої мети" (тобто допомагає встановити причинно наслідковий зв'язок).

Оцінки, отримані на основі обробки, є випадковими об'єктами, тому однією з найважливіших задач процесу обробки є визначення їх надійності.

1.9 Байєсова мережі довіри

Байєсівські мережі довіри [13] - Bayesian Belief Network — мережі, що застосовуються в тих областях, які характеризуються успадкованою невизначеністю. Така невизначеність може виникнути внаслідок:

- неповних знань;
- часткового розуміння предметної області;
- випадкового характеру задачі.

Таким чином, байєсівські мережі довіри (БСД) використовують для імітації ситуацій, що мають у деякому сенсі невизначеність. Для байєсівських мереж довіри іноді застосовується ще одна назва причинно-наслідкової мережі, в якій випадкові події з'єднуються причинно-наслідковими зв'язками.

З'єднання за допомогою методу причин і наслідків дозволяють простішим способом оцінити ймовірності подій. У сучасному світі оцінювання найчастіше здійснюється в напрямку від "спостерігача" до "спостереження", або від "ефекту" до "наслідку", яке в загальному випадку важче оцінити, ніж напрямком "наслідок > ефект", тобто в напрямку від наслідку.

На рис.1.3 розглянемо приклад мережі, в якій ймовірність перебування вершини e в різних станах (e_k) буде залежати від станів (c_i, d_j) вершин d і c та визначається виразом

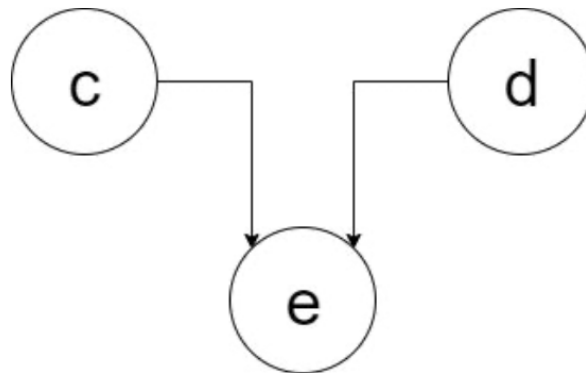


Рисунок 1.3 — Приклад найпростішої байєсівської мережі довіри

$$p(e_k) = \sum_i \sum_j p(e_k | c_i, d_j) \times p(c_i, d_j) \quad (16)$$

де $p(e_k | c_i, d_j)$ - ймовірність перебування в стані e_k залежно від станів c_i, d_j . Так як події, що зображені вершинами c і d незалежні, то

$$p(e_k | c_j, d_j) = p(c_i) \cdot p(d_j) \quad (17)$$

Розглянемо також приклад більш складної мережі (див. рис.1.4). Даний рисунок демонструє умовну незалежність подій. Щоб оцінити вершини c і d використовуються ті ж висловлювання, що й для обчислення $p(e_k)$, тоді:

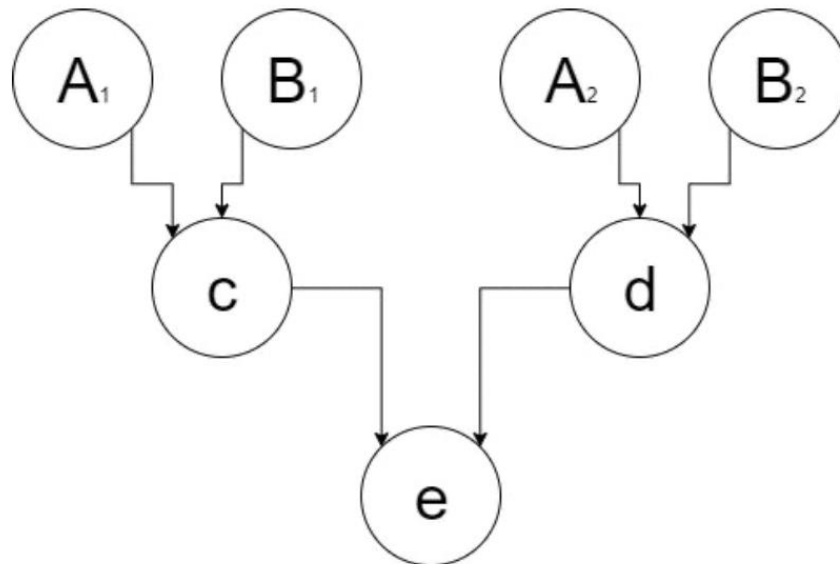


Рисунок 1.4 — Дворівнева БМД

$$p(c_i) = \sum_m \sum_n p(c_i | A_{1m}, B_{1n}) \times p(A_{1m}) \times p(B_{1n}) \quad (18)$$

$$p(d_j) = \sum_m \sum_n p(d_j | A_{2m}, B_{2n}) \times p(A_{2m}) \times p(B_{2n}) \quad (19)$$

З даних виразів бачимо, що вершина «e» умовно є незалежною від вершин A_1 , A_2 , B_1 , B_2 , так як немає стрілок, що безпосередньо з'єднують ці вершини. Розглянувши приклади спробуємо більш точно дати визначення основним поняттям, які застосовуються в БМД. Байєсовські мережі довіри - це спрямований ациклічний граф, який має наступні властивості:

— всі вершини являють собою події, які описуються випадковою величиною, що може мати кілька станів;

— всі вершини, які пов'язані з "батьківськими" визначаються за допомогою таблиці умовних ймовірностей (ТУЙ) або функції умовних ймовірностей (ФУЙ);

— для вершин, які не пов'язані з "батьківськими", ймовірності її станів є маргінальними (безумовними).

Іншими словами, вершини в байєсовських мережах довіри є випадковими змінними, а дуги — імовірнісними залежностями, які визначаються через таблиці умовних ймовірностей. Таблиця умовних

ймовірностей кожної вершини містить ймовірності станів цієї вершини за умови станів її "Батьків".

Також розглянемо випадок побудови найпростішої байєсівської мережі довіри. Візьмемо за приклад невелику яблучну плантацію «яблучного Джона». Одного разу Джон помітив, що його яблуня втратила листя. Тому він хоче визначити причину чому це сталося. Джон знає, що листя може опадати у випадку:

- якщо недостатня кількість води і дерево засихає;
- якщо дерево хворіє чи ушкоджене паразитами.

Дану ситуацію можемо змодельовати за допомогою байєсівської мережі довіри (рис.11.5), яка містить 3 вершини: «Хворіє», «Засохло» і «Опало».

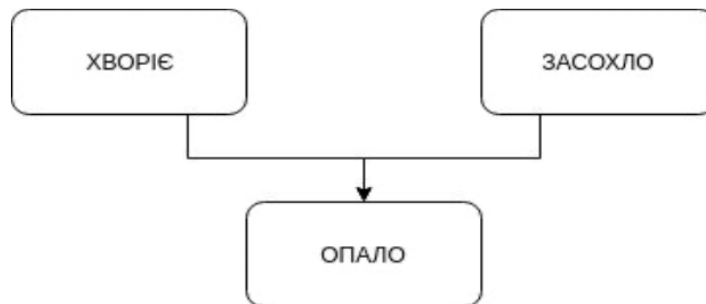


Рисунок 1.5 — Приклад байєсівської мережі довіри з трьома подіями

У такому випадку розглянемо ситуацію, коли кожна вершина може прийняти лише два можливих стани і, як наслідок знаходиться в одному з них (див. таб.1.3):

Таблиця 1.3 — Перелік вершин БМД

Вершина (подія) БМД	Стан 1	Стан 2
"Хворіє"	«хворіє»	«ні»
"Засохло"	«засохло»	«ні»
"Опало"	«так»	«ні»

Під вершиною "Хворіє" мається на увазі, що дерево захворіло, будучи в

стані «Хворіє», в іншому випадку воно буде знаходитись у стані «ні». Аналогічно і для інших двох вершин. Вже згадана байєсова мережа довіри, моделює той факт, що є причинно-наслідкова залежність від події "Хворіє" до події "Опало" і від події "Засохло" до події "Опало". На байєсівській мережі довіри це відображено стрілками.

Коли є причинно-наслідкова залежність від вершини А до іншої вершини В, то ми очікуємо, що коли А знаходиться в деякому певному стані, це впливає на стан В. Варто зауважити, коли моделюється залежність в байєсовських мережах довіри іноді зовсім не очевидно, який напрямок повинна мати стрілка.

Наприклад, у розглянутому прикладі, ми говоримо, що існує залежність від "Хворіє" до "Опало", оскільки хвороба дерева може викликати опадання листя. Таким чином, опадання листя є наслідком хвороби, а не хвороба - наслідком опадання листя.

На вищенаведеному рисунку відображено графічне уявлення байєсівської мережі довіри. Проте, це лише якісне представлення байєсівської мережі довіри. Перед тим, як назвати це повністю байєсівською мережею довіри необхідно визначити кількісне уявлення, тобто множину таблиць умовних ймовірностей (таб.1.4 - таб.1.6):

Таблиця 1.4 — Априорна ймовірність $p(\text{«Хворіє»})$

Хворіє – «хворіє»	Хворіє – «ні»
0,1	0,9

Таблиця 1.5 — Априорна ймовірність $p(\text{«Засохло»})$

Засохло – «засохло»	Засохло – «ні»
0,1	0,9

Таблиця 1.6 — Таблиця умовних ймовірностей p («Опало» | «Хворіє», «Засохло»)

	Засохло – «засохло»		Засохло – «ні»	
	Хворіє = «хворіє»	Хворіє = «ні»	Хворіє = «хворіє»	Хворіє = «ні»
Опало = «так»	0,95	0,85	0,90	0,02
Опало = «ні»	0,05	0,15	0,10	0,98

Наведені таблиці демонструють ТУЙ для трьох вершин байєсівської мережі довіри. Варто зауважити, що всі три таблиці показують ймовірність перебування деякої вершини в певному стані, обумовленим станом її батьківських вершин. Але так як вершини "Хворіє" та "Засохло" не мають батьківських вершин, то їх ймовірності є маргінальними (не обумовленими), тобто ні від чого не залежать .

На даному прикладі ми розглянули як описуються дуже прості вершини байєсівської мережею довіри. Сучасні програмні засоби (такі як Hugin, MSBN та ін.) забезпечують інструментарій для побудови цих мереж, а також здатність використовувати байєсовські мережі довіри для уведення нових свідчень та отримання рішення (висновку) за рахунок перерахунку нових можливостей у всіх вершинах, відповідних знову введенням свідчень.

Припустимо у нашому прикладі відомо, що дерево скинуло листя. Це свідчення вводиться за допомогою вибору стану «так» у вершині "Опало". Після цього можна дізнатися ймовірності того, що дерево засохло. Для наведених вище вихідних даних, результати виведення шляхом поширення ймовірностей по БСД будуть: p («Хворіє» = «хворіє» | «Опало» = «так») = 0,47; p («Засохли» = «засохло» | «Опало» = «так») = 0,49

1.10 Діаграми впливу

Діаграми впливу застосовуються для прийняття рішень. Фактично діаграми впливу — це байєсовські мережі довіри, які розширені поняттями користі (Utility) та рішення (decisions). Якщо байєсовські мережі довіри містять лише один тип вершин, які ми назвемо вершинами шансів, і які відповідають станом випадкових змінних, то в діаграмах впливу використовуються ще, як мінімум, два типи вершин (рис.1.6): вершини користі, позначені в діаграмах впливу у вигляді ромба та вершини рішення, позначені в діаграмах впливу прямокутниками.

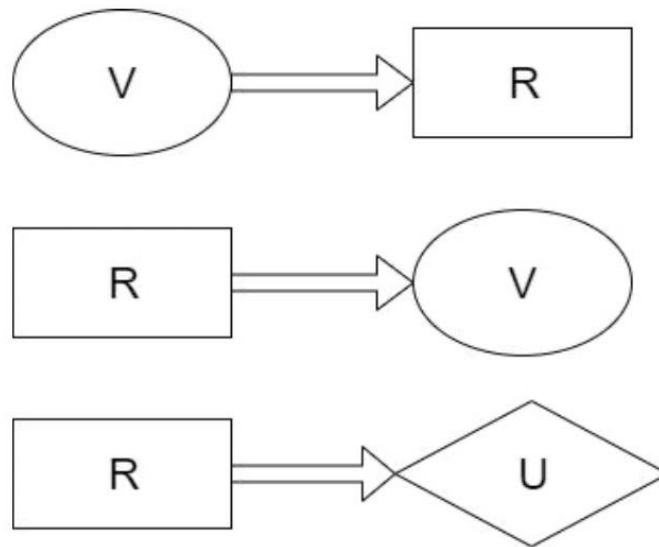


Рисунок 1.6 — Приклад вершин рішень

Вершини рішення, а точніше вказівки, які містяться в них, визначають тимчасове старшинство:

— стрілка від вершини шансів (випадкової змінної) до вершини рішення (змінної рішення) вказує, що значення випадкової змінної відомо на момент прийняття рішення;

— стрілка від змінної рішення до будь-якої іншої змінної вказує час, який впорядкований рішенням.

При цьому мережа має бути ациклічною і повинен існувати безпосередній шлях, що містить всі вершини рішення в мережі.

У процесі прийняття рішень важливим є не просто знайти рішення, а знайти в якомусь сенсі найкраще рішення. Тому з цією метою в діаграмах впливу «вершини користі» зв'язуються зі станом мережі.

Кожна з вершин користі містить функцію корисності, яка пов'язує кожну конфігурацію стану її батьків з корисністю. Оскільки вершини користі не мають спадкоємців, то стрілка може бути спрямована лише до них (рис.1.7).

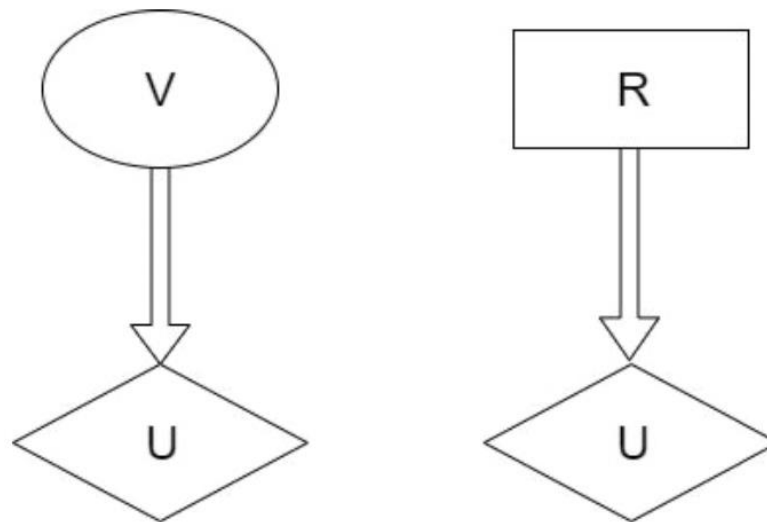


Рисунок 1.7 — Приклад вершин рішень

Тобто приймаючи рішення ми виходимо з ймовірності конфігурації мережі. Тому можемо обчислити очікувану корисність кожної альтернативи та обрати ту альтернативу, яка має найбільш очікувану корисність. Такий принцип має назву максимальної очікуваної корисності. Діаграма впливу може містити в собі декілька вершин користі. При цьому загальна функція корисності являє собою сукупність всіх локальних функцій корисності.

$$F(x_1, \dots, x_n) = F(x_1, \dots, x_n) = \sum^k f_1(x_1, \dots, x_n) \quad (20)$$

Процедура прийняття рішень за допомогою використання діаграми впливу здійснюватиметься в наступному алгоритмі:

— після спостереження значень змінних, які є батьками першої вершини рішення, ми хочемо дізнатись максимальну корисність для альтернатив;

— ICET буде враховувати ці корисності в припущенні, що всі майбутні рішення будуть зроблені оптимально з використанням всіх наявних свідчень у моменті прийняття кожного з рішень.

