

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ

В.о. завідувача кафедри інтелектуальних
інформаційних систем, канд. техн. наук, доцент

_____ Є. В. Сіденко

« ____ » _____ 2023р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

СИСТЕМА ПЕРВИННОЇ МЕДИЧНОЇ ДІАГНОСТИКИ НА ОСНОВІ НЕЧІТКОЇ ЛОГІКИ

Спеціальність 124 «Системний аналіз»

124 – МКР – 607.21710110

Виконав студент 6-го курсу, групи 607

_____ *Е. В. Ергарт*

«17» лютого 2023 р.

Керівник: канд. техн. наук, доцент

_____ *О. В. Козлов*

«17» лютого 2023 р.

Миколаїв – 2023

Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Освітньо-кваліфікаційний рівень **магістр**

Галузь знань **12 «Інформаційні технології»**

(шифр і назва)

Спеціальність **124 «Системний аналіз»**

(шифр і назва)

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри інтелектуальних
інформаційних систем, канд. техн. наук, доцент

_____ Є. В. Сіденко

« _____ » **2022 р.**

ЗАВДАННЯ

на магістерську кваліфікаційну роботу

Ергарту Едуарду Володимировичу

(прізвище, ім'я, по батькові)

1. Тема магістерської кваліфікаційної роботи «Система первинної медичної діагностики на основі нечіткої логіки».

Керівник роботи Козлов Олексій Валерійович, к.т.н., доцент.

Затв. наказом Ректора ЧНУ ім. Петра Могили від «03» листопада 2022 р. № 197

2. Строк подання студентом роботи 17 лютого 2023 р.

3. Вхідні (початкові) дані до роботи: симптоми користувача які являються основою первинної діагностики пацієнта; пріоритетність критеріїв. Очікуваний результат: система оцінювання та визначення найбільш точного діагнозу пацієнта на основі первинної діагностики, що базується на методах нечіткої логіки.

4. Зміст пояснювальної записки (перелік питань, які потрібно розглянути):

– аналіз сучасних систем первинної медичної діагностики на основі на нечіткої логіки;

– огляд існуючих методів реалізації систем діагностики на основі нечіткої логіки;

– розробка експертної системи для визначення діагнозу пацієнта на основі нечіткої логіки;

– порівняльний аналіз підходів та методів, реалізація системи первинної діагностики.

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: Захист від іонізуючих випромінювань.

7. Консультанти:

| Розділ | Прізвище, ініціали та посада консультанта | Підпис |
|------------------------------------|---|--------|
| Спеціальна частина з охорони праці | <u>докт. біол. наук, проф. Григор'єва Л. І.</u> | |
| Методична частина | <u>к.т.н., доцент. Козлов О. В.</u> | |

Керівник роботи к.т.н., доцент. Козлов О. В.
(наук. ступінь, вчене звання, прізвище та ініціали)

(підпис)

Завдання прийнято до виконання Ергарт Е. В.
(прізвище та ініціали)

(підпис)

Дата видачі завдання « 06 » листопада 2022 р.

КАЛЕНДАРНИЙ ПЛАН

Виконання магістерської кваліфікаційної роботи

Тема: Система первинної медичної діагностики на основі нечіткої логіки

| № | Найменування роботи | Початок | Закінчення | Примітки |
|----|---|------------|------------|----------|
| 1 | Визначення керівника і теми МКР. Подання заяви на затвердження теми МКР | 01.09.2022 | 20.10.2022 | Виконано |
| 2 | Отримання завдання на виконання МКР | 21.10.2022 | 10.11.2022 | Виконано |
| 3 | Складання календарного плану на період виконання МКР | 11.11.2022 | 15.11.2022 | Виконано |
| 4 | Огляд літератури за темою дослідження | 16.11.2022 | 04.12.2022 | Виконано |
| 5 | Проходження переддипломної практики, збір та аналіз матеріалів до МКР | 05.12.2022 | 25.12.2022 | Виконано |
| 6 | Аналіз предметної області та розробка технічного завдання. Моделювання результатів | 26.12.2022 | 12.01.2023 | Виконано |
| 7 | Опис фахової частини МКР, зокрема дослідження публікацій щодо діагностування пацієнтів на основі нечітких систем, огляд існуючих архітектур, експертних систем, розробка системи первинної діагностики на основі нечіткої логіки. | 13.01.2023 | 25.01.2023 | Виконано |
| 8 | Розробка спеціальної частини з охорони праці та методичної частини | 26.01.2023 | 02.02.2023 | |
| 9 | Попередній захист МКР на засіданні комісії кафедри | 03.02.2023 | 03.02.2023 | |
| 10 | Корегування роботи за результатами попереднього захисту | 04.02.2023 | 06.02.2023 | |
| 11 | Остаточне оформлення пояснювальної записки та слайдів доповіді для захисту | 07.02.2023 | 09.02.2023 | |
| 12 | Подання МКР рецензенту | 09.02.2023 | 10.02.2023 | |
| 13 | Рецензування МКР | 11.02.2023 | 12.02.2023 | |
| 14 | Подання МКР, її електронної копії та інших документів (відгуку, рецензії) до захисту | 16.02.2023 | 17.02.2023 | |
| 15 | Захист МКР перед екзаменаційною комісією (ЕК) | 24.02.2023 | 24.02.2023 | |

Розробив студент Ергарт Е.В. _____
(прізвище та ініціали) (підпис)

Керівник роботи к.т.н., доцент. Козлов О.В _____
(наук. ступінь, вчене звання, прізвище та ініціали) (підпис)

«12» листопада 2022 р.

АНОТАЦІЯ

до магістерської кваліфікаційної роботи
студента групи 607 ЧНУ ім. Петра Могили

Ергарта Едуарда Володимировича

на тему: “ **СИСТЕМА ПЕРВИННОЇ ДІАГНОСТИКИ НА ОСНОВІ
НЕЧІТКОЇ ЛОГІКИ**”

Актуальність теми обумовлена тим, що у зв'язку зі швидким розвитком інформаційних технологій та інтелектуальних систем постає питання модернізації методів обслуговування, діагностування хвороб та лікування пацієнтів, які дозволяють оптимізувати всі процеси, що проходять між хворим та медичним закладом.

Об'єктом дослідження є процеси автоматизації діагностування захворювань пацієнтів.

Предметом дослідження є методи діагностування захворювань за допомогою експертно-діагностичних систем.

Метою роботи є підвищення ефективності процесу діагностування ймовірних захворювань пацієнтів за рахунок побудови діагностичної системи, що містить актуальну базу знань взаємозв'язків між симптомами і захворюваннями.

В результаті виконання роботи було досліджено систем нечіткого логічного виведення з їх практичним застосуванням в задачах з діагностики потенційних захворювань пацієнтів та розроблено експертно-діагностичну систему.

Дана робота складається з шести розділів. Кожен розділ відповідно присвячений: аналізу предметної області, математичним моделям і методам, використаним у магістерській роботі, моделюванню і проектуванню системи первинної діагностики, аналізу отриманих результатів, охороні праці, методичній частині магістерської роботи. Загальний обсяг роботи – 91 сторінок. Магістерська кваліфікаційна робота містить один додаток, 65 рисунків, 10 таблиць і посилання на 30 літературних джерел.

Ключові слова: нечітка логіка, експертна система, система первинної діагностики.

ABSTRACT

to the master's qualification work by the student of the group 607 of Petro Mohyla
Black Sea National University

Eduard Erhart

“PRIMARY DIAGNOSTIC SYSTEM BASED ON FUZZY LOGIC”

A relevance of the topic is due to the fact that in connection with the rapid development of information technologies, the issue of modernization of service methods, diagnosis of diseases and treatment of patients arises, and fuzzy logic plays an important role in decision-making with the help of intelligent systems.

An object of the study is the process of automating the diagnosis of patients' diseases.

A subject of research is methods of diagnosing diseases using expert diagnostic systems.

A purpose of the work is to increase the efficiency of the process of diagnosing probable diseases of patients due to the construction of a diagnostic system containing an up-to-date knowledge base of the relationships between symptoms and diseases.

As a result of the work, fuzzy logical inference systems were investigated with their practical application in the tasks of diagnosing potential diseases of patients, and an expert diagnostic system was developed.

This work consists of six chapters. Each of them is devoted to: analysis of the subject area, mathematical models and methods used in the master's work, modeling and design of the primary diagnosis system, analysis of the obtained results, labor protection, methodical part of the master's work. The total volume of work is 91 pages. The master's thesis contains one appendix, 65 figures, 10 tables and references to 30 literary sources.

Keywords: fuzzy logic, expert system, primary diagnosis system.

ЗМІСТ

| | |
|--|-----------|
| ПЕРЕЛІК СКОРОЧЕНЬ | 3 |
| ВСТУП..... | 5 |
| 1 АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ ПЕРВИННОЇ ДІАГНОСТИКИ ЗАХВОРЮВАНЬ | 7 |
| 1.1 Опис процесу діагностування пацієнтів..... | 7 |
| 1.2 Останні дослідження та публікації..... | 9 |
| 1.3 Огляд існуючих аналогів..... | 10 |
| 1.4 Постановка задачі..... | 14 |
| Висновки до розділу 1..... | 17 |
| 2 МОДЕЛІ ТА МЕТОДИ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ | 18 |
| 2.1 Основи теорії нечітких множин та нечіткої логіки | 18 |
| 2.2 Нечіткі системи логічного виведення | 22 |
| 2.3 Постановка задачі з точки зору математичної складової..... | 23 |
| 2.4 Нечіткі експертні системи..... | 27 |
| 2.5 Опис технологій..... | 29 |
| Висновки до розділу 2..... | 31 |
| 3 ТЕХНІЧНЕ ЗАВДАННЯ ТА МОДЕЛЮВАННЯ ІС | 32 |
| 3.1 База даних та модель системи..... | 32 |
| 3.2 Архітектура MVC | 41 |
| Висновки до розділу 3..... | 43 |
| 4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА РОЗРОБКА ДОКУМЕНТАЦІЇ..... | 45 |
| 4.1 Обґрунтування та вибір технологій розробки ПЗ..... | 45 |
| 4.2 Програмні засоби для розробки ПЗ | 49 |
| 4.3 Опис програмної реалізації..... | 53 |
| 4.4 Керівництво користувача..... | 59 |
| Висновки до розділу 4..... | 65 |
| ВИСНОВКИ | 67 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ | 69 |

ПЕРЕЛІК СКОРОЧЕНЬ

| | |
|------|---|
| БД | – база даних |
| МКР | – магістерська кваліфікаційна робота |
| ЕС | – Експертна система |
| МОЗ | – Міністерство охорони здоров'я |
| НСЗУ | – національна служба здоров'я України |
| ПЗ | – програмне забезпечення |
| СУБД | – система управління базами даних |
| НТТР | – (Hypertext Transfer Protocol) протокол передачі гіпертексту |
| MVC | – (Model-View-Controller) архітектурний шаблон, який використовується для розробки програмного забезпечення |
| ТСР | – (Transmission Control Protocol) протокол управління передачею даних |
| UI | – (User interface) інтерфейс користувача |

Кафедра інтелектуальних інформаційних систем
Система первинної медичної діагностики на основі нечіткої логіки

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему:

«СИСТЕМА ПЕРВИННОЇ МЕДИЧНОЇ ДІАГНОСТИКИ НА ОСНОВІ НЕЧІТКОЇ ЛОГІКИ»

Спеціальність 124 «Системний аналіз»

124 – МКР – 607.21710110

Виконав студент 6-го курсу, групи 607

_____ Е. В. Ергарт

«17» лютого 2023 р.

Керівник: канд. техн. наук, доцент

_____ О. В. Козлов

«17» лютого 2023 р.

Миколаїв – 2023

ВСТУП

Питання здоров'я та медицини завжди відігравали основну роль в житті людей. Здоров'я – основа щасливого життя людей. Це питання є актуальним навіть у розвинутих країнах, оскільки існує багато різноманітних хвороб, які важко виявити та діагностувати. Тому тема медицини є доволі актуальною і на сьогоднішній день.

З поширенням методів штучного інтелекту їх почали впроваджувати і в сферу медицини. В світі, що постійно розвивається в сфері інформатики та науки про інформацію, нечітка логіка відіграє важливу роль в прийнятті рішень, та за допомогою інтелектуальних принципів в діагностичних системах дозволяє підвищувати точність діагностики за рахунок використання нечітких правил та нечітких значень. Це може відіграти важливу роль в запобіганні помилок в процесі діагностики, а також у підвищенні ефективності систем.

