Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій та технічного захисту інформації
(повна назва)

Кафедра Радіотехнологій інформаційно-комунікаційних систем
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ГЮІК.ХХХХХХ.012 ПЗ
Дослідження експертної системи діагностування стану людини в умовах надзвичайних ситуацій
(тема)

Виконав: студент 2 курсу, групи ІКТм-18-1
Шило Родіон Сергійович
(прізвище, ініціали)
спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма
інформаційно-комунікаційні технології
(повна назва освітньої програми)

Керівник д.т.н., професор Кузьомін О. Я.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Цопа О. І.
(підпис)(прізвище, ініціали)

2019 р

Не містить відомостей заборонених для відкритого публікування.

Керівник д.т.н., професор Кузьомін О. Я.

Студент Шило Р.С.

Харківський національний університет радіоелектроніки

Факультет Інформаційних радіотехнологій та технічного захисту інформації

Кафедра__радіотехнологій інформаційно-комунікаційних систем ___

Рівень вищої освіти_____другий (магістерський)_____

Спеціальність _____122 Комп'ютерні науки_____
(код і повна назва)

Тип програми _____освітньо-професійна_____
(освітньо-професійна або освітньо-наукова)

Освітня програма ____інформаційно-комунікаційні технології_____
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові_____Шило Родіону Сергійовичу_____
(прізвище, ім'я, по батькові)

1. Тема роботи _Дослідження експертної системи діагностування стану людини в умовах надзвичайних ситуацій_____
_____
затверджена наказом по університету від __21 _____11_____2019 р. № ___1730_____
2. Термін подання студентом роботи до екзаменаційної комісії _20___грудня____ 2019 р.
3. Вихідні дані до роботи ____Науково-технічні публікації та інтернет джерела з тематики атестаційної роботи_____
_____
_____
_____
_____
_____
_____
_____
4. Перелік питань, що потрібно опрацювати в роботі _вступ, аналіз предметної області та постановка задачі, загальні відомості про експертні системи, технології для розробки____ системи медичного експерту на основі чат-бота, аналіз та проектування чат-бота, _____ висновки_____
_____
_____
_____
_____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п. 5 включається до завдання за рішенням випускової кафедри) _Слайди презентації_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	Проф.кафр.інформатики Кузьомін О.Я.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Об'єктний аналіз поставленої задачі	10.10.19	виконано
2	Аналіз предметної галузі	25.10.19	виконано
3	Аналіз існуючих алгоритмів	08.11.19	виконано
4	Дослідження існуючих методів	20.11.19	виконано
5	Пошук та вирішення недоліків алгоритмів та методів	03.12.19	виконано
6	Підготовка пояснювальної записки.	11.12.19	виконано
7	Підготовка презентації та доповіді	11.12.19	виконано
8	Попередній захист	18.12.19	виконано
9	Нормоконтроль, рецензування	19.12.19	виконано
10	Занесення диплома в електронний архів	19.12.19	виконано
11	Допуск до захисту у зав. кафедри	19.12.19	виконано

Дата видачі завдання __21___ _листопада__ 2019 р.

Студент _____
(підпис)

Керівник роботи _____ _____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до магістерської атестаційної роботи містить: 93с., 4 розділи, 17 рис., 6 табл.,14 джерел.

ЕКСПЕРТНА СИСТЕМА, ЧАТ-БОТ, CLIPS, TELEGRAM, БАЗА ЗНАНЬ, БАЄСОВА МЕРЕЖА, PYTHON.

В роботі виконано огляд обгортки експертної системи CLIPS. Проаналізовано чинні експертні системі у предметній області.

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

ABSTRACT

The explanatory note to the master's certification work contains:
93 pages, 4 sections, 17 figures, 6 tables, 14 sources.

EXPERT SYSTEM, CHAT-BOT, CLIPS, TELEGRAM, KNOWLEDGE, BAYESIAN BELIEF NETWORK, PYTHON.

This paper provides an overview of the wrapper of the CLIPS expert system. Existing expert systems in the subject area are analyzed.

In the course of the study the following results were obtained: the existing software of expert systems were identified, the architecture studied, the methods of design, and the scope of use of these systems. Developed project plan: defined list of works, deadlines.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ЕС – експертна система;

ІДС – інтелектуальна діагностична система;

БД – база даних;

БЗ – база знань;

АП – аналітичні показники;

ФП – фізичні показники;

БМД – байєсівської мережі довіри;

ТУЙ – таблицею умовних ймовірностей;

ШІ – штучний інтелект;

ФУЙ - функцією умовних ймовірностей.

ВСТУП

Досвід розробки та експлуатації комп'ютерних обчислювальних систем різного призначення показав, що їх ефективність визначається не лише якістю закладених алгоритмів, але й обсягом знань, які вони використовують під час свого функціонування. Реалізація даної концепції привела до розвитку спеціалізованих програмних систем, які є, свого роду, «експертами» у певній предметній галузі — експертних систем.

Найважливіша ознака експертних систем полягає в орієнтації на процеси накопичення та організації знань. Стратегії експертних систем базуються на використанні високоякісних елітних знань людей – висококваліфікованих спеціалістів-експертів у певній предметній області. Такі знання сконцентровані в базах знань експертних систем та використовуються для розв'язання поточних практичних задач. Таким чином, експертні системи виконують роль асистента-радника для особи, яка приймає рішення в процесі управління або обслуговування складних технічних, соціальних або природних системам, у тому числі – медичних.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Сучасні проблеми

Мета цього проекту - розробити систему медичних експертів на основі чатів. Лікарям при надзвичайній ситуації доводиться працювати довгі години і бачити сотні пацієнтів на день. Багато лікарів витрачають дорогоцінний час на збір особистої та медичну історію пацієнтів щодо їх демографії, сучасних та минулих симптоми.

Мета проекту - посилити першу взаємодію лікар-пацієнт, коли лікар задає пацієнтові певні питання, щоб занотувати історію хвороби пацієнта. Щоб звільнити лікаря та зекономити його дорогоцінний час, можна взяти основну історію хвороби у пацієнта за допомогою розмови з доброзичливим чатом-ботом.

Цей проект має на меті лише діяти як проміжна людина між пацієнтом та лікарем. Він працює, дозволяючи пацієнтові спілкуватися із системою, щоб він / вона могла повідомляти про свою персональну та медичну історію ботові, за яким комп'ютер веде облік. Потім система спробує встановити діагноз із цієї взаємодії і повідомити про це безпосередньо лікареві, надіславши лікарю електронний лист.

1.2 Передумови та основи

Експертна система [1] - це комп’ютерна система, яка імітує здатність людини приймати рішення. Як описано вище, більшість лікарів при

надзвичайній ситуації перевантажена кількістю пацієнтів, яких їм доводиться бачити. Пропонується нове рішення цієї проблеми, створивши медичного експерта на базі системи, яка може вести дружню розмову з пацієнтом, щоб відзначити його історію хвороби.

З цією метою використовується додаток Telegram для обміну повідомленнями як інтерфейс для пацієнта. Telegram-bot - це широко використовуваний, крос-платформний, відкритий і високозахищений додаток для чату, який дозволяє ботам вести бесіди з користувачами. Це робиться за допомогою RESTful API, з яким може взаємодіяти будь-який чат-бот.

1.3 Огляд існуючих реалізацій

Експертні системи, засновані на чат-ботах, існують з останніх 30 років [2]. Примітним прикладом є знаменитий “Eliza” чат-бот. Чат-боти, засновані на онтології, - це постійні зусилля, засновані на семантичній мережі і моделях NLP (Natural Language Processing) [3] для підтримка діалогових взаємодій користувач-машина. Система працює, використовуючи 3 модуля - засновані на знаннях система, інтерпретація питань і Natural Language Generation. Онтологія - це формальне іменування і визначення типів, властивостей і взаємозв'язків між сутностями.

Важливою подією в створенні експертних систем на основі чату є розробка системи Temporal Reasoning. Тимчасові міркування важливі, тому що остаточна діагностика сильно залежить від послідовність симптомів. Тимчасові міркування і логіка - це система правил для представлення та міркування про судженнях кваліфіковані з точки зору

часу, наприклад, я ЗАВЖДИ голодний. Стаття [3] досліджує тимчасові відносини між симптомами, заснованих на теорії нечітких множин.

В іншій оглядовій статті розкриваються цікаві способи моделювання процесу прийняття медичних рішень [4]. Вона аналізує AI і експертні системи, щоб познайомити читача з полем і запропонувати способи, за допомогою яких дослідження в кінцевому підсумку будуть застосовуватися для подальшого медичного моніторингу. У ній говориться, що експертна система є система, яка містить і може застосовувати спеціалізовані знання. Вона використовує ці знання для припущень користувачів, які можуть не мати повного спектру доступних знань.

Різниця між експертними системами і системами, заснованими на знаннях, полягає в тому, що експертна система здатна пояснити її поведінку користувачеві. У пояснювальній записці також аналізуються різні підходи, прийняті для реалізації медичної експертної системи.

1.4 Аналіз методів розробки експертних систем

Розвиток нових методів діагностики, а також розширення можливостей вже існуючих методів було і залишається актуальним завданням в медицині [6]. Поява нових діагностичних і лікувальних технологій вимагають використання методів штучного інтелекту для обробки та інтерпретації даних з можливістю накопичення, зберігання і багаторазового використання медичних даних. Одним з найбільш ефективних засобів у цій галузі є експертні системи. Вони дозволяють автоматизувати процес прийняття рішення при огляді пацієнтів, підвищуючи рівень кваліфікації користувача до рівня досвідчених

експертів. Тому необхідно, щоб експертні системи володіли можливістю гнучкої постановки завдань, були здатні до всіх областей біології і медицини, володіли великою інформаційною ємністю і перешкодостійкістю, не потребували тривалому часу для розробки.

Безперервний розвиток засобів обчислювальної техніки розширює потенційні можливості подібних систем, в зв'язку з чим необхідно постійно поповнювати знання в даній області.

Експертні системи протягом тривалого часу застосовуються в медицині для діагностики захворювань. Кожна система при цьому володіє обмеженою сферою застосування, в зв'язку з початковою спрямованістю розробки застосування цих експертних систем в областях, для яких вони спочатку не були призначені, утруднене і часто просто неможливо, в тому числі тому, що вони обмежені вбудованими засобами виведення.

Рішення задач розробки експертних систем можна поділити на дві великі категорії:

- завдання, які вирішуються за явними алгоритмам;
- завдання прийняття рішень на основі досвіду і знань.

Відповідно, можна виділити 2 способи вирішення завдання - логічний і інтуїтивний. Логічний метод оперує набором формальних правил, інтуїтивний - накопиченим досвідом. При вирішенні завдання першим способом завдання зазвичай розбивається на підзадачі, кожна з яких, в свою чергу, розбивається на кілька елементарних функцій з відомим алгоритмом обчислень. Знаючи алгоритм кожної елементарної функції, можна вирішувати найскладніші завдання, поєднуючи елементарні функції в потрібній послідовності. При спробі застосувати такий метод до вирішення інтуїтивних завдань програміст може зіткнутися

з неможливістю виділити алгоритм елементарної функції або взагалі розбити задачу на такі функції.

Існують традиційні системи прийняття рішень, що базуються на явних правилах виведення, створені, як правило, групою фахівців, в числі яких - математики, програмісти і предметні фахівці, що ставлять завдання. Можливості настройки таких систем для кінцевого споживача часто недостатні.

Самонавчальна медичні експертні системи (ЕС) прийняття рішень повинні відповідати таким вимогам [6]:

- індивідуалізація (настройка на традиції клінічних шкіл, геосоціальні особливості регіону застосування, набори медико-біологічних даних, особливості лікувально-діагностичних технологій, індивідуальний досвід і знання фахівця);
- динамічний розвиток (накопичення досвіду системи в процесі функціонування, слідуючи змін в пунктах, зазначених у попередньому вимозі); можливість перенастроювання при різкій зміні умов, наприклад, при перенесенні в інший регіон;
- здатність до екстраполяції результату - вимога, зворотне індивідуальності. Система не повинна різко втрачати якість роботи при зміні умов;
- можливість конструювання з нуля кінцевим користувачем (фахівець повинен мати можливість придумати абсолютно нову ЕС і мати можливість просто і швидко створити її);
- «нечіткий» характер результату. Рішення, що видається системою, не повинно бути остаточним. Воно може бути імовірнісним або пропонувати відразу кілька варіантів на вибір. Це дає можливість

фахівцеві критично оцінювати рішення системи і не позбавляє його ініціативи в ухваленні остаточного рішення;

- ЕС є тільки порадником фахівця, не претендуючи на абсолютну точність рішення. Вона повинна накопичувати досвід і знання і значно прискорювати доступ до них, моделювати результат при зміні умов завдання. Відповідальність за рішення завжди лежить на фахівця.

1.5 Приклади існуючих експертних систем

Більшість розроблених експертних систем задовольняють цим вимогам. Перелічимо найбільш відомі з них [6]:

- PUFF - експертна система, що здійснює діагностику легеневих захворювань на основі легеневих функціональних тестів;
- SPE - проводить діагностику станів при запальних процесах;
- ABEL - здійснює діагностику кислотних і електролітних захворювань;
- AI / RHEUM - діагностика захворювань сполучних тканин;
- CADUCEOS - діагностика внутрішніх захворювань загального профілю;
- BLUE FOX - діагностика і лікування депресивних станів;
- CASNET / GLACOMA - діагностика і лікування очних захворювань, пов'язаних з глаукомою;
- MYCIN - діагностика і лікування інфекційних захворювань;
- ONCOCIN - лікування хворих на рак хіміотерапією і спостереження за ними;
- PIP - діагностика захворювань нирок;

- МОДІС-2 - діагностика симптоматичної гіпертонії;
- GUIDON - навчальна система діагностики і лікування інфекційних захворювань.

Кожна з цих систем має вузьку областю застосування і може використовуватися за призначенням тільки в ній.

1.6 Підходи до обробки інформації

Медичні експертні системи будуються на основі декількох підходів до обробці інформації. Зазвичай використовуються такі підходи [12]:

- побудова дерева рішень;
- статистична обробка даних;
- використання елементів штучного інтелекту.

1.6.1 Побудова дерева рішень

При використанні цього підходу в програмі протоколюється послідовність питань, що задаються лікарем при вирішенні діагностичної проблеми. Такий протокол структурується у вигляді дерева. Кожна вершина такого дерева являє собою певне питання, що задається хворому, а розгалуження, які виходять із вершини, відповідають альтернативним відповідями на питання і ведуть, в свою чергу, до нових питань. Програма здійснює перехід від питання до питання до тих пір, поки не буде знайдено рішення або вичерпані можливі переходи. Недоліком такого підходу можна вважати наступне:

- при спробі побудувати подібні дерева для вирішення складних діагностичних задач кількість вершин і розгалужень стає настільки великим, що аналіз логічного дерева виявляється вкрай складним;

- найменші зміни, що вносяться до логіку програми, призводять до необхідності будувати дерево заново і перепрограмувати всю задачу.

У той же час такий підхід вкрай зручний, оскільки дозволяє уявити в програмі логіку складання послідовності питань лікарем при вирішенні діагностичної задачі в клінічних умовах. Цей підхід дозволяє імітувати процес прийняття рішення лікарем при постановці діагнозу.

У міру того, як проводиться все більше досліджень в різних областях, база знань (КБ) продовжує розширюватися поступово, а в деяких випадках експоненціально. Використання такої великої бази знань і висновок рішення, засновані на таких великих даних, складні і вимагають великої кількості зусиль і вивчення Рішення, які приймаються на основі таких існуючих знань, складаються з логічних тверджень. Система яка досліджується використовує існуючі знання в певних областях і допомагає користувачеві в прийняття рішень на його основі. Система буде спілкуватися з користувачем через платформу (Telegram Чат бот) і надавати рішення, засновані на знаннях.

Знання будуть структуровані, і по цій структурі будуть прийматися рішення з урахуванням від користувача про те, яке рішення або інформація необхідна. Наша система дуже модульна і може застосовуватися до будь-якого полю. Нами була вибрана галузь медицини, оскільки вона має широке коло соціальних вигоди. Сфера медицини має величезну базу знань про хвороби, станах, симптомах, ліки і сезонні умови. Ця система буде використовувати медичну галузь в якості свого знання бази і намагається змоделювати перші взаємодія, яке пацієнт має з лікарем.