2 ТЕХНОЛОГІЇ ДЛЯ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ МЕДИЧНОГО ЕКСПЕРТУ НА ОСНОВІ ЧАТ-БОТА

2.1 Перспектива та функції продукту

Інформаційні системи експертного типу існують вже досить давно, але їх застосування в дослідженнях історії хвороби є новими. Ця система буде розроблена з нуля, черпаючи натхнення від різних системних експертів і чат-ботів, які були там раніше.

Медична експертна система на основі чат-боту пацієнту буде повідомлено, а його симптоми будуть зафіксовані. Ці симптоми потім будуть повідомлені вашому лікарю для подальшого аналізу.

Система є інструментом, який допомагає лікарям збирати історію хвороби, щоб краще використовувати час, який часто витрачається на це завдання.

Зі сторони користувача, характеристики продукту можна записати як:

— основним інтерфейсом пацієнта є текстовий чат-бот.

— користувачі взаємодіють із системою через додаток Telegram, що дозволяє їм вводити текст і надсилати його на веб-сервер.

Є два основні клієнти — пацієнт і лікар. Пацієнти зв'яжуться з системою, щоб повідомити про свої симптоми, і система буде реагувати, щоб викликати реакцію та подальше тестування на симптоми. Завданням системи буде діагностика захворювання пацієнта, а також збір відповідних даних щодо інших особистих параметрів, таких як вік, стать, зріст і вага.

Іншими користувачами системи будуть лікарі. Вони будуть взаємодіяти з системою електронної пошти. Повідомлення електронною поштою з детальною інформацією про симптоми пацієнта та можливі діагнози на основі зібраної інформації будуть надіслані лікарям, коли пацієнт успішно завершить чат із системою.

Третій користувач системи — адміністратор. Роль адміністратора полягає в тому, щоб дозволити лікарю вивчити деталі конкретного пацієнта.

Оскільки ваша історія хвороби дуже особиста, лише адміністратор (який буде довіреною особою компанії для впровадження цієї послуги) зможе видавати схвалення лікаря. Адміністратор може спілкуватися з системою через командний рядок.

2.2 Класи та функції користувача

В основному це буде 3 класи користувачів:

1) користувачі;

— основний користувач системи, пацієнти спілкуватимуться з системою через інтерфейс чату;

— вони повідомляють про свої симптоми в систему, і система задає їм відповідні запитання для встановлення діагнозу;

— вони спілкуватимуться з системою та братимуть участь у чатах, щоб визначити точне захворювання, пацієнти є найважливішими користувачами системи;

2) лікарі;

— лікарі використовуватимуть систему для доступу до даних пацієнтів.

— вони зможуть переглядати всіх пацієнтів, до яких мають доступ, а також детальну інформацію про кожного пацієнта;

3) адміністратор;

— адміністратор відповідає за обслуговування та утримання системи;

— адміністратор буде єдиною особою, яка зможе встановити конкретні авторизації даних пацієнтів для лікарів.

2.3 Обмеження проєктування та провадження

Головна турбота — збереження конфіденційності даних користувачів. Тільки конкретні лікарі повинні мати доступ до даних про симптоми певних пацієнтів, і це також за їх згодою. Доступ до цих даних буде обмежено головним технічним персоналом системи. Таким чином, дані повинні

зберігатися в дуже безпечний спосіб, їх не можна поширювати без явної юридичної згоди пацієнтів.

Інтерфейс системи буде залежати від мобільного додатка Telegram. Нам потрібно відповідати стандартам Telegram для клієнт-серверного чату, серверний код буде написаний на Python, а база даних буде Redis. В основному ми будемо використовувати HTTP для отримання та розміщення запитів на надсилання та отримання даних через мережу.

2.4 Вимоги до зовнішнього інтерфейсу

Telegram створив інтерфейс між пацієнтом та системою, який ми будемо використовувати для отримання інформації про пацієнтів та передачі їм відповідної інформації.

Він має текстове поле для введення тексту, вікно чату, яке відображає попередні розмови, клавіатуру для введення відповідей і кнопку «відправити» для надсилання цієї інформації.

Єдина інформація, яку лікар очікує від системи – це інформація про своїх пацієнтів, вона буде доставлено їм через їх електронну пошту і це — інтерфейс між лікарем та системою.

За пацієнтами адміністратор закріплює лікарів. Це буде зроблено через інтерфейс командного рядка — інтерфейс між адміністратором та системою.

Апаратним інтерфейсом (основним інтерфейсом) між пацієнтом і системою, а також лікарем і системою має бути смартфон, на якому можна запусити додаток Telegram. Тому це має бути телефон, який може працювати на відповідній версії Android, iOS або Windows Phone. Телефон повинен підтримувати підключення до Інтернету та відповідати останнім стандартам HTTP.

Звичайно, лікарю знадобиться подібний комп'ютер, підключений до Інтернету та електронної пошти.

2.5 Інтерфейси програмного забезпечення

Проект складається з двох основних частин — однієї на клієнті (інтерфейс чату) і іншої на сервері (чат-бот). Клієнтська частина системи покладається на мобільний додаток Telegram, який має встановити двосторонній зв'язок із програмним забезпеченням на сервері, щоб система працювала належним чином. Також є база даних чатів ботів, які беруть участь. Більшість вимог до програмних інтерфейсів відповідають цій архітектурі. Вони поділяються на дві категорії: ті, які використовуються клієнтами, і ті, які використовуються серверами.

2.5.1 Інтерфейси програмного забезпечення на стороні клієнта

Інтерфейси програмного забезпечення на стороні клієнта:

- операційна система на платформі Android (v3.1 або новішої) або на платформі iOS (v7.0 або новішої) або Windows Phone (v8 або новішої), та desktop версії Windows (7 або новішої), MacOS (v11 або новішої);

- telegram v6.1 або новішої версії (Android), v1.15 або новішої версії (Windows Phone), > v6.2.1 (iOS);

- мережевий стек Http/https (TCP / IP);

- клієнт надсилатиме текстові повідомлення з пакетом HTTP на сервер і отримуватиме текст і розкладку клавіатури телеграм у вигляді відповідей.

2.5.2 Інтерфейс ICET на стороні сервера

Інтерфейс ICET на стороні сервера:

- екземпляр UNIX з підтримкою Інтернету, який підтримує HTTP, автономно або в хмарі;

- redis;

- різні бібліотеки Python, такі як python-telegram-api, багатопроцесорні завдання тощо;

Сервер отримує дані у вигляді HTTP-пакета, що містить JSON, який буде переносити текстове повідомлення та облікові дані користувача. Він надсилає HTTP-пакети, що містять JSON, з деталями відповіді з чату бота.

2.6 Особливості системи

2.6.1 Чат-бот

Це ядро проекту з найвищим пріоритетом. Це серце проекту, який буде спілкуватися з пацієнтами природною мовою. Він забезпечить умови для отримання тексту від користувача, обробки накопичення його відповідей і відправки його по мережі.

Інгредієнти високого ризику. Це єдиний спосіб для пацієнтів ввести дані; будь-які помилки тут відобразатимуться по всій системі.

2.6.2 Стимули послідовностей відповіді

Чат системи з пацієнтом наступний.

Стимул — пацієнт надсилає текстове повідомлення з описом свого стану.

Відповідь — чат-бот отримує інформацію про користувача з бази даних в активну пам'ять для подальшої роботи та обробки. Система запускає чат і намагається діагностувати захворювання пацієнта. Це передбачає читання/запис копій бази даних і модулів бізнес-логіки. Після того, як чат досягне стану "зроблено", користувачеві надсилається повідомлення про це.

Стимул — пацієнт відповідає на поставлені системою запитання за допомогою клавіатури Telegram у додатку.

Відповідь — чат-бот записує цю відповідь і після виконання модуля з бізнес-логікою надсилає відповідну відповідь користувачеві. Це може бути ще одне запитання, яке пацієнт повинен запитати для отримання додаткової інформації або оголосити розмову закінченою.

Взаємодія системи з лікарем виглядає так.

Стимул — пацієнт, який дозволив конкретному лікарю переглядати свої дані, щойно успішно завершив розмову з системою..

Відповідь — система отримає відповідну інформацію про історію хвороби пацієнта та поверне її лікареві в доступному та зрозумілому форматі. Інформація буде надіслана лікарю електронною поштою. Відповідь зазвичай полягає в тому, щоб отримати більше інформації про конкретного пацієнта.

Взаємодія адміністративної системи наступна.

Стимул — адміністратор хоче дати можливість лікарям терпляче отримувати інформацію про це.

Відповідь — система пов'язує дані пацієнтів із пацієнтами та надає лікарям доступ до інформації про пацієнтів.

2.7 Інтерфейс чату та функціональні вимоги

Серверний модуль відповідає за отримання та відправку повідомлень шляхом взаємодії з сервером Telegram. Крім того, він відповідає за відправку повідомлень модулям інформаційної системи системи та отримання на них відповідей;

Модуль зв'язку бази даних відповідає виключно за читання/запис реплік бази даних. Модуль бази даних надає чат-боту відповідні дані користувача та зберігає отримані дані в базі даних;

Модуль диспетчеризації виконує роль маршрутизатора. У самостійній розмові виділяє предмет, зберігає дані запитань і відповідей. Це статичне сховище даних для зберігання даних запитань і відповідей;

Основний модуль, який знає стан людини, створюється на основі сеансу, і знає стан сеансу в будь-який момент.

Модуль експертної системи розташований між базою та модулем Doctor Sky Net. Цей модуль оброблятиме діалог та стан зв'язку між Core та Doctor SkyNet;

Модуль Doctor SkyNet — базовий модуль, який містить різні важливі алгоритми для реальної роботи системи.

Це друга найважливіша частина проєкту, окрім самого чату бота. Інтерфейс чату буде шлюзом для входу користувачів до системи. Зокрема, для цього будемо використовувати мобільний додаток «Телеграм». Telegram дозволяє нам змінити інтерфейс чат-бота, який може надсилати та отримувати повідомлення користувача. Однією з його основних функцій є те, що він дозволяє користувачеві відображати спеціальну клавіатуру, яка може містити лише певні кнопки. Це корисно для обмеження введення користувача до дуже конкретних відповідей і забезпечує важливу взаємодію керування між користувачем і чат-ботом.

2.7.1 Базові функції

Доступні базові функції:

- приймати необроблений текст, який ввів користувач;
- перетворювати отриманий текст в JSON і передавати його по мережі в чат-боті;
- отримувати відповіді на текстові повідомлення від чат-ботів і надсилати їх користувачам;
- отримувати макет спеціальної клавіатури від чату-бота та подавати її користувачеві;
- отримувати відповіді на цій спеціальній клавіатурі та передавати її чат-боту.

2.7.2 Ризики

Більшість ризиків, пов'язаних з цією частиною проєкту, пов'язані з програмою Telegram. Наприклад, що робити, якщо компанія, що стоїть за Telegram, вирішить заборонити взаємодію з чат-ботами? Нам потрібно знайти інший спосіб увімкнути інтерфейс чату.

Інший ризик — втрата вашого інтернет-з'єднання. Але жоден з цих ризиків не буде фатальним для проєкту, оскільки чат-бот розроблений як агностична передня частина (у цьому випадку Telegram). Він легко

адаптується до веб-інтерфейсу навіть за допомогою спеціальних мобільних додатків.

2.7.3 Стимули послідовності відповідей

Коли пацієнт починає розмову.

Стимул — пацієнт або лікар висловлюють намір почати розмову, відкриваючи програму Telegram і починаючи вводити текст у текстовому полі.

В: Інтерфейс чекає, поки користувач не натисне кнопку «Надіслати» у програмі. Як тільки це буде зроблено, текст, введений за допомогою його/її облікового запису користувача, буде упаковано в пакети HTTP і надіслано по мережі до чат-бота.

Бесіда пацієнта вже триває.

Стимул — пацієнт або лікар через інтерфейс чату отримує повідомлення від чат-бота, яке пропонує йому ввести відповідь. Користувач повинен відповідати за допомогою спеціальної клавіатури і може надсилати лише конкретні відповіді.

Відповідь — інтерфейс чату надсилає цю відповідь у вигляді HTTP-пакета чат-боту, де чат-бот може виконувати свою роботу.

2.8 Інші нефункціональні вимоги

2.8.1 Вимоги до продуктивності

Продуктивність здебільшого стосується користувачів, які можуть отримати відповідь на свій внесок у розумний проміжок часу. Щоб вийти з системи чату, потрібно близько 5 секунд.

Час доставки тут не враховується, оскільки залежить лише від користувача та швидкості мережі. Таким чином, алгоритми та апаратне забезпечення, що використовуються в системі, повинні забезпечити відповідь системи протягом 5 секунд після отримання повідомлення користувача.

2.8.2 Вимоги до безпеки

Під час обробки та захисту даних пацієнтів необхідно дотримуватись максимальної безпеки. Особливо важливо забезпечити доступ до цих даних таким чином, щоб їх могли прийняти лише лікарі, пов'язані з пацієнтом.

Це занепокоєння пов'язано насамперед із юридичними та етичними обмеженнями, які виникають під час обробки персональних даних та даних історії хвороби.

Надзвичайно важливо повністю переконатись у тому що особисті дані пацієнта, які будуть зібрані та збережені у системі, є дуже захищені і до них не має несанкціонованого доступу.

Оскільки додаток Telegram використовує цей інтерфейс, використовуватиметься унікальне ім'я користувача Telegram.

2.9 Моделювання проєктних процесів

Для реалізації проєкту буде доречною модель інкрементного процесу. Тому що ми плануємо почати тестування на прості захворювання та симптоми, а потім збільшувати тип і складність захворювань. Ми очікуємо надавати ітерації програмного забезпечення шляхом підвищення продуктивності або додавання нових функцій на кожній ітерації.

Дослідження та дослідження комунікацій — ці заходи включають опитування поточних досліджень та існуючих методологій, а також спілкування з керівниками проєктів для покращення концепцій проєкту.

Завдання — дослідіть ідею та дослідіть існуючі підходи. Оцініть плюси і мінуси. Збірник відповідних журналів/наукових праць. Поясніть ідею до посібника з проєкту та прокоментуйте її.

Результати — порівняйте всі існуючі методи та виберіть метод, використаний у проєкті.

Планування — планування проєкту, включаючи розробку розкладів, завдань космічної техніки, аналіз доцільності та ризику тощо.

Наступне завдання — реалізуйте частину проєкту із заходами протягом певного періоду часу, розподіліть обов'язки між членами команди та придбайте ресурси та інструменти програмування, необхідні для реалізації проєкту.

Результати — таблиця ризиків, техніко-економічне обґрунтування з використанням математики, вибір симулятора, хронологічна діаграма проєкту.

Проєктування та моделювання, тобто система, що розробляється, повинна бути розроблена з використанням діаграм UML.

Завдання — декомпозиція системи для ідентифікації компонентів, ідентифікація акторів системи, ознак, які повинні і не повинні бути присутніми, взаємодії між компонентами, даних потоку, ідентифікації класів.

Результати — діаграма архітектури системи, схема компонентів, варіант використання, діаграма класів, блок-схема.

Програмування, як схема компонентів, діаграма архітектури системи, блок-схема, діаграма класів.

Завдання — відповідні члени команди виконують кодування компонентів, взаємодію компонентів та тестування на основі проєктів, затверджених у попередніх заходах.

Результати — основна реалізація програмного забезпечення на першій ітерації, результати тестування функцій, виявлення помилок і збоїв (якщо є).