Актуальність теми обумовлена тим, що у зв'язку зі швидким розвитком інформаційних технологій та інтелектуальних систем постає питання модернізації методів обслуговування, діагностування хвороб та лікування пацієнтів, які дозволяють оптимізувати всі процеси, що проходять між хворим та медичним закладом.

Об'єктом дослідження є процеси автоматизації діагностування захворювань пацієнтів.

Предметом дослідження є методи діагностування захворювань за допомогою експертно-діагностичних систем.

Метою роботи є підвищення ефективності процесу діагностування ймовірних захворювань пацієнтів за рахунок побудови діагностичної системи, що містить актуальну базу знань взаємозв'язків між симптомами і захворюваннями.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- аналіз сучасного стану проблеми первинної діагностики захворювань на основі експертних систем;

- аналіз та узагальнення існуючих інтелектуальних моделей та підходів, що використовуються для первинної медичної діагностики;
- розробка системи первинної медичної діагностики на основі нечіткої логіки, що дає можливість встановити точний діагноз хворого;
- дослідження системи первинної медичної діагностики на основі нечіткої логіки, що дає можливість встановити точний діагноз хворого.

В процесі розробки буде використано такі технології та мови програмування:

- Laravel – це відкритий веб-фреймворк на мові програмування PHP для розробки веб-додатків. Це зручний і потужний інструмент, який допомагає розробникам зосередитися на реалізації бізнес-логіки, а не витратити час на написання складного коду;

- MySQL – це безкоштовна система управління базами даних з відкритим вихідним кодом, яка використовує мову запитів SQL. Вона широко використовується для зберігання та управління даними на веб-сайтах. MySQL дозволяє створювати та модифікувати таблиці, виконувати запити до бази даних, забезпечувати безпеку даних та інші функції;

- PHP – це мова програмування, яка широко використовується для розробки веб-додатків. Вона працює на стороні сервера та може бути використана для створення складних веб-додатків. PHP має вбудовану підтримку для роботи з базами даних та HTTP-запитами, що дозволяє створювати динамічні веб-сторінки;

- Bootstrap – це популярний фреймворк для розробки веб-інтерфейсів, що дозволяє швидко та ефективно створювати респонсивні та мобільні версії веб-сайтів та веб-додатків. Він містить набір готових CSS, JavaScript та HTML-компонентів та шаблонів, що можуть бути використані для побудови різноманітних веб-проектів.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ ПЕРВИННОЇ ДІАГНОСТИКИ ЗАХВОРЮВАНЬ

1.1 Опис процесу діагностування пацієнтів

Діагностування пацієнтів - це процес визначення причин хвороби, на підставі аналізу клінічних даних та інших даних, що отримані з різних джерел. Основні концепції, що застосовуються при діагностуванні пацієнтів, включають наступне:

- анамнез: Отримання історії хвороби, а також інших факторів, що можуть вплинути на здоров'я пацієнта. Анамнез дозволяє отримати інформацію про симптоми, що переживає пацієнт, тривалість та інтенсивність симптомів, фактори, що підвищують або знижують симптоми, а також інформацію про історію хвороби, що допоможе зробити точну діагноз;
- фізичний огляд: Включає перевірку стану здоров'я пацієнта, що включає огляд тіла, вимірювання температури тіла, пульсу, тиску, а також огляд зовнішніх симптомів;
- лабораторні дослідження: Включає проведення аналізу крові, сечі, фекалій, а також інших матеріалів для виявлення патогенних мікроорганізмів або ознаки патології;
- інструментальні дослідження: Включає проведення рентгенівських, ультразвукових, магнітно-резонансних та інших досліджень для визначення стану внутрішніх органів та систем органів;
- консультація спеціалістів: У випадках, коли діагноз є складним або неочевидним, пацієнта може бути направлено до спеціаліста, який має досвід та знання.

Діагностування пацієнтів в онлайн системах може бути обмежене у порівнянні з традиційним діагностуванням, оскільки лікар не може провести фізичний огляд пацієнта або зібрати зразки для лабораторних досліджень. Однак,

з використанням спеціалізованих програм та телемедицини можливо здійснити ефективне діагностування пацієнтів онлайн.

Для діагностування пацієнтів в онлайн системах лікар може використовувати відеоконференції для спілкування з пацієнтом та збирання історії хвороби. Також, лікар може надіслати пацієнта на лабораторні дослідження в місцеві лабораторії або відправити тестові комплекти для виконання тестів в домашніх умовах. У деяких випадках, лікар може замовити діагностичні процедури, наприклад рентген, зображення магнітною резонансною томографією (МРТ), комп'ютерною томографією (КТ) або ультразвуком, якщо це можливо в конкретній онлайн системі.

Онлайн системи для діагностування пацієнтів зазвичай забезпечують доступ до медичних записів пацієнтів, дозволяючи лікарю відстежувати хронологію симптомів та лікування, а також проводити моніторинг стану пацієнта в режимі реального часу.

Важливо зазначити, що онлайн діагностування не може замінити традиційне діагностування, особливо в тяжких випадках хвороби, де потрібні додаткові діагностичні методи та лікування.

Система медичної діагностики на основі нечіткої логіки концепції діагностування базується на теорії нечіткої логіки та використовується для вирішення завдань, пов'язаних з діагностикою захворювань та прийняттям рішень про лікування. Ця система діагностики враховує не тільки експертні знання лікарів, а й враховує нечіткість та невизначеність в медичних даних, що може виникнути при діагностуванні пацієнта.

Для того, щоб забезпечити більш точну діагностику, система медичної діагностики на основі нечіткої логіки збирає велику кількість інформації про пацієнта, включаючи його історію захворювань, результати лабораторних досліджень, зображення медичних досліджень та інші медичні дані. Ці дані аналізуються за допомогою алгоритмів нечіткої логіки, які враховують невизначеність та нечіткість у медичних даних.

Система медичної діагностики на основі нечіткої логіки також використовує експертні знання лікарів та експертів, які можуть внести свої висновки та рекомендації до системи. Для цього в системі може бути використана база знань, що містить правила та експертні знання для діагностики різних захворювань.

Після аналізу медичних даних та експертних знань, система медичної діагностики на основі нечіткої логіки може зробити діагноз та надати рекомендації щодо лікування.

1.2 Останні дослідження та публікації

Нечітка логіка — це математична основа для представлення та обробки невизначеної інформації, яка застосовувалася до широкого спектру експертних діагностичних систем. В останні роки було проведено ряд досліджень і публікацій, присвячених розробці та оцінці експертних діагностичних систем на основі нечіткої логіки. Одним із прикладів є використання нечіткої логіки в медичній діагностиці. Дослідники запропонували системи на основі нечіткої логіки для діагностики різних захворювань, таких як діабет, хвороби серця та рак. Ці системи використовують нечіткі правила та базу знань для аналізу симптомів і результатів медичних тестів і створення списку можливих діагнозів. Інша область застосування - використання нечіткої логіки в діагностиці технічних проблем. Наприклад, дослідники розробили системи на основі нечіткої логіки для діагностики несправностей у транспортних засобах, системах електроживлення та системах управління промисловими процесами. Були також проведені дослідження щодо інтеграції нечіткої логіки з іншими технологіями, такими як штучні нейронні мережі та еволюційні алгоритми, для покращення продуктивності та надійності експертних діагностичних систем. Таким чином, використання нечіткої логіки в експертних діагностичних системах продовжує бути активною областю досліджень, і останнім часом у цій галузі було досягнуто багато досягнень. Нижче наведено кілька прикладів публікацій та досліджень на тему медичних експертних систем діагностики пацієнта:

- "Розробка медичної експертної системи для діагностики захворювань": ця публікація описує розробку медичної експертної системи для діагностики різних захворювань, зокрема серцево-судинних та респіраторних. Система використовує нейронні мережі та інші алгоритми машинного навчання для забезпечення точності діагностики;
- "Експертна система діагностики та лікування захворювань": це дослідження описує розробку експертної системи для діагностики та лікування різних захворювань, зокрема раку, діабету та хвороб нирок. Система використовує базу знань, що включає клінічні дані, медичну літературу та рекомендації фахівців, для забезпечення точності діагностики та лікування;
- "Розробка медичної експертної системи для діагностики інфекційних захворювань": ця публікація описує розробку медичної експертної системи для діагностики інфекційних захворювань, зокрема COVID-19. Система використовує клінічні дані та медичні зображення, щоб допомогти фахівцям у діагностиці та лікуванні цих захворювань;
- "Експертна система діагностики раку шкіри на основі зображень": ця публікація описує розробку експертної системи для діагностики раку шкіри на основі зображень. Система використовує нейронні мережі та інші алгоритми машинного навчання.

1.3 Огляд існуючих аналогів

Для аналізу існуючих аналогів даної системи було обрано наступні програмні продукти:

- «Експертна система Internist-I»;
- «Система Casnet»;
- «Експертна система – MYCIN».

Internist-I – відома експертна система в галузі медичної діагностики. Вона була розроблена в 1980-х роках групою дослідників і лікарів і вважається однією з перших успішних експертних систем у галузі медицини.

Система Internist-I призначена для допомоги лікарям у діагностиці внутрішніх захворювань. Він використовує базу експертних медичних знань і набір правил висновку для аналізу симптомів, результатів медичних тестів та іншої інформації про пацієнта, а також створення списку можливих діагнозів.

База знань Internist-I організована як серія дерев рішень, які представляють процес прийняття рішень лікарем-експертом. Для встановлення остаточного діагнозу система використовує комбінацію двійкових і багатозначних рішень на основі симптомів пацієнта та результатів тестів.

Internist-I широко використовувався та оцінювався в клінічних умовах і був визнаний корисним інструментом для діагностики внутрішніх захворювань. Він також використовувався як еталон для оцінки інших експертних діагностичних систем у галузі медицини.

Загалом Internist-I є добре відомою та шанованою експертною системою в галузі медичної діагностики, яка продовжує використовуватися та вивчатися як модель для розробки подібних систем у майбутньому. На рисунку 1.1 – зображено

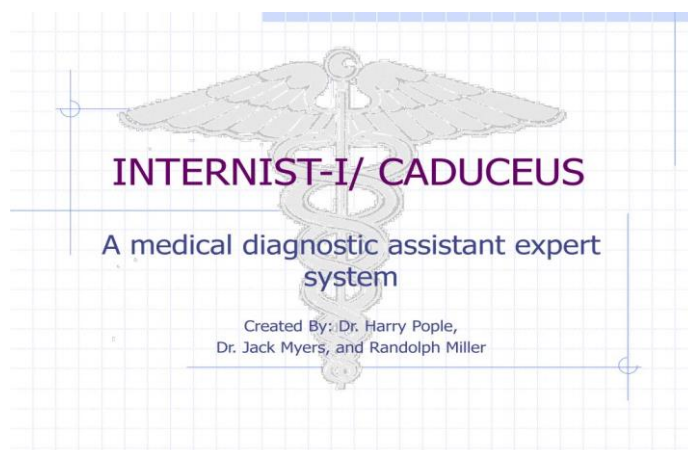


Рисунок 1.1 – Логотип Експертної системи Internist-I

Casnet — експертна система для діагностики несправностей систем управління промисловими процесами. Вона була розроблена в 1990-х роках і вважається однією з перших успішних експертних систем в галузі промислової діагностики.

Система Casnet використовує базу експертних знань про системи керування процесами та набір правил висновків для аналізу симптомів та іншої інформації про систему та створення списку можливих несправностей. База знань організована у вигляді набору дерев рішень, які представляють процес прийняття рішень експертом у галузі промислової діагностики.

Система Casnet використовує комбінацію двійкових і багатозначних рішень на основі симптомів системи, щоб отримати остаточний діагноз. Система також може використовувати додаткову інформацію, наприклад результати тестів, для уточнення свого діагнозу.

Casnet широко використовувався та оцінювався в промислових умовах і був визнаний корисним інструментом для діагностики несправностей у системах керування процесами. Він також використовувався як еталон для оцінки інших експертних діагностичних систем у галузі промислової діагностики.

Загалом Casnet є добре відомою та шанованою експертною системою в галузі промислової діагностики, яка продовжує використовуватися та вивчатися як модель для розробки подібних систем у майбутньому. На рисунку 1.2 зображено логотип експертної системи Casnet.