Ця система не замінює і не замінить лікарів, але полегшить діагностику і зробити перший візит пацієнта до лікаря більш продуктивним. Ми розробили систему з метою поліпшення історій хвороб,

отримання досвіду і надання можливості пацієнтам почати діагностику віддалено.

1.6.2 Статистична обробка даних

Цей підхід полягає в застосуванні методів математичної статистики. Він грунтується на обробці великих масивів інформації, зібраних по захворюваннях, що підлягають машинній діагностиці. Оброблена інформація може використовуватися різним чином.

Частина медичних експертних систем базується на використанні теорії розпізнавання образів. При цьому необхідно мати деякий безліч конкретних історій хвороб з відомими діагнозами. Такі безлічі аналізуються з метою визначення статистично «типовою» для кожного захворювання картини - образа.

Визначаються ті ознаки захворювання, які найбільш характерні для нього, виходячи із зібраної інформації, поданої статистичної обробки. «Типові» картини захворювань використовуються при аналізі історії кожного конкретного пацієнта, для визначення того, наскільки «схожий» розглянутий випадок на «типовий».

Оцінюючи «відстань» між порівнюваними картинами, програма формує рішення про діагноз. Ряд методів комп'ютерної діагностики заснований на застосуванні теореми Байеса. Такий підхід дозволяє визначити ймовірності захворювання у людини і базується на встановлення частоти появи ознак при захворюванні. Наприклад, припустимо, що D_i - одне з n захворювань, що підлягають діагностиці, а E - ознаки, які спостерігаються у хворого. Тоді якщо $P(D_i)$ - ймовірність появи i-го захворювання, то ймовірність i-го захворювання за умови наявності ознак E буде

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

Безперечними перевагами статистичних методів є те, що з їх допомогою робиться спроба об'єктивізувати наявну інформацію про захворювання. Присутні у них також і суттєві недоліки.

Перший з них пов'язаний з тим, що інформація, необхідна для побудови статистичних моделей, часто відсутня. Для її накопичення, зберігання і обробки потрібні спеціальні засоби обчислювальної техніки, створення банків даних по захворювань. Кількість систематизованої інформації по окремим областям медицини ще мало. Та й сама побудова статистичних моделей є непростою справою.

Використання теореми Байеса [14] має деякі обмеження, пов'язані з тим, що воно грунтується на використанні деяких припущень. Насамперед, передбачається, що кожне захворювання має свої непересічні набори симптомів, в той час як на практиці часто зустрічається ситуація, коли однакові симптоми зустрічаються у різних захворювань. Інша допущення використовується при визначенні умовних ймовірностей появи ознак при захворюванні і засноване на незалежності симптомів між собою. Такі залежності, тим не менш, є і накладають суттєві обмеження на застосовність теореми.

Ще один істотний недолік цих методів полягає в тому, що внесення в моделі нової інформації являє собою значні труднощі, тому що веде до зміни самої моделі і перерахунку всіх ймовірностей.

Також недоліком є неможливість пояснення лікаря і хворого того, яким саме чином отриманий даний результат. Причиною цього є те, що даний результат є наслідком математичних операцій і не має нічого

спільного зі звичайним процесом прийняття рішення лікарем при обстеженні пацієнта.

1.6.3 Використання елементів штучного інтелекту

Обробка даних, побудована на основі використання елементів штучного інтелекту [14]. У таких системах робиться спроба моделювання здатності людини до розгляду предметної області та здійснення умовиводів з відкиданням найменш перспективних напрямків пошуку. Для цього використовуються набори правил, що задаються апріорно і є, по суті, знаннями експертів в певної проблемної області. Саме якість знань визначає «компетентність» проблемно-орієнтованої експертної системи. Знання про проблемної області, формалізовані певним чином, зберігаються окремо від інших програм, що дозволяє модифікувати роботу системи ШІ, змінюючи набір знань і не зачіпаючи інших компонентів системи. Сукупність таких правил прийнято називати базою знань, в той час як сукупність даних про пацієнтів - базою даних.

Існує два основні підходи до організації баз знань, що виражаються в різної структурі відповідних систем:

- опис відносин між описаними об'єктами;
- уявлення обробки знань у вигляді процедур.

Серед найбільш відомих способів подання знань згадаємо продукційний. При цьому підході реалізується чіткий поділ між даними, операціями і управлінням. Можна виділити якусь глобальну базу даних (не слід її плутати з базою даних в традиційному розумінні), над якою виконуються деякі дії, описувані сукупністю правил продукції. Управління цим процесом відбувається відповідно до деякої глобальної стратегією управління.

Глобальна база даних - це структура даних, анализируемая і перетворюються системою ШІ. Вид бази залежить від розв'язування задачі і може бути простим (як, наприклад, спискова структура) або складним.

Правила продукції складаються з двох частин: умова (ліва частина) і дію (Права частина). Умова встановлює застосовність правила до бази даних. Дія полягає в зміні інформації в глобальній базі даних.

Система управління вибирає, яке з кількох застосованих правил повинно бути використано, а також визначає момент припинення подальших дій шляхом перевірки термінальних умов на базі даних.

Пошук рішення здійснюється на основі двох основних підходів. Метод «від даних до мети »полягає в послідовному зміні бази даних усіма правилами, які можуть бути застосовані, і в пошуку після зміни нових застосованих правил. Пошук закінчується при наявності твердження, що є рішенням, або при виконанні умови припинення пошуку. Метод «від мети до даних» зводиться до перевірці деяких тверджень, які можуть виступати в якості найбільш ймовірних рішень. При цьому проводиться пошук правих частин правил з метою виявлення шуканого затвердіння і перевірка лівих частин відповідних правил на базі даних - при підтвердженні або гіпотеза вважається дійсною, або ліва частина розглядається в якості нової гіпотези

2 ЗАГАЛЬНІ ВІДОМОСТІ ПРО ЕКСПЕРТНІ СИСТЕМИ

2.1 Основні режими роботи експертних систем

В роботі ЕС [6] можна виділити два основні режими: режим придбання знань і режим рішення задачі (режим консультації або режим використання). В режимі придбання знань спілкування з ЕС здійснює експерт (за допомогою інженера знань).

Використовуючи компонент придбання знань, експерт описує проблемну область у вигляді сукупності фактів і правил. Іншими словами, «наповнює» ЕС знаннями, які дозволяють їй самостійно вирішувати завдання з проблемної області. Цього режиму при традиційному підході до програмування відповідають етапи: алгоритмізації, програмування і налагодження, виконувані програмістом. Таким чином, на відміну від традиційного підходу в разі ЕС розробку програм здійснює не програміст, а експерт, який не володіє програмуванням.

У режимі консультацій спілкування з ЕС здійснює кінцевий користувач, якого цікавить результат і (або) спосіб його отримання. Необхідно відзначити, що в залежності від призначення ЕС користувач може:

- не бути фахівцем в даній галузі, і в цьому випадку він звертається до ЕС за результатом, який не вміє отримати сам;
- бути фахівцем, і в цьому випадку він звертається до ЕС з метою прискорення отримання результату, покладаючи на ЕС рутинну роботу.

Слід зазначити, що на відміну від традиційних програм ЕС при рішенні завдання не тільки виконують запропоновану алгоритмом послідовність операцій, а й сама попередньо формує її.

Добре побудована ЕС має можливість самонавчатися на розв'язуваних завданнях, поповнюючи автоматично свою БЗ результатами отриманих висновків і рішень.

2.2 Відмінність експертних систем від традиційних програм

Особливості ЕС, що відрізняють їх від звичайних програм, полягають в тому, що вони повинні володіти [12]:

1) Компетентністю, а саме:

- досягати експертного рівня рішень (тобто в конкретній предметній області мати той же рівень професіоналізму, що й експерти-люди).
- бути вмілої (тобто застосовувати знання ефективно і швидко, уникаючи, як і люди, непотрібних обчислень).
- мати адекватну робастної (тобто здатність лише поступово знижувати якість роботи у міру наближення до границь діапазону компетентності або допустимої надійності даних).

2) Можливістю до символьних міркувань, а саме:

- представляти знання в символьному вигляді
- переформулювати символьні знання. На жаргоні штучного інтелекту символ - це рядок знаків, стосується вмісту деякого поняття. Символи об'єднують, щоб висловити відносини між ними. Коли відносини представлені в ЕС вони називаються символьними структурами.

3) Глибиною, а саме:

- працювати в предметній області, що містить важкі завдання
- використовувати складні правила (тобто використовувати або складні конструкції правил, або велику їх кількість)

4) Самосвідомістю, а саме:

- дослідити свої міркування (тобто перевіряти їх правильність)
- пояснювати свої дії

Існує ще одна важлива відмінність ЕС. Якщо звичайні програми розробляються так, щоб кожен раз породжувати правильний результат, то ЕС розроблені з тим, щоб вести себе як експерти. Вони, як правило, дають правильні відповіді, але іноді, як і люди, здатні помилятися.

Традиційні програми для вирішення складних завдань, теж можуть робити помилки. Але їх дуже важко виправити, оскільки алгоритми, що лежать в їх основі, явно в них не сформульовані. Отже, помилки нелегко знайти і виправити. ЕС, подібно людям, мають потенційну можливість вчитися на своїх помилках

2.3 Технологія розробки експертних систем

Технологія їх розробки ЕС, включає в себе шість етапів (рис.2.1): етапи ідентифікації, концептуалізації, формалізації, виконання, тестування, дослідної експлуатації. Розглянемо більш докладно послідовності дій, які необхідно виконати на кожному з етапів.

1) На етапі ідентифікації необхідно виконати наступні дії:

- визначити задачі, що виникають і цілі розробки,
- визначити експертів і тип користувачів.

Рисунок 2.1 - Алгоритм розробки експертних систем

2) На етапі концептуалізації:

- проводиться змістовний аналіз предметної області,
- виділяються основні поняття і їх взаємозв'язки,
- визначаються методи розв'язання задач.

3) На етапі формалізації:

- вибираються програмні засоби розробки ЕС,
- визначаються способи подання всіх видів знань,
- формалізуються основні поняття.

4) На етапі виконання (найбільш важливому і трудомісткий) здійснюється наповнення експертом БЗ, при якому процес придбання знань розділяють:

- на "витяг" знань з експерта,
- на організацію знань, що забезпечує ефективну роботу ЕС,
- на подання знань у вигляді, зрозумілому для ЕС.

Процес придбання знань здійснюється інженером по знаннях на основі діяльності експерта.

5) На етапі тестування експерт і інженер по знаннях з використанням діалогових і пояснювальних засобів перевіряють компетентність ЕС. процес тестування триває до тих пір, поки експерт не вирішить, що система досягла необхідного рівня компетентності.

6) На етапі дослідної експлуатації перевіряється придатність ЕС для кінцевих користувачів. За результатами цього етапу можлива істотна модернізація ЕС.

Процес створення ЕС не зводиться до суворої послідовності цих етапів, так як в ході розробки доводиться неодноразово повертатися на більш ранні етапи і переглядати прийняті там рішення.

2.4 Виявлення знань від експертів

Ефективність початкових етапів розробки ЕС (етапів ідентифікації та концептуалізації) багато в чому визначається успішним формуванням авторитетної групи експертів і отриманням від них якісних знань, що складають основу будь-який ЕС.

Суть процесу виявлення знань полягає в організації проведення експертами інтуїтивно-логічного аналізу проблемної області з кількісної оцінкою сформульованих ними суджень. На цьому етапі експерти:

- формують об'єкти і поняття предметної області (цілі, рішення, альтернативні ситуації і т.д.);
- проводять вимірювання характеристик (ймовірності звершення подій, коефіцієнти значущості цілей, перевагу рішень і т.д.).

Експертне оцінювання являє собою процес вимірювання, який можна визначити як процедуру порівняння об'єктів за обраними показниками (Ознаками). У цьому визначенні фігурують три поняття:

об'єкт, показник (Ознака) і процедура порівняння. Об'єктами можуть бути предмети, явища, рішення. Як показники порівняння можуть використовуватися просторово-часові, фізичні, психічні та інші властивості і характеристики об'єктів. процедура порівняння включає в себе:

- визначення причинно-наслідкового зв'язку між об'єктами;
- а також встановлення ступеня впливу одних об'єктів на інші.

Остання обставина вимагає проведення порівняння об'єктів, що визначають будь-якої результат, за ступенем їх впливу на нього. Вступ конкретних показників порівняння дозволяє експертам встановлювати відносини між об'єктами, наприклад, "більше", "краще", "більш ніж", "гірше", "однакові", "краще" і т.д.

Для формального опису безлічі об'єктів і відносин між ними вводиться поняття емпіричної системи з відносинами

$$M = < O; R >, \qquad (2)$$

де $O = (O_1, O_2 ... O_n)$ - множина об'єктів предметної області,

$R = (R_1, R_2 ... R_n)$ - множина відносин між ними.

Запис $O_i R_k O_j$ означає, що об'єкт O_i знаходиться в відношенні R_k до об'єкта O_j .

Таке ставлення називається бінарним (двомісний) оскільки пов'язує два об'єкта. Відносини можуть бути тримісні і т.д. У загальному випадку k-місцевими, по числу об'єктів, які вони пов'язують. Визначимо основні властивості відносин:

- відношення R рефлексивно, якщо $O_i R_k O_j$ - це правда,
- відношення R симметрично, якщо з $O_i R_k$ слідує $R_k O_j$;

- відношення R транзитивне, якщо з $O_i R\, O_j$ і $O_i R\, O_k$ слідує $O_i R\, O_k$.

Ставлення, яке має властивості рефлексивності, симетричності і транзитивності називається відношенням еквівалентності.

$$O_i \sim O_j \tag{3}$$

При експертному оцінюванні, крім відносини еквівалентності використовується відношення порядку. Це ставлення може означати, наприклад: "раніше ніж", "більш ніж "," сильніше ніж "," краще ніж "і т.д. Ставлення порядку антирефлексивне, транзитивно і позначається:

$$O_i > O_j \tag{4}$$

для випадку, якщо O_i краще O_j .

Експертне оцінювання в процесі виявлення знань від експертів, використовує поряд з емпіричною, також числову систему з відносинами

$$H = < N; S >, \tag{5}$$

де N - множина дійсних чисел, а S - множина відносин між числами. Саме ця система дозволяє не тільки встановлювати прічіннослеДственние зв'язку між об'єктами предметної області, а й визначати їх взаємозв'язку і вплив.

2.5 Зв'язок емпіричних і числових систем.

При експертному оцінюванні предметної області важливим є можливість для емпіричної системи з відносинами побудови числової системи з відносинами, що описують вплив об'єктів і відносини між ними з допомогою чисел.

Для того, щоб числова система зберігала властивості і відносини об'єктів, необхідно, щоб вона була изоморфной емпіричної системі. для пояснення цього поняття визначимо поняття подібності двох систем. Дві системи з відносинами

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

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

називаються подібними, якщо:

- число відносини (заданих на множині об'єктів і дійсних чисел) однаково, тобто $k = m$;
- місцевість відносин однакова (наприклад R_i і S_i двомісні відносини).

Визначивши поняття подібності, ми можемо тепер дати визначення изоморфности двох систем (числовий і емпіричної). Числова система з відносинами $H = (N; S_1, S_2 ... S_m)$ ізоморфна емпіричної системі $M = (O; R_1, R_2 ... R_k)$ якщо ці системи подібні і 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, що відображають об'єкти на числової осі.

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

Проблема єдиності визначає: скількома способами можна описати дану емпіричну систему різними ізоморфними числовими системами, і як ці числові системи пов'язані між собою. Ця проблема формулюється, як проблема визначення типу шкали. Шкалою називається сукупність:

- емпіричної системи;

- числової системи;
- відображення, тобто $< M, H, f >$.

Нехай $< M, H, f >$ і $< M, H, f >$ дві шкали з різними відображеннями, тоді виникає 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 запишемо за допомогою функції φ: $r_i = \varphi(r'_i)$ або $f(O_i) = \varphi\,[g(O_i)]$.

Функція φ називається допустимим перетворенням шкали. Єдиність опису емпіричної системи числовими системами виражається у властивостях допустимого перетворення шкали, тобто у властивостях функції φ.

2.5.1 Методи вимірювання ступеня впливу об'єктів

До найбільш часто використовуваних при експертному оцінюванні методів належать: ранжування, парне порівняння, безпосередня оцінка. При описі кожного з перерахованих методів будемо вважати, що є кінцеве число вимірюваних об'єктів і сформульований один або кілька ознак порівняння, за якими вивчається ступінь впливу об'єктів на результат.