ЗАНАЛІЗ ТА ПРОЄКТУВАННЯ ЧАТ-БОТА

3.1 Діаграми архітектури інформаційної системи

Діаграма архітектури системи показує модулі та взаємодії всієї системи з одного погляду (рис. 3.1). Зауважте, що користувачі ніколи не мають прямого контакту з лікарями, а різні взаємодії між лікарями та користувачами завжди відбуваються через чат-ботів.

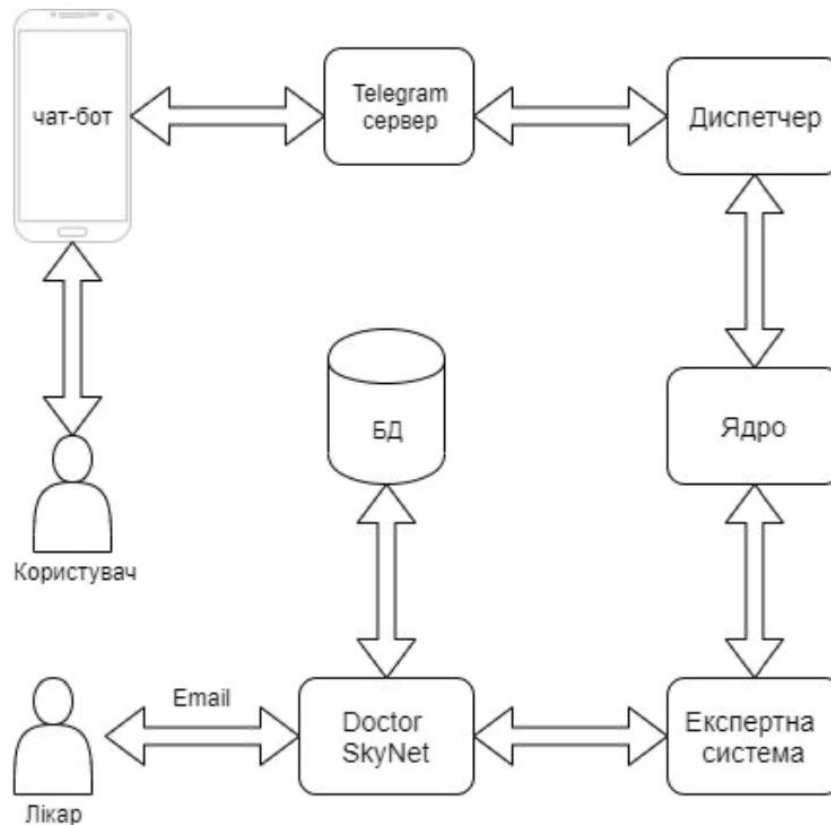


Рисунок 3.1 — Діаграма архітектури системи

Діаграма, що показує загальну архітектуру системи та взаємодії, які необхідно розмістити між різними модулями. Користувачі ініціюють розмови з системою, надсилаючи повідомлення, які фіксуються серверами Telegram, а потім надсилаються чат-боту через різні модулі. Почніть розмову з користувачем, а потім припиніть розмову, коли лікар отримає електронний лист від чат-бота про користувача

3.1.1 Діаграми використання

"Use Case" діаграма розкриває візуально взаємодію між різними модулями та суб'єктами системи (рис.3.2). У цьому випадку ви можете побачити найважливіші модулі чату бота та акторів, з якими вони взаємодіють, основними системними модулями, з якими взаємодіють актори, є інтерфейс Telegram, база даних Redis.

Діагностика ІСЕТ-системи лікаря та експерта SkyNet. Головними дійовими особами є користувач, лікар і адміністратор.

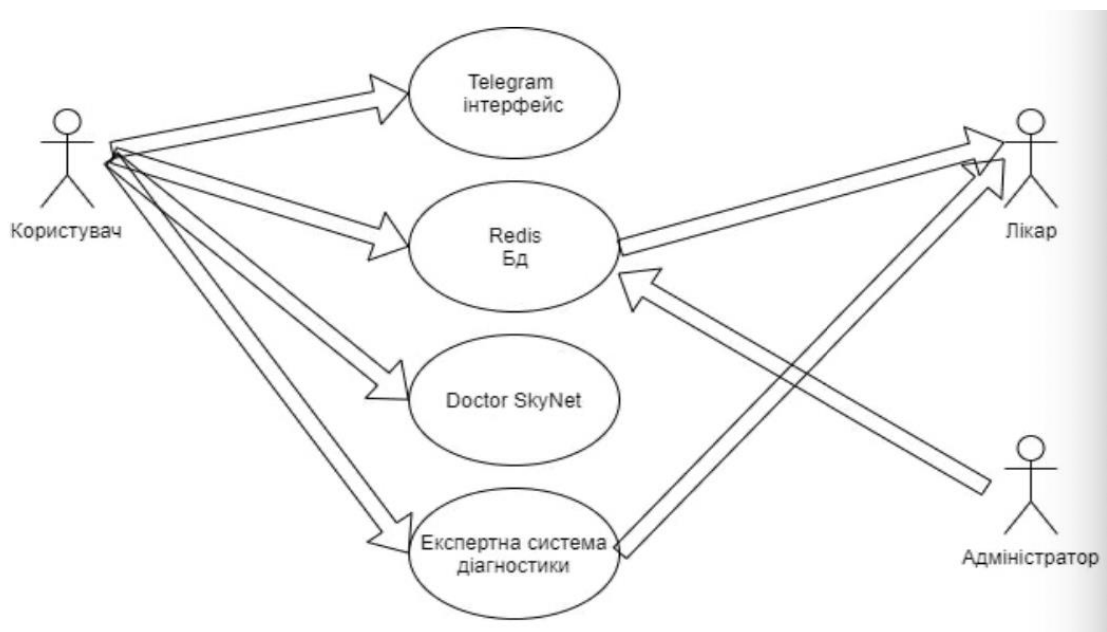


Рисунок 3.2 — Діаграма взаємодії використання

Користувач є найважливішим суб'єктом системи, без нього система не може працювати і чат-бот ніколи не створить екземпляр. Пацієнт взаємодіє з усіма модулями з моменту нормального перебігу, діалог робить використання всіх модулів дуже важливим. Роль лікаря в системі зводиться до отримання інформації про пацієнта, яку він може переглядати пізніше відповідно до своїх вимог. Тому лікар отримує інформацію лише від діагностичної системи Redis та модуля експертної системи.

Єдина робота адміністратора — контролювати використання системи та призначати права доступу лікарям, тому адміністратору необхідно

взаємодіяти з базою даних лише в разі необхідності зміни будь-якої інформації.

3.1.2 Діаграми класів

Діаграма класів системи, що показує взаємодії між різними класами та потік даних між ними (рис. 3.3). Кожен клас містить поведінку, властивості та асоціації «один з одним» або «один до багатьох».

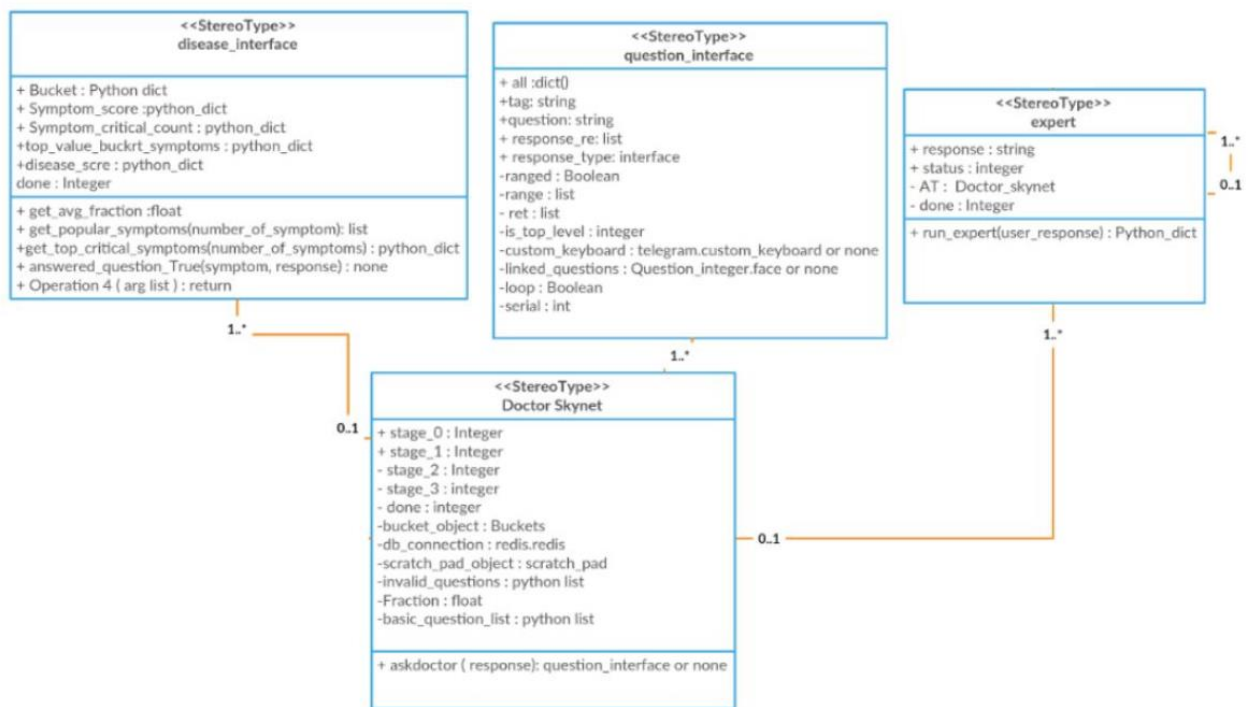


Рисунок 3.3 — Діаграма класів

Діаграма класів складається в основному з класів і модулів, які вже визначені на схемі архітектури системи. Крім того, певні класи, важливі для реалізації тут, також були визначені як функціональні можливості основних класів. Наприклад, `question_interface` та `disease_interface` класи, які є важливою для роботи класу `DoctorSkyNet`, оскільки питання, які потрібно поставити пацієнту, і форма інтерпретації захворювань, які можуть виникнути у пацієнта, є важливою частиною визначення наступного запитання, яке потрібно поставити користувачеві.

3.1.3 Діаграми стану та переходу

Діаграма переходу стану пацієнта показує переходи станів, які відбуваються, коли пацієнт продовжує спілкуватися з системою (рис. 3.4). Умови (поля) представляють дії, які виконує пацієнт у певному стані, а стрілки представляють дії, які пацієнт повинен зробити, щоб досягти цього стану.

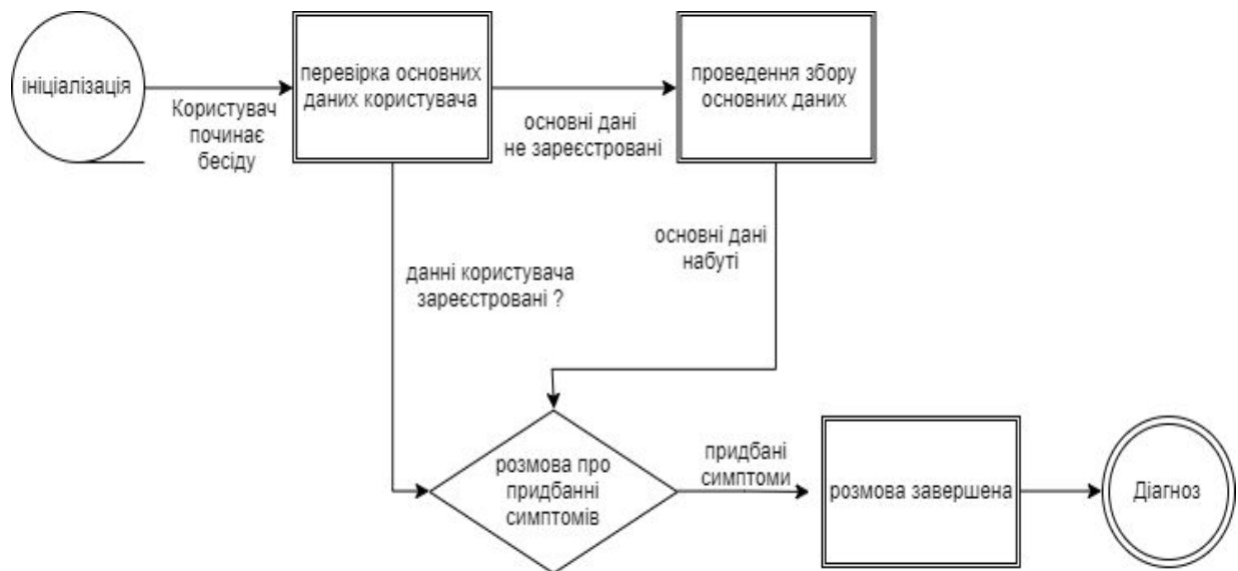


Рисунок 3.4 — Діаграма стану та переходу

Наведені вище сценарії переходу описують дії, які може зробити пацієнт, і що він чи вона повинні зробити, щоб перейти від одного набору дій до іншого. Виходячи з графіка, зрозуміло, що пацієнт повинен спочатку підтвердити свої основні дані, перш ніж продовжити розмову з системою, що дозволить йому/їй повідомити про свої симптоми чат-бота. З діаграми станів видно, що система спочатку проведе розмову, щоб отримати основну інформацію від користувача, перш ніж дозволити йому почати розмову про набуття симптомів. Як тільки ви успішно отримаєте симптоми, ви можете припинити розмову.

3.1.4 Діаграми розгортання

Діаграма, що показує, як система буде розгорнута в реальній комп'ютерній системі (рис. 3.5). Діаграми розгортання допомагають нам

зрозуміти, як саме розгортається вся система, і дозволяють побачити, як різні компоненти інтегруються один з одним у реальному світі.

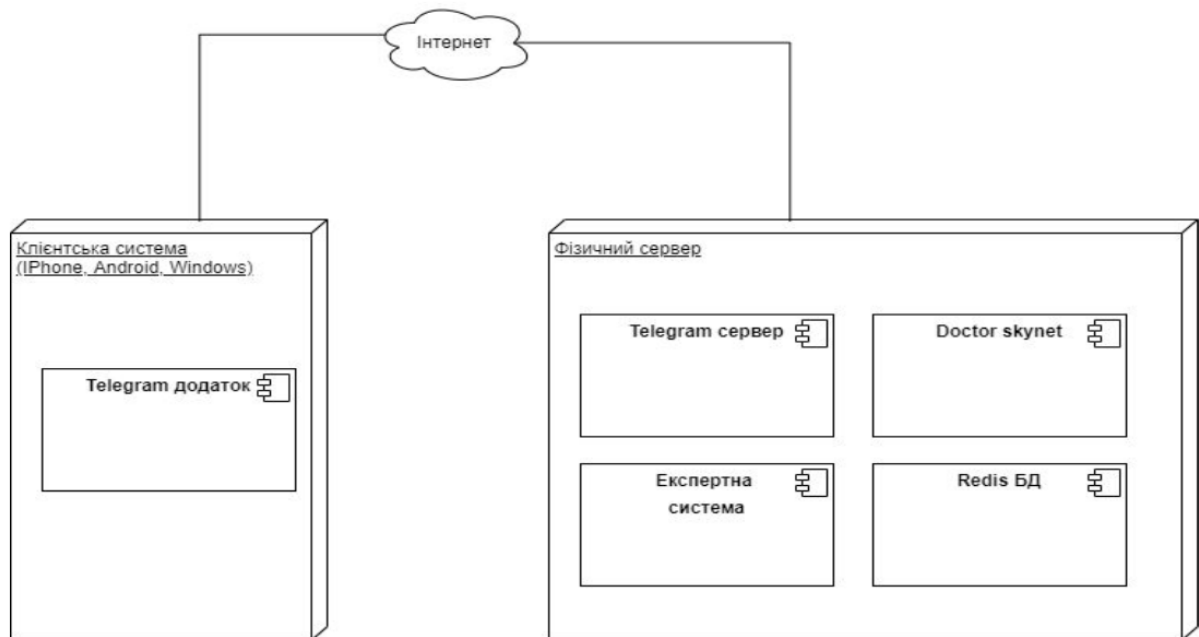


Рисунок 3.5 — Діаграма розгортання

Як ви можете бачити на зображенні вище, система складається в основному з двох розгортань: клієнтської системи, яка запускає програму чату Telegram, яка може бути будь-яким, від iPhone до ПК з Windows, і фізичного сервера, який фактично запускає чат-бота через Додаток Telegram Отримувати введення користувача. Обидві зони розгортання підключені через Інтернет. Як описано в цьому документі, фізичним сервером може бути будь-який пристрій UNIX, а клієнтська система має бути будь-яким середовищем, яке підтримує програму чату Telegram.