Рисунок 1.2 – Логотип експертної системи Casnet

MYCIN – відома експертна система в галузі медичної діагностики. Вона була розроблена в 1970-х роках групою дослідників і лікарів і вважається однією з перших успішних експертних систем у галузі медицини. Система MYCIN

призначена для допомоги лікарям у діагностиці та лікуванні інфекційних захворювань. Він використовує базу експертних медичних знань і набір правил висновків для аналізу симптомів, результатів медичних тестів та іншої інформації про пацієнта, а також для створення списку можливих діагнозів і рекомендацій щодо лікування.

База знань MYCIN організована як набір правил «якщо-тоді», які представляють процес прийняття рішень лікарем-експертом. Система використовує комбінацію двійкових і зважених рішень на основі симптомів пацієнта та результатів тестів, щоб отримати остаточний діагноз і рекомендації щодо лікування.

MYCIN широко використовувався та оцінювався в клінічних умовах, і було визнано корисним інструментом для діагностики та лікування інфекційних захворювань. Він також використовувався як еталон для оцінки інших експертних діагностичних систем у галузі медицини.

Загалом MYCIN є добре відомою та шанованою експертною системою в галузі медичної діагностики та лікування, яка продовжує використовуватися та вивчатися як модель для розробки подібних систем у майбутньому. На рис. 1.3 зображено логотип експертної системи MYCIN.



Рисунок 1.3 – Логотип експертної системи MYCIN

Це лише декілька прикладів медичних експертних систем для діагностики. Сфера медичної діагностики постійно розвивається, постійно розробляються нові системи. Використання експертних систем для діагностики стає все більш важливим, оскільки галузь медицини стає все складнішою, а потреба в швидкій і точній діагностиці стає все гострішою. Також є ще декілька прикладів експертних систем:

- DXplain: заснована на знаннях система для діагностики захворювань, вперше розроблена в 1980-х роках;
- клінічна довідкова бібліотека Cerner Multum: база даних інформації про ліки, хвороби та інструменти для підтримки діагностичних рішень;
- Isabel: веб-інструмент діагностики, який допомагає клініцистам поставити диференційний діагноз, беручи до уваги симптоми, результати тестів та іншу інформацію про пацієнта;
- Ada: система штучного інтелекту, яка використовує алгоритми машинного навчання для діагностики захворювань на основі інформації про пацієнтів і медичних записів.

1.4 Постановка задачі

Актуальність теми обумовлена тим, що у зв'язку з швидким розвитком інформаційних технологій постає питання модернізації методів обслуговування, діагностування хвороб та лікування пацієнтів, а нечітка логіка відіграє важливу роль в прийнятті рішень за допомогою інтелектуальних систем.

Об'єктом дослідження є процеси автоматизації діагностування захворювань пацієнтів.

Предметом дослідження є методи діагностування захворювань за допомогою експертно-діагностичних систем.

Метою роботи є підвищення ефективності процесу діагностування ймовірних захворювань пацієнтів за рахунок побудови діагностичної системи, що містить актуальну базу знань взаємозв'язків між симптомами і захворюваннями.

Для досягнення поставленої мети необхідно виконати наступні завдання:

- провести дослідження в галузі нечіткої логіки та її застосування у медицині;
- розробити математичну модель системи діагностики на основі нечіткої логіки;
- створити програмний продукт для реалізації системи діагностики;
- провести тестування та налаштування системи з використанням реальних клінічних даних;
- оцінити ефективність системи порівняно з традиційними методами діагностики;
- підготувати необхідну документацію для запровадження системи у медичну практику.

Використання даної системи спрямовано на оптимізації та автоматизації робочих процесів медичних закладів при роботі з пацієнтами та їх діагностуванням.

Інформаційна система повинна бути реалізована у вигляді web-додатку, що має весь необхідний функціонал, що сприяє комфортній роботі всіх користувачів системи. Web-додаток має складатися із основних структурних розділів, які мають певний функціонал.

Для підтримки ІС й експлуатації інтерфейсу достатньо володіти початковими навичками користування комп'ютером та браузером.

Вся основна інформація про лікарів та пацієнтів має зберігатися в базі даних.

Користувачі системи мають такий розподіл на такі групи відповідно до прав доступу:

- лікарі;
- пацієнти;
- адміністратор.

Лікарі загалом мають особистий кабінет з переліком пацієнтів та їх симптомами та з історією діагностувань.

Пацієнти мають доступ до особистого кабінету, де можуть пройти первинну діагностику та отримати результати діагностування та відповідно від результатів діагностування можуть отримати рекомендацію до якого лікаря звернутися.

Доступ до системи здійснюється за допомогою авторизації користувачів.

Веб-додаток повинен мати такі розділи та функціональні можливості:

- головна сторінка – містить досить короткий опис про саму систему;
- діагностування – даний розділ містить перелік кроків для отримання результатів діагностування та встановлення остаточного діагнозу користувача (пацієнта);
- діагностування пацієнтів – містить основні етапи проходження первинної діагностики пацієнта та визначення його діагнозу;
- особистий кабінет пацієнта – слугує для відображення його основних можливостей та всіх даних, що стосуються скарг та симптомів та історія результатів діагностування;
- особистий кабінет лікаря – слугує для відображення всіх пацієнтів з можливістю діагностування та корегування остаточного діагнозу.

Інтерфейс системи повинен бути інтуїтивно зрозумілий користувачам та мати досить простий функціонал для швидкого запису та отримання результатів діагностики.

Система повинна забезпечувати навігацію по ключовим розділам ІС і відображати відповідну інформацію. Елементи навігації повинні бути гарно оформленими та чітко структурованими. Посилання на сторінки повинні містити заголовок, що відповідає змісту розділу та функціональності розділу.

Система повинна забезпечувати навігацію по всіх доступних користувачеві

ресурсам. Для навігації повинна використовуватися меню-шапка. Меню повинне являти собою блок (список гіперпосилань) у верхній частині сторінки, котрий розміщений по всій ширині головного вікна додатку.

Відображення даних для всіх сторінок повинне відбуватися за допомогою отримання інформації з БД .

Основний функціонал, що відповідає за адміністрування системи повинен міститися на адміністративній панелі сторінки адміністратора, даний функціонал повинен передбачати можливість перегляду всіх лікарів системи, видалення необхідного лікаря із системи та додавання нового лікаря до БД, можливість вносити корективи щодо експертної системи, та її даних.

Висновки до розділу 1

В даному розділі було проаналізовано основні концепції діагностування пацієнтів та їх подальше коригування діагнозу в залежності від симптомів та історії хвороб. Визначено основні особливості діагностування пацієнтів на основі нечіткої логіки, досліджено останні публікації та аналоги, що стосуються даної теми, визначено основні концепції та підходи створення веб-додатку для діагностування пацієнтів на основі нечіткої логіки, поставлено основні задачі та вимоги до технічного завдання. В даному розділі надана коротка інформації про необхідні дані від лікарів та користувачів.

2 МОДЕЛІ ТА МЕТОДИ ДЛЯ ВИРШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Основи теорії нечітких множин та нечіткої логіки

Теорія нечітких множин - це математичний підхід до представлення та обробки нечіткої інформації. Основні поняття в теорії нечітких множин включаються такі як нечіткі множини, функції належності та операції над ними.

Нечітка множина - це множина, елементи якої мають нечітку приналежність до множини. Наприклад, множина "високі люди" може мати елементи, які належать до неї з різною ступенем приналежності, такі як "дуже високий", "трохи високий" або "не дуже високий". Формула для нечіткої множини може бути записана наступним чином:

$$A = \{(x, \mu_A(x)) \mid x \in X\}, \quad (2.1)$$

де A - нечітка множина;

X - універсум (загальна множина елементів);

$\mu_A(x)$ - функція належності елемента x до множини A ;

x - елемент універсуму.

Функція належності - це функція, яка визначає ступінь приналежності елемента до множини. Функція належності може бути задана у вигляді числа від 0 до 1, де 0 означає, що елемент не належить до множини, а 1 - що елемент повністю належить до множини. Наприклад, функція належності для множини "високі люди" може бути записана наступним чином:

$$\mu_A(x) = 1 / (1 + \exp(-(x - a) / b)), \quad (2.2)$$

де a - середня висота;

b - стандартне відхилення.

Операції над нечіткими множинами:

- об'єднання;
- перетин;
- доповнення.

Об'єднання двох нечітких множин A та B визначається як множина C , яка містить елементи, що належать хоча б одній з множин A або B . Формула для об'єднання множин A та B виглядає наступним чином:

$$C = \{(x, \max(\mu_A(x), \mu_B(x))) \mid x \in X\}, \quad (2.3)$$

Перетин двох нечітких множин A та B визначається як множина C , яка містить елементи, які належать як до множини A , так і до множини B . Формула для перетину множин A та B виглядає наступним чином:

$$C = \{(x, \min(\mu_A(x), \mu_B(x))) \mid x \in X\}, \quad (2.4)$$

Доповнення нечіткої множини A визначається як множина, що містить елементи, які не належать множині A . Формула для доповнення множини A виглядає наступним чином:

$$A' = \{(x, 1 - \mu_A(x)) \mid x \in X\}, \quad (2.5)$$

Ці операції можуть використовуватись для різних задач в теорії нечітких множин, наприклад, для розрахунку ступеня приналежності елемента до певної множини, або для визначення нечіткої логіки, яка може бути використана для прийняття рішень в нечітких середовищах.

Розглянемо прямокутну систему координат, в якій на осі ординат будемо відкладати значення $\mu_A(x)$, а на вісі абсцис елементи E будуть розташовані в довільному порядку. Якщо E впорядкована, то даний порядок потрібно зберегти в

розташуванні елементів на осі абсцис. Такі поєднання дозволяють отримати гарну візуалізацію простих операцій над нечіткими множинами. Наприклад, маємо нечітку множину A з певною функцією належності, що зображена на рис. 2.1.

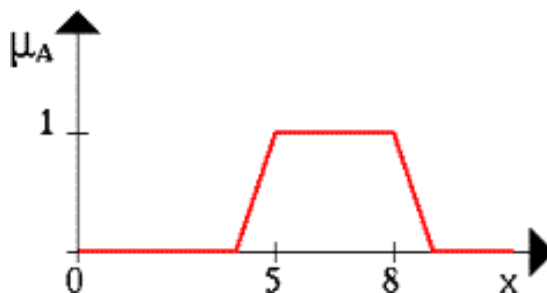


Рисунок 2.1 – Функція належності певної нечіткої множини в інтервалі від 5 до 8

На рис. 2.2 зображена нечітка множина B та нечітке число, що в межах 4, з функцією належності.

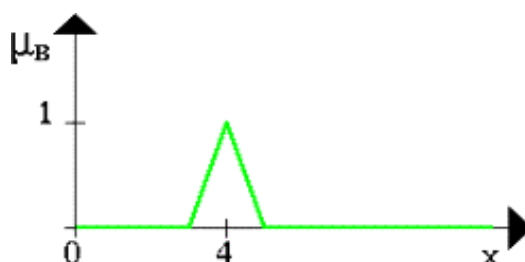


Рисунок 2.2 – Функція належності нечіткої множини, що в межах значення 4

На рис 2.3 зображено функцію належності перетину множин A та B .

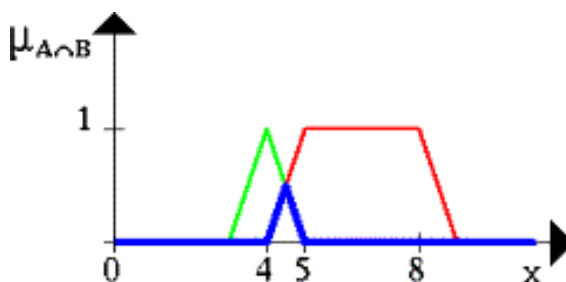


Рисунок 2.3 – Функція належності перетину двох множин

На рис. 2.4 зображено функцію належності суми множин А та В.

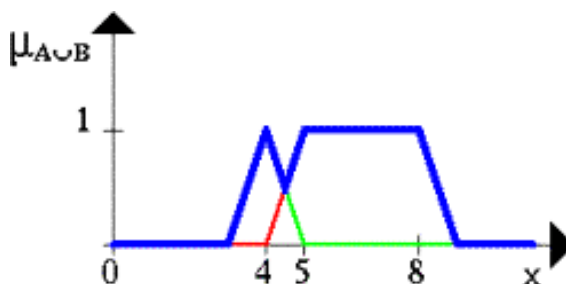


Рисунок 2.4 – Функція належності суми

Приклад операції доповнення можна побачити на рис. 2.5 (доповнення нечіткої множини А).