Отже, методи вимірювання будуть відрізнятися лише процедурою порівняння об'єктів. Ця процедура включає:

- побудова відносин між об'єктами емпіричної системи;
- вибір функції f, що відображає об'єкти емпіричної системи на числову систему;
- визначення шкали вимірювань.

Розглянемо докладно всі ці питання, що виникають при використанні кожного з методів вимірювань.

2.5.2 Метод ранжирування

Ранжування - це процедура впорядкування об'єктів за ступенем їх впливу на результат, виконується експертом в процесі виявлення його знань [10]. На основі своїх знань і досвіду експерт має об'єкти в порядку переваги, керуючись одним або декількома показниками порівняння. Залежно від виду відносин між об'єктами можливі різні варіанти впорядкування об'єктів.

Нехай серед об'єктів немає еквівалентних за ступенем впливу на результат. Ветом випадку між об'єктами існує відношення строгого порядку, що володіє властивостями:

- несиметричності (якщо $O_i > O_j$, то $O_j > O_i$);
- транзитивності (якщо $O_i > O_j$, $O_j > O_k$, то $O_i > O_k$);
- зв'язності (для будь-яких двох об'єктів, або $O_i > O_j$, або $O_j > O_i$).

В результаті порівняння всіх об'єктів по відношенню строгого порядку експертсоставляет впорядковану послідовність:

$$O_1 > O_2 > ... > O_n \qquad (8)$$

де об'єкт з номером один є найкращим з усіх об'єктів, об'єкт зі номером два менш кращий ніж перший, але краще усіх інших і т.д.

Отримана система з відношенням порядку $< O, >>$ утворює серію. Для серії доведено існування числової системи:

- елементами якої є числа;
- а відношення порядку f є ставлення "більше ніж", "краще ніж".

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

$$f(O_1) > f(O_2) > \ldots > f(O_n). \qquad (9)$$

У практиці експертного ранжування найчастіше використовується послідовність натуральних чисел

$$r_1 = f(O_1) = 1 > r_2 = f(O_2) = 2 > \ldots > r_n = f(O_n) = n \qquad (10)$$

Числа r_1, r_2, ..., r_n называются рангами. найбільш кращого присвоюється ранг 1, другому - ранг 2 і т.д. На практиці, серед об'єктів можуть бути і еквівалентні за ступенем їх впливу на результат. Наприклад, упорядкування може мати вигляд

$$O_1 > O_2 > O_3 \sim O_4 \sim O_5 \ldots > O_{n-1} \sim O_n \qquad (11)$$

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 прийнято призначати однакові ранги, рівні середнім арифметичним значенням рангів, приписуваних однаковим об'єктам. Такі ранги отримали назву пов'язаних рангів. для прикладу упорядкування у разі n = 10 ранги об'єктів O_3, O_4 і O_5 будуть однаковими і рівними:

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

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

Як видно з прикладу пов'язані ранги можуть бути дробовими. зручність використання 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 цю суму. Ця обставина істотно спрощує обробку результатів ранжування при груповій експертній оцінці.

2.5.3 Метод парних порівнянь

Парне порівняння являє собою процедуру встановлення переваги об'єктів при порівнянні всіх можливих пар. На відміну від ранжирування,

при якому здійснюється впорядкування всіх об'єктів відразу, парне порівняння представляє для експертів більш просту задачу. При порівнянні кожної пари об'єктів можливі відносини або порядку, або еквівалентності. парне порівняння є вимір в шкалі порядку.

В результаті порівняння кожної пари об'єктів O_j та O_i експерт повинен привести до ладу цю пару, висловлюючи, 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:

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

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

Нарешті, якщо об'єкти еквівалентні, то природно припущення, що $f(O_j) = f(O_i)$. Найбільш часто в практиці експертного оцінювання використовують наступні числові уявлення (таб.2.1)

Таблиця 2.1 - Результати порівняння експертом всіх пар об'єктів

Эмпирична 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 цієї таблиці (таб.2.2) випливає, 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 об'єктів у всіх можливих парах не

дає повного упорядкування всіх об'єктів. Тому виникає задача про ранжування об'єктів на основі парного порівняння.

Таблиця 2.2 - Результати результатів проведеного парного порівняння п'яти об'єктів при використанні числового уявлення (таб.2.1)

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

2.5.4 Метод безпосередньої оцінки

Безпосередня оцінка являє собою процедуру приписування об'єктам числових значень в шкалі інтервалів. Ці значення відповідає ступеня впливу того чи іншого об'єкта на спостережуваний результат.

У процесі виявлення знань експерт повинен поставити у відповідність кожному об'єкту точку на безперервної числової осі, наприклад, на відрізку [0,1]. природно зажадати, щоб еквівалентним по впливів об'єктів приписувалося б одне і теж число.

Вимірювання переваги в шкалі інтервалів можна виконати з високою ступенем довіри тільки при хорошій інформованості експертів про властивості об'єктів і предметної області.

У ряді випадків, з метою ослаблення цих умов, але, природно, за рахунок зменшення точності вимірювання замість безперервної числової осі розглядають велику оцінку, яка використовує 5, 10, 100 - бальні шкали.

Однак безпосередня оцінка не завжди повинна використовувати числові шкали. Наприклад, колір об'єкта неможливо уявити у вигляді будь-якого числового значення, а перехід до значень частот спектра в багатьох випадках скрутний для експерта.

2.6 Формування і оцінки компетентності групи експертів

При формуванні групи експертів на стадії виявлення знань необхідно враховувати такі характеристики експертів як:
компетентність - ступінь кваліфікації експерта в цій галузі знань;

- креативність - здатність вирішувати творчі завдання;
- ставлення до експертизи - негативний або пасивне ставлення, або -зайнятість істотно впливає на якість роботи експерта в групі;
- конформізм - схильність до впливу авторитетів, при якому думка
- авторитету може пригнічувати осіб, що володіють більш високою -компетентністю;
- колективізм і самокритичність.

Розглянемо один з можливих шляхів кількісного опису характеристик експерта, заснований на обчисленні відносних коефіцієнтів компетентності за результатами висловлювання фахівців про склад експертної групи.

Суть методики зводиться до того, що ряду фахівців пропонується висловити думка про обліковому складі експертної групи. Якщо в цьому

списку з'являються особи, які не ввійшли в вихідний список, їм теж пропонується назвати фахівців для участі в експертизі. Після кількох етапів буде отриманий досить повний список кандидатів в групу.

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

При цьому експерт може включати себе або не включати в експертну групу (Тобто x_{ij} = 0 або x_{ij} = 1). По даній таблиці можна обчислити відносні коефіцієнти компетентності, використовуючи алгоритм рішення задач про лідера.

2.7 Характеристика та режими роботи групи експертів

Основними характеристиками групи експертів є достовірність експертизи та витрати на неї. Обидві ці характеристики визначають кількість експертів в групі і їх якість.

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

$$D_i = \frac{Nn_i}{N_i} \qquad (i = 1, 2...m) \tag{14}$$

де - Nn_i число опитувань, коли експерт дав прийнятне практикою рішення, N_i - загальне число випадків участі і-ого експерта в експертизі.

Можна також врахувати внесок кожного експерта в достовірність всієї групи. Відносна достовірність виражається у вигляді:

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

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

Для вирішення проблем з високим рівнем інформаційного потенціалу знань, збільшення кількості експертів в групі призводить, як це випливає з теорії обробки спостережень, до монотонного зростанню достовірності експертизи.

Експериментальні дослідження підтверджують цю залежність (рис.2.2).

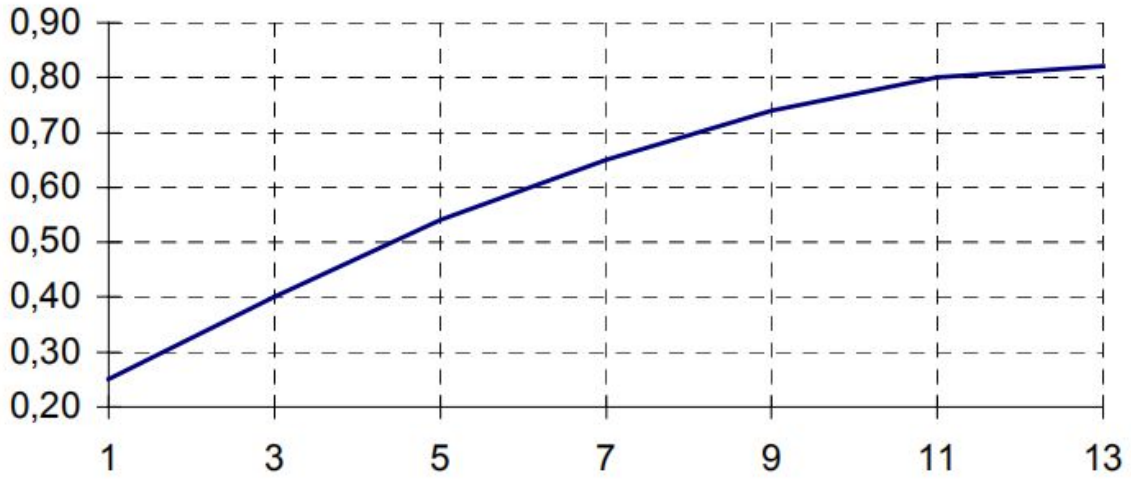

Рисунок 2.2 - Достовірність експертизи від кількості експертів

Опитування - основний етап спільної роботи групи експертів. при колективної експертизі використовуються такі види опитування:

- дискусія;
- анкетування і інтерв'ювання;
- метод колективної генерації ідей;
- мозковий штурм.

Результатом опитування є інформація, що виражає перевагу експертів і змістовне обгрунтування цих переваг. Наявність як числових даних, так і змістовних висловлювань експертів, призводить до необхідності застосування якісних і кількісних методів обробки результатів групового експертного оцінювання.

2.8 Обробка експертних оцінок

Залежно від цілей експертного оцінювання і методу обліку експертних оцінок виникають такі основні завдання:

- побудова узагальненої оцінки понять і об'єктів на основі індивідуальних оцінок експертів;
- побудова узагальненої оцінки на основі парного порівняння об'єктів -кожним з експертів;
- визначення відносних ваг взаємозв'язку об'єктів;
- визначення залежностей між ранжування;
- визначення узгодженості думок експертів;
- оцінка надійності обробки результатів.

При вирішенні багатьох завдань недостатньо впорядкування об'єктів по одному або групі показників. Необхідно мати числові значення для кожного об'єкта, що визначають його перевагу перед іншими об'єктами. Наявність таких оцінок дозволить визначити узагальнену оцінку для всієї групи експертів.

Визначення узгодженості думок експертів проводиться шляхом обчислення числової заходи, що характеризує ступінь близькості індивідуальних думок. Аналіз значення заходи узгодження сприяє

виробленню правильного судження про загальний рівень знань по розв'язуваної проблемі і виявлення угруповань думок експертів.

Обробка експертних оцінок дозволяє розкрити пов'язані показники порівняння і здійснити угруповання за ступенем зв'язку. Так, наприклад, якщо показники порівняння - різні цілі, а об'єкти порівняння - засоби досягнення цих цілей, то встановлення взаємозв'язку між ранжування, упорядочивающими кошти з точки зору досягнення цілей, дозволяє обґрунтовано відповісти на питання: "якою мірою досягнення однієї мети при даних засобах сприяє досягненню інших цілей "(тобто встановити причинно наслідковий зв'язок).

Оцінки, одержані на основі обробки, є випадкові об'єкти, тому однією з найважливіших завдань процедури обробки є визначення їх надійності.

2.9 Байєсова мережі довіри

Байєсівські мережі довіри [13] - Bayesian Belief Network - використовуються в тих областях, які характеризуються успадкованою невизначеністю. Ця невизначеність може виникати внаслідок:

- неповного розуміння предметної області;
- неповних знань;
- коли задача характеризується випадковістю.

Таким чином, байєсівські мережі довіри (БСД) застосовують для моделювання ситуацій, що містять невизначеність в деякому сенсі. Для байесовских мереж довіри іноді використовується ще одна назва причинно-наслідковий мережу, в яких випадкові події з'єднані причинно-наслідковими зв'язками.

З'єднання методом причин і наслідків дозволяють більш просто оцінювати ймовірності подій. У реальному світі оцінювання найбільш часто робиться в напрямку від "спостерігача" до "спостереження", або від "ефекту" до "слідству", яке в загальному випадку більш складно оцінити, ніж напрямок "наслідок -> ефект ", тобто в напрямку від слідстві.

Розглянемо приклад мережі (рис.2.3), в якій ймовірність перебування вершини e в різних станах (e_k) залежить від станів (c_i , d_j) вершин d і c і визначається виразом

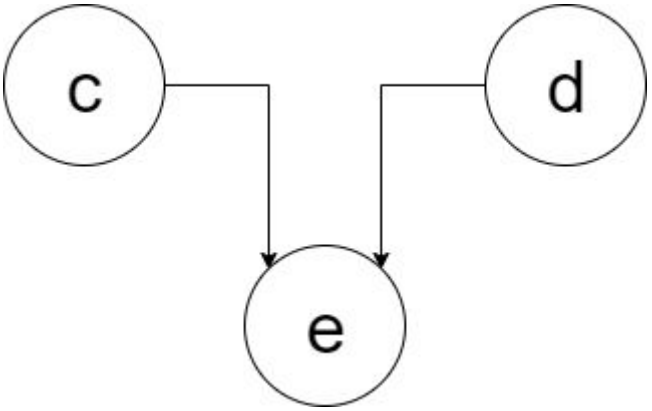

Рисунок 2.3 - Приклад найпростішої байєсівської мережі довіри

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

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

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

Розглянемо приклад більш складної мережі (рис.2.4). даний рисунок ілюструє умовну незалежність подій. Для оцінки вершин c і d використовуються ті самі висловлювання, що й для обчислення $p(e_k)$, тоді:

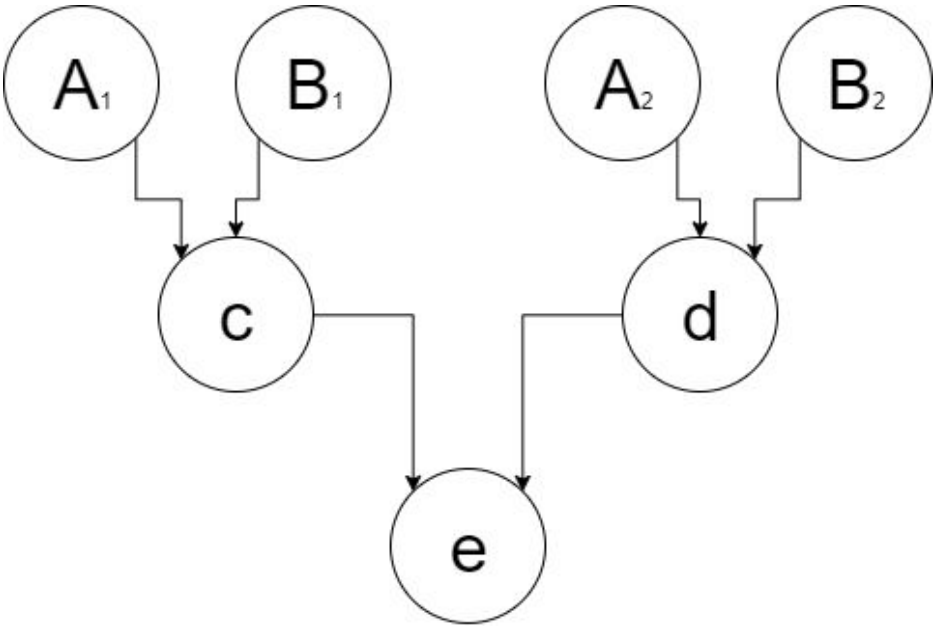

Рисунок 2.4 - Дворівнева БМД.

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

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

З цих виразів видно, що вершина «е» умовно не залежить від вершин A_1 , A_2 , B_1 , B_2 , так як немає стрілок безпосередньо з'єднують ці вершини.

Розглянувши ці приклади спробуємо тепер більш точно визначити основні поняття, що використовуються в БМД. Баєсовські мережі довіри - це спрямований ациклический граф, що володіє наступними властивостями:

- кожна вершина являє собою подія, що описується випадковою величиною, яка може мати кілька станів;
- всі вершини, пов'язані з "батьківськими" визначаються таблицею умовних ймовірностей (ТУЙ) або функцією умовних ймовірностей (ФУЙ);

- для вершин без "батьків" ймовірності її станів є безумовними (Маргінальними).