3.1.5 Діаграми послідовності

На діаграмі послідовності показано тимчасові дії користувача над модулем чат-бота (рис. 3.6). Основні модулі включають додаток Telegram, сервер та інформаційну систему.

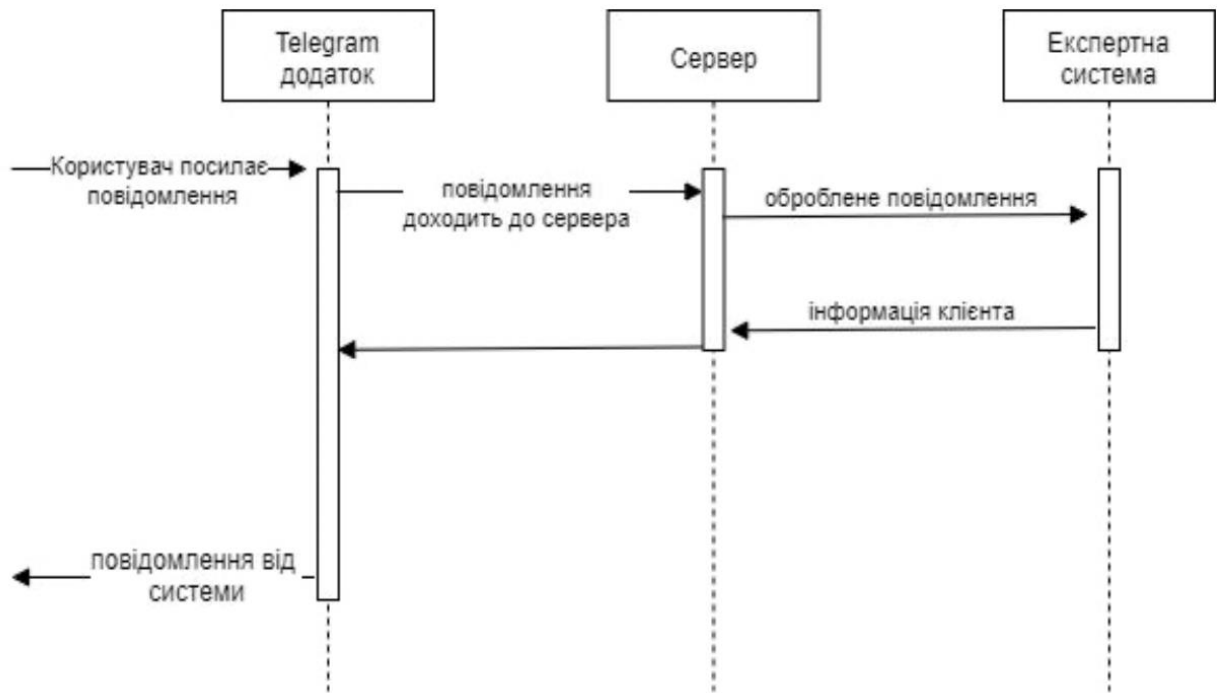


Рисунок 3.6 — Діаграма послідовності

Як видно з малюнка, пацієнту необхідно спочатку взаємодіяти з програмою Telegram, щоб почати розмову з системою. Потім програма Telegram пакує повідомлення у власному форматі та надсилає його на сервер, який потім надсилає його в систему обміну повідомленнями. Інформаційна система відповідає за глибше розуміння цього повідомлення в контексті інших повідомлень, раніше надісланих чат-боту, на основі збережених даних користувача (якщо такі є). Він генерує відповідну відповідь на надіслане повідомлення на основі цих параметрів і знову надсилає його через Сервер і додаток Telegram, щоб досягти користувача.

3.2 Схема бази даних

Вся система спирається на два типи даних — закодовані дані та динамічні дані. Вони виглядають так:

- а) кодовані дані;

1) стосуються симптомів, захворювань або проблем, вони дуже важливі дані і мають бути в якомусь форматі і є головним сховищем знань інформаційних систем;

2) зберігаються у формі власних структур даних Python, таких як словники, списки, рядки та числа — все у вибраному файлі Python і причина зберігати жорстко зашифровані дані замість будь-якої зовнішньої бази даних полягає в тому, щоб повністю виключити можливість небажаних елементів, які вносять зміни в дані, які можуть бути шкідливими для всієї системи;

3) два основних типи жорстко закодованих даних є такими;

— ці захворювання дають інформацію про кожну хворобу та її складові симптоми;

— внутрішньо кожен симптом даного захворювання оцінюється як **ВАЖЛИВО**, **КРИТИЧНИЙ** та **ОПЦІОНАЛЬНЕ**, дати інформаційній системі уявлення про найважливіші та найменш важливі симптоми даного захворювання;

— дані запитань — це питання, яке буде задано щодо кожного симптому, щоб підтвердити, чи є у пацієнта симптоми;

— одне питання дорівнює одному симптому, поділяються на основні та конкретні запитання;

— ключові запитання — питання щоб підтвердити, чи є у пацієнта певні симптому, і конкретні запитання, щоб запитати подробиці конкретного симптому;

— кожне запитання позначено назвою симптому, мітки відіграють важливу роль у визначенні ієрархії симптомів, оскільки вони завантажуються в різні структури даних інформаційної системи, такі як `scratch_pad` та `symptom_validity_table`;

б) динамічні дані, надані системі реальними людьми в результаті їх розмови з системою, зберігаються в базі даних Redis як пара ключів-значень для кожного користувача.

3.3 Сервера

Серверний модуль є основним шлюзом для зв'язку системи із зовнішнім світом. По суті, це асинхронна, неблокуюча, багатопотокова, багатопроцесорна програма на Python, яка опитує сервер чату Telegram щодо вхідних повідомлень і передає їх базовій логіці, яка фактично обробляє та повертає результати користувачеві.

Бібліотека `python-telegram-api` служить корисною обгорткою Python, яка робить спілкування з Telegram дуже простим. Це дозволяє легко налаштувати гачки методів, які запускаються щоразу, коли відбувається певна дія в розділі чат-бота Telegram. Це також дозволяє нашій програмі опитувати сервер Telegram кожні 0,1 секунди за допомогою методу `start_polling()`. Наприклад, щоб викликати метод `accept_message` щоразу, коли повідомлення є «обробником команд телеграм», запитайте сервер telegram.

```

updater.addTelegramMessageHandler(accept_message)
# start polling the Telegram server...
updater.start_polling(0.1)

```

Лістинг 3.1 — Обробник команди Telegram

Метод `accept_message` внутрішньо використовує спільну чергу (екземпляр `multiprocessing.QUEUE`), яка використовується всіма трьома процесами, які використовуються сервером. Він просто передає відповідні дані з повідомлень, отриманих у черзі, і гарантує, що інші вхідні повідомлення не блокуються через перебої в мережі/затримки, на які впливають будь-які повідомлення. Всередині метод `accept_message` виглядає так:

```

d = { # populate dict with useful
      information about message }
MESSAGE_QUEUE.put(d)

```

Лістинг 3.2 — Загальна черга “MESSAGE_QUEUE”

Окремий процес безперервно запитує очікувані повідомлення з черги, яка надсилає їх в інформаційну систему для обробки. Це робиться, спочатку відправивши його в модуль «менеджер». Код для цього можна підсумувати як:

```
While True:
    if messages_present_in_queue:
        user_info = get_most_recent_message_from_queue
        m = send_message_to_dispatcher_for_processing(user_info)
        add_machine_response_to_dispatch_queue(m, user_info)
```

Лістинг 3.3 — Відправка повідомлення до диспетчера

Як бачимо, як тільки процес виявляє, що повідомлення знаходиться в черзі, воно негайно надсилається диспетчеру для обробки, а повернуте повідомлення додається до черги надсилання, яка є чергою, спільною для іншого процесу. Таким чином, обробка повідомлень не залежить від затримки мережі, яка може виникнути під час надсилання повідомлень користувачам.

Третій процес є багатопоточним процесом, єдиним завданням якого є запитувати загальну чергу надсилання повідомлень і надсилати їх відповідному користувачеві Telegram. Кожне повідомлення в черзі надсилається через інший потік, так що мережа, яка працює для одного повідомлення, не впливає на повідомлення, доступні в черзі.

3.4 Диспетчер

Менеджер — це модуль, який знаходиться безпосередньо за сервером для отримання та обробки повідомлень від користувачів. Сервер створює один екземпляр менеджера, і повідомлення надсилаються цьому екземпляру шляхом виклику методу `depecher.run_dispatcher ()`.

Внутрішньо менеджер підтримує словник Python про поточні активні розмови, які він веде. Цей словник називається `object_list`. Ключовими словами цього диктанту є ідентифікатори чату користувача, які є унікальними для кожного користувача, а значення — об'єкти типу ядра. Основний клас — це той, який фактично відповідає за обробку повідомлень.

Кожен раз, коли менеджер отримує нове повідомлення, він шукає `object_list`, якщо ідентифікатор чату вхідного повідомлення вже присутній. Якщо так, це інтерпретується як відповідь на повідомлення, яке раніше було надіслано користувачеві, і повідомлення надсилається безпосередньо до основного об'єкта, з яким пов'язаний ідентифікатор чату. Якщо ні, у словнику `object_list` створюється новий запис, який пов'язує ідентифікатор чату повідомлення з останнім створеним основним об'єктом.

Як ми незабаром побачимо, основний клас відповідає за відстеження поточної розмови, зберігаючи деякі змінні, які визначають поточний стан розмови. Принадність цього підходу полягає в тому, що стає дуже легко стежити за розмовою, просто зберігаючи основні об'єкти в пам'яті, поки розмова не закінчиться.

Після завершення розмови основний об'єкт, пов'язаний з ідентифікатором чату, видаляється з диктанту і, таким чином, остаточно видаляється з пам'яті за допомогою кошика Python.

Приблизно найважливішу функцію менеджера, як було підсумовано вище, можна представити наступним фрагментом коду:

```
object_list=dict()
if conversation_with_chat_id_over(chat_id):
    object_list[chat_id]=assign_new_core_object()
    core_object=object_list[chat_id]
    core_object.process_message_and_return_reply()
```

Лістинг 3.4 — Узагальнений функціонал диспетчера

3.5 Ядро

Головний модуль викликається диспетчером. Для кожної незалежної розмови, що відбувається, є один екземпляр базового класу. Ядро містить змінні та методи, які дозволяють відстежувати розмову з верхнього рівня, тобто воно може знати лише, увімкнено чи закінчено розмову. Проміжні стани не впливають на ядро.

Створює екземпляр інформаційної системи експертного типу та керує відправленням та отриманням з неї даних, які в кінцевому підсумку потрапляють назад у клас керування та надсилаються назад користувачеві. Вся попередня обробка вхідних повідомлень, така як перевірка орфографії, пунктуація тощо, виконується ядром до того, як повідомлення навіть передається в інформаційну систему. Повертає повідомлення про диктовку Python, після завершення виклику повертається «None».

Основну логіку, що відбувається, можна підсумувати так:

```
Expert = expert_system() # instantiate expert system
module if expert.done == 1: # conversation is over return
None else: # conversation still in progress
return_message=expert.run_expert_system(incoming_message)
return return_message
```

Лістинг 3.5 — Узагальнений функціонал диспетчера

3.6 Дані запитань та відповідей

Для того, щоб інформаційна система функціонувала належним чином, важливо мати сховище знань. Знання нашого чат-бота представлені файлами Python, що зберігаються в папці data/. Кожне запитання має певні властивості, які надають додаткову інформацію, наприклад, відповідь, яка має з'явитися на спеціальній клавіатурі Telegram, коли запитання представляється користувачеві. Різні властивості проблеми можуть бути такими:

— панель запитань — це ваше власне запитання, яке буде надіслано користувачеві;

— відповідь — це список відповідей, який буде відобразитися користувачеві, вказуючи відповіді, які він може надіслати в систему за допомогою спеціальної клавіатури Telegram;

— послідовний — це внутрішнє ціле число, що вказує порядок, у якому задаються питання.

Користувачеві можна задати два типи запитань:

а) ключові запитання щодо даних;

1) задаються основні запитання щодо збору основних персональних даних кожного пацієнта, таких як вік, стать, вага та зріст;

2) це питання, які потрібно задати, коли пацієнт вперше починає спілкуватися з системою;

3) запитання зберігаються як словник Python у файлі `basic_data.py`;

б) питання, пов'язані з симптомами;

1) запитання, щодо симптомів, — це запитання, які будуть задані пацієнтам для коректного підтвердження того чи проявляють вони ознаки певного симптому чи ні;

2) вони здатні збирати так / ні стилістичні чи конкретні відповіді на основі заздалегідь визначених параметрів;

3) ці питання також поділяються на два типи;

— головні запитання — головне питання, чи підтверджує це наявність певного симптому;

— пов'язані запитання — це запитання, які служать для запиту додаткової інформації про певний симптом, якщо пацієнт відповідає на основне питання цього симптому;

4) оскільки пов'язані запитання стосуються збору додаткової інформації про конкретну симптоматику, вони розміщуються в окремих файлах, кожен з яких названий на честь симптому, який обговорюється у файлі і тому питання отримання додаткової інформації про біль у тілі розміщено у файлі під назвою `body_pain.py`.

3.7 Інформаційно-експертна система

Цей модуль, який містить власне експертну систему в інформаційній системі для відображення змісту та контексту відповіді. Інше питання вирішується в модулі `DoctorSkyNet`. Експертна система визначає стан

розмови, відстежуючи її за допомогою змінної стану. Більшість базової логіки експертної системи міститься в методі `run_expert()`.

Якщо статус інформаційно-експертної системи буде дорівнювати одиниці, то це може означати, що інформаційна система знаходиться в тому стані, в якому щойно надійшло найперший запит від юзера, і він/вона не вибрав жодного варіанту в питаннях системи. Цей статус важливий, тому що він формулює, чи відповідь буде відправлено в DoctorSkyNet. Після того як користувач дав відповідь, то система переходить у відповідно другий стан.

У стані 2 експертна інформаційна система чекає від юзера відповідь, яку він відправить в DoctorSkyNet в наступної обробки. Коли DoctorSkyNet повертає `None`, інформаційно-експертна допускає, що виклик закінчено та виходить до третього, що користувач повинен відправити запит, що тестування закінчено.

Загальне функціонування інформаційно-експертної системи можна визначити так.

```
If user_has_begun_conversation():
    status=1
    if status == 1:# First question is being asked.
        status = 2# First question asked.
        get_first_question_from_doctor_skynet()
    if status == 2:# System expecting a reply from user.
        send_to_doctor_skynet_and_get_next_question()
        if no_more_questions_remain():
            state=3
            done=1# Indicates
            return_conversation_done_to_core()
        return_question_to_core()
```

Лістинг 3.6 — Загальний функціонал інформаційно-експертної системи

3.8 Лікар SkyNet

Ця магія є справжньою магією. Його головна мета — використовувати інтелектуальні алгоритми для зчитування відповідей користувача, відстеження симптомів, на які користувач уже відповів, і задання їм найбільш актуальних питань на основі підтримуваної бази даних захворювань.