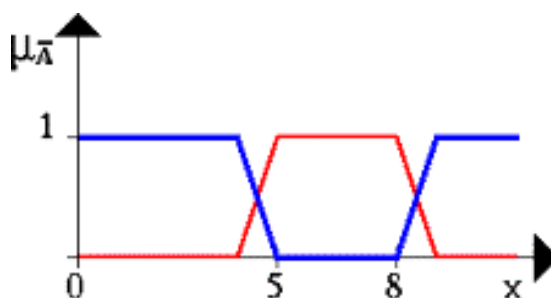


Рисунок 2.5 – Функція належності доповнення нечіткої множини А

Функції належності - це важлива концепція в теорії множин, логіці та інших галузях математики. Вони використовуються для визначення, чи належить елемент певній множині, і можуть бути використані для побудови складніших функцій та виразів.

Функції належності зазвичай задаються у вигляді математичних формул, де визначається, які елементи належать до множини, а які - ні. Наприклад, функція належності для множини натуральних чисел може бути визначена як "x належить до множини натуральних чисел, якщо x є додатнім цілим числом".

Функції належності також використовуються для побудови теорії множин, яка включає в себе операції об'єднання, перетину, різниці та комплементу.

множин. Ці операції можуть бути використані для визначення складніших функцій належності.

Крім того, функції належності мають широке застосування в інших галузях математики та науки, включаючи теорію ймовірностей, математичну статистику, кібернетику та інформатику.

Тому функції належності є важливим інструментом для роботи з множинами та визначення відношень між елементами множин. Вони мають широке застосування в різних галузях математики та науки і є необхідними для побудови складніших функцій та виразів.

Нечітка логіка не безпосередньо пов'язана з числами трикутної форми, оскільки це поняття належить до області теорії чисел. Однак, можна застосувати нечітку логіку для розуміння властивостей та використання чисел трикутної форми.

Наприклад, можна використовувати нечітку логіку для визначення того, наскільки далеко дане число знаходиться від числа трикутної форми. Це може допомогти при прийнятті рішень засобами нечіткої логіки, коли потрібно визначити, наскільки близьким є результат до ідеального значення.

Також можна використовувати нечітку логіку для знаходження аналогій між числами трикутної форми та іншими послідовностями чисел. Наприклад, можна порівнювати числа трикутної форми з числами Фібоначчі та знаходити спільні властивості та закономірності.

Таким чином, нечітка логіка може бути корисною для розуміння та аналізу чисел трикутної форми, а також для прийняття рішень на основі цих чисел.

2.2 Нечіткі системи логічного виведення

Нечіткі системи логічного виведення (НСЛВ) - це математичні моделі, які дозволяють здійснювати логічне виведення на основі нечіткої логіки. Вони

використовуються для прийняття рішень в умовах невизначеності та нечіткості даних.

Основними елементами НСЛВ є вхідні змінні, правила, база знань та механізм виведення. Вхідні змінні - це параметри, які впливають на прийняття рішення. Правила - це умови, які пов'язують вхідні змінні з вихідними. База знань - це набір правил, який використовується для прийняття рішень. Механізм виведення - це процес застосування правил з бази знань до вхідних змінних з метою отримання вихідних змінних.

Прикладом застосування НСЛВ є система керування температурою в приміщенні. В цій системі вхідні змінні - це температура в приміщенні та задана температура, а вихідна змінна - це потужність обігрівачів. Правила містять умови, які пов'язують вхідні змінні з вихідною, наприклад, якщо температура в приміщенні нижче заданої, то потужність обігрівачів повинна збільшуватись. База знань містить набір таких правил. Механізм виведення застосовує ці правила до вхідних змінних з метою визначення вихідних змінних.

2.3 Постановка задачі з точки зору математичної складової

Розглянемо множину захворювань D , що складається з m елементів, і множину симптомів F із n елементів, що мають відношення до захворювань із множини D . Зазвичай $n \gg m$. Тоді:

$$D = \{d_1, d_2, d_3, \dots, d_m\},$$

$$F = \{f_1, f_2, f_3, \dots, f_n\}.$$

Для уточнення стану пацієнта, необхідно визначити рівень інтенсивності усіх симптомів із множини F - тоді кожному симптому пацієнта буде присвоєно нечітке значення (рівень належності), яке обирається з множини L :

$$L = \{\text{Дуже низький, низький, помірний, високий, дуже високий}\}.$$

Тоді описати симптом можна парою <нежить, помірний>. Визначивши інтенсивність усіх n симптомів пацієнта з множини F та призначивши відповідне нечітке значення (рівень належності) кожному симптому, множина симптомів S пацієнта буде виглядати наступним чином:

$$S = \{ \langle f_1, v_1 \rangle, \langle f_2, v_2 \rangle, \langle f_3, v_3 \rangle, \dots, \langle f_n, v_n \rangle \},$$

де v_i , - нечітке значення, яке відповідає рівню належності симптому f_i при перевірці пацієнта, $i = 1, \dots, n$.

Досвід лікаря-експерта щодо сукупності розглянутих захворювань з множини D відображається у вигляді таблиць, у кожній з яких вказується так званий «профіль» певного захворювання. Ми розглядаємо три нечіткі множини «Так», «Ймовірно» та «Ні», які зображені на рисунку 2.6, щоб представити рівень впевненості у наявності захворювання. Записи в таблицях профілів захворювання базуються на цих нечітких множинах.

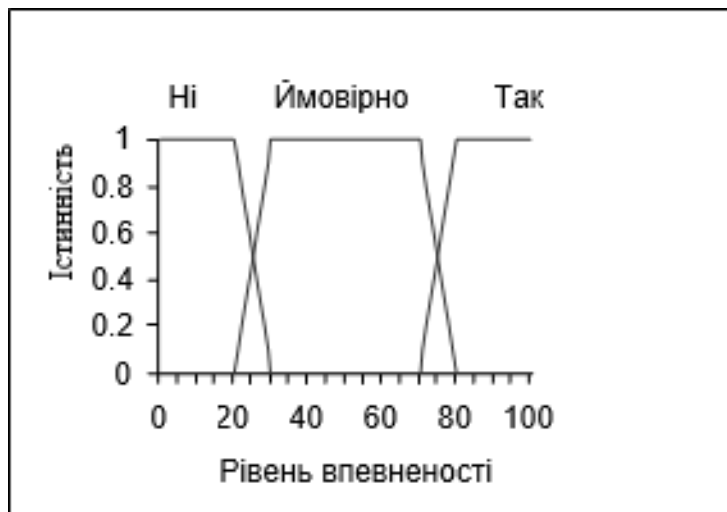


Рисунок 2.6 – Нечіткі множини які описують впевненість наявності захворювання

Для деякої хвороби dx існує множина R відповідних $k \leq n$ симптомів, що є підмножиною загальної множини симптомів F .

У таблиці 2.1 представлена порожня таблиця для профілю захворювання dx . Він вказує п'ять нечітких значень для кожної відповідного симптому g_i , $i = 1, \dots, k$.

Таблиця 2.1 – Приклад порожнього профілю захворювання

| Симптом | Нечітке значення симптому | | | | |
|---------|---------------------------|---------|----------|---------|--------------|
| | Дуже низький | Низький | Помірний | Високий | Дуже високий |
| r_1 | | | | | |
| r_2 | | | | | |
| ... | | | | | |
| r_k | | | | | |

Завданням лікаря-експерта є запропонувати відповідні значення для кожного запису в таблиці профілів захворювання, виходячи з його досвіду. Це слід зробити для кожного захворювання з множини D . У таблиці 2.3 наведено профіль для Грипу, а у таблиці 2.4 наведено приклад запису симптомів пацієнта з відповідними значеннями нечіткими значеннями інтенсивності.

Таблиця 2.2 – Профіль для Грипу

| Рівень належності Симптоми | Дуже низький | Низький | Помірний | Високий | Дуже високий |
|------------------------------|--------------|----------|----------|----------|--------------|
| 1. Нежить | Ні | Ні | Ймовірно | Так | Так |
| 2. Лихоманка | Ні | Ймовірно | Так | Так | Так |
| 3. Кашель | Ні | Ймовірно | Так | Так | Так |
| 4. Сильні болі в тілі | Ні | Ймовірно | Так | Так | Так |
| 5. Головний біль | Ні | Ймовірно | Так | Так | Так |
| 6. Кон'юнктивіт | Ні | Ні | Ймовірно | Так | Так |
| 7. набряк лімфатичних вузлів | Ні | Ні | Ймовірно | Ймовірно | Ймовірно |
| 8. Слабкість у тілі | Ні | Ні | Ймовірно | Так | Так |
| 9. Блювота | Ні | Ні | Так | Ймовірно | Ймовірно |
| 10. Біль у горлі | Ні | Ні | Так | Так | Так |

Таблиця 2.3 – Таблиця симптомів пацієнта

| Симптом | Інтенсивність |
|------------------------------|---------------|
| 1. Нежить | Високий |
| 2. Лихоманка | Середній |
| 3. Кашель | Середній |
| 4. Сильні болі в тілі | Низький |
| 5. Головний біль | Високий |
| 6. Кон'юнктивіт | Високий |
| 7. набряк лімфатичних вузлів | Низький |
| 8. Слабкість у тілі | Середній |
| 9. Втрата апетиту | Середній |
| 10. Чханя | Середній |

Припустимо, що певне захворювання має 10 симптомів, які мають однакову вагу при прийнятті діагнозу. Тоді:

$$k_i = 10, w_{ij} = 1, \quad (2.6)$$

де $j = 1, \dots, 10$.

Використовуючи рівняння для прийняття діагностичних рішень σ_j ($j = 1, \dots, 10$), симптоми та їх відповідні рівні інтенсивності з таблиці 2.4, можна отримати наступний результат: 7 «Так», 2 «Ймовірно» і 1 «Ні». Тоді загальне рішення можна розрахувати так:

$$\sigma_i = (7 \text{ «Так»} + 2 \text{ «Ймовірно»} + 1 \text{ «Ні»}) / 10, \quad (2.7)$$

Після отримання загального рішення σ_i необхідно розрахувати чітке значення ймовірності наявності захворювання у пацієнта. Існує багато методів дефазифікації, однак найбільш поширеними є центроїд (CENTROID) та максимум (MAXIMUM).

Нечітка множина загального рішення σ_i показана на рисунку 2.7.

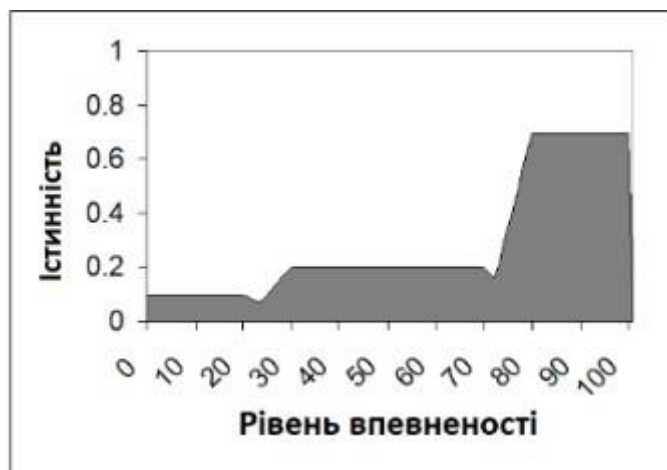


Рисунок 2.7 – Нечітка множина загального рішення σ_i

Використовуючи метод центроїд маємо:

- c_i - середнє арифметичне (центроїд) нечіткої множини σ_i ;
- $c_{\text{Так}}$ - середнє арифметичне (центроїв) рівня належності «Так»;
- q_i - значення ймовірності наявності симптому.

Для цього прикладу значення c_i та $c_{\text{Так}}$ відповідно дорівнюють 0.67 і 0.87.

Тоді:

$$q_i = \left(\frac{c_i}{c_{\text{Так}}} \right) * 100\% = \left(\frac{0.67}{0.87} \right) * 100\% = 77.1\%, \quad (2.8)$$

2.4 Нечіткі експертні системи

Нечіткі експертні системи - це системи, які використовують нечітку логіку для прийняття рішень на основі даних, які можуть містити нечітку інформацію або нечіткість. Ці системи зазвичай складаються з трьох основних компонентів: база знань, модуль інференції та інтерфейс користувача.

База знань - це централізоване сховище інформації, яке містить знання, отримані від експертів у певній області. Знання можуть бути відображені у вигляді правил, фактів, гіпотез та інших форм.