Іншими словами, в байесовских мережах довіри вершини представляють собою випадкові змінні, а дуги - імовірнісні залежності, які визначаються через таблиці умовних ймовірностей. Таблиця умовних ймовірностей кожної вершини містить ймовірності станів цієї вершини за умови станів її "Батьків".

2.10 Приклад побудови найпростішої байєсівської мережі довіри

У цьому прикладі розглядаємо невелику яблучну плантацію «яблучного Джека ». Одного разу Джек виявив, що його прекрасне яблучне дерево втратило листя. Тепер він хоче з'ясувати, чому це сталося. Він знає, що листя часто опадає, якщо:

- дерево засихає в результаті нестачі вологи;
- або дерево хворіє.

Дана ситуація може бути змодельована байєсівської мережею довіри (рис.2.5) , містить 3 вершини: «Хворіє», «Засохли» і «Опало».

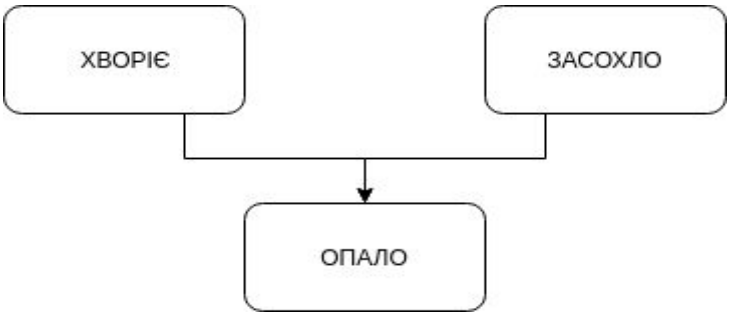

Рисунок 2.5 - Приклад байєсівської мережі довіри з трьома подіями.

В даному найпростішому випадку розглянемо ситуацію, при якій кожна вершина може приймати всього лише два можливих станів і, як наслідок знаходиться в одному з них (таб.2.3):

Таблиця 2.3 - Перелік вершин БМД

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

Вершина "Хворіє" говорить про те, що дерево захворіло, будучи в стані «Хворіє», в іншому випадку вона знаходиться в стані «ні». аналогічно для інших двох вершин. Вже згадана баєсова мережа довіри, моделює той факт, що є причинно-наслідковий залежність від події "Хворіє" до події "Опало" і від події "Засохли" до події "Опало". це відображено стрілками на байєсівської мережі довіри.

Коли є причинно-наслідковий залежність від вершини А до іншої вершини В, то ми очікуємо, що коли А знаходиться в деякому певному стані, це впливає на стан В. Слід бути уважним, коли моделюється залежність в байесовских мережах довіри. Іноді зовсім не очевидно, яке напрямок повинна мати стрільця.

Наприклад, в розглянутому прикладі, ми говоримо, що є залежність від "Хворіє" до "Опало", так як коли дерево хворіє, це може викликати опадання його листя. Опадання листя є наслідком хвороби, а не хвороба - наслідком опадання листя.

На наведеному вище рисунку дано графічне представлення байєсівської мережі довіри. Однак, це тільки якісне уявлення байєсівської мережі довіри. Перед тим, як назвати це повністю байєсівської мережею довіри необхідно визначити кількісне уявлення, тобто множину таблиць умовних ймовірностей (таб.2.4 - таб.2.6):

Таблиця 2.4 - Апріорна ймовірність p("Хворіє")

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

Таблиця 2.5 - Апріорна ймовірність p("Засохло")

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

Таблиця 2.6 - 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 ілюструють ТУЙ для трьох вершин байєсівської мережі довіри. Зауважимо, що всі три таблиці показують ймовірність перебування деякої вершини в певному стані, обумовленим станом її батьківських вершин. Але так як вершини Хворіє і засохли не мають батьківських вершин, то їх ймовірності є маргінальними, тобто не залежить (НЕ обумовлені) ні від чого.

На даному прикладі ми розглянули, що і як описується дуже простий байєсівської мережею довіри. Сучасні програмні засоби (такі як MSBN, Hugin і ін.) Забезпечують інструментарій для побудови таких мереж, а

також можливість використання байесовских мереж довіри для введення нових свідоцтв та отримання рішення (висновку) за рахунок перерахунку нових можливостей у всіх вершинах, відповідних знову введенням свідченнями.

У нашому прикладі нехай відомо, що дерево скинуло листя. це свідоцтво вводиться вибором стану «так» в вершині "Облетіло". 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

2.10 Діаграми впливу

Діаграми впливу використовуються для прийняття рішень. Фактично діаграми впливу - це байєсовські мережі довіри розширені поняттями користі (Utility) і рішення (decisions). Якщо байєсовські мережі довіри містили тільки один тип вершин, які ми назвемо вершинами шансів, і які відповідали станом випадкових змінних, то в діаграмах впливу використовуються ще, як мінімум, два типи вершин (рис.2.6): вершини рішення, що позначаються в діаграмах впливу прямокутниками і вершини користі, що позначаються в діаграмах впливу у вигляді ромба

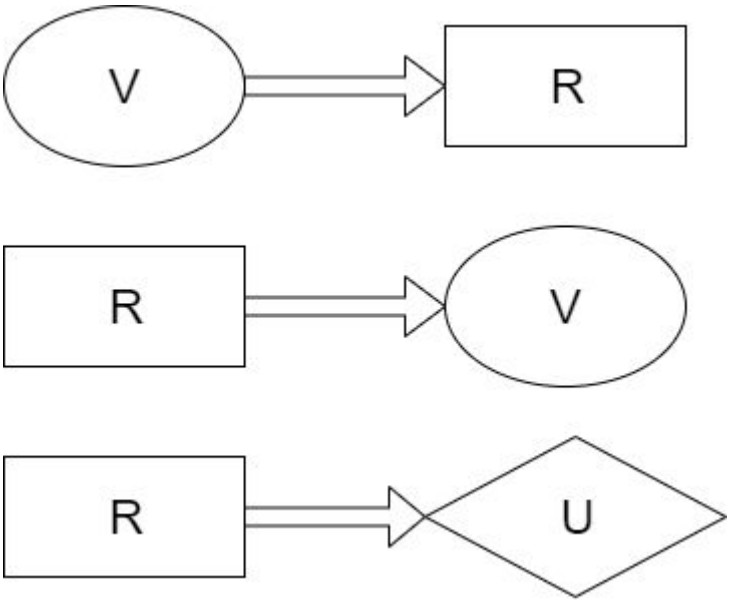

Рисунок 2.6 - Приклад вершин рішень

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

- стрілка від випадкової змінної (вершини шансів) до змінної рішення (Вершині рішення) вказує, що значення випадкової змінної відомо на момент прийняття рішення;
- стрілка від змінної рішення до будь-якої іншої змінної вказує час, впорядковане рішенням.

При цьому мережа повинна залишатися ациклической і повинен існувати безпосередній шлях, що містить всі вершини рішення в мережі.

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

Кожна вершина користі (корисності) містить функцію корисності, яка пов'язує кожну конфігурацію стану її батьків з корисністю. Вершини корисності не мають спадкоємців (а, отже, стрілка може бути спрямована тільки до них) (рис.2.7), тобто

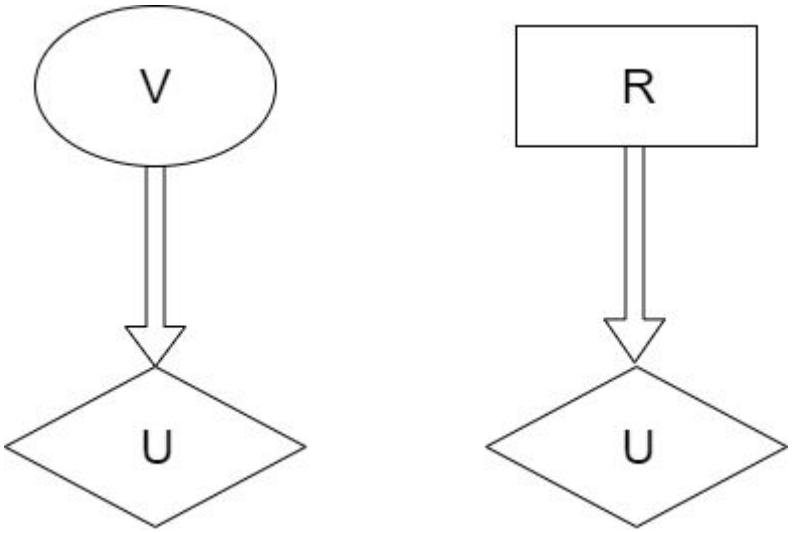

Рисунок 2.7 - Приклад вершин рішень

Приймаючи рішення ми виходимо ймовірності конфігурації мережі. Тому можна обчислити очікувану корисність кожної альтернативи і вибрати альтернативу з найбільшою очікуваною корисністю. Це принцип максимальної очікуваної корисності. Діаграма впливу може містити кілька вершин корисності. При цьому загальна функція корисності являє собою суму всіх локальних функцій корисності.

$$F(x_1, ..., x_n) = \sum^{k} f_i(x_1, ..., x_n) \qquad (20)$$

Процес прийняття рішення з використанням діаграм впливу буде здійснюватися в наступному порядку:

- після спостереження значень змінних, які є батьками першої вершини рішення ми хочемо знати максимальну корисність для альтернатив;
- ЕС вирахує ці корисності в припущенні, що всі майбутні рішення будуть зроблені оптимально, використовуючи всі наявні свідчення в момент кожного рішення.

3 ТЕХНОЛОГІЇ ДЛЯ РОЗРОБКИ СИСТЕМИ МЕДИЧНОГО ЕКСПЕРТУ НА ОСНОВІ ЧАТ-БОТА

3.1 Перспектива продукту

Експертні системи існують дуже давно, але їх застосування для медико-історичних досліджень є нові. Ця система буде розроблена з самого початку, беручи натхнення у різних експертів системи та чати-боти, що були раніше.

Система медичних експертів, що базується на чаті, буде спілкуватися з пацієнтом і відзначати його симптоми. Ці симптоми потім будуть передані лікареві для подальшого аналізу.

Ця система є інструментом для надання допомоги лікарям в збору анамнезу, щоб час, який часто витрачається на це завдання можна було краще використовувати.

3.2 Функції продукту

Функції продукту з точки зору користувачів можна записати так:

- основним інтерфейсом для пацієнтів є текстовий чат.
- користувачі взаємодіють із системою через додаток Telegram, який дозволяє їм вводити текст та надсилати його на веб-сервер.
- є два основні клієнта це пацієнт та лікар. Пацієнти будуть спілкуватися з системою, щоб звітувати їх симптоми захворювання і система реагуватиме так, щоб викликати відповідь та подальше зондування симптомів. Завданням системи буде діагностувати хворобу

пацієнта, а також збір відповідних даних про інші особисті параметри, наприклад вік, стать, ріст і вага.

- іншими користувачами системи будуть лікарі. Вони будуть взаємодіяти із системою електронною поштою. Електронні листи з деталізацією симптомів та можливої діагностикою пацієнтів на основі зібраної інформації буде направлена лікарям, коли пацієнт успішно завершить чат із системою.

- третім користувачем системи є адміністратор. Роль адміністратора полягає в тому, щоб дозволити лікаря на огляд реквізити конкретного пацієнта. Оскільки історія хвороби носить дуже особистий характер, лише адміністратор (який буде довіреною особою компанії розгортає цю послугу) зможе встановити дозволи для лікарів. Адміністратор може взаємодіяти з системою за допомогою командної строки

3.3 Класи та характеристики користувача

В основному це буде 3 класи користувачів:

1) Пацієнти

- первинні користувачі системи. Пацієнти спілкуватимуться із системою через інтерфейс чату.
- вони повідомляють про свої симптоми системі, і система запитає їх про відповідні запитання для з'ясування діагнозу.
- вони будуть спілкуватися із системою та брати участь у чаті для визначення точної хвороби. Пацієнти є найважливішими користувачами системи.

2) Лікарі

- лікарі будуть використовувати систему для доступу до даних про пацієнтів.
- вони зможуть переглядати всіх пацієнтів, до даних яких вони мають доступ, а також переглядати конкретні подробиці про кожного пацієнта.

3) Адміністратор

- адміністратор несе відповідальність за технічне обслуговування та обслуговування системи.
- адміністратор буде єдиною особою, яка зможе встановити дозволи лікарям до даних конкретного пацієнта.

3.4 Обмеження проектування та впровадження

Основна проблема - збереження конфіденційності даних користувачів. Тільки конкретні лікарі повинні мати доступ до доступу дані, пов'язані з симптомами певних пацієнтів, і це теж за їх згодою. Доступ до цих даних буде обмежено головними технічними працівниками системи. Таким чином, дані доведеться зберігати в дуже безпечному режимі, їх не можна поширювати без конкретного юридичного дозволу пацієнтів.

Інтерфейс системи залежатиме від мобільного додатку Telegram. Нам потрібно відповідати стандартам, встановленим Telegram для спілкування в чаті між клієнтом і сервером код сторони сервера буде записаний на Python, а база даних буде Redis. Ми в першу чергу будемо використовувати HTTP отримувати та розміщувати запити на надсилання та отримання даних по мережі.

3.5 Вимоги до зовнішнього інтерфейсу

3.5.1 Інтерфейс між пацієнтом та системою

Цей інтерфейс вже створений Telegram, і ми будемо використовувати той самий для прийняття інформації від пацієнта і передавати їм відповідну інформацію.

У ньому є текстове поле для введення тексту, вікно чату, яке відображає попереднє бесіди, клавіатура для введення відповідей та кнопка "надіслати", на яку надсилається ця інформація система.

3.5.2 Інтерфейс між лікарем та системою

Єдина інформація, яку лікар очікує від системи, - це інформація про його / її пацієнтів. Це буде доставлено їм через їх електронну пошту.

3.5.3 Інтерфейс між адміністратором та системою

Адміністратор піклується про призначення лікарів пацієнтам. Це буде зроблено через інтерфейс командної строки.

3.5.4 Апаратні інтерфейси

Передбачений основний інтерфейс між пацієнтом та системою та лікарем та системою бути смартфоном, який зможе запустити додаток Telegram. Таким чином, це має бути телефон, який може працює на відповідній версії Android, iOS або Windows Phone. Телефон повинен підтримувати підключення до Інтернету і повинен бути сумісний з найновішим HTTP стандарти.

Звичайно, лікарям знадобиться аналогічний комп’ютер, підключений до Інтернету, та електронна пошта.

3.6 Інтерфейси програмного забезпечення

Проект складається з двох основних частин - тієї, що знаходиться на стороні клієнта (інтерфейс чату) та іншої який знаходиться на сервері (чат-бот). Клієнтська частина системи покладається на мобільний додаток Telegram, який повинен встановити двосторонній зв'язок із програмним забезпеченням на сервері для роботи системи належним чином. Існує також база даних, яка бере участь у чаті бота. Більшість вимог програмного інтерфейсу випливає з цієї архітектури. Вони поділяються на 2 категорії, ті, якими користуються клієнт та ті, якими користується сервер

3.6.1 Інтерфейси програмного забезпечення на стороні клієнта

- операційна система Android (v2.2 і новіших версій) або iOS (v6.0 або новішої версії) або WindowsPhone (v8 або v10).
- telegram v3.2 або новішої версії (Android), v1.15 або новішої версії (Windows Phone), > v3.2.1 (iOS).
- мережевий стек HTTP (TCP / IP).
- клієнтська сторона надсилатиме на сервер текстові повідомлення, упаковані в пакети HTTP, і отримує текстові та телеграмові розкладки клавіатури як відповіді.

3.6.2 Інтерфейси програмного забезпечення на стороні сервера

- підключений до Інтернету та готовий для HTTP екземпляр UNIX, незалежно, або в хмарі;
- redis;
- різні бібліотеки пітона, такі як python-telegram-api, багатопроцесорна робота тощо;

- сервер отримує дані у вигляді HTTP-пакетів, які будуть містити JSON, у яких буде розміщено текстове повідомлення та облікові дані користувача.Він надсилає HTTP-пакети, що містять JSON, які містять деталі відповіді з чату бот.

3.7 Особливості системи

3.7.1 Чат-бот

Це серце проекту, що має найвищий пріоритет. Це серцевина проекту, яка буде взаємодіяти з пацієнтом на природній мові. У ньому будуть положення про прийняття тексту від користувача, обробку його накопичення відповідей та надсилання її по мережі.

Компонент високого ризику. Це єдиний метод введення даних для пацієнта; будь-який помилка тут буде відображаються у всій системі.