Кожен діагноз представлений у вигляді «кошика» (з використанням модуля «Кошки», який ми дізнаємося нижче). Кожен кошик має певні симптоми. Алгоритми, яку застосовує DoctorSkyNet, зчитують статус цих кошиків і задають користувачеві найбільш актуальне запитання. Це може допомогти нам суттєво зменшити кількість запитань, які система має поставити, щоб представити ймовірний діагноз.

Алгоритми, що використовується в DoctorSkyNet, можна узагальнити таким чином:

а) мінус перший алгоритм;

1) функція — кластерний ідентифікатор симптоматики;

2) видає симптоми, від якого можуть страждати ВСІ користувачі;

3) використовує Redis;

4) не зв'язано безпосередньо з кошиком і має етапи;

— отримати весь симптом від Redis через ORM «так»;

— знайти симптоматику яка найбільше відповідає «Так»;

— повернути симптоматику;

б) нульовий алгоритм;

1) функція — найімовірніше пошук симптомів заснований на історії хвороби пацієнта;

2) виявляє симптом;

3) використовує Redis;

4) не зв'язано безпосередньо з кошиком і має етапи;

— використовувати Redis через ORM для отримання історії симптомів пацієнта;

— знайти симптоми з найбільшою кількістю «так»;

— повернімося до цього симптому;

в) перший алгоритм;

1) функція — знайдіть у кошику симптоматики із найбільшим балом;

2) знайдіть симптоми, які найбільше заповняють кошик;

3) не задіяно із кошиками і має етапи;

— отримати оцінку симптомів;

— знайдіть симптом з найвищим балом;

— повернути цей симптом;

— оновлений кошик, щоб видалити симптоми з усіх точок;

г) другий алгоритм;

1) функція — відшукати критичний стан із найбільшим балом у кошику;

2) якщо відповідь негативна, шукайте симптоми, які приберуть більшість кошиків;

3) не зв'язано безпосередньо з кошиком і має етапи;

— ретельна оцінка симптомів;

— знайдіть симптом з найвищим балом;

— повернути цей симптом;

— оновлений кошик, щоб видалити симптоми з усіх точок;

г) третій алгоритм;

1) функція — знайдіть симптом із максимальним сукупним балансом у кожному кошику;

2) якщо відповідь негативна, шукайте симптоми, які приберуть більшість кошиків;

— отримати кошик;

— знайдіть симптом з найвищим балом у кошику;

— запишіть симптом, оцініть його в структурі даних;

— якщо кошик має бути перезапущений;

— знайти симптом з найвищим балом у структурі даних;

— повернімося до цього симптому.

Вся робота модуля контролюється в два етапи. Кожен крок вказує, де зараз перебуває система в розмові. На початковому етапі проводиться попередня перевірка, чи отримані саме базові дані пацієнта. Саме це відбувається шляхом перевірки головних даних в базі Redis. Якщо відсутнє, бот ініціює діалог з користувачем, щоб краще зрозуміти його основну особисту інформацію, таку як зріст, вік, вагу та стать.

Встановлює змінну "done" 1, коли система отримує оптимальну кількість інформації про симптоматику. Загальну роботу DoctorSkyNet можна підсумувати так. Після встановлення вихідних даних DoctorSkyNet виходить до наступного етапу та виходить запиту користувача щодо симптоматики на базі вищевказаних алгоритмів.

Клас інтерфейсу запитань зберігає подробиці всіх запитань та оптимізує роботу для пришвидшення доступу для бідь-якого класу іншого, якому потрібно отримати доступ до запитів в правильному форматі. Він використовує ті налаштування, що й будь-який словник запитів у базі даних.

```
While conversation_in_progress():
    update_bucket_fractions()
if basic_questions_unanswered():
    init_basic_questions_conversation()
if symptom_queries_not_over():
    init_symptom_query_conversation()
```

Лістинг 3.7 — Загальний вигляд системи DoctorSkyNet

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ СИСТЕМИ

4.1 Тестування графічного інтерфейсу

Зображення нижче (рис. 4.1 - рис. 4.4) є зображеннями програми iPhone Telegram iOS версії на відмінних етапах оцінювання. Поетапне демонстрування діалогу системи з потенційним користувачем (пацієнтом) яку він може провести з чат-ботом. Зверніть увагу на зміни клавіатури та відповіді на кожному зображенні, це повинно дати вам гарне уявлення та враження про систему чат-бота.

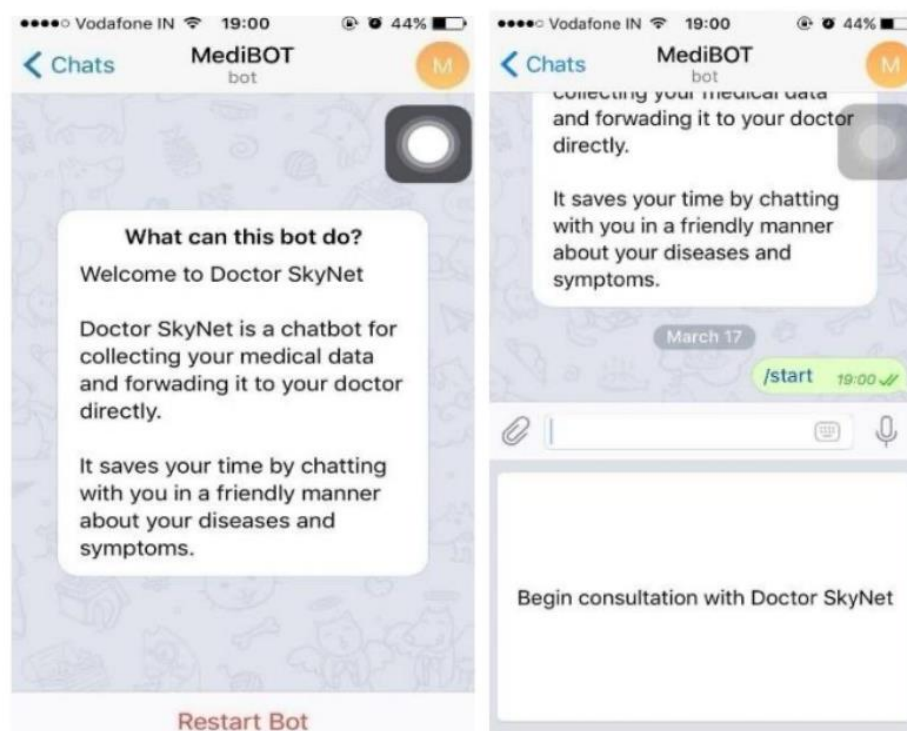


Рисунок 4.1 — Старт роботи чат-боту та привітальне повідомлення

4.2 Тестування роботи серверу

На зображенні терміналу наведеного нижче можемо дізнатись результат, який відображає сервер, коли він одержує повідомлення телеграм від клієнта. Червоною лінією вказується на те, що об'єкт User не буває раніше створений та саме створюється і це буде означати, що з користувачем



Рисунок 4.2 — Початок діалогу з чат-ботом

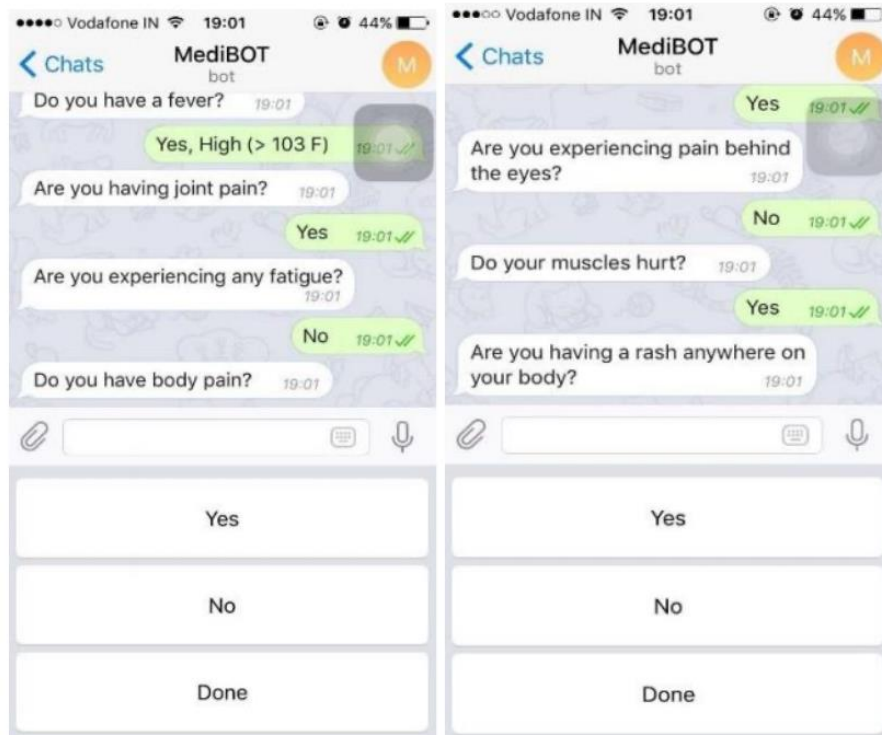


Рисунок 4.3 — Симптоматичні запитання

розпочався новий діалог, а також ввідне повідомлення не пересилається

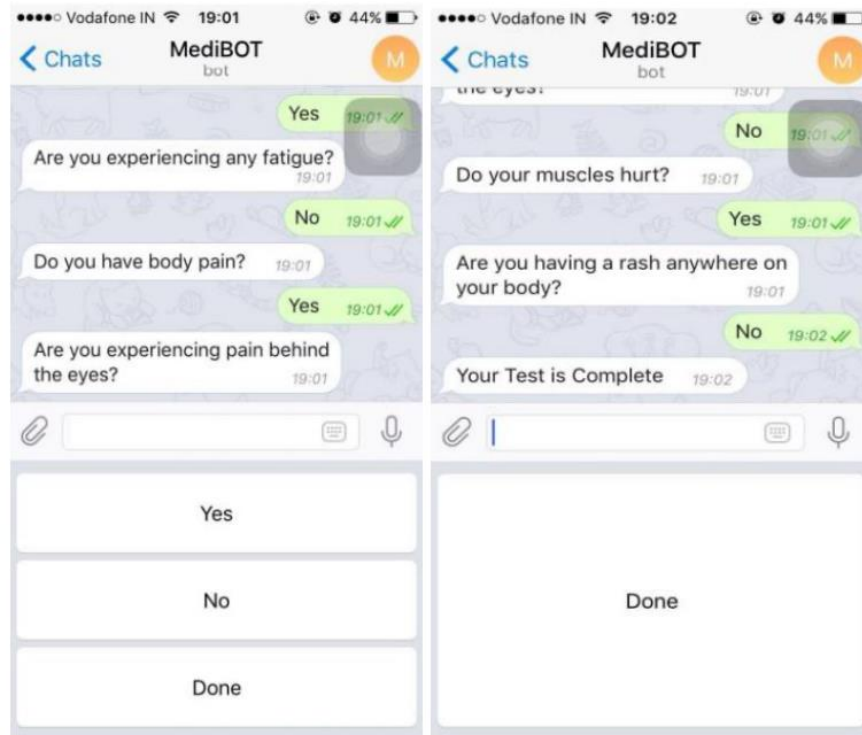


Рисунок 4.4 — Завершення отримання інформації користувача

Цифри детально показують цю поведінку. Ви можете побачити, як діалог починається з (рис. 4.5), потім прогресує (рис. 4.6), а потім завершується (рис. 4.7).

```

Activities | Terminal | Thu 19:01
/run.sh
File Edit View Search Terminal Tabs Help
/run.sh | ipython | pry
14012:M 17 Mar 18:58:36.963 # Creating Server TCP listening socket *:6379: bind: Address already in use
User with chat id90417424 not found , creating new object
USER OBJECT CREATED WITH CHAT ID: 90417424

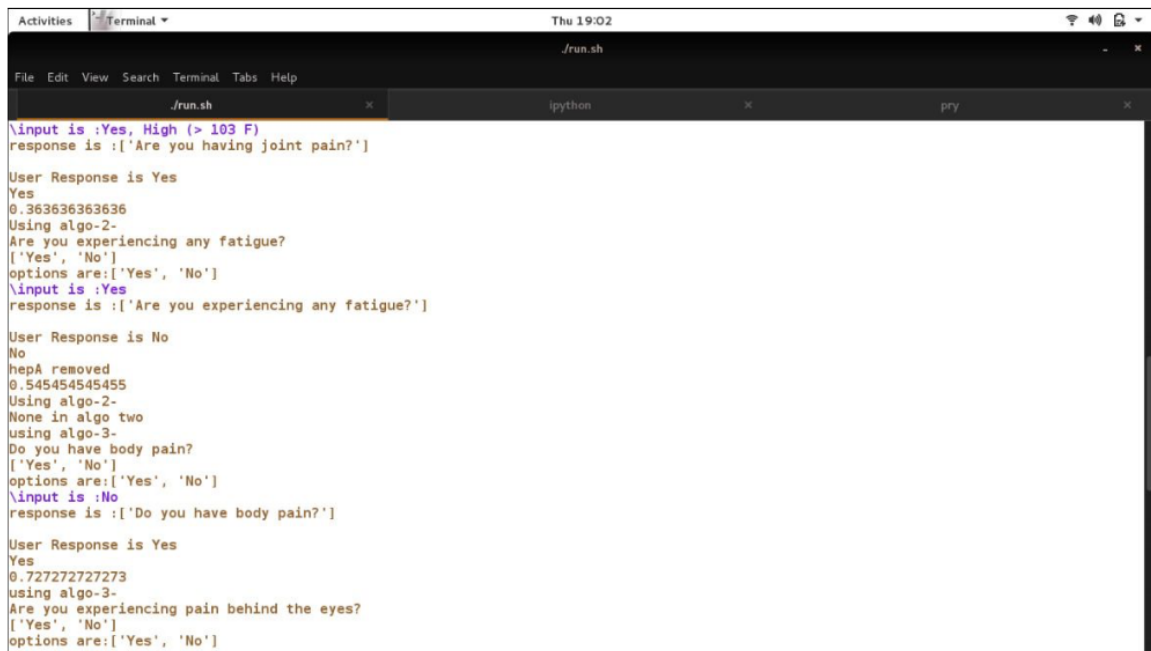
hepA : LOADED
dengue : LOADED
User Response is Begin consultation with Doctor SkyNet
Begin consultation with Doctor SkyNet
0.0
using algo-1-
Do you have a fever?
['Yes, High (> 103 F)', 'Yes, Mild (101-103 F)', 'Yes, Very Mild (99 - 101 F)', 'No']
\input is :Begin consultation with Doctor SkyNet
response is :['Do you have a fever?']

User Response is Yes, High (> 103 F)
Yes, High (> 103 F)
0.181818181818
using algo-1-
Are you having joint pain?
['Yes', 'No']
options are:['Yes', 'No']
\input is :Yes, High (> 103 F)
response is :['Are you having joint pain?']

User Response is Yes
Yes
0.363636363636
Using algo-2-
Are you experiencing any fatigue?
['Yes', 'No']
options are:['Yes', 'No']
\input is :Yes