Модуль інференції - це програмний модуль, який забезпечує логічне виведення на основі даних з бази знань. Він використовує правила та гіпотези, збережені у базі знань, щоб виконувати логічні операції та приймати рішення.

Інтерфейс користувача - це компонент, який дозволяє користувачам взаємодіяти з системою. Він може бути графічним інтерфейсом, текстовим інтерфейсом або комбінованим інтерфейсом. Інтерфейс дозволяє користувачам ввести запити, переглядати результати та змінювати параметри системи.

Одним з прикладів застосування нечітких експертних систем є системи підтримки прийняття рішень. Наприклад, нечітка експертна система може використовуватися для прийняття рішень щодо підтримки або припинення виробництва певного продукту, в залежності від різноманітних факторів, таких як прибуток, ризику, витрати та інші.

Нечіткі експертні системи в медицині використовуються для підтримки медичних діагностичних процесів та прийняття рішень. Ці системи базуються на знаннях експертів та використовують математичні методи нечіткої логіки.

Наприклад, нечітка медична діагностична система може бути використана для діагностики захворювань шкіри. Вона базується на знаннях дерматологів та використовує нечіткі правила, які описують зв'язки між симптомами та захворюваннями. Наприклад, правило "Якщо пацієнт має плями на шкірі, які змінюють колір під впливом сонця, то це може свідчити про ліхен планус" може бути використане в системі. Система отримує дані про симптоми пацієнта та застосовує правила для визначення діагнозу. Чим більше симптомів збігається з описом захворювання, тим вища вага присвоюється цьому діагнозу.

Нечіткі експертні системи можуть також використовуватись для прийняття рішень у медицині. Наприклад, система може допомогти лікарю визначити оптимальну дозу лікарства для пацієнта на основі його віку, статі та стану здоров'я. Система використовує нечіткі правила та базується на знаннях експертів для прийняття оптимального рішення.

Основна перевага нечітких експертних систем полягає у їх здатності обробляти нечітку та невизначену інформацію, що є типовим для медичної діагностики.

Одним з прикладів експертної системи медичного діагностування, що базується на базі правил, може бути система, яка допомагає лікарям при діагностиці та лікуванні захворювань шкіри. Наприклад, така система може мати наступний набір правил:

- якщо у пацієнта є свербіж та почервоніння на шкірі, а також симптоми дерматиту, то діагноз може бути atopічний дерматит;
- якщо у пацієнта є білі або червоні плями на шкірі, а також симптоми зуду та сухості, то діагноз може бути псоріаз;
- якщо у пацієнта є свербіж та висип на шкірі, а також симптоми подібні до віспи, то діагноз може бути вітряна віспа;
- якщо у пацієнта є почервоніння, болісні висипи та виразки на шкірі, а також симптоми грибової інфекції, то діагноз може бути дерматофітоз.

Система може питати лікаря певні запитання щодо симптомів та ознак, а потім порівнювати їх з правилами, щоб дійти до підтвердження діагнозу. Вона також може надавати рекомендації щодо лікування та допомагати лікареві при виборі найкращого підходу до лікування.

2.5 Опис технологій

Перед початком реалізації додатку маємо обрати найбільш доцільні технології та методи, що будуть використовуватися під час розробки проекту. Згідно поставленої мети та задачі, оптимальними варіантами технології та методів є:

- HTML (Hypertext Markup Language) — стандартна мова розмітки, яка використовується для створення веб-сторінок. Вона забезпечує

- структуру та вміст веб-сторінки, визначає заголовки, абзаци, зображення, посилання та інші елементи [1];
- CSS (каскадні таблиці стилів) — це мова, яка використовується для оформлення стилів веб-сторінок. Він контролює представлення вмісту, наприклад колір, макет, розмір шрифту та інтервал між елементами на сторінці [2];
 - Laravel – безкоштовний з відкритим кодом, необхідний для розробки веб-додатків з архітектурною моделлю MVC [3];
 - JavaScript — це мова програмування, яка використовується для додавання інтерактивності та динамічних функцій веб-сторінкам. Це дозволяє веб-розробникам створювати інтерактивні елементи, такі як спливаючі вікна, спадні меню та слайд-шоу. JavaScript також можна використовувати для зв'язку з веб-серверами та маніпулювання даними на веб-сторінці [4];
 - Bootstrap – це безкоштовний набір інструментів з відкритим кодом, призначений для створення веб-сайтів та додатків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript. Він спрощує розробку динамічних веб-сайтів і веб-додатків [5];
 - Angular Material - це набір готових компонентів та стилів для Angular, що дозволяє розробникам швидко створювати естетичний та зручний користувацький інтерфейс. Angular Material містить компоненти для роботи з формами, таблицями, кнопками, меню та багатьма іншими елементами інтерфейсу, що дозволяє розробникам швидко створювати функціональний та привабливий користувацький інтерфейс;
 - AJAX – підхід до побудови користувацьких інтерфейсів веб-додатків, за яких веб-сторінка, не перезавантажуючись, у фоновому режимі надсилає запити на сервер і сама звідти довантажує потрібні користувачу дані [6];

- PHP – інтерпретована мова програмування, що має властивість виконувати скрипти та генерувати HTML сторінки з серверної сторони;
- MySql – вільна реляційна система керування базами даних [7].

Висновки до розділу 2

В даному розділі наведено основні моделі систем первинної діагностики на базі нечіткої логіки. Надано приклади методологій за допомогою яких здійснюється побудова експертної системи на базі правил для медичної системи діагностики пацієнта. Описано основні концепції діагностування пацієнта за допомогою нечіткої логіки. Надано інформацію про використання протоколів у фреймворку Laravel. Описано основні технології, що використовуються при створенні інформаційних систем спрямованих на архітектуру веб-додатків. Описано модель системи з математичної точки зору. Наведено основні компоненти системи та процес діагностування в теоретичній формі.

3 ТЕХНІЧНЕ ЗАВДАННЯ ТА МОДЕЛЮВАННЯ ІС

3.1 База даних та модель системи

Архітектура системи являє із себе клієнт-серверну модель, що зазвичай використовується для створення веб-систем, які містять в собі процеси обміну даними між компонентами системи. Основні складові моделі:

- сервер – приймає запити від користувачів системи та надає відповіді у вигляді HTML сторінок;
- користувачі системи, що звертаються до сервера за необхідним ресурсом;
- мережа котра надає можливість учасникам системи обмінюватися даними.

На рис. 3.1 можна побачити приклад клієнт-серверної архітектури.

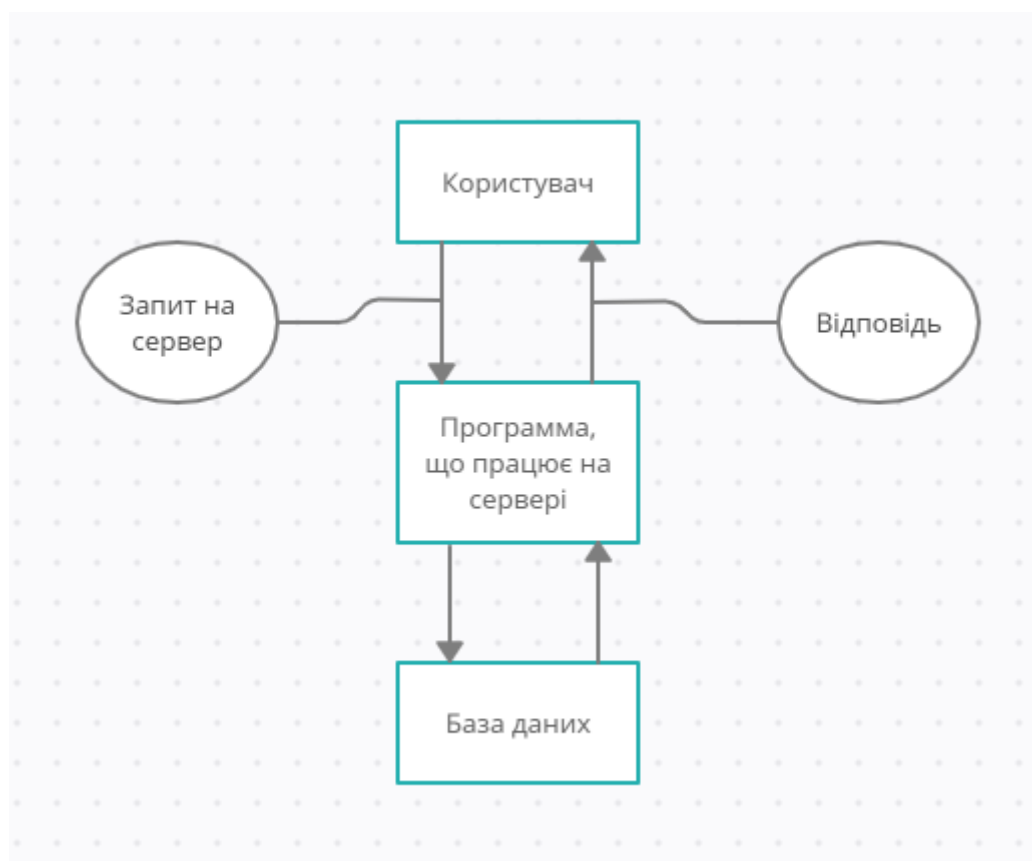


Рисунок 3.1 – Схема, що описує взаємодію користувача з сервером

Для загального представлення моделі самої системи наведемо схему, що словесно описує архітектуру веб-додатку з її функціоналом. На рис. 3.2 можна побачити основні структурні компоненти, що утворюють систему електронного запису до лікарів.

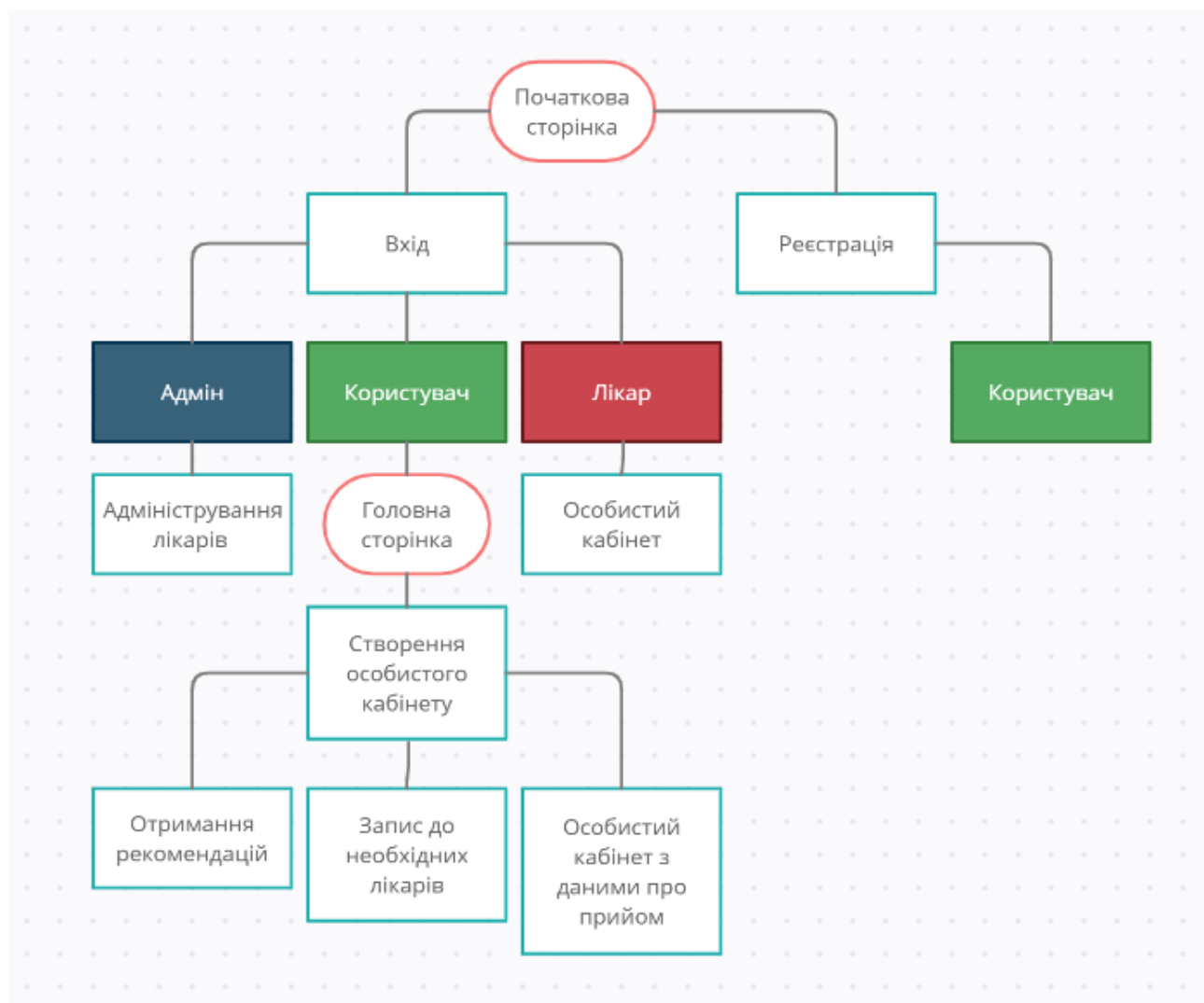


Рисунок 3.2 – Схема основних структурних розділів системи

Вищенаведена схема показує основні структурні розділи веб-додатку, які може використовувати користувач системи відповідно до своїх прав зазначених в базі даних. Кожного користувача системи можна поділити на три групи користувачів, що представлені у вигляді схеми, котра зображена на рис. 3.3.