3.7.2 Стимули послідовності відповідей

Чат пацієнтів із системою:

Стимул: пацієнт надсилає текстове повідомлення з описом своєї недуги.

Відповідь: Чат-бот витягує інформацію користувачів з бази даних в активну пам'ять для подальшої роботи, обробки. Система ініціює чат і спробує діагностувати хворобу пацієнта. Це передбачає читання / запис реплік до бази даних та модулів бізнес-логіки. Як тільки чат досягне стану "закінчено", користувачеві буде надіслано відповідне повідомлення із зазначенням цього.

Стимул: Пацієнт відповідає за допомогою клавіатури Telegram в додатку на запитання, яке йому задає система.

Відповідь: чат-бот записує цю відповідь, і після обробки виконується модуль з бізнес логікою, відповідна відповідь надсилається користувачеві. Це може бути ще одне запитання, яке потрібно підказати пацієнт для отримання додаткової інформації або повідомлення про те, що розмова закінчена.

Взаємодія лікаря-системи:

Стимул: Пацієнт, дані якого дозволено переглядати конкретний лікар, щойно успішно завершив розмову з системою.

Відповідь: Система отримає відповідні реквізити пацієнта про анамнез та поверне його до лікаря в читаному та зрозумілому форматі. Інформація буде надіслана лікареві за допомогою електронної пошти. Відповіді, як правило, стосуються отримання більшої інформації про конкретного пацієнта.

Взаємодія адміністративної системи:

Стимул: Адміністратор хоче дозволити лікаря мати можливість отримувати інформацію про дану інформацію терплячий.

Відповідь: система пов'язує дані пацієнта з пацієнтом та надає доступ лікаря інформація про пацієнта.

3.8 Функціональні вимоги

- серверний модуль. Цей модуль відповідає за отримання та надсилання повідомлень шляхом взаємодії з сервером Telegram. Крім того, він також несе відповідальність за відправлення повідомлень модуля експертної системи та отримання відповідей від них;
- модуль зв'язку бази даних. Це модуль, який повністю відповідає за читання / запис реплік до бази даних. Модуль бази даних

надасть відповідні дані користувача чатовому боту а отримані дані зберігає в базі даних;

- модуль диспетчеризації. Диспетчер діє як маршрутизатор. Він виділяє один об'єкт у незалежну розмову;
- зберігання даних запитань та відповідей. Це статичний сховище даних, який зберігає дані про запитання та відповіді;
- основний модуль. Цей модуль, якому відомо про стан людини, який створюється на основі розмови, усвідомлюючи стан, в якому розмова ведеться в будь-який момент часу;
- експертний системний модуль. Модуль Expert System розташований між базовим і модулем Doctor Sky Net. Цей модуль буде занятий турботою про стан розмов та спілкуванням між Core та Doctor SkyNet;
- модуль Doctor SkyNet. Це основний модуль, який містить різні важливі алгоритми для фактичного функціонування системи;

3.9 Інтерфейс чату

Це друга найважливіша частина проекту, окрім самого бота-чату. Інтерфейс чату буде шлюзом користувачів у систему. Зокрема, ми будемо використовувати для цієї мети мобільний додаток "Telegram". Telegram дозволяє нам змінювати інтерфейс чат-бота, який може надсилати та отримувати повідомлення користувача. Однією з його головних особливостей є те, що він дозволяє демонструвати користувачеві спеціальну клавіатуру, яка може містити лише певні кнопки. Це корисно для обмеження входу користувача до дуже конкретних відповіді та

здійснюють значну контрольну взаємодію між користувачем та чатом-ботом.

3.9.1 Основні функції

- прийняття необроблений текст, введений користувачем;
- перетворення отриманого тексту у JSON та передання його по мережі в чат-бот;
- отримання відповіді на текстове повідомлення від чата-бота та подання його користувачеві;
- отримання макету спеціальної клавіатури від чату-бота та подання її користувачеві;
- прийняття відповідь на цій спеціальній клавіатурі та тексту і передання її чатовому боту.

3.9.2 Ризики

Більшість ризиків, пов'язаних з цією частиною проекту, пов'язані з програмою Telegram. Наприклад, що робити, якщо компанія, що стоїть за Telegram, вирішить заборонити взаємодіяти з чат-ботами? Нам потрібно буде придумати альтернативний спосіб, щоб дозволити інтерфейс чату.

Ще одним ризиком є втрата підключення до Інтернету.

Але жоден з цих ризиків не був би фатальним для проекту, оскільки чат-бот розроблений як агностичний переднього кінця (в даному випадку Telegram). Він може легко адаптуватися до веб-інтерфейсу навіть на замовлення мобільний додаток.

3.9.3 Стимули послідовності відповідей

Пацієнт хоче почати розмову:

Стимул: пацієнт або лікар висловлюють намір розпочати розмову, відкривши Telegram додаток і почніть щось вводити в текстове поле.

Відповідь: Інтерфейс чекає, поки користувач натисне кнопку "надіслати", присутню в додатку. Як тільки це буде зробленим текст, введений разом з його / її обліковими записами користувачів, упакуються в HTTP-пакет і відправляться по мережі до чату-боту.

Бесіда пацієнта вже триває:

Стимул: пацієнт або лікар отримує повідомлення від чат-бота через інтерфейс чат спонукаючи його / її ввести відповідь. Користувач повинен відповісти за допомогою спеціальної клавіатури і може надсилати лише конкретні відповіді.

Відповідь: Інтерфейс чату надсилає цю відповідь у вигляді пакету HTTP на чат-бот, де чат-бот може зробити свою роботу.

3.10 Інші нефункціональні вимоги

3.10.1 Вимоги до продуктивності

Ефективність роботи стосується головним чином користувачів, які можуть отримати відповідь на свій вклад через, розумну кількість часу. Для виходу із системи чат-відповіді встановлено приблизно 5 секунд.

Тут не вважаєть час доставки повідомлення, оскільки це залежить виключно від користувача та швидкість мережі. Таким чином, алгоритми та апаратні засоби, що використовуються в системі, повинні забезпечити відповідь системи протягом 5 секунд, з моменту отримання повідомлення користувача.

3.10.2 Вимоги безпеки

Необхідно підтримувати максимальну безпеку під час обробки та захисту даних, зібраних від пацієнтів. Особливо важливо забезпечити доступ до цих даних таким чином, щоб його могли приймати лише лікарі, які пов'язані з пацієнтом.

Таке занепокоєння головним чином виникає з-за правових та етичних обмежень, які виникають при обробці даних як особистих, так і в 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 інтерфейс використовується додатком Telegram, для цього буде використано унікальне ім’я користувача від Telegram.

3.11 Моделювання проектних процесів

Для реалізації цього проекту буде придатною модель поступового процесу. Тому що ми плануємо почати реалізацію з виявлення простих захворювань та симптомів, а потім згодом наростити види та складність захворювань. Ми очікуємо доставку ітерацій програмного забезпечення за допомогою поліпшення продуктивності або додавання нових особливостей у кожній ітерації.

1) Опитування досліджень та комунікацій: Ця діяльність включає опитування поточних досліджень та існуючі підходи, а також спілкування з керівництвом проектів, вдосконалення ідеї проекту.

Набори завдань

Завдання: Провести дослідження ідеї та вивчити існуючі підходи. Оцініть плюси і мінуси. Збір відповідних журналів / наукових робіт. Поясніть ідею посібникам проекту та зазначте їх коментарі.

Результати: Порівняння всіх існуючих підходів та вибір методології для використання у проекті.

2) Планування: планування проекту, що передбачає підготовку часової шкали, технічні завдання для простору, аналіз доцільності та ризику тощо.

Набори завдань

Завдання: Розділ заходів з впровадження впродовж певного часу проекту, призначення відповідальність членів групи, отримання ресурсів програмування та інструментів, необхідних для реалізувати проект.

Результати: Таблиця ризиків, аналіз техніко-економічного обґрунтування з використанням математики, вибір імітатора, діаграма часової шкали проекту

3) Проектування та моделювання: Система, що розробляється, повинна бути розроблена за допомогою UML-діаграм

Набори завдань

Завдання: розбивка системи для виявлення компонентів, ідентифікації учасників системи, особливостей, які мають бути присутні, та не присутні, взаємодія між компонентами, потоковими даними, ідентифікація класів.

Результати: діаграма архітектури системи, діаграма компонентів, випадки використання, діаграма класів, блок-схема

4) Програмування: Програмування компонентів системи, зв'язування компонентів разом і тестування виконується.

Набори завдань

Завдання: Кодування компонентів відповідними членами групи відповідно до проекту, затвердженого в попередня діяльність, взаємодію компонентів та тестування.

Результати: Основна реалізація програмного забезпечення в першій ітерації, результат тестових особливостей, виявлення помилок і збоїв, якщо такі є.

4 АНАЛІЗ ТА ПРОЕКТУВАННЯ ЧАТ-БОТА

4.1 Діаграма архітектури системи

Діаграма архітектури системи показує модулі та взаємодії всієї системи між ними з першого погляду (рис.4.1). Зауважте, що Користувач ніколи безпосередньо не контактує з лікарем різного роду взаємодії між лікарем та користувачем завжди відбуваються через чат-бот.

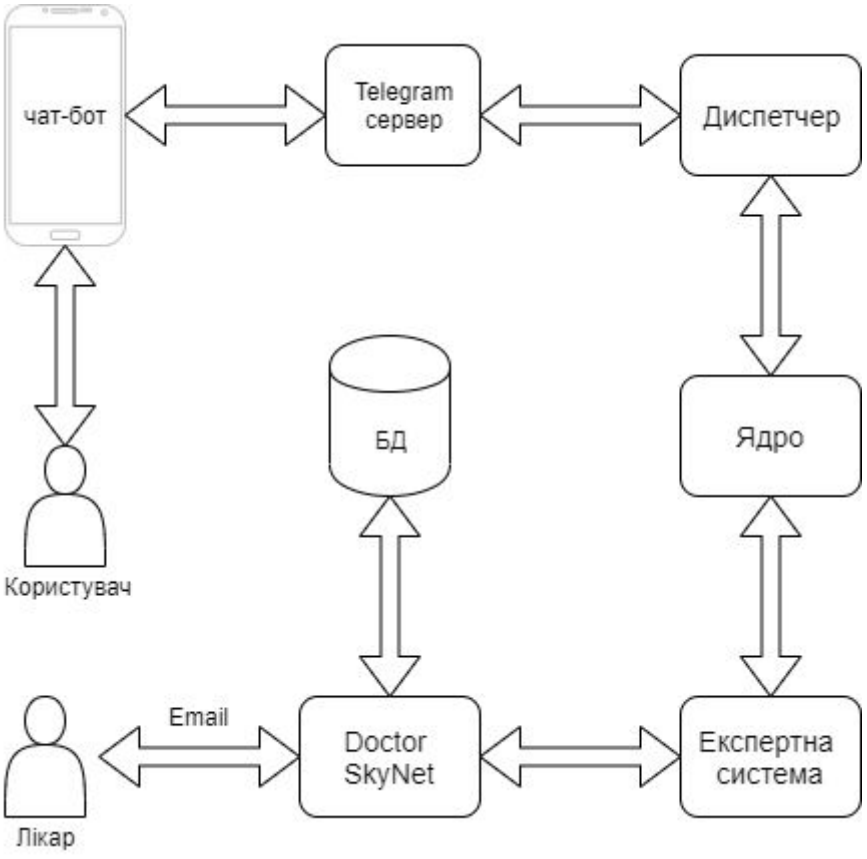

Рисунок 4.1 - Діаграма архітектури системи

Діаграма, що показує архітектуру всієї системи в цілому, разом із взаємодією, яка потребує розміщення між різними модулями. Користувач ініціює розмову з системою, відправивши повідомлення, яке захоплюється

сервером Telegram і потім надсилається через різні модулі до чат-бота. Ініціюється розмова з Користувачем, яка згодом припиняється коли Лікар отримує електронного листа від чату-бота про Користувача

4.2 Діаграми використання

Діаграма "Use Case" показує взаємодію між різними суб'єктами та модулями системи (рис.4.2). В цьому випадку можно побачити найважливіші модулі чат боту та акторів, з якими вони взаємодіють

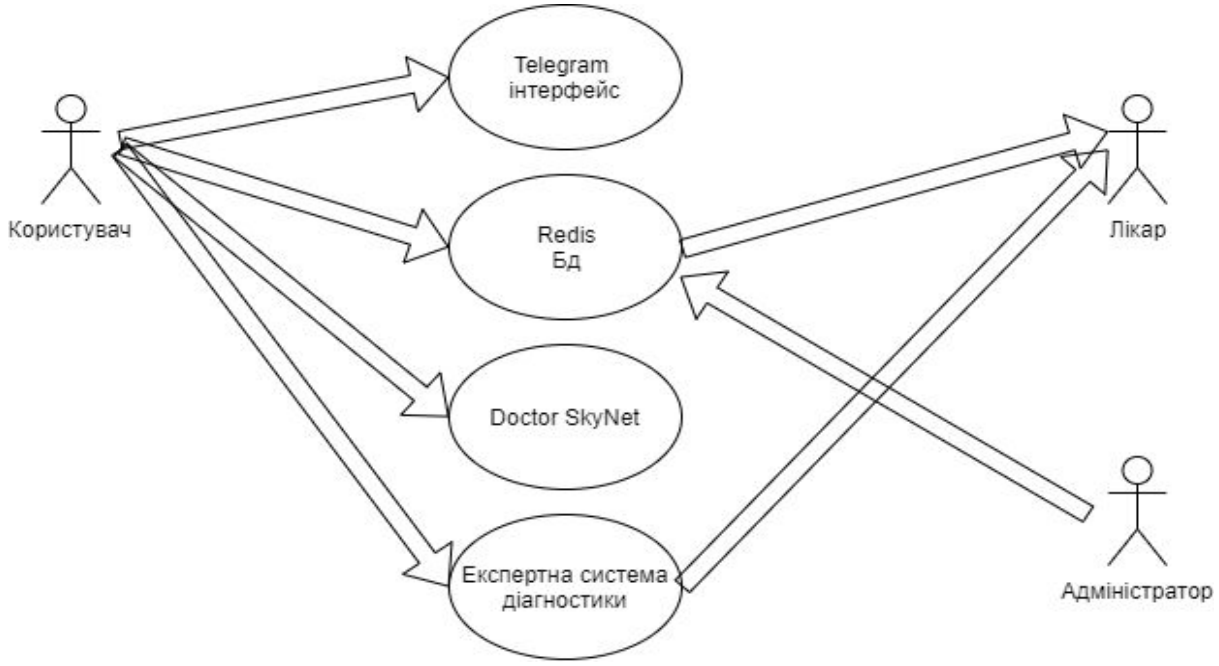

Рисунок 4.2 - Діаграма використання

Основними модулями системи, з якою взаємодіють актори, є інтерфейс Telegram, база даних Redis, Діагностика лікаря SkyNet та експертної системи. Головні дійові особи - користувач, лікар та адміністратор.

Користувач є найважливішим суб'єктом системи, без якого система не функціонувала бот-чат ніколи не буде створювати інстанцію. Пацієнт

взаємодіє з усіма модулями з моменту нормального потоку, розмова робить використання всіх модулів дуже важливим. Роль лікаря стосовно системи обмежується отриманням інформації про пацієнта, яку він може розглянути пізніше відповідно до його / її вимог. Таким чином, лікар отримує інформацію лише з діагностики системи Redis та Expert System модулі. Єдина робота адміністратора - контролювати використання системи та призначати лікарям права доступу, отже, адміністратору потрібно взаємодіяти з базою даних, лише коли йому потрібно змінити будь-яку інформацію.