```

Рисунок 4.5 – Початок діалогу з користувачем



```

Activities | Terminal | Thu 19:02
Terminal
/run.sh
File Edit View Search Terminal Tabs Help
/run.sh | ipython | pry
\input is :Yes, High (> 103 F)
response is :['Are you having joint pain?']

User Response is Yes
Yes
0.363636363636
Using algo-2-
Are you experiencing any fatigue?
['Yes', 'No']
options are:['Yes', 'No']
\input is :Yes
response is :['Are you experiencing any fatigue?']

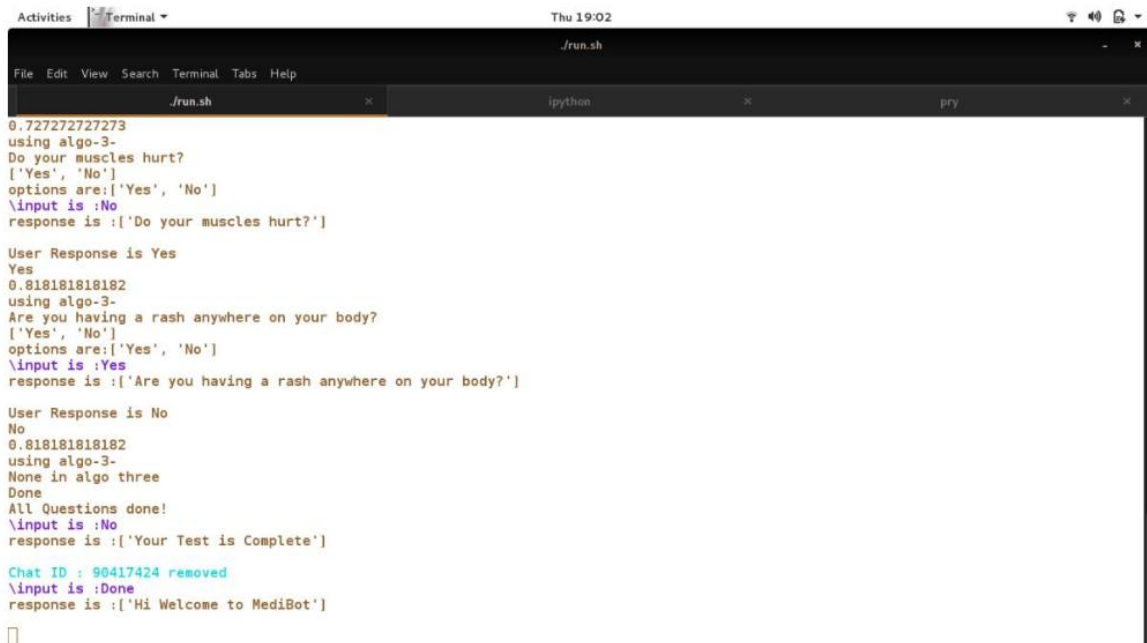
User Response is No
No
hepA removed
0.545454545455
Using algo-2-
None in algo two
using algo-3-
Do you have body pain?
['Yes', 'No']
options are:['Yes', 'No']
\input is :No
response is :['Do you have body pain?']

User Response is Yes
Yes
0.727272727273
using algo-3-
Are you experiencing pain behind the eyes?
['Yes', 'No']
options are:['Yes', 'No']

```

Рисунок 4.6 – Перехідний етап діалогу з користувачем

у відповідь на останнє повідомлення-відповідь. Записувач не буде створювати нові об'єкти, коли користувач отримує іншу відповідь, яка вказує



```

Activities | Terminal | Thu 19:02
Terminal
/run.sh
File Edit View Search Terminal Tabs Help
/run.sh | ipython | pry
0.727272727273
using algo-3-
Do your muscles hurt?
['Yes', 'No']
options are:['Yes', 'No']
\input is :No
response is :['Do your muscles hurt?']

User Response is Yes
Yes
0.818181818182
using algo-3-
Are you having a rash anywhere on your body?
['Yes', 'No']
options are:['Yes', 'No']
\input is :Yes
response is :['Are you having a rash anywhere on your body?']

User Response is No
No
0.818181818182
using algo-3-
None in algo three
Done
All Questions done!
\input is :No
response is :['Your Test is Complete']

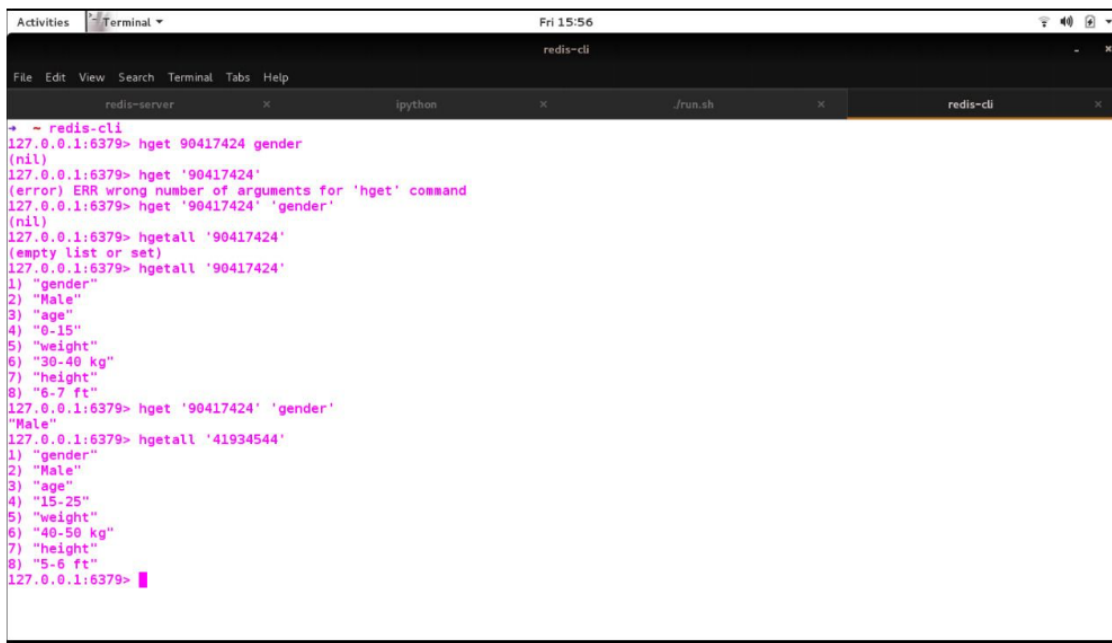
Chat ID : 90417424 removed
\input is :Done
response is :['Hi Welcome to MediBot']

```

Рисунок 4.7 – Завершення розмови з користувачем

на те, що той саме об'єкт пам'ятає статус користувача під час з'єднання.

На скриншоті нижче (рис. 4.8) показана вигразка бази Redis (дамп) для діалогу, який показаний на (рис. 4.5 — рис. 4.7). Відображає значення, що зберігаються в дампі з ідентифікатором користувача.



```
Activities Terminal Fri 15:56
redis-cli
File Edit View Search Terminal Tabs Help
redis-server ipython /run.sh redis-cli
+ ~ redis-cli
127.0.0.1:6379> hget 90417424 gender
(nil)
127.0.0.1:6379> hget '90417424'
(error) ERR wrong number of arguments for 'hget' command
127.0.0.1:6379> hget '90417424' 'gender'
(nil)
127.0.0.1:6379> hgetall '90417424'
(empty list or set)
127.0.0.1:6379> hgetall '90417424'
1) "gender"
2) "Male"
3) "age"
4) "0-15"
5) "weight"
6) "30-40 kg"
7) "height"
8) "6-7 ft"
127.0.0.1:6379> hget '90417424' 'gender'
"Male"
127.0.0.1:6379> hgetall '41934544'
1) "gender"
2) "Male"
3) "age"
4) "15-25"
5) "weight"
6) "40-50 kg"
7) "height"
8) "5-6 ft"
127.0.0.1:6379> █
```

Рисунок 4.8 – Вигразка бази даних бота Redis

5 ЕКОНОМІЧНА ЧАСТИНА

Виконання дослідницької роботи передбачає певні витрати. Зазвичай це витрати на створення та впровадження конкретного програмного забезпечення. Такі витрати необхідно постійно зменшувати, адже у цьому полягає прогрес будь-якого суспільства. Без такого прогресу ніяка науково-технічна розробка не буде реалізована на практиці, адже така розробка не буде ефективнішою за існуючі на ринку аналоги.

На основі економічних розрахунків можна довести економічну доцільність та ефективність впровадження отриманих результатів виконаних науково-дослідних робіт у виробництво, тобто здійснити комерціалізацію наукових розробок [15]. Дана магістерська кваліфікаційна робота носить експериментальний характер, тобто прогнозується впровадження науково-технічної розробки чат-боту безпосередньо розробником.

5.1 Проведення технологічного аудиту інформаційної системи

Метою оцінки потенціалу комерційного розвитку є оцінка потенціалу комерційного розвитку, що випливає з науково-технічних досліджень. За результатами оцінки розболяться висновки про напрямки (особливості) організації в майбутньому її впровадження з урахуванням встановленої оцінки. У цілях проведення технологічного аудиту було залучено 3 незалежних експерти. Такими експертами є к.т.н., доц. Муращенко Олександр Геннадійович, к.т.н., доц. Черняк Олександр Іванович та к.т.н., доц. Крупельницький Леонід Віталійович. Комерційний потенціал інвестицій буде оцінюватись відповідно до дванадцяти критеріїв, наведених у таблиці 5.1.

На основі таблиці різні експерти, у нашому випадку керівник магістерської роботи та викладачі кафедри ОТ, визначають різні результати.

Таблиця 5.1 — Оцінювання комерційного потенціалу розробки

Критерії оцінювання та бали (за 5-бальною шкалою)					
Критерій	0	1	2	3	4
1	Достовірність концепції є підтверджена	Підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту
2	Багато аналог. на мал. ринку	Мало аналогів на мал. ринку	Багато аналогів на вел. ринку	Один аналог на вел. ринку	Не має аналогів на вел. ринку
3	Ціна продукту значно вища аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно як ціна аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту звано нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експл. витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
6	Ринок малий і без позитивної динаміки	Ринок малий, з позитивною динамікою	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Конкуренція на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
8	Відсутні фахівці, як з технічної, так і з комерційної реалізації ідеї	Наймати фахівців або витратити значні кошти на навчання	Незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань, як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фін. ресурси, які відсутні	Незначні фін. ресурси. Джер. фінансування відсутні	Потрібні значні фін. ресурси. Джерела фінансування є	Потрібні незначні фін. ресурси. джер. фінанс. є	Не потребує додаткового фінансування
10	Необхідна розробка нов. матеріалів	Матеріали, що використовують ся у ВПК	Потрібні дорогі матеріали	Досяжні та дешеві матеріали	Матеріали відомі та давно використов.
11	Термін реалізації ідеї більше за 10 років	Термін більше за 5р., окупність інвестицій більше 10 р.	Термін від 3-х до 5-ти років, окупність інвестицій більше 5р.	Термін менше 3 р. Термін окупності інвестицій від 3-5 р.	Термін реалізації менше 3р., окупність менше 3 р.
12	Необхідно регламентні документи та велика кількість дозвіл. док. на виробництво продукту	Необхідно отримання великої кількості дозвіл. док. на виробництво та реалізацію продукту	Процедура отримання дозвіл. док. для виробн. та реаліз. продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органами, про виробництво та реалізацію продукту	Відсутні будь-які регламенти обмеження на виробництво та реалізацію продукту

Результати цієї оцінки комерційного потенціалу узагальнено у таблиці 5.2.

Таблиця 5.2 — Результати оцінювання комерційного потенціалу

Критерії	Експерт (ПБ, посада)		
	1.Мурашенко О.Г., к.т.н., доц. кафедри ОТ	2.Черняк О.І., к.т.н., доц. кафедри ОТ	3.Крупельницький Л.В., к.т.н., доц. кафедри ОТ
	Бали:		
1.Технічна здійсненність концепції	3	3	2
2.Ринкові переваги (наявність аналогів)	4	3	3
3.Ринкові переваги (ціна продукту)	3	3	3
4.Ринкові переваги (технічні властивості)	3	3	4
5.Ринкові переваги (експлуатаційні витрати)	3	3	3
6.Ринкові перспективи (розмір ринку)	3	3	4
7.Ринкові перспективи (конкуренція)	3	3	3
8.Практична здійсненність (наявність фахівця)	4	3	3
9.Практична здійсненність (наявність фінансів)	2	2	2
10.Практична здійсненність (необхід. нових матеріалів)	4	3	4
11.Практична здійсненність (термін реалізації)	4	4	3
12.Практична здійсненність (розробка документів)	3	3	3
Сума балів	СБ ₁ = 39	СБ ₁ = 37	СБ ₁ = 37
Середньо-арифметична сума балів СБ _с	$СБ_{с} = \frac{\sum_{i=1}^3 СБ_{i}}{3} = \frac{35+38+36}{3} = 38$		

Відповідно до таблиці 5.2, а також відповідно до рекомендацій, наведених у таблиці 5.3, можна зробити висновок про рівень потенціалу комерційного розвитку.

Таблиця 5.3 – Рівні комерційного потенціалу розробки

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

З урахуванням середніх арифметичних балів $СБ_c = 38$, які були визначені експертами, можна зробити висновок, що рівень комерційного потенціалу цієї розробки буде вище середнього. На ринку дуже мало якісних програм аналогів для визначення та постановки медичного діагнозу.

Програмами аналогами є Symptomate та Helzy. Веб-застосунок Symptomate, де зібрали групи відомих лікарів, науковців, психологів та письменників, щоб донести свої думки та ідеї. Присутнє проходження заздалегідь підготовлених опитувань та генерування висновку на основі пройденого раніше.

Також є можливість знайти собі спеціаліста в медичному напрямку для подальшого лікування. Це рішення уже більш професійне в плані діагностування та лікування, тут присутні онлайн консультації у відомих терапевтів, також в наявності опитувальники, які націлені на меданаліз людини та генерування висновку на результаті цього, також більш зручний та зрозумілий інтерфейс.

По-перше, один із головних мінусів застосунку, те що тести працюють тільки у веб-застосунку безпосередньо на комп'ютері, в час коли користувачі все більш використовують мобільні пристрої, смартфони і планшети, це я суттєвим зауваженням. По-друге після 3-5 хвилин опитувань, що триває досить не мало, маємо результативне уточнення, а лише рекомендації до якого лікаря звернутись (терапевта, психіатра, тощо). По-третє лише після опитування буде запропоноване платне отримання результату, шляхом купівлі

ліцензії для користувача. Має вартість 2300 грн., в яку входить купівля ліцензії додатку для користувача на весь час.

Helzy — веб-застосунок який має схожий принцип, але відрізняється тип, що для постановки медичного діагнозу використовую базу медичних рішень з відповідних симптомів.

Серед недоліків маємо схожу проблему – застосунок працює виключно з комп'ютера та не підтримує мобільні пристрої, смартфони, планшети, що суттєво може погано сказатись на оперативності постановки діагнозу. Також має складний, «недружній» інтерфейс до користувача і довгий час опитування (близько 10—15 хв.). Платне отримання результату, вартістю 2700 грн.

Таблиця 5.4 – Порівняння характеристик розробки із аналогом

Показники	Власна розробка	Аналог1	Аналог2
Функціонал	7	7	8
Швидкодія	8	8	7
Надійність	8	8	8
Метод розповсюдження	9	7	6
Інтерфейс, простота використання	8	7	6
Ціна	-	2300 грн	2700 грн – докладний результат (немає можливості повноцінної купівлі)

Програма, яка розробляється, буде виконувати такі функції:

- проведення опитування для визначення діагнозу;
- мати зручний та простий у використанні інтерфейс;
- швидке опрацювання даних;
- підтримка на всіх платформах засобами чат-бота;
- можливість виведення статистичних даних для розробника;

— гнучке конфігурування та можливість оновлюватись.