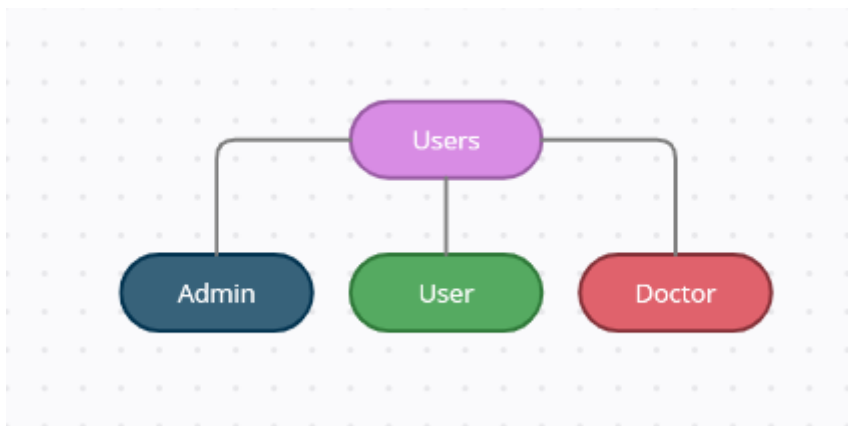


Рисунок 3.3 – Розподілення користувачів системи на ролі [8]

Малюнок рис. 3.3 зображує схему, що відповідає розподілу користувачів за їх правами у системі [9]. Відповідно до права користувача, який авторизувався в системі надається відповідний функціонал та інтерфейс. Далі на рис. 3.4 зображено відповідна таблиця, яка містить дані про користувачів системи.

| T→ | id | email | email_verified_at | password | remember_token | created_at | updated_at | role | | |
|--|----|-------|-------------------|---|----------------|---|---------------------|---------------------|---------------------|--------|
| Клікніть випадаючу стрелку для переключення видимості столбця. | | | NULL | \$2y\$10\$Yg1gj7g3lqX7m7xHelMh/gwAE7AYq5dpdjZYicRT... | NULL | 2021-06-11 05:42:03 | 2021-06-11 05:42:03 | admin | | |
| | | | NULL | \$2y\$10\$8dKm.3obmOtaxFFSdfk.3arFzwmahDqs6fYUV38c... | NULL | 2021-06-15 08:47:12 | 2021-06-15 08:47:12 | user | | |
| | | | 56 | patient2@gmail.com | NULL | \$2y\$10\$TR0Jckwran4oBRXMm0bJdeHX/LMzP6nRgSdDOLLMC55... | NULL | 2021-06-15 08:57:55 | 2021-06-15 08:57:55 | user |
| | | | 57 | patient3@gmail.com | NULL | \$2y\$10\$1GmnoSMQ0OV2Xb4uWkqbeShGuG4L0wM3ed9njfJQ2... | NULL | 2021-06-15 08:59:19 | 2021-06-15 08:59:19 | user |
| | | | 58 | doctor1@gmail.com | NULL | \$2y\$10\$SHKIZLir1VP3AeCCfEWP0ugAnvVe8xkU8XQqMxfY7.N... | NULL | 2021-06-15 09:01:47 | 2021-06-15 09:01:47 | doctor |
| | | | 60 | doctor2@gmail.com | NULL | \$2y\$10\$VodF4Cxmq1Fy3CGNXz0xx.4sAIARQGxfSAqNA/cSTOQ... | NULL | 2021-06-15 09:04:10 | 2021-06-15 09:04:10 | doctor |
| | | | 61 | doctor3@gmail.com | NULL | \$2y\$10\$8AI55xBCKmsFrKNf6yw1L.yMTnCbz9vzIXQ8jqY86rS... | NULL | 2021-06-15 09:11:11 | 2021-06-15 09:11:12 | doctor |
| | | | 62 | doctor4@gmail.com | NULL | \$2y\$10\$K4WSzeFWrHoAr24gHcRRCOloqAOR7IEjehZvIKKJg... | NULL | 2021-06-15 09:13:03 | 2021-06-15 09:13:03 | doctor |
| | | | 63 | doctor5@gmail.com | NULL | \$2y\$10\$ey6pJ5ow/ri2onQPmAYseAYc4zj/p8h5Lp1.lao6Vv... | NULL | 2021-06-15 09:14:56 | 2021-06-15 09:14:56 | doctor |
| | | | 64 | doctor6@gmail.com | NULL | \$2y\$10\$06lyBzO2HuB0z5vR7RIIReet.MptMZ/aeJUzIKXrJP8... | NULL | 2021-06-15 09:16:30 | 2021-06-15 09:16:31 | doctor |
| | | | 65 | doctor7@gmail.com | NULL | \$2y\$10\$otM8B33Zya8bHdkohbpm0u62E7NFEBSLidifu3MXQG... | NULL | 2021-06-15 09:17:46 | 2021-06-15 09:17:46 | doctor |
| | | | 66 | doctor8@gmail.com | NULL | \$2y\$10\$Qt489q2Hh7IDJfPEHOTcxu8GMHXcxyQ2KP4Usc0gp6E... | NULL | 2021-06-15 09:20:07 | 2021-06-15 09:20:07 | doctor |
| | | | 67 | test@gmail.com | NULL | \$2y\$10\$ntGGSkL9BGAKJbFzZVO9S0q/KCvCC5WfI9r.Jng7xEdd... | NULL | 2021-06-16 16:11:21 | 2021-06-16 16:11:21 | user |
| | | | 68 | edik.ergart2017@gmail.com | NULL | \$2y\$10\$B2ZQScRP8eabxRE9wd1wleewl83n02UyVWnk1JorGp... | NULL | 2023-02-15 12:41:05 | 2023-02-15 12:41:05 | user |

Рисунок 3.4 – Дані про користувачів системи в БД

На малюнку рис. 3.4 можна побачити стовпчик з назвою «role», який відповідає за розподіл прав користувачів. На рис. 3.5 можна побачити загальну структуру таблиці користувачів [10].

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|---|-------------------|--------------|--------------------|----------|------|--------------|-------------|----------------|----------|
| 1 | id | bigint | | UNSIGNED | Нет | Нет | | AUTO_INCREMENT | ✎ ⌵ |
| 2 | email | varchar(255) | utf8mb4_unicode_ci | | Нет | Нет | | | ✎ ⌵ |
| 3 | email_verified_at | timestamp | | | Да | NULL | | | ✎ ⌵ |
| 4 | password | varchar(255) | utf8mb4_unicode_ci | | Нет | Нет | | | ✎ ⌵ |
| 5 | remember_token | varchar(100) | utf8mb4_unicode_ci | | Да | NULL | | | ✎ ⌵ |
| 6 | created_at | timestamp | | | Да | NULL | | | ✎ ⌵ |
| 7 | updated_at | timestamp | | | Да | NULL | | | ✎ ⌵ |
| 8 | role | varchar(10) | utf8mb4_unicode_ci | | Нет | user | | | ✎ ⌵ |

Рисунок 3.5 – Структура таблиці користувачів системи

Таблиця «users», основні поля:

- id – унікальний ідентифікатор;
- email – пошта користувача;
- password – пароль користувача;
- role – роль користувача в системі для надання необхідних прав.

Для того, щоб увійти в систему спочатку потрібно в ній зареєструватись.

Приклад процесу реєстрації зображений у вигляді схеми на рис. 3.6.



Рисунок 3.6 – Процес реєстрації користувача [11]

Реєструватися можуть лише користувачі системи, які є пацієнтами і будуть проходити діагностування на відміну від самих лікарів. Для лікарів діє дещо інший механізм, який не включає в себе процес реєстрації. Додаванням лікарів до бази даних та створенням їх обліковим записом займається адміністратор сайту та тише він має на це право. Всі користувачі можуть легко авторизуватися в системі. Процес авторизації, що зображений на рис. 3.7, один для всіх користувачів.

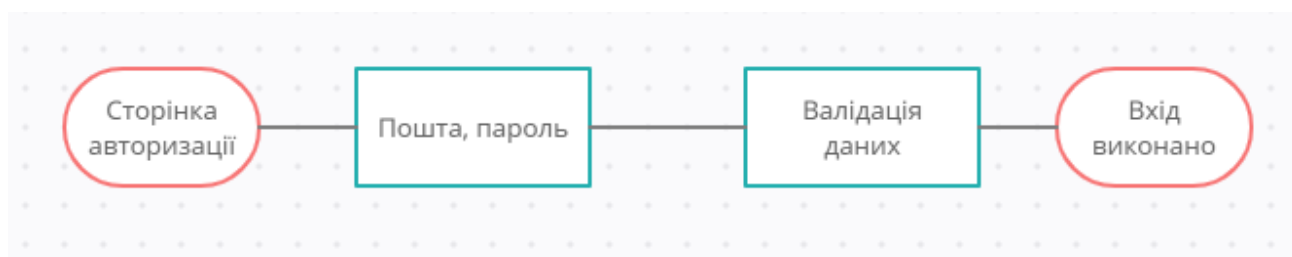


Рисунок 3.7 – Процес авторизації користувача на сайт [12]

Система електронного запису до лікарів передбачає збереження необхідних даних користувача для подальшого їхнього використання в процесі запису на прийом до лікаря. Тому після процесу авторизації користувач має створити власний кабінет надавши свої дані системі для їх подальшої обробки. Якщо користувач створить власний кабінет він не зможе пройти діагностування та потім записатися на прийом до лікаря. Для створення особистого кабінету необхідно внести такі дані:

- ім'я пацієнта;
- прізвище пацієнта;
- ім'я по батькові пацієнта;
- номер телефону пацієнта.

Коли дані користувача внесено до системи, створюється особистий кабінет пацієнта де можна отримати докладну інформацію про пацієнта та список скарг та історію результатів діагностувань й список його записів на прийом до лікарів. Якщо користувач надав свої дані то при вході в свій обліковий запис в нього

будуть доступні всі можливості системи, інакше система буде пропонувати внести дані для подальшого користування системою. На малюнку рис. 3.8 зображено таблиця, яка відповідає за збереження даних пацієнтів, які створили власний кабінет.

| | id | patient_id | name | I_name | patronymic | phone_num | created_at | updated_at |
|--|----|------------|--------|-----------|---------------|---------------|---------------------|---------------------|
| | 54 | 55 | Микола | Коваленко | Миколайович | +380682608833 | 2021-06-15 08:57:39 | 2021-06-15 08:57:39 |
| | 55 | 56 | Іван | Іванов | Іванович | +380508280011 | 2021-06-15 08:58:56 | 2021-06-15 08:58:56 |
| | 56 | 67 | Едуард | Ергарт | Володимирович | +380994059262 | 2021-06-16 17:47:54 | 2021-06-16 17:47:54 |

Рисунок 3.8 – Таблиця «patients» та дані, що в ній зберігаються

На рис. 3.9 описана структура таблиці «patient».