4.3 Діаграми класів

Класова схема системи, що показує взаємодію між різними класами та потоками даних між ними їх (рис.4.3). Кожен клас містить поведінку та атрибути та асоціації «один до одного» або «один на багато»

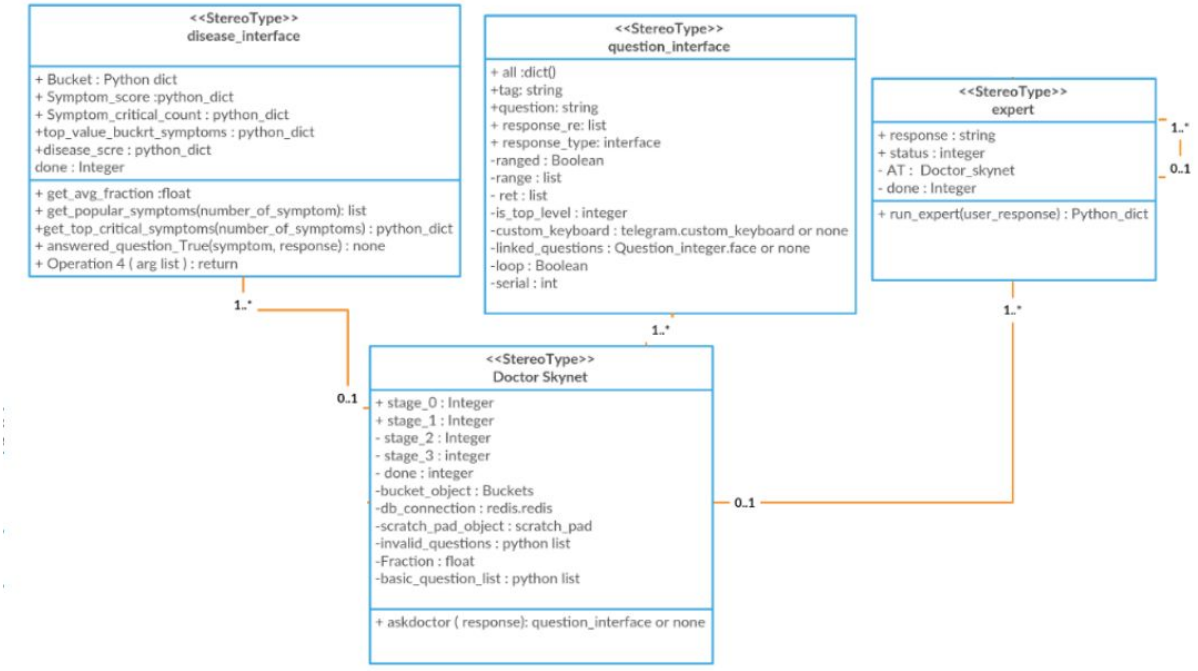

Рисунок 4.3 - Діаграма класів

Діаграма класів складається здебільшого з класів та модулів, які вже були визначені в діаграмі архітектури системи На додаток до них, певні класи, які мають важливе значення для реалізації тут також була визначена функціональність основних класів. Наприклад, `question_interface` та `disease_interface` класи які є важливими для роботи класу DoctorSkyNet, оскільки питання, які слід задати пацієнтові, і форма інтерпретації захворювань, які може виникнути у пацієнта, є важливою частиною визначення наступного питання, яке потрібно задати користувачеві.

4.4 Діаграми стану та переходу

Діаграма переходу стану пацієнта позначає переходи в стані пацієнта, які мають місце як він / вона продовжує вести розмову з системою (рис.4.4). Стани (поля) позначають діяльність, яку пацієнт виконує в певному стані, а стрілки позначають дію, яку повинен вжити пацієнт, щоб досягти цього стану

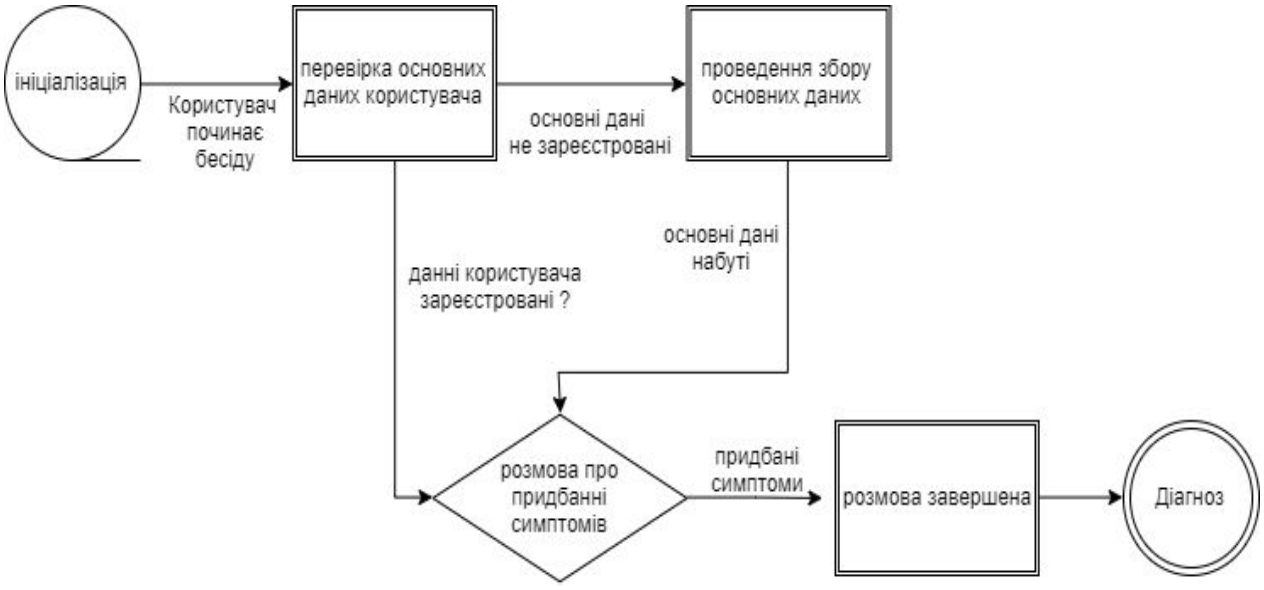

Рисунок 4.4 - Діаграма стану та переходу

Наведена вище схема переходу стану позначає огляд дій, які пацієнт може виконати, та дій, які він повинен вжити, щоб перейти від виконання набору дій до іншого набору дій. Згідно діаграми, зрозуміло, що пацієнт повинен спочатку підтвердити свої основні дані, щоб продовжити розмову з системою, яка дозволить йому / їй повідомити про свої симптоми чат-боту. З діаграми стану ясно, що система спочатку проведе розмову для отримання основних даних від користувача, перш ніж вона навіть дозволить їм розпочати розмову про придбання симптомів. Після того, як симптоми будуть успішно набуті, розмову можна припинити.

4.5 Діаграми розгортання

Діаграма, що показує, як система буде розгорнута в реальній комп'ютерній системі (рис.4.5). Діаграма розгортання допомагає нам точно зрозуміти умови, в яких буде розгорнута вся система, і дозволяє нам бачити, як різні компоненти можуть інтегруватися один з одним у реальному світі.

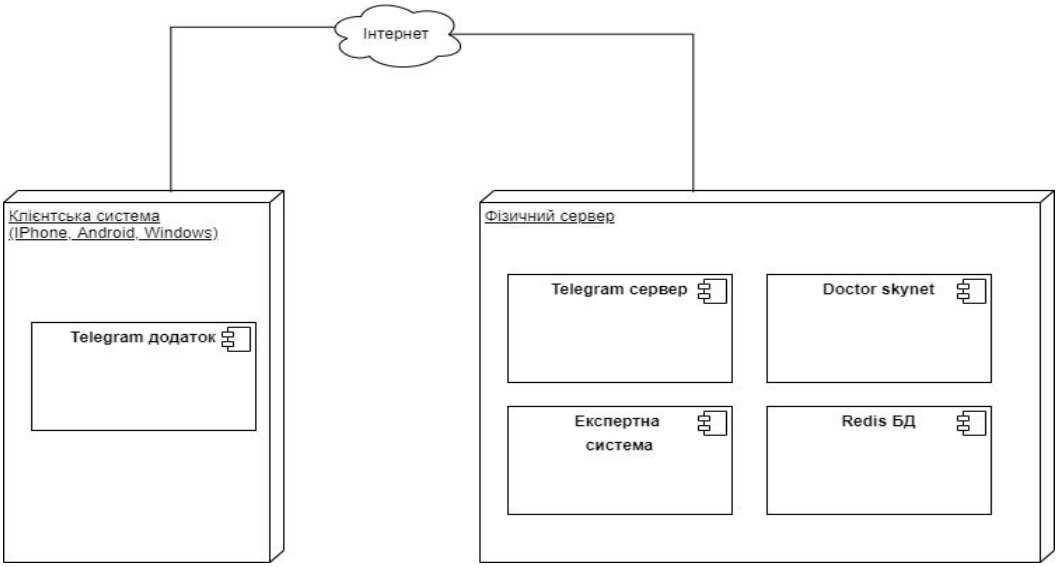

Рисунок 4.5 - Діаграма розгортання

Як видно з діаграми вище, система складається, головним чином, з двох розгортань, системи клієнтів, яка запускає програму чату Telegram, яка може бути будь-яка, від iPhone до ПК з Windows, і фізичного сервера, який фактично запускає чат бот, приймаючи введення користувача через додаток Telegram. Обидві ці області розгортання з'єднані за допомогою Інтернету. Як детально описано в цьому документі, Фізичний сервер може бути будь-яким пристроєм UNIX, а Система клієнтів повинна бути будь-яким середовищем, яке підтримує програму чату Telegram.

4.5 Діаграми послідовності

Діаграма послідовностей показує часові дії користувача щодо модулів чат-бота (рис.4.6). Основні модулі складаються з додатків Telegram, сервера та експертної системи.

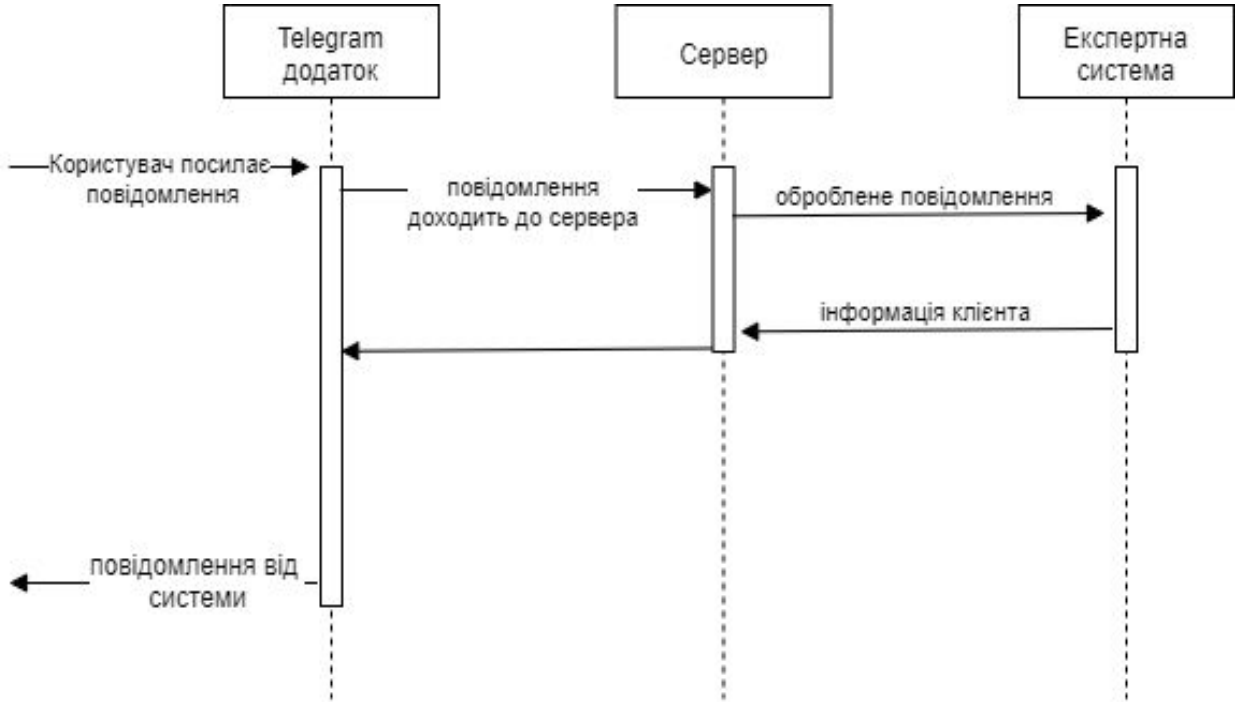

Рисунок 4.6 - Діаграма послідовності

Як видно з наведеної діаграми, пацієнт повинен спочатку взаємодіяти з програмою Telegram, щоб розпочати розмову з системою. Потім додаток Telegram пакує повідомлення у власному форматі та надсилає його на Сервер, який потім надсилає його до Експертної системи. Експертна система несе відповідальність за глибше розуміння повідомлення в контексті інших повідомлень, які раніше були надіслані чатовому боту та відповідно до вже збережених даних користувача, якщо такі є. Він генерує відповідну відповідь на надіслане повідомлення на основі цих параметрів і знову надсилає його через додаток Сервер і Телеграма, щоб воно могло охопити користувача.

4.5 Схема бази даних

Система в цілому покладається на два види даних - кодовані дані та динамічні дані. Вони представлені так:

а) Кодовані дані:

1) Ці дані стосуються симптомів, захворювань або питань. Це надзвичайно важливі дані, які повинні бути у певному форматі. Це первинний сховище знань експертної системи;

2) Вони зберігається у вигляді нативних структур даних Python, таких як словники, списки, рядки та числа - все у виділених файлах python. Причиною збереження жорстко зашифрованих даних і не в жодній зовнішній базі даних є повне зведення нанівець можливості небажаних елементів внесення змін до даних, що може виявитися згубним для всієї системи;

3) Два основних типи жорстко кодованих даних наступні:

- дані захворювання дають інформацію про кожне захворювання та його складові симптоми;

- внутрішньо кожен симптом певного захворювання оцінюється як КРИТИЧНИЙ, ВАЖЛИВО та ОПЦІОНАЛЬНЕ, щоб дати експертній системі уявлення про те найбільш і найменш важливі симптоми конкретного захворювання;

- дані запитань - це питання, яке буде задаватися кожному симптому, щоб підтвердити наявність симптому у пацієнта;

- одне питання відповідає одному симптому. Ці питання поділяються на основні питання та конкретні запитання;

- найважливіші запитання - для підтвердження, якщо пацієнт виявляє ознаки певного симптому, а конкретні запитання - для запитання про конкретні деталі конкретного симптому;

- кожне питання позначається назвою симптому. Мітки відіграють важливу роль у визначенні ієрархії симптомів, коли вони завантажуються в різні структури даних експертної системи, такі як `symptom_validity_table` та `scratch_pad`.

б) Динамічні дані:

1) Динамічні дані включають дані, які надаються системі реальними користувачами людини в результаті розмов, які вони мають із системою. Ці дані зберігаються в базі даних Redis як пари ключових значень на основі кожного користувача.

4.6 Сервер

Серверний модуль є основним шлюзом, в якому система може спілкуватися із зовнішнім світом. Це в основному асинхронна, не

блокуюча, багатопотокова програма багатопроцесорного пітона, що опитує сервер чату Telegram для вхідних повідомлень передає їх до основної логіки фактичної обробки та повертає результат користувачеві.