Продукт може просуватись за допомогою реклами в соціальних мережах, пошукових системах та багато інших інструментах джерел Інтернету. Використовуючи аналітику цих сервісів, можна буде націлити рекламу на цільову групу захисників інформації.

Наукова новизна дослідження полягає в тому, що вперше буде запропоновано та впроваджено комплексне використання методів штучного інтелекту, типу чат-бот на платформі відомого та розповсюдженого в країні месенджера, для максимального охопту користувачів у зручній для них вид пристрою (стаціонарного комп'ютера, ноутбука, смартфона та планшета на різних операційних системах) які підтримує месенджер Telegram.

Даний рівень було досягнуто за рахунок покращення та розширення функціональних можливостей нової інформаційної системи порівняно з аналогічними розробками, існуючими в цей час на ринку.

5.2 Розрахунок витрат на здійснення науково-дослідної роботи

У магістерській роботі розглядається програмний продукт, який через збір відповідних даних (симптомів), алгоритмічним методом чат-боту розраховує можливий діагноз. Тому значну частину складають витрати пов'язані з розробкою, а не виробництвом та відтворенням [16]. Відповідно, є певна специфіка розрахунків. Витрати складаються з оплати праці, соціальних відрахувань, витрат на сировину та матеріали, амортизації обладнання, програмних засобів та приміщень, витрат на електроенергію, накладних (загальновиробничих) витрат та інших витрат, тощо.

Основна заробітна плата розробників, що працюють над проектом, визначена у формулі:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (5.1)$$

де k — кількість посад дослідників, залучених до процесу досліджень;

M_{ni} — місячний посадовий оклад конкретного дослідника, грн;

T_p — середня кількість робочих днів в місяці, $T_p = 20 \dots 23$ днів, обрано 20 днів;

t_i — кількість днів роботи конкретного дослідника, днів.

Уад створенням розробки працював менеджер проекту та інженер програмного забезпечення, тому ми вионаємо для них усі необхідні розрахунки і вносимо їх до таблиці 5.5:

$$З_{о.к.} = \frac{14500 \cdot 32}{20} = 6000 \text{ (грн)}$$

$$З_{о.к.} = \frac{10000 \cdot 32}{20} = 21800 \text{ (грн)}$$

Додаткова винагорода ($З_{дод.}$) усіх розробників та працівників, які брали участь у цьому етапі роботи, обчислюється як 10...12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_о + З_р) \cdot \frac{N_{дод.}}{100\%} \quad (5.2)$$

де $N_{дод.}$ — норма нарахування додаткової заробітної плати.

$$З_{дод.к.} = \frac{12 \cdot 5800}{100} = 696 \text{ (грн)}$$

$$З_{дод.в.} = \frac{12 \cdot 16000}{100} = 1920 \text{ (грн)}$$

$$З_{дод} = З_{дод.к.} + З_{дод.в.} = 2616 \text{ (грн)}$$

Витрати на основну заробітну плату робітників за відповідними найменуваннями робіт відсутні, тобто $З_р = 0$.

Таблиця 5.5 — Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн
Менеджер проєкту	14500	725	9	6000
Старший інженер-програміст	10000	500	32	16000
Всього				22000

Заробітна плата робітників відсутня, тому $Z_p = 0$. Нарахування на заробітну плату дослідників та нарахування на заробітну плату працівників, які брали участь у цьому етапі роботи, розраховується як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{дод}}) \cdot \frac{N_{\text{зп.}}}{100\%} \quad (5.3)$$

де $N_{\text{зп}}$ — норма нарахування на заробітну плату.

$$Z_n = (22000 + 0 + 2616) \cdot \frac{22\%}{100\%} = 5371,52 \text{ (грн.)}$$

Витрати на матеріали (M), у вартісному вираженні розраховується окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n N_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{\text{в}j}, \quad (5.4)$$

де N_j — кількість матеріалу j -го виду, шт.;

n — кількість видів матеріалу;

C_j — ціна матеріалу j -го виду, грн.;

K_j — коефіцієнт транспортних витрат, $K_j = (1,1 \dots 1,15)$, обираємо $K_j 1,15$;

B_j — маса відходів j -го найменування, кг;

$C_{\text{в}j}$ — вартість відходів j -го найменування, грн/кг.

Результати розрахунків занесено до таблиці 5.6.

Таблиця 5.6 – Витрати на матеріали

Найменування комплектуючих	Ціна за 1 штуку, грн.	Кількість матеріалу, шт.	Величина відходів, кг	Ціна відходів грн/кг	Вартість витраченого матеріалу, грн.
Стирус	24,91	1	0	0	24,91
Карта пам'яті	550,00	1	0	0,00	550,00
Пачка офісного папіру	105,00	1	0,5	18,75	86,25
Всього (з урахуванням транспортних витрат)					727,28

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, можуть бути розраховані з використанням прямолінійного методу амортизації за формулою:

$$A_{\text{обл}} = \frac{Ц_{\text{б}}}{T_{\text{в}}} \cdot \frac{t_{\text{вик}}}{12} \quad (5.5)$$

Таблиця 5.7 — Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, місяців	Амортизаційні відрахування, грн
ЕОМ	15000	2	1,5	937,5
Приміщення	100000	20	1,5	625,0
Всього				1562,5

Витрати на силову електроенергію (B_e) розраховують за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{впi}}{\eta_i}, \quad (5.6)$$

де W_{yi} — встановлена потужність обладнання на певному етапі розробки, кВт;

t_i — тривалість роботи обладнання на етапі дослідження, год;

C_e — вартість 1 кВт-години електроенергії, грн (вартість електроенергії визначається за даними енергопостачальної компанії), $C_e = 4,52$ [17];

$K_{впi}$ — коефіцієнт, що враховує використання потужності, $K_{впi} < 1$; обираємо $K_{впi} 0,7$;

η_i — коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = \sum_{i=1}^1 \frac{0,07 \cdot 256 \cdot 4,62 \cdot 0,7}{0,8} = 72,44 \text{ (грн.)},$$

Проведені розрахунки необхідно занести до таблиці 5.8.

Таблиця 5.8 — Витрати на електроенергію

Найменування обладнання	Балансова потужність, кВт	Тривалість роботи, год	Сума, грн
ЕОМ	0,07	256	72,44
Всього			72,44

5.2.1 Накладні (загальновиробничі) витрати та інші витрати

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

$$I_B = (3_o + 3_p) \cdot \frac{H_{iB}}{100\%} \quad (5.7)$$

де H_{iB} — норма нарахування за статтею «Інші витрати».

$$I_{B.K.} = 5800 \cdot \frac{50\%}{100\%} = 2900 \text{ (грн.)},$$

$$I_{B.B.} = 16000 \cdot \frac{50\%}{100\%} = 8000 \text{ (грн.)},$$

$$I_B = I_{B.K.} + I_{B.B.} = 10900 \text{ (грн.)}.$$

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

$$V_{H3B} = (Z_o + Z_p) \cdot \frac{H_{H3B}}{100\%} \quad (5.8)$$

де H_{H3B} — норма нарахування за статтею «Накладні (загальновиробничі) витрати».

Беремо норму нарахування 150%.

$$V_{H3B.K.} = 5800 \cdot \frac{150\%}{100\%} = 8700 \text{ (грн.)},$$

$$V_{H3B.B.} = 16000 \cdot \frac{150\%}{100\%} = 24000 \text{ (грн.)},$$

$$V_{H3B} = V_{H3B.K.} + V_{H3B.B.} = 32700 \text{ (грн.)}.$$

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

$$V_{\text{заг}} = Z_o + Z_{\text{дод}} + Z_n + M + K_B + V_{\text{спец}} + V_{\text{прг}} + A_{\text{обл}} + V_e + V_{\text{св}} + V_{\text{сп}} + I_B + V_{H3B} \quad (5.9)$$

У нашому випадку:

$Z_p = 0, K_B = 0, V_{\text{спец}} = 0, V_{\text{прг}} = 0, V_{\text{св}} = 0, V_{\text{сп}} = 0$, тому отримаємо:

$$\begin{aligned} V_{\text{заг}} &= Z_o + Z_{\text{дод}} + Z_n + M + A_{\text{обл}} + V_e + I_B + V_{H3B} \\ &= 21800 + 2616 + 5371,52 + 727,28 + 1562,5 + 72,44 + 10900 + 32700 \\ &= 75749,74 \text{ (грн.)}. \end{aligned}$$

Загальні витрати ZB на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за $ZB = \frac{V_{\text{заг}}}{\eta}$, формулою:

де η — коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, обираємо його 0,5

$$ЗВ = \frac{75749,74}{0,5} = 151499,48 \text{ (грн).}$$

5.3 Розрахунок економічної ефективності науково-технічної розробки за її впровадження безпосередньо розробником (замовником)

Для оцінки та прогнозування зростання чистого прибутку від впровадження розробником нової науково-технічної продукції на своєму підприємстві (організації) використаємо метод прямої оцінки зростання чистого прибутку підприємства від впровадження результатів наукової розробки [18]. У цьому випадку збільшення чистого прибутку підприємства $\Delta\Pi_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою 5.11:

$$\Delta\Pi_i = (\Delta\Pi_{\text{я}} \cdot N + \Pi_{\text{я}} \cdot \Delta N)_i, \quad (5.11)$$

де $\Delta\Pi_{\text{я}}$ — покращення основного якісного показника від впровадження результатів розробки у даному році;

N — основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN — покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$\Pi_{\text{я}}$ — основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

N — кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

Припустимо, що в результаті впровадження наукових досліджень за допомогою реклами кількість учасників товариства зростає, що дозволяє

проводити більшу кількість учасників проведення тестування за рік з 100 до 150, за другий рік з 150 до 300, за третій рік, з 30 до 40, а плату за організацію підвищити на 200 грн.

Орієнтовно — поповнення бюджету розробника (замовника) до впровадження проєкту складало близько 15000 грн, кількість учасників проведення тестування на рік на 100.

Тоді збільшення чистого прибутку товариства $\Delta\Pi_1$ протягом першого року складе:

$$\Delta\Pi_1 = 200 * 100 + (1500 + 200) * 50 = 105000$$

Збільшення чистого прибутку товариства $\Delta\Pi_2$ протягом другого року (відносно базового року, тобто року до впровадження результатів наукового дослідження) складе:

$$\Delta\Pi_2 = 200 * 100 + (1500 + 200) * (50 + 150) = 360000$$

Збільшення чистого прибутку товариства $\Delta\Pi_3$ протягом третього року (відносно базового року, тобто року до впровадження результатів наукового дослідження) складає:

$$\Delta\Pi_3 = 200 * 100 + (1500 + 200) * (50 + 150 + 100) = 530000$$

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.12)$$

де $\Delta\Pi_i$ — збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T — період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$, обираємо $\tau 0,1$;

t — період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$ПП = \frac{105000}{(1 + 0,1)^2} + \frac{360000}{(1 + 0,1)^3} + \frac{530000}{(1 + 0,1)^4} = 719247,32 \text{ (грн.)}$$

Далі розраховують величину початкових інвестицій PV , які розробник (замовник) має вкласти для здійснення науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{\text{ИНВ}} \cdot 3B, \quad (5.13)$$

де $k_{\text{ИНВ}}$ — коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, які можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо. Зазвичай $k_{\text{ИНВ}} = 2 \dots 5$, але може бути і більшим. Обираємо даний коефіцієнт 2;

$3B$ — загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 \cdot 151499,48 = 302998,96 \text{ (грн.)}$$

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

$$E_{\text{абс}} = ПП - PV \quad (5.14)$$

де $ПП$ — приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн;

PV — теперішня вартість початкових інвестицій, грн.

$$E_{\text{абс}} = 719247,32 - 302998,96 = 416248,17 \text{ (грн.)}$$

Оскільки $E_{\text{абс}} > 0$, то результат від провадження наукових досліджень та їх впровадження принесе прибуток.

Внутрішня економічна дохідність інвестицій E_B , які можуть бути вкладені розробником (замовником) у впровадження науково-технічної розробки, розраховується за формулою:

$$E_B = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} - 1, \quad (5.15)$$

де $E_{\text{абс}}$ — абсолютний економічний ефект вкладених інвестицій, грн;

PV — теперішня вартість початкових інвестицій, грн;

$T_{\text{ж}}$ — життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, роки.

$$E_B = \sqrt[3]{1 + \frac{416248,17}{302998,96}} - 1 = 0,35$$

Далі розраховуємо період окупності інвестицій $T_{\text{ок}}$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{\text{ок}} = \frac{1}{E_B} \quad (5.16)$$

де E_B — внутрішня економічна дохідність вкладених інвестицій.

$$T_{\text{ок}} = \frac{1}{0,35} = 2,8 \text{ року}$$

Оскільки $T_{\text{ок}} = 2,8$ року, тоді розвиток доречний.

ВИСНОВКИ

Під час виконання магістерської кваліфікаційної роботи, вивчивши різні інформаційні системи експертного типу, чат-боти та отримавши відгуки від людей, які займаються медичною професією, можна зробити висновок, що:

— медична інформаційна система на основі чатів може бути дуже корисною для зменшення стресу, який лікарям доводиться щодня переживати;

— однак результати системи все ж повинні бути перевірені законним лікарем, перш ніж вони будуть надані пацієнту;

— таку систему можна вдосконалити, щоб вона включала більше запитань, для цього потрібно розширити базу знань та набрати більшу кількість експертів.

Загалом отримані результати під час аналізу сфер використання та технологій розробки готових інформаційних систем у медичній галузі свідчать про попит на продукцію. Наростаючий що дня, інтерес до новітніх технологій та швидке зростання галузі мобільних додатків та чат-ботів свідчать про актуальність проведених досліджень та їх конкурентну спроможність на ринку сучасних інформаційних технологій.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

- 1.Євтухівський О.В. Інформаційна експертна система постановки медичного діагнозу. URL: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2022/paper/view/15272>
- 2.Кохане, Ісаак Самуїл. Тимчасові міркування в медичних експертних системах. Бостонський ун-т, МА (США), 1987 рік.
- 3.Шортліфф, Едвард Х. та Лоуренс М. Фаган. "Дослідження інформаційних систем: моделювання процесу прийняття медичних рішень". (1982).
- 4.Хілл, Дженніфер, У. Рендольф Форд та Інгрід Г. Фаррерас. "Справжні розмови зі штучним інтелектом: порівняння між Інтернет-розмовами людини та людини та бесідами між людиною та чатом". Комп'ютери в поведінці людини 49 (2015): 245-250.
- 5.Дутта, Сумітра. "Тимчасові міркування в медичних експертних системах". Інженерія комп'ютерних медичних систем, 1988., Матеріали симпозіуму. ІЕЕЕ, 1988.
- 6.Уотерман, Д. Будівництво інформаційних систем / ред. Ф. Хейес-Рот, Д. Уотерман, Д. Ленат. - М.: Мір, 2013. - 441 с.
- 7.Комп'ютерні та нелінійні явлення. Інформатика та сучасне естествознавство / ред. А.А. Самарський. - М.: Наука, 2017. - 192 с.
- 8.Белоко Льюнг, Л. Ідентифікаційна система / Л. Льюнг. - М.: Наука. Головна редакція фізико-математичної літератури, 2015. - 432 с.
- 9.Любарський, Ю.Я. Інтелектуальні інформаційні системи / Ю.Я. Любарський. - М.: Наука, 2013. - 232 с.
- 10.Нариньяни, А.С. Моделювання мовної роботи в інтелектуальних системах / ред. А.Е. Кибрик, А.С. Нариньяни. - М.: Наука. Головна редакція фізико-математичної літератури, 2012. - 280 с.
- 11.Нейлор, К. Як побудувати свою інформаційно-експертну систему / К. Нейлор. - М.: Энергоатомздат, 2013. - 286 с.