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|---|------------|-------------|--------------------|----------|------|--------------|-------------|----------------|----------|
| 1 | id | bigint | | UNSIGNED | Нет | Нет | | AUTO_INCREMENT | |
| 2 | patient_id | bigint | | UNSIGNED | Да | NULL | | | |
| 3 | name | varchar(25) | utf8mb4_unicode_ci | | Нет | Нет | | | |
| 4 | I_name | varchar(25) | utf8mb4_unicode_ci | | Нет | Нет | | | |
| 5 | patronymic | varchar(25) | utf8mb4_unicode_ci | | Нет | Нет | | | |
| 6 | phone_num | varchar(15) | utf8mb4_unicode_ci | | Нет | Нет | | | |
| 7 | created_at | timestamp | | | Да | NULL | | | |
| 8 | updated_at | timestamp | | | Да | NULL | | | |

Рисунок 3.9 – Структура таблиці «patients»

Короткий опис основних полів таблиці:

- id – унікальний ідентифікатор пацієнта;
- patient_id – ідентифікатор, який посилається на ідентифікатор користувача в таблиці «users»;
- name – ім'я пацієнта;
- I_name – прізвище пацієнта;
- patronymic – ім'я по батькові пацієнта;
- phone_num – номер телефону пацієнта.

Дані, які потребує система не є вибагливими та не несуть шкоди користувачу, зібрані дані потрібні лише для комфортної роботи системи та

можливості здійснити ефективну діагностику та встановити якомога точніший діагноз для ефективнішого лікування пацієнта [13].

Аналогічним чином, система первинної медичної діагностики передбачає внесення даних лікаря, щоб пацієнти могли отримати всю необхідну та докладну інформацію про нього. Відповідно до зазначених вище вимог дані лікаря мають бути наступними:

- ім'я;
- прізвище;
- ім'я по батькові;
- професія;
- номер телефону;
- адреса лікарні де працює.

Всі дані, що надає лікар вносяться в систему адміністратором за допомогою адміністративної панелі, яка містить весь необхідний функціонал для додавання та видалення даних лікарів із системи [14].

Відповідно вищезазначених даних, місце де зберігається дана інформація є таблиця «doctors», що зображена на рис. 3.10.

| + Параметри | | | | | | | | | | | |
|-------------|---|----|-----------|-----------|-----------|------------|---------------|---------------|--|---------------------|---------------------|
| ← T → | | | | | | | | | | | |
| | | id | doctor_id | name | I_name | patronymic | specialty | phone_num | address | created_at | updated_at |
| ■ | ✎ | 21 | 58 | Микола | Коваль | Іванович | Хірург | +380978866351 | м. Миколаїв, вул. Корабелів, 14 | 2021-06-15 09:01:47 | 2021-06-15 09:01:47 |
| ■ | ✎ | 23 | 60 | Сергій | Шевченко | Іванович | Травматолог | +380999444882 | 54020, м.Миколаїв, вул. Корабелів 14Б | 2021-06-15 09:04:10 | 2021-06-15 09:04:10 |
| ■ | ✎ | 24 | 61 | Марія | Олійник | Іванівна | Дієтолог | +380668284797 | вул. Адмірала Макарова, 1, місто Миколаїв | 2021-06-15 09:11:12 | 2021-06-15 09:11:12 |
| ■ | ✎ | 25 | 62 | Ольга | Кравченко | Іванівна | Кардіолог | +380668196672 | 54058, м. Миколаїв, вул. Київська, 1 | 2021-06-15 09:13:03 | 2021-06-15 09:13:03 |
| ■ | ✎ | 26 | 63 | Валентина | Мороз | Миколаївна | Психотерапевт | +380676318249 | 54029, м. Миколаїв, вул. Рюміна, 5 | 2021-06-15 09:14:57 | 2021-06-15 09:14:57 |
| ■ | ✎ | 27 | 64 | Тетяна | Ткаченко | Іванівна | Окуліст | +380935874948 | 54000 м.Миколаїв, вул. Космонавтів, буд.97 | 2021-06-15 09:16:31 | 2021-06-15 09:16:31 |
| ■ | ✎ | 28 | 65 | Анатолій | Шевченко | Іванович | Стоматолог | +380672998809 | 54020, м.Миколаїв, вул. Корабелів 14Б | 2021-06-15 09:17:46 | 2021-06-15 09:17:46 |
| ■ | ✎ | 29 | 66 | Микола | Шевченко | Петрович | Стоматолог | +380961788739 | 54000 м.Миколаїв, вул. Космонавтів, буд.97 | 2021-06-15 09:20:07 | 2021-06-15 09:20:07 |

Рисунок 3.10 – Дані таблиці «doctors»

Структуру таблиці «doctors» можна почати на рис. 3.11.

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|----|------------|--------------|--------------------|----------|------|--------------|-------------|----------------|----------|
| 1 | id | bigint | | UNSIGNED | Нет | Нет | | AUTO_INCREMENT | ✎ ⌵ |
| 2 | doctor_id | bigint | | UNSIGNED | Да | NULL | | | ✎ ⌵ |
| 3 | name | varchar(25) | utf8mb4_unicode_ci | | Нет | Нет | | | ✎ ⌵ |
| 4 | l_name | varchar(25) | utf8mb4_unicode_ci | | Нет | Нет | | | ✎ ⌵ |
| 5 | patronymic | varchar(25) | utf8mb4_unicode_ci | | Нет | Нет | | | ✎ ⌵ |
| 6 | specialty | varchar(25) | utf8mb4_unicode_ci | | Нет | Нет | | | ✎ ⌵ |
| 7 | phone_num | varchar(15) | utf8mb4_unicode_ci | | Нет | Нет | | | ✎ ⌵ |
| 8 | address | varchar(100) | utf8mb4_unicode_ci | | Нет | Нет | | | ✎ ⌵ |
| 9 | created_at | timestamp | | | Да | NULL | | | ✎ ⌵ |
| 10 | updated_at | timestamp | | | Да | NULL | | | ✎ ⌵ |

Рисунок 3.11 – Структура таблиці «doctors»

Короткий опис основних полів таблиці:

- id – ідентифікатор лікаря;
- doctor_id – поле, що посилається на ідентифікатор користувача в таблиці «users»;
- name – ім'я лікаря;
- l_name – прізвище лікаря;
- patronymic – ім'я по батькові лікаря;
- specialty – назва професії лікаря;
- phone_num – номер телефону лікаря;
- address – адреса робочого місця лікаря.

Система запису до лікарів повинна містити дані про візит пацієнта до лікаря, а саме:

- дані лікаря;
- дані пацієнта;
- час візиту;
- дата візиту.

Після того, як користувач записується на прийом до лікаря створюється відповідний запис в базі даних про даний візит. Після внесення даних стосовно візиту в БД, з'являється рядок повідомлення про успішний запис та при переході в особистий кабінет користувача, відображаються дані про нещодавно створений

візит [15]. Аналогічно в особистому кабінеті лікаря з'являється блок з інформацією пацієнта про запис на прийом. За зберігання даних всіх візитів до лікаря відповідає таблиця, що зображена на рис. 3.12.

| id | doctor_id | patient_id | time | day | created_at | updated_at |
|-----|-----------|------------|-------|----------|---------------------|---------------------|
| 54 | | 54 | 12:00 | 21-06-21 | 2021-06-16 12:29:24 | 2021-06-16 12:29:25 |
| 54 | | 54 | 13:00 | 18-06-21 | 2021-06-16 12:29:40 | 2021-06-16 12:29:40 |
| 103 | 21 | 54 | 13:00 | 21-06-21 | 2021-06-16 12:29:53 | 2021-06-16 12:29:53 |
| 105 | 25 | 55 | 12:00 | 23-06-21 | 2021-06-16 12:30:29 | 2021-06-16 12:30:29 |
| 106 | 29 | 55 | 14:00 | 17-06-21 | 2021-06-16 12:30:45 | 2021-06-16 12:30:45 |

Рисунок 3.12 – Дані таблиці «visits»

На малюнку рис. 3.13 можна побачити структуру таблиці «visits».

| # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно | Действие |
|---|------------|--------------|--------------------|----------|------|--------------|-------------|----------------|----------|
| 1 | id | bigint | | UNSIGNED | Нет | Нет | | AUTO_INCREMENT | |
| 2 | doctor_id | bigint | | UNSIGNED | Да | NULL | | | |
| 3 | patient_id | bigint | | UNSIGNED | Да | NULL | | | |
| 4 | time | varchar(255) | utf8mb4_unicode_ci | | Нет | Нет | | | |
| 5 | day | varchar(255) | utf8mb4_unicode_ci | | Нет | Нет | | | |
| 6 | created_at | timestamp | | | Да | NULL | | | |
| 7 | updated_at | timestamp | | | Да | NULL | | | |

Рисунок 3.13 – Структура таблиці «visits»

Короткий опис основних полів таблиці:

- id – унікальний ідентифікатор візиту;
- doctor_id – поле, що посилається на ідентифікатор лікаря в таблиці «doctors»;
- patient_id – поле, що посилається на ідентифікатор пацієнта в таблиці «patients»;
- time – час візиту;
- day – дата візиту.

База даних, що містить всі вище перелічені таблиці має назву «Diploma», та має схему зв'язків між таблицями, котра зображена на малюнку рис. 3.14.

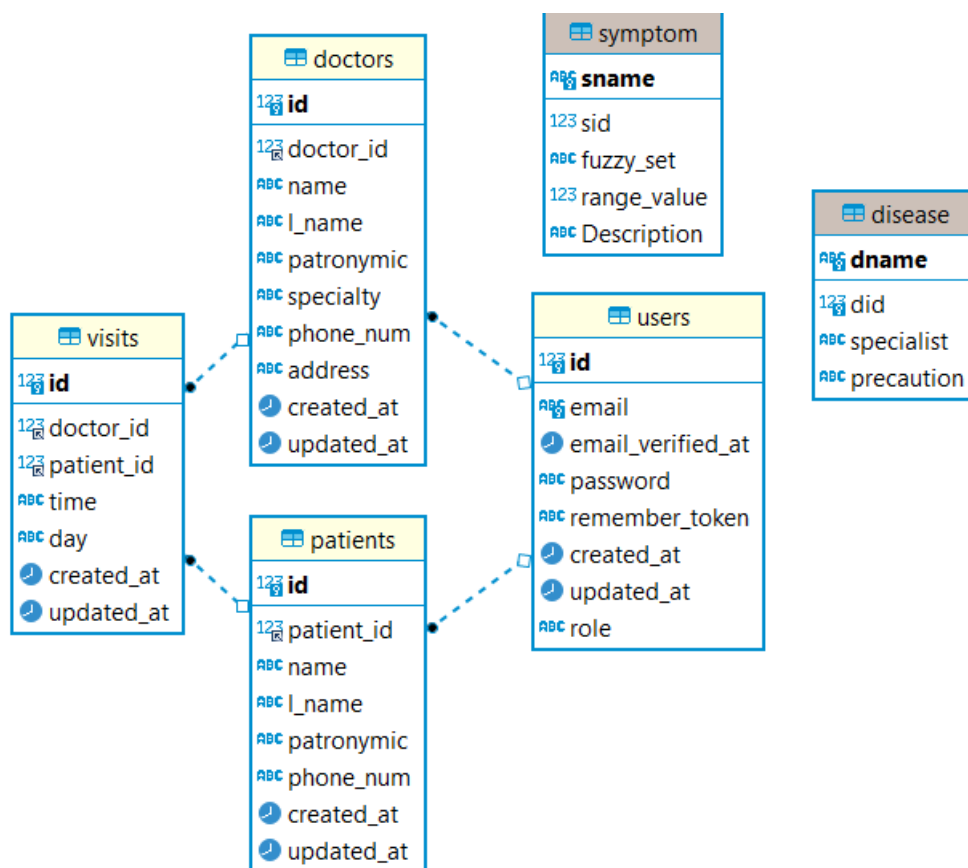


Рисунок 3.14 – Схема зв'язків таблиць в базі даних «Diploma»

Схема на рис. 3.14 чітко описує зв'язок даних пацієнта та лікаря з їхніми даними, що необхідні для входу в обліковий запис, а таблиця «visits» демонструє зв'язок між лікарем та його пацієнтом, що записаний на прийом [16].

3.2 Архітектура MVC

MVC відомий як архітектурний шаблон, який втілює три частини Model, View та Controller, або, точніше кажучи, він ділить додаток на три логічні частини: частину моделі, вигляду та контролер [17]. Він використовувався для графічних інтерфейсів настільних комп'ютерів, але в даний час використовується для проектування мобільних додатків та веб-програм [18].

MVC - це архітектурний шаблон, це означає, що він керує всією архітектурою програми [19]. Незважаючи на те, що часто він відомий як шаблон дизайну, ми можемо помилитися, якщо позначати його лише як шаблон дизайну, оскільки шаблони дизайну використовуються для вирішення конкретної технічної проблеми, тоді як шаблон архітектури використовується для вирішення архітектурних проблем, тому він впливає на архітектуру нашого додатку [20].