Використовується бібліотека python-telegram-api як корисна обгортка Python, яка робить спілкування з Telegram дуже простим. Це дозволяє легко встановлювати гачки методу, які спрацьовують кожного разу а конкретна дія відбувається в частині чату-бота Telegram. Він також дозволяє нашій програмі опитувати сервер Telegram кожні 0,1 секунди методом start_polling (). Наприклад, для виклику методу accept_message щоразу, коли повідомлення, яке є "обробником команд Telegram", запитуйте сервер Telegram

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

Лістінг 4.1. - Обробник команд Telegram

Метод accept_message внутрішньо використовує спільну чергу (екземпляр multiprocessing.QUEUE), яка поділяється між усіма трьома процесами, якими користується сервер. Він просто передає відповідні дані з отриманих повідомлень всередині черги та виходить, щоб інші вхідні повідомлення не були заблоковані внаслідок розбиття / відставання в мережі, яке зазнає будь-яке повідомлення. Всередині метод `accept_message` виглядає приблизно так:

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

Лістінг 4.2. - Загальна черга "MESSAGE_QUEUE"

Черга постійно запитується на очікувані повідомлення окремим процесом, який надсилає їх для обробки в експертну систему. Це робиться, попередньо надсилаючи його до модуля «диспетчер». Код, який це робить, можна коротко 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)
```

Лістінг 4.3.- Відправка повідомлення до диспетчера

Як ви бачите, як тільки процес виявить, що повідомлення присутнє у черзі, воно негайно надсилається диспетчеру для обробки, а повернене повідомлення додається до черги відправлення, яка є чергою, спільною для іншого процесу, Єдиною завданням є надсилання повідомлень користувачеві. Таким чином, обробка повідомлень відбувається незалежно від мережевих лагів, які можуть виникнути під час надсилання повідомлень користувачам.

Третій процес - це багатопотоковий процес, єдиним завданням якого є опитування спільної черги відправки на повідомлення та надсилання їх зацікавленому користувачеві Telegram. Кожне повідомлення у черзі надсилається через інший потік, так що мережа, досвідчена для одного повідомлення, не впливає на повідомлення, наявні в черзі.

4.7 Диспетчер

Диспетчер - це модуль, який сидить прямо за сервером, щоб приймати та обробляти повідомлення від користувачів. Сервер створює

один екземпляр диспетчера, а повідомлення надсилаються цьому екземпляру за допомогою виклику методу `depecher.run_dispatcher ()`.

Внутрішньо диспетчер підтримує словник Python поточно активних розмов, які він має. Цей словник називається `object_list`. Ключовими словами цього диктату є ідентифікатори чату користувачів, які є унікальні для кожного користувача, а значення - це об'єкти ядра типу. Основним класом є той, який насправді відповідає за обробку повідомлень.

Щоразу, коли диспетчер отримує нове повідомлення, він здійснює пошук у `object_list`, якщо ідентифікатор чату вхідного повідомлення вже присутній. Якщо так, це тлумачиться як отримання відповіді на повідомлення, яке було попередньо надсилається користувачеві, і повідомлення безпосередньо надсилається в основний об'єкт, з яким асоціюється ідентифікатор чату. Якщо ні, то новий запис створюється в словнику `object_list`, пов'язуючи ідентифікатор чату повідомлення з нещодавно виготовленим основним об'єктом.

Як ми побачимо незабаром, основний клас несе відповідальність за відстеження бесіди, яку він робить, зберігаючи деякі змінні, що визначають поточний стан розмови. Краса такого підходу полягає в що стає дуже просто відстежувати розмову, просто зберігаючи основні об'єкти в пам'яті, поки розмова не закінчиться.

Коли розмова закінчена, основний об'єкт, пов’язаний з ідентифікатором чату, видаляється з диктату, і, таким чином, врешті-решт видаляється з пам'яті колектором сміття Python.

Приблизно, найважливіша функція диспетчера, як узагальнено вище, може бути представлена 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()
```

Лістінг 4.4. - Узагальнений функціонал диспетчера

4.7 Ядро

Основний модуль викликає диспетчер. Один екземпляр основного класу буде існувати для кожної незалежної бесіди, яка ведеться. Ядро містить змінні та методи, які дозволяють йому відстежувати розмову з точки зору верхнього рівня, тобто він може знати, лише якщо розмова ведеться чи закінчена. Проміжні стани не стосуються ядра.

Він інстанціює експертну систему та управляє надсиланням та отриманням даних з неї, які в кінцевому підсумку переходить назад до диспетчерського класу для відправки назад користувачеві. Вся попередня обробка вхідних повідомлень, таких як перевірка орфографії, видалення пунктуації тощо, проводиться ядром до того, як повідомлення навіть буде передане експертній системі. Він повертає повідомлення у формі диктанту Python, і якщо розмова закінчена, повертається "None".

Основна логіка, яка змушує це статися, може бути зведена як:

```
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
```

Лістінг 4.5. - Узагальнений функціонал диспетчера

4.8 Дані запитань та відповідей

Щоб експертна система працювала, як очікувалося, важливо мати сховище знань. Знання для нашого чату-бота представлені файлами python, що зберігаються в даних / папці. Кожне запитання має певні атрибути, які розповідають більше інформації про це питання, як відповідь, яка повинна з’являтися на спеціальній клавіатурі Telegram після того, як питання буде представлено користувачеві. Різні атрибути запитання можуть бути:

- рядок запитань - це власне питання, яке буде надіслано користувачеві.
- відповідь - це перелік відповідей, який буде показаний користувачеві, що позначає відповіді, які він / вона може надіслати в систему за допомогою спеціальної клавіатури Telegram.
- послідовне - це внутрішнє ціле число, яке позначає порядок, в якому слід задавати питання.

Є два типи питань, які можна задати користувачеві:

а) Основні запитання щодо даних:

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

2) Це питання, які необхідно задати, коли пацієнт вперше розпочинає розмову із системою;

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

б) Питання, пов'язані з симптомами:

1) Питання, пов'язані з симптомами, - це питання, які будуть задані пацієнтам для підтвердження того, проявляють вони ознаки певного симптому чи ні;

2) Вони здатні зібрати або так / ні стильові відповіді, або конкретні відповіді на основі попередньо визначених параметрів;

3) Ці питання далі поділяються на два типи:

- основні питання - Основне питання - це питання, яке підтверджує наявність певного симптому чи ні;
- пов’язані запитання - це запитання, які використовуються для запиту додаткової інформації про певний симптом, якщо пацієнт відповідає позитивно на головне питання цього симптому;

4) Оскільки пов'язані питання стосуються збору більшої інформації про певний симптом, вони розміщуються в окремих файлах, кожен з яких названий тим симптомом, про який йдеться у файлі. Тому питання, які отримують більше інформації про біль у тілі, розміщуються у файлі під назвою `body_pain.py`.

4.9 Експертна система

Це модуль, який містить фактичну експертну систему для виведення змісту та контексту відповідей. Вирішення наступного питання відбувається в модулі DoctorSkyNet. Експертна система визначає стан розмови, відстежуючи її зі змінною стану. Більша частина основної логіки експертної системи лежить всередині методу `run_expert()`.

Якщо статус експертної системи дорівнює 1, це означає, що експертна система перебуває у стані, коли перше повідомлення від користувача щойно надійшло і він / вона фактично не відповів на жодне запитання, розміщене системою. Цей стан необхідний, оскільки він визначає, чи буде відповідь надіслана модулю DoctorSkyNet. Після

надання першої відповіді користувачеві експертна система переходить у стан 2.

Перебуваючи у стані 2, експертна система очікує від користувача відповіді, яку вона надішле до DoctorSkyNet для подальшої обробки. Якщо DoctorSkyNet повертає None, експертна система припускає, що розмова закінчена і переходить до 3, що означає, що користувачеві слід надіслати повідомлення про те, що тест закінчений.