12.Нільсон, Н. Принципи штучного інтелекту / Н. Нільсон. - М .: Радіо і зв'язь, 2014. - 373 с.

13.Джарратано Д. Інформаційні системи експертного типу: принципи розробки та програмування [Текст] / Д. Джарратано, Г. Райлі; пер. з англ. - М.: Вільямс, 2007., - 1152 с. : Ил. - ISBN 978-5-8459-1156-8.

14.Джексон П. Введення в інформаційні системи [Текст]: пер. з англ. /П. Джексон. - 617-622. - ISBN 5-8459-0150-2.

15.Кавецький В.В. Економічне обґрунтування інноваційних рішень: Практикум/В.В. Кавецький, В.О. Козловський, І.В. Причепа. — ВНТУ, 2013. — 100 с.

16.Адлер О.О. Методичні вказівки до підготовки та написання курсової роботи з дисципліни «Економічне обґрунтування інноваційних рішень» / Уклад. О.О. Адлер, І.В. Причепа, Н.М. Тарасюк — Вінниця, ВНТУ, 2014. — 38 с.

17.Тарифи на електроенергію [Електронний ресурс]. Режим доступу: <http://index.minfin.com.ua/tarif/electric.php>. Дата звернення: Лютого 11, 2022.

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

ДОДАТОК А

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ, д.т.н., проф.


_____ Азаров О.Д.

«__» _____ 2022 року


ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи
«Актуальність інформаційно-аналітичних систем електронної торгівлі під
час карантину»
08-23.МКР.002.00.000 ТЗ

Виконав: студент 2-го курсу, групи КІ-20м

_____  Свтухівський О.В.

Науковий керівник: к.т.н., доцент каф. ОТ

_____  Муращенко О.Г.

Вінниця 2022

1 Найменування та галузь застосування

Магістерська кваліфікаційна робота має назву «Інформаційна система постановки медичного діагнозу».

Галузь застосування — інформаційно-експертні системи медицини.

2 Підстава для розробки

Підставою для виконання магістерської кваліфікаційної роботи (МКР) є індивідуальне завдання на МКР та наказ ректора ВНТУ №65 від 24 березня 2022 року та рішення засідання кафедри обчислювальної техніки (протокол № 11 від 08 лютого 2022 року).

3 Мета та призначення розробки

Метою і завдання магістерської роботи є розробка інформаційної системи, яка має допомогти людині, у рекомендованому форматі проінформувати про можливі проблеми зі здоров'ям або проінформувати про вірогідний діагноз, згідно вказаних людиною симптомів.

Призначення роботи — розробка ефективної інформаційної системи на базі чат-боту платформи Telegram, з можливістю оновлюватись, вдосконалюватись задля покращення ефективності алгоритмічних виводів кінцевому користувачу.

4 Вихідні дані для проведення МКР

База даних можливих симптомів, база даних можливих діагнозів, структура інформаційних систем, задачі інформаційно-експертних систем, вимоги до функціональних можливостей, структурні можливості та використання чат-ботів.

5 Вимоги до виконання МКР

В МКР необхідно:

- проаналізувати особливості використання інформаційних систем та інформаційно-експертних систем;
- розглянути види сучасних інформаційно-експертних систем;
- застосувати сучасні технології розробки інтернет-інтегрованих інформаційних систем сумісними з максимальним різновидом типів пристрою та ОС;
- обрати та використати технології програмування чат-боту;
- оцінити функціональні можливості інформаційної системи.

6. Матеріали, що подаються до захисту МКР

До захисту подаються: пояснювальна записка МКР, графічні і ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відгук наукового керівника, відгук рецензента, анотації до МКР українською та іноземною мовами, довідка про відповідність оформлення МКР діючим вимогам.

7. Вимоги до оформлення роботи

7.1 При оформлюванні МКР використовуються:

- ДСТУ 3008 : 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;
- ДСТУ 8302 : 2015 «Бібліографічні посилання. Загальні положення та правила складання»;
- ГОСТ 2.104-2006 «Єдина система конструкторської документації. Основні написи»;
- документами на які посилаються у вище вказаних.

7.2 Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21».

7.3 При розробці програмних засобів слід дотримуватись уніфікації і ДСТУ.

8. Стадії та етапи виконання роботи


Стадії та етапи виконання роботи занесені в таблицю А.1

Таблиця А.1 — Етапи виконання магістерської кваліфікаційної роботи

№ з/п	Назва етапів дипломного роботи	Строк виконання етапів роботи	Примітка
1	Аналіз поставленої задачі	01.10.2021	Виконано
2	Огляд існуючих реалізацій	02.10.2021 – 14.11.2021	Виконано
3	Аналіз методів розробки інформаційних систем та їх використання	15.11.2021 – 05.12.2021	Виконано
4	Вивчення зв'язку емпіричних і числових систем	06.12.2021 – 02.01.2022	Виконано
5	Вивчення технологій для розробки системи на основі чат-бота	03.01.2022 – 06.02.2022	Виконано
6	Розрахунок перспективи продукту та його функціонал	07.02.2022 – 14.03.2022	Виконано
7	Проектування, розробка чат-бота та його тестування	15.03.2022 – 24.04.2022	Виконано
8	Розрахунок економічної доцільності системи та оформлення економічної частини в пояснювальній записці	25.04.2022 – 13.05.2022	Виконано
9	Перевірка якості виконання магістерської кваліфікаційної роботи та усунення недоліків	16.05.2022	Виконано

9. Порядок контролю та прийому

Виконання етапів магістерської кваліфікаційної роботи контролюється керівником згідно з графіком виконання роботи. Захист магістерської кваліфікаційної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком.

Технічне завдання до виконання отримав  Євтухівський О.В.

ДОДАТОК Б

Алгоритм розробки інформаційно-експертної системи



Рисунок Б.1 — Блок схема інформаційно-експертної системи

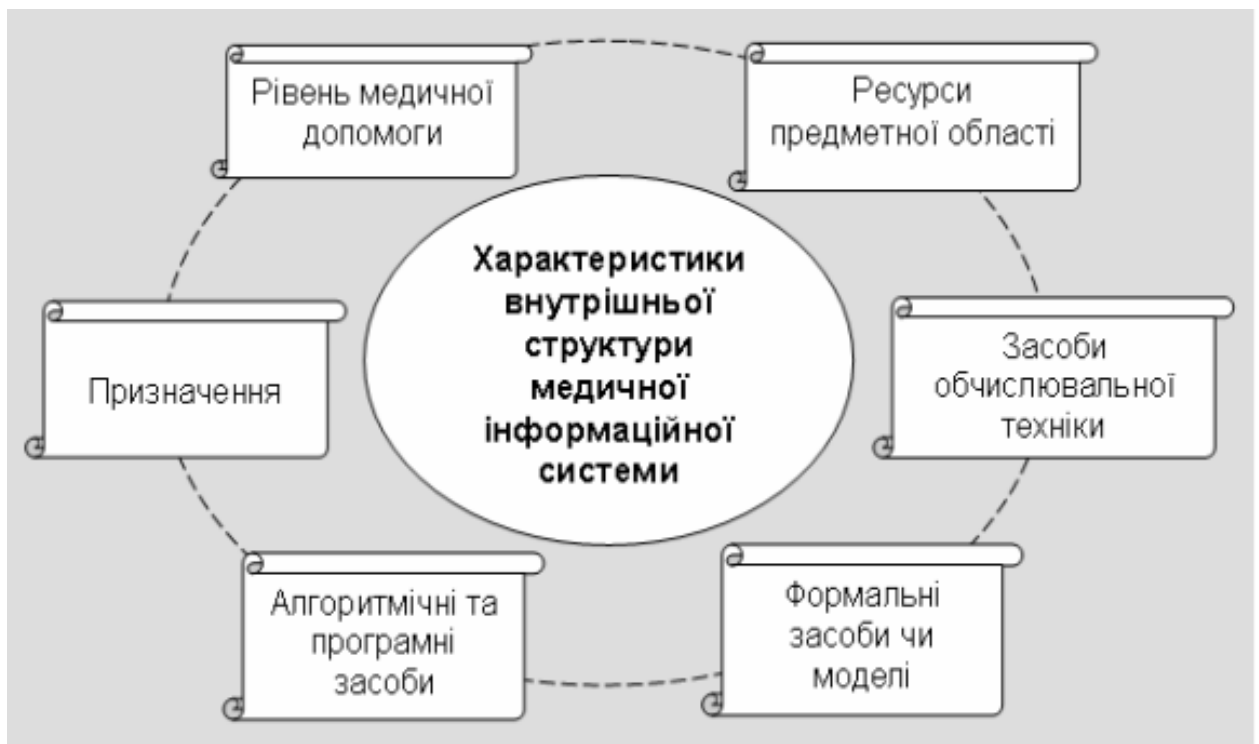




Рисунок Б.2 — Блок схема характеристики внутрішньої структури МІС

					08-23.МКР.002.00.001			
Змн.	Лист	№ докум.	Підпис	Дата				
Розроб.	Євтухівський О.В.				<i>Додаток Б — Алгоритм розробки інформаційно- ..</i>	Літ.	Арк.	Аркушіє
Перевір.	Муращенко О.Г.							
Н. Контр.	Швець С.І.					Гр. КІ-20м		
Реценз.	Дудатъев А.В.							
Затверд.	Азаров О.Д.							

ДОДАТОК В

Лістинг чат-бота

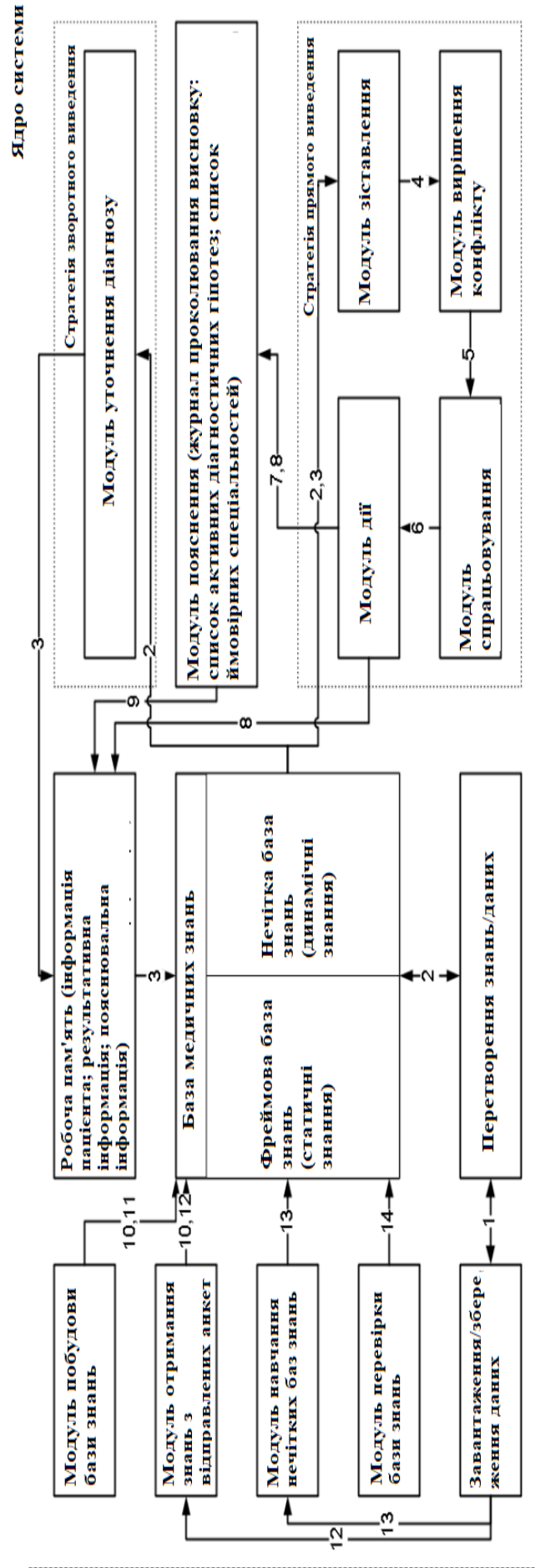
```

updater.addTelegramMessageHandler (accept_message)
# start polling the Telegram server...
Updater.start_polling (0.1)
//Обробник команд Telegram
d = { # populate dict with useful
information about message }
MESSAGE_QUEUE.put (d)
//Загальна черга "Message_queue"
While True:
if messages_present_in_queue:
user_info = get_most_recent_message_from_queue
m =send_message_to_dispatcher_for_processing(user_info)
add_machine_response_to_dispatch_queue (m, user_info)
//Віправка повідомлень до диспетчера
object_list=dict ()
if conversation_with_chat_id_over(chat_id):
object_list [chat_id] =assign_new_core_object ()
core_object=object_list [chat_id]
core_object.process_message_and_return_reply()
//Узагальнений функціонал диспетчера
Expert = expert_system ()# instantiate expert system
module if expert.done == 1:# conversation is over return
None else:# conversation still in progress
return_message=expert.run_expert_system (incoming_message)
return return_message
//Узагальнений функціонал диспетчера
If user_has_began_conversation() :
```

```
status=1
if status == 1:# First question is being asked.
status == 2# First question asked.
get_first_question_from_doctor_skynet ()
if status == 2:# System expecting a reply from user.
send_to_doctor_skynet_adn_get_next_question ()
if no_more_question_remain ():
state=3
done=1# Indicates
return_conversation_done_to_core ()
return_question_to_core ()
//Загальний функціонал інформаційної системи
While conversation_in_progress ():
update_bucket_fractions ()
if basic_questions_unanswered ():
init_basic_questions_conversation ()
if symptom_queries_not_over ():
init_symptom_query_conversation ()
//Загальний вигляд системи DoctorSkyNet
```



ДОДАТОК Г

Структурна схема ядра системи



- 1 - Медичні дані
- 2 - Медичні знання
- 3 - Факти
- 4 - Потенційне захворювання
- 5 - Активне захворювання
- 6 - Дігностична гіпотеза
- 7 - Список активних діагностичних гіпотез
- 8 - Список можливих спеціальностей
- 9 - Пояснювальна інформація
- 10 - Знання та досвід групи лікарів
- 11 - Фреймової та нечітко-продукційної моделі
- 12 - Надіслані анкети
- 13 - Навчальні нечіткі знання та навчальні вибірки
- 14 - Тестуючі вибірки

Рисунок Г.1 — Структурна схема ядра системи

					<i>08-23.МКР.002.00.002</i>			
<i>Змн.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	<i>Додаток Г — Структурна схема ядра системи</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркуші</i>
<i>Розроб.</i>	<i>Євтухівський О.В.</i>	<i>Муращенко О.Г.</i>						
<i>Перевір.</i>	<i>Муращенко О.Г.</i>	<i>Швець С.І.</i>						
<i>Н. Контр.</i>	<i>Швець С.І.</i>	<i>Дудатьєв А.В.</i>				<i>Гр. КІ-20м</i>		
<i>Реценз.</i>	<i>Дудатьєв А.В.</i>	<i>Азаров О.Д.</i>						
<i>Затверд.</i>	<i>Азаров О.Д.</i>							

ДОДАТОК І

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Інформаційна система постановки медичного діагнозу

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра обчислювальної техніки

(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 85,9 Схожість 14,1

Аналіз звіту подібності (відмітити потрібне):

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

Особа, відповідальна за перевірку _____
(підпис)

Захарченко С.М.
(прізвище, ініціали)


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

Автор роботи


(підпис)

Євтухівський О.В.
(прізвище, ініціали)

Керівник роботи


(підпис)

Муращенко О.Г.
(прізвище, ініціали)