Загальне функціонування експертної системи можна позначити 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()
```

Лістінг 4.6. - Загальне функціонал експертної системи

4.10 Лікар SkyNet

У цьому модулі відбувається справжня магія. Його основна мета - використовувати інтелектуальні алгоритми для зчитування відповіді від користувача, відстеження симптомів, про які користувач вже відповів, та подання запитання користувачеві, які є найбільш актуальними для нього на основі бази даних захворювань що підтримується.

Кожна хвороба представлена як "корзина" (використовуючи модуль "Корзини", як ми побачимо далі). Кожна корзина пов’язане з певними симптомами. Алгоритми, які використовує DoctorSkyNet, читають стан цих корзин і подають користувачеві найбільш відповідне питання. Це допомагає нам звузити кількість питань, які необхідно задати системі, щоб поставити можливий діагноз.

Алгоритми, які використовує DoctorSkyNet, можна узагальнити так:

а) Алгоритм мінус один

1) Роль: Кластерний ідентифікатор симптомів;

2) Виявляє симптом, від якого страждають BCI пацієнти;

3) Використовує Redis;

4)Не взаємодіє з корзинами;

4) Кроки:

- отримайте весь симптом: «Так» від Redis через ORM;
- знайдіть симптом з найбільшим числом Так;
- поверніть цей симптом.

б) Алгоритм нульовий

1) Роль: Найбільш вірогідний пошук симптомів, заснований на анамнезі пацієнта;

2) Виявляє симптом;

3) Використовує Redis;

4) Не взаємодіє з корзинами;

5) Кроки:

- отримайте історію симптомів пацієнта, з Redis через ORM;
- знайдіть симптом з найбільшим числом “Так”;
- поверніть цей симптом.

в) Алгоритм один

1) Роль: Знайдіть симптом з найвищим балом у корзинах;

2) Знаходить симптом, який заповнить найбільшу кількість корзин;

3) Не взаємодіє безпосередньо з корзинами;

4) Кроки:

- отримайте бальну оцінку симптомів;
- знайдіть симптом з найвищим балом;
- поверніть цей симптом;
- оновіть корзини, видаліть симптом з усіх балів

г) Алгоритм другий

1) Роль: Знайдіть критичний симптом із найвищим балом у корзинах;

2) Знаходить симптом, який видалить найбільшу кількість корзин, якщо відповісти негативно;

3) Не взаємодіє з корзиною безпосередньо;

4) Кроки:

- отримайте критичну оцінку симптомів;
- знайдіть симптом з найвищим балом;
- поверніть цей симптом;
- оновіть корзини, видаліть симптом з усіх балів.

ґ) Алгоритм третій

1) Роль: Знайдіть симптом з найвищим сукупним рештою балів у кожній корзині;

3) Знаходить симптом, який видалить найбільшу кількість корзини, якщо відповісти негативно;

4) Кроки:

- отримайте корзину;
- знайдіть симптом з найвищим балом у корзині;
- запишіть цей симптом, оцініть у структурі даних;
- якщо корзина залишилися почати заново;
- знайдіть симптом з найвищим балом у структурі даних;

- поверніть цей симптом.

Загальне функціонування модуля контролюється на 2 етапи. Кожен етап позначає, в якій частині розмови зараз знаходиться система. На першому етапі проводиться первинна перевірка, чи були отримані основні дані пацієнта. Це робиться шляхом перевірки основних даних у базі даних redis. Якщо ні, то він ініціює розмову з пацієнтом, щоб отримати більше уявлень про їх основні особисті дані, такі як вік, стать, вага та ріст.

Після того, як основні дані встановлені, DoctorSkyNet переходить до другого етапу і переходить до запиту пацієнта щодо симптомів на основі вищезазначених алгоритмів. Він встановлює змінну "зроблено" на 1, коли система отримає достатньо даних про симптоми. Загальну роботу DoctorSkyNet можна підсумувати наступним чином:

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

Лістінг 4.7. - Загальне вигляд системи DoctorSkyNet

Клас інтерфейсу питань зберігає деталі кожного питання та робить його швидко доступним для будь-якого іншого класу, який потребує доступу до питань у відповідному форматі. Він має ті самі атрибути, що і будь-який словник запитань у сховищі даних

4.11 Основні знімки графічного інтерфейсу

Наведені нище рисунки (рис.4.7 - рис.4.10) - це знімки програми Telegram iPhone на різних етапах оцінки. Вони намагаються показати повну розмову, яку може мати користувач із системою. Зауважте зміни клавіатур та відповідей у кожному знімку, він повинен дати вам хороше уявлення про те, що таке власне враження.

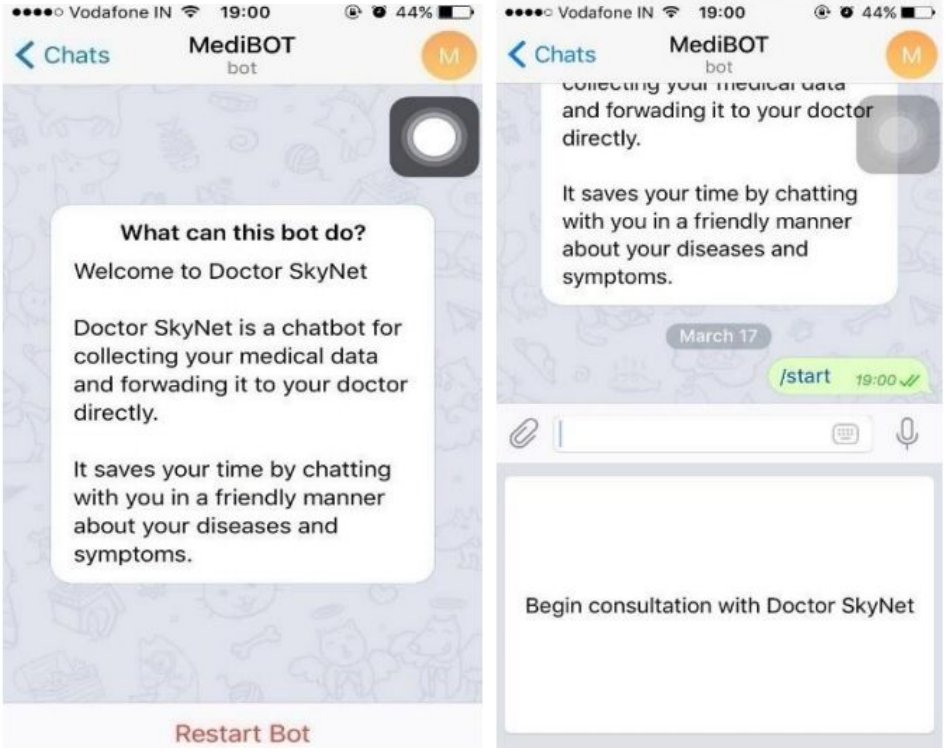

Рисунок 4.7 - Привітання та початок роботи бота

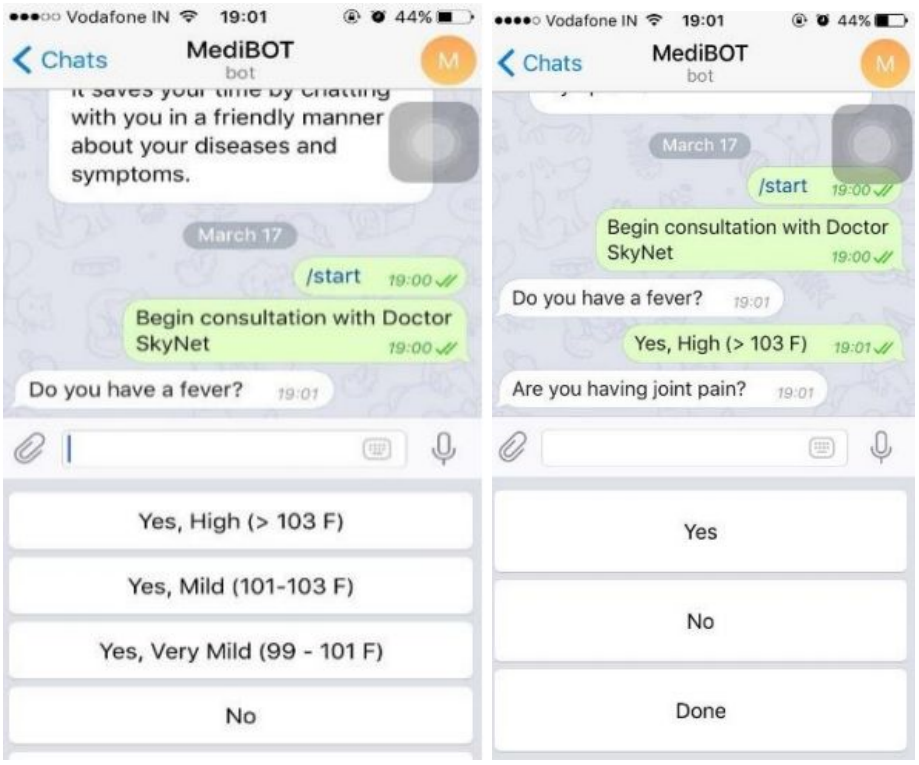

Рисунок 4.8 - Початок розмови з ботом

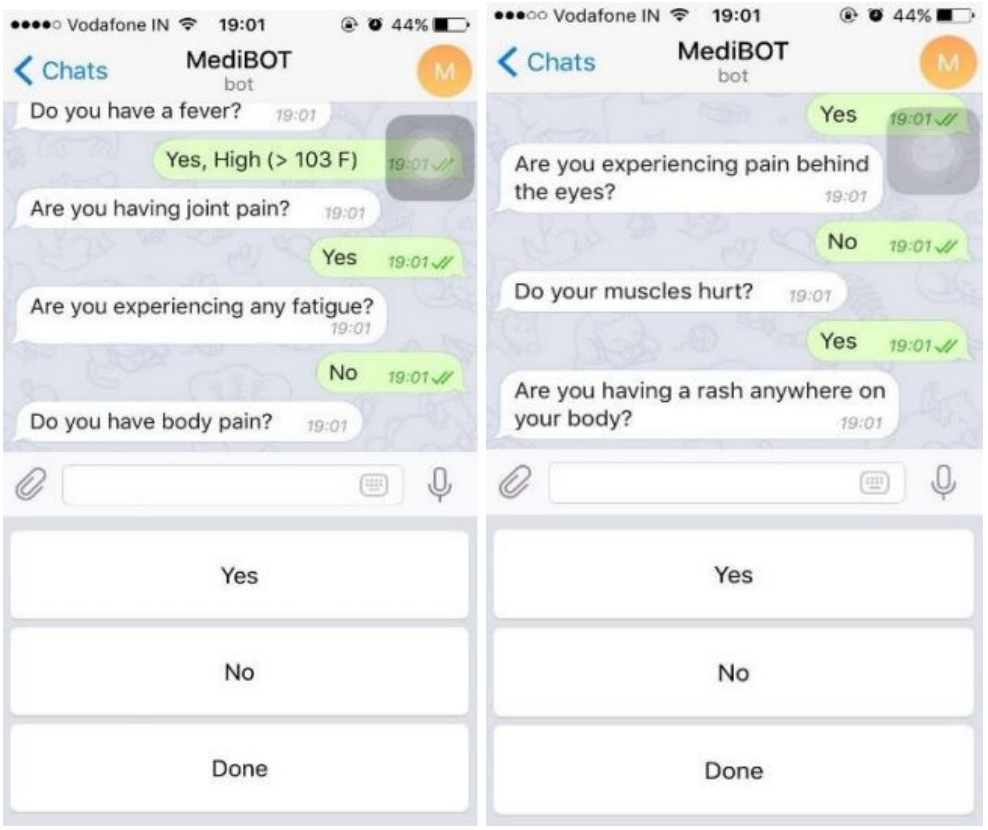

Рисунок 4.9 - Уточнення симптомів

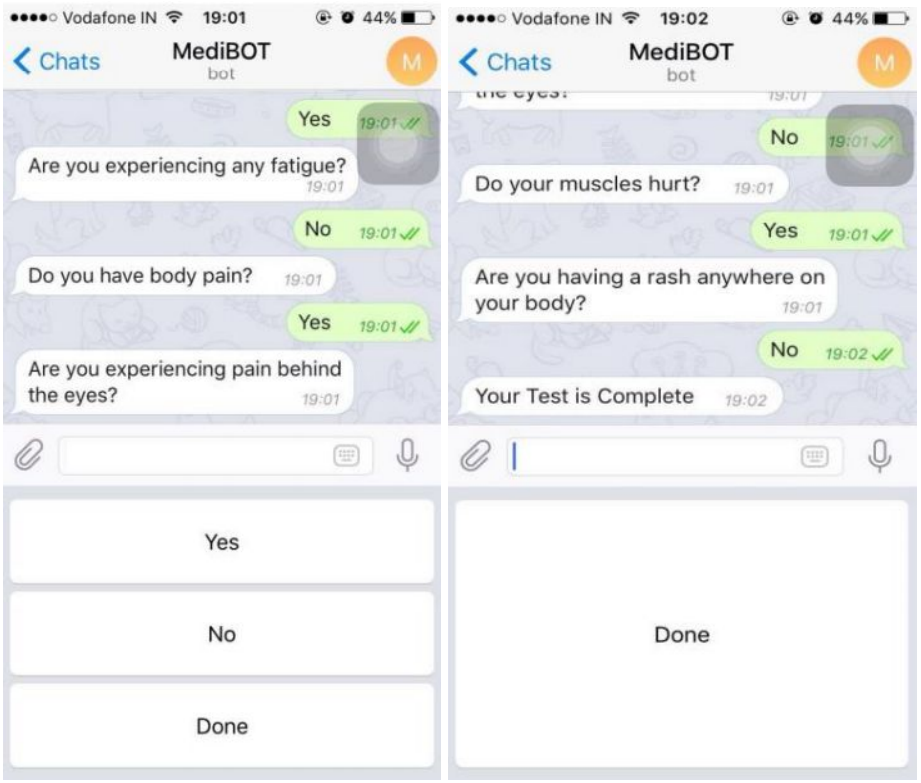

Рисунок 4.10 - Закінчення збору інформації у пацієнта

4.12 Знімки з терміналу

На знімку терміналу нижче ви можете побачити вихід, який відображається сервером, коли він отримує повідомлення від телеграми від користувача. Червона лінія вказує, що об’єкт користувача вже не був у пам'яті, і він створюється, це означає, що почалася нова розмова з користувачем, а отримане повідомлення не надсилається як відповідь на попереднє повідомлення відповіді. Реєстратор не створює нових об'єктів, коли від користувача отримується інша відповідь, яка показує, що той самий об’єкт зберігає стан користувача, поки триває розмова.

На рисунках детально показана така поведінка. Ви можете бачити, як розмова починається з (рис. 4.7), потім просувається в (рис. 4.8), а потім закінчується (рис. 4.9).

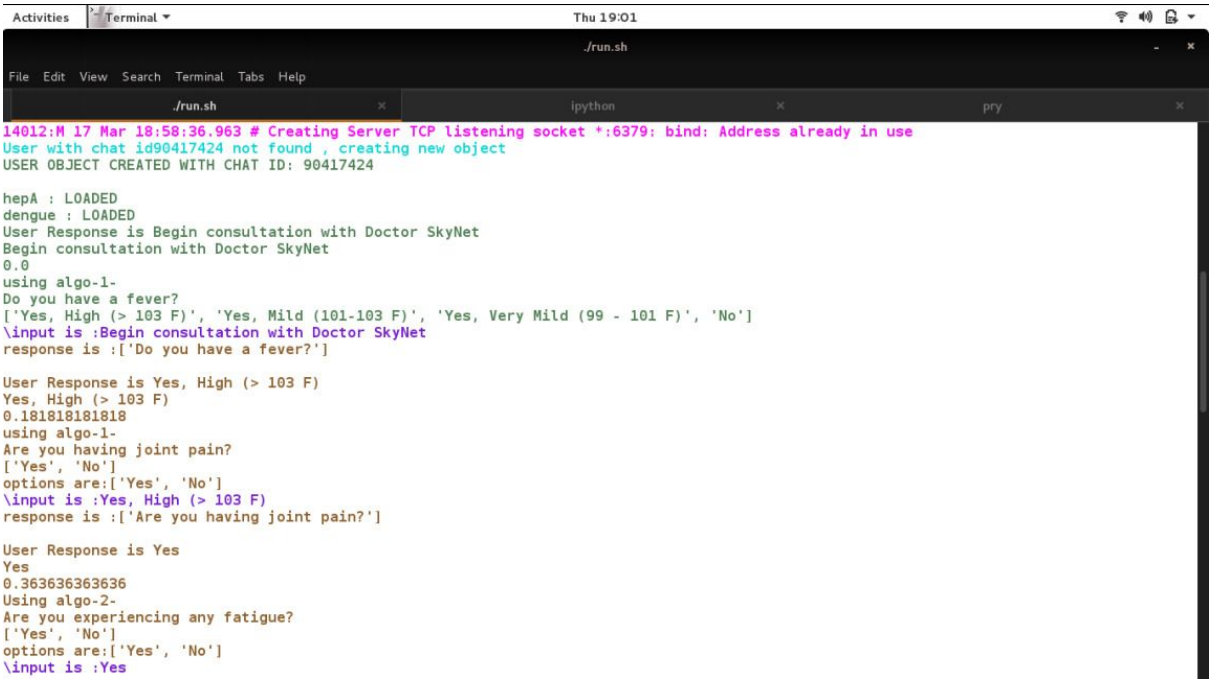

Рисунок 4.7 - Початок розмови з пацієнтом

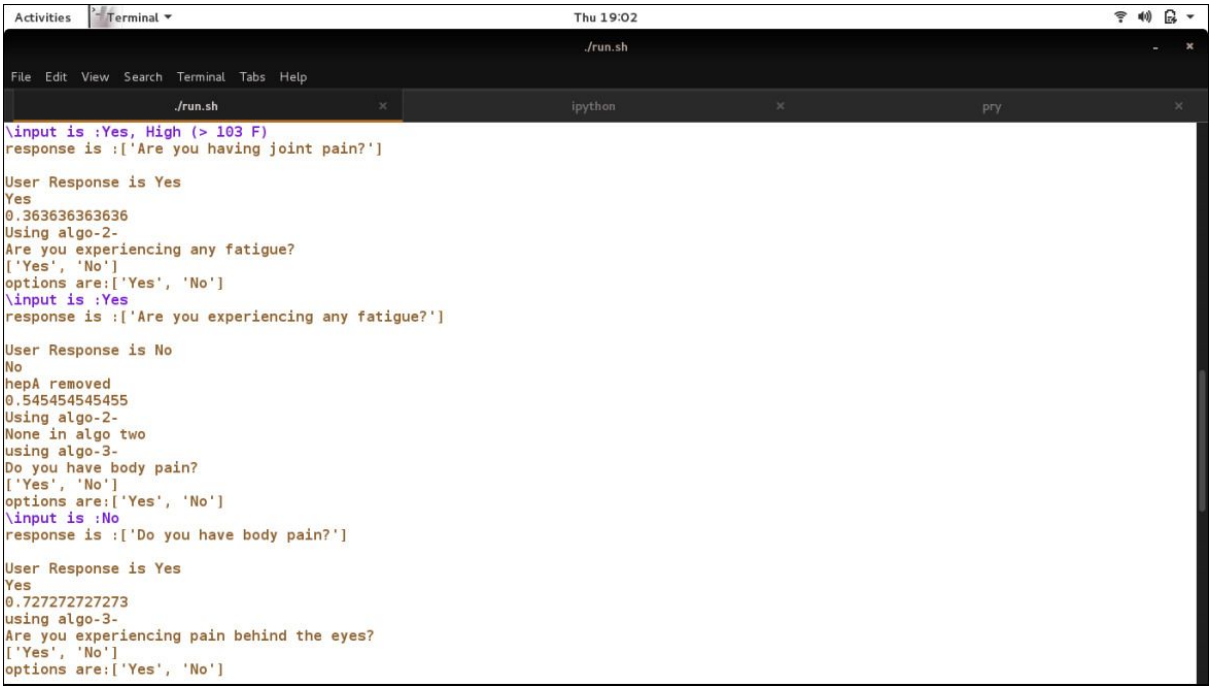

Рисунок 4.8 - Проміжний етап розмови з пацієнтом

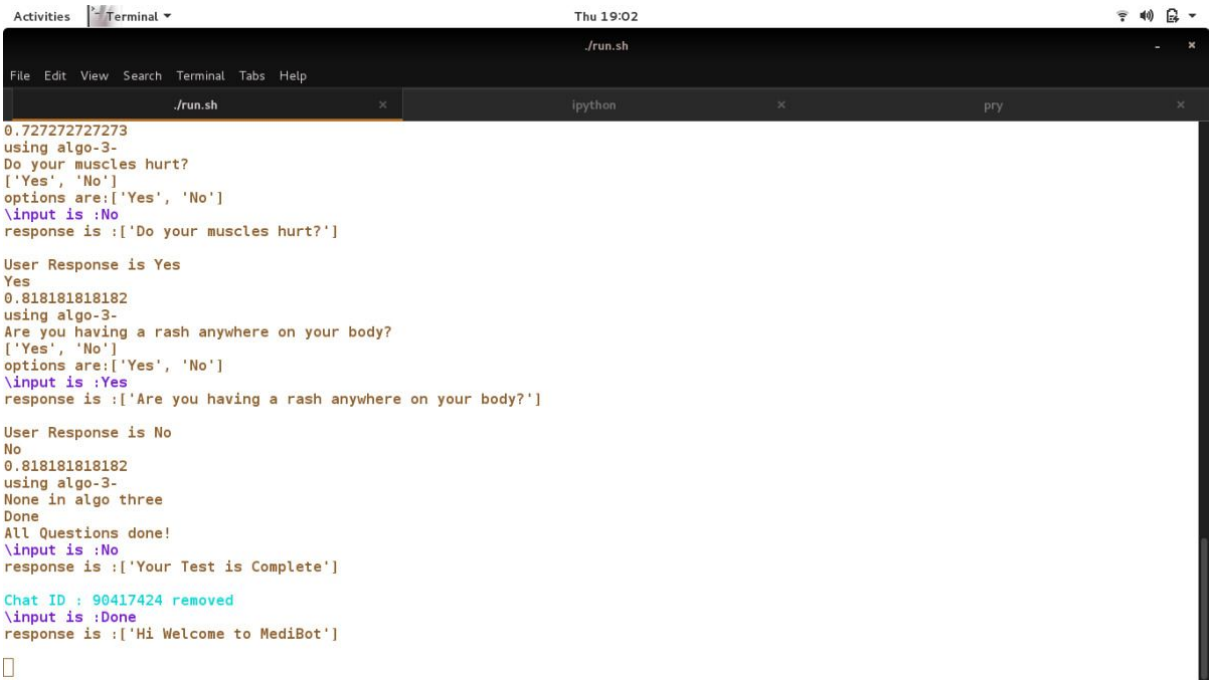

Рисунок 4.9 - Кінець розмови з пацієнтом

На рисунку нижче (рис.4.10) показан дамп бази даних Redis для розмови, який був показаний на (рис.4.7 - рис.4.9). Він показує значення, які зберігалися в базі даних відповідно до ідентифікатора користувача.

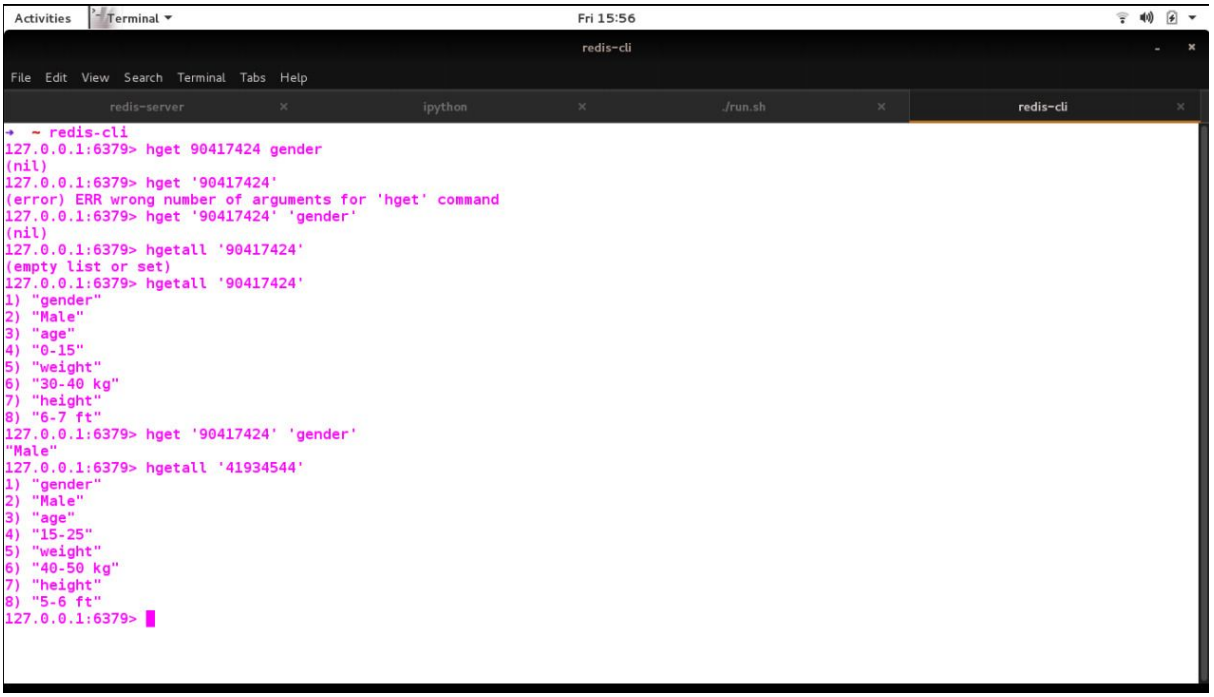

Рисунок 4.10- Дамп бази Redis

ВИСНОВКИ

Вивчивши різні експертні системи, чат-боти та отримавши відгуки від людей, які займаються медичною професією, можна зробити висновок, що:

- медична експертна 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 експертних систем у медичній галузі свідчать про попит на продукцію. Наростаючій 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. Халлілі, Амін. "Назустріч онтологічному чатові, наділеному обробкою та породженням природної мови". 26-а Європейська літня школа з логіки, мови та інформації. 2014 рік.

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 експертних 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. IEEE, 1988.

6. Уотерман, Д. Будування експертних 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. 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 експертну систему / К. Нейлор. - М .: Энергоатомиздат, 2013. - 286 с.

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

13. Джарратано Д. Експертні системи: 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 експертні системи [Текст]: пер. з англ. /П.Джексон. - 617-622. - ISBN 5-8459-0150-2.