Він має три основні компоненти:

- модель;
- вид;
- контролер.

Основні причини, за якими використовується MVC: По-перше, даний шаблон не дозволяє нам повторюватись, а по-друге, це допомагає створити надійну структуру наших веб-додатків. На рис. 3.15 зображено MVC шаблон.

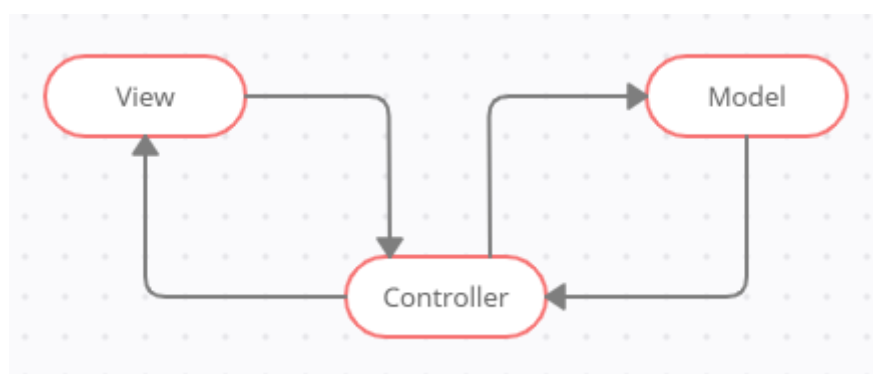


Рисунок 3.15 – Архітектура MVC

Модель. Відома, як найнижчий рівень, це означає, що вона відповідає за збереження даних. Обробляйте дані логічно, щоб вони в основному мали справу з даними [21]. Модель насправді підключена до бази даних, тому це все, що ви робите з даними. Додавання або отримання даних здійснюється в компоненті моделі. Він відповідає на запити контролера, оскільки контролер ніколи не розмовляє з базою даних сам. Модель розмовляє з базою даних туди-сюди, а

потім передає необхідні дані контролеру. Примітка: модель ніколи не спілкувалася безпосередньо з видом.

Вид або представлення. Представлення даних здійснюється компонентом подання [22]. Представлення насправді генерує користувальницький інтерфейс або інтерфейс користувача для користувача. Тож у веб-додатках, коли ви думаєте про компонент подання, просто подумайте про Html / CSS. Перегляди створюються даними, які збирає компонент моделі, але ці дані беруться не безпосередньо, а через контролер, тому подання розмовляє лише з контролером.

Контролер. Він відомий як головна людина, оскільки контролер - це компонент, який забезпечує взаємозв'язок між Представленням та моделлю, тому він виступає посередником [23]. Контролеру не потрібно турбуватися про обробку логіки даних, він просто вказує моделі, що робити. Отримавши дані від моделі, вона обробляє їх, а потім бере всю ту інформацію, яку надсилає у подання, і пояснює, як представляти користувача. Примітка: Перегляди та моделі не можуть розмовляти безпосередньо.

Переваги MVC [24]:

- архітектура MVC відділить користувальницький інтерфейс від бізнес-логіки;
- компоненти багаторазові;
- легке обслуговування;
- різні компоненти програми в MVC можна незалежно розгортати та підтримувати;
- ця архітектура допомагає протестувати компоненти самостійно.

Недоліки MVC:

- висока складність архітектури;
- не підходить для невеликих застосувань.

Висновки до розділу 3

Даний розділ описує модель системи та її архітектуру. Надано схеми та діаграми структури системи, яка складається з різних розділів. Надано детальний опис необхідних даних системи для її функціонування. Описано структуру кожної таблиці та приведено приклади даних, що містяться в базі даних. Описано загальну структуру зв'язків таблиць в базі даних. Схематично наведено розподіл користувачів в залежності від їх прав доступу. Зображено у вигляді діаграм процеси, що відповідають за авторизацію та реєстрацію користувачів у системі.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА РОЗРОБКА ДОКУМЕНТАЦІЇ

4.1 Обґрунтування та вибір технологій розробки ПЗ

Для створення веб-системи «система первинної діагностики на базі нечіткої логіки» було обрано технології, які спрямовані на розробку веб-застосунків, що дають змогу створювати веб-сторінки з визначеним дизайном та функціоналом.

Для створення дизайну системи була обрана бібліотека Bootstrap, яка містить в собі необхідні компоненти для створення гарного та простого у використанні UI. Загалом весь дизайн побудований за допомогою даної бібліотеки. На малюнках рис. 4.1, рис. 4.2, рис. 4.3, можна побачити приклади компонентів бібліотеки та їх реалізацію у веб-додатку.

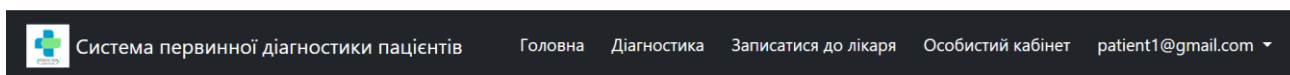
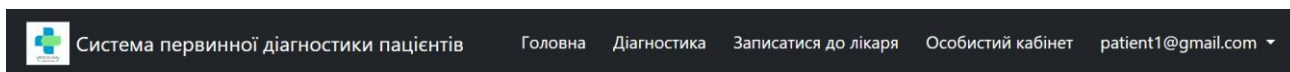


Рисунок 4.1 – Навігаційна панель системи



Контактна інформація лікарів

| Прізвище | Ім'я | По батькові | Професія | Номер телефону | Пошта | Адреса |
|-----------|-----------|-------------|---------------|----------------|-------------------|--|
| Микола | Коваль | Іванович | Хірург | +380978866351 | doctor1@gmail.com | м. Миколаїв, вул. Корабелів, 14 |
| Сергій | Шевченко | Іванович | Травматолог | +380999444882 | doctor2@gmail.com | 54020, м.Миколаїв, вул.Корабелів 14В |
| Марія | Олійник | Іванівна | Дієтолог | +380668284797 | doctor3@gmail.com | вул. Адмірала Макарова, 1, місто Миколаїв |
| Ольга | Кравченко | Іванівна | Кардіолог | +380668196672 | doctor4@gmail.com | 54058, м. Миколаїв, вул. Київська, 1 |
| Валентина | Мороз | Миколаївна | Психотерапевт | +380676318249 | doctor5@gmail.com | 54029, м. Миколаїв, вул. Рюміна, 5 |
| Тетяна | Ткаченко | Іванівна | Окуліст | +380935874948 | doctor6@gmail.com | 54000 м.Миколаїв, вул. Космонавтів, буд.97 |
| Анатолій | Шевченко | Іванович | Стоматолог | +380672998809 | doctor7@gmail.com | 54020, м.Миколаїв, вул.Корабелів 14В |
| Микола | Шевченко | Петрович | Стоматолог | +380961788739 | doctor8@gmail.com | 54000 м.Миколаїв, вул. Космонавтів, буд.97 |

Рисунок 4.2 – Таблиці з даними

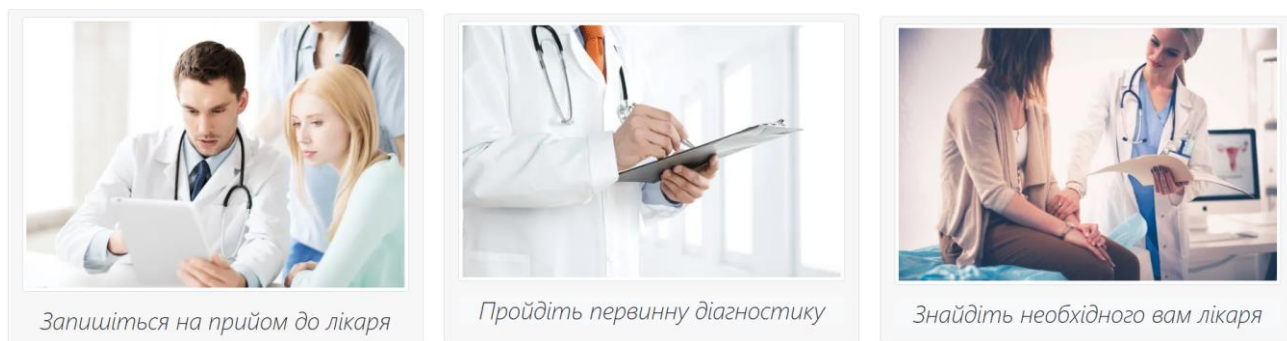


Рисунок 4.3 – Представлення інформації у вигляді карток

Значна частина проекту побудована за допомогою веб-фреймворку Laravel, який дозволяє вести швидку розробку веб-орієнтованих застосунків, має досить комфортну архітектуру та легкий у використанні. Laravel надає таку низку можливостей, як створення представлень, що відповідають за відображення змісту сторінок з їх визначеним дизайном, створення моделей які містять зв'язок із базою даних, та визначення контролерів, що взаємопов'язують модель із представленням. На рис. 4.3 можна побачити короткий приклад коду представлення, що оформлений у вигляді звичайної веб-сторінки.

```

107
108 <div class="container-fluid">
109 <div class="row mb-3">
110 <hr class="bg-info">
111 <h1 class="text-center p-3"><em>Пройдіть швидко та невідкладну діагностику</em></h1>
112 <hr class="bg-info">
113 </div>
114 <div class="row" style="display:flex; align-items:center">
115 <div class="col">
116 <div class="card" style="">
117 <div class="card-header">
118 
119 <h4 class="text-center fs-3 mt-3 display-6 mb-2"><em>Запишіться на прийом до лікаря</em></h4>
120 </div>
121 </div>
122 </div>
123 <div class="col">
124 <div class="card">
125 <div class="card-header">
126 
127 <h4 class="text-center fs-3 mt-3 display-6 bg-light mb-3"><em>Пройдіть первинну діагностику</em></h4>
128 </div>
129 </div>
130 </div>
131 <div class="col">
132 <div class="card">
133 <div class="card-header">
134 
135 <h4 class="text-center fs-3 mt-3 display-6 bg-light mb-3"><em>Знайдіть необхідного вам лікаря</em></h4>
136 </div>
137 </div>

```

Рисунок 4.4 – Приклад коду представлення

На малюку рис. 4.4 можна побачити приклад створення контролера у вигляді фрагменту коду.

```
if(date('l',strtotime("+". $i . "day"))== "Saturday") {
    $i += 2;
} elseif(date('l',strtotime("+". $i . "day"))== "Sunday") {
    $i +=1;
}
$days[$i] = date('d-m-y',strtotime("+". $i . "day"));
}

for ($i=0; $i < 7; $i++) {
    $time[$i] = date('H:i',mktime(9+$i, 0));
}

return view('visit.create',['user'=>$request->input('user'),'doctor_id'=>$request->input('doctor_id'),'days' => $days,'time'=>$time]);
}

public function create(Request $request) {
    $patient = Patient::where('patient_id',$request->input('user'))->first();
    $doctor = Doctor::where('id',$request->input('doctor_id'))->first();

    if(Visit::where(['patient_id',$patient->id],['doctor_id',$doctor->id])>count() == 1) {
        $message = "Ви вже записані на прийом до цього лікаря!";
    } else {
        $doctor->visitDoctor()->create([
            'doctor_id'=> $doctor->id,
            'time' => $request->input('time'),
            'day' => $request->input('day')
        ]);
    }
}
```

Рисунок 4.5 – Короткий фрагмент коду контролера

Зазвичай у контролері знаходиться вся логіка веб-додатку, де приймаються дані надіслані користувачем з виду, потім проходять валідацію або через алгоритм, після чого вносяться відповідні зміни у базу даних та повертається результат у вид, згідно внесених змін у БД. Наприклад на рис. 4.4 зображено медот «index()», що відповідає за відображення всіх лікарів до яких можна записатися користувачу при переході в розділі навігації на вкладку «Записатися до лікаря».

На малюнку рис. 4.5 можна побачити приклад моделі, що пов'язана з певною таблицею в базі даних та відповідає за всі процеси, що проходять між контролером та цією таблицею.

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Visit extends Model
{
    protected $fillable = [
        'time', 'day'
    ];

    public function visitDoctor() {
        return $this->belongsTo(
            Doctor::class,
            'doctor_id',
            'id'
        );
    }

    public function visitPatient() {
        return $this->belongsTo(
            Patient::class,
            'patient_id',
            'id'
        );
    }
}
```

Рисунок 4.5 – Фрагмент коду моделі, яка пов’язана з таблицею «visits»

Основною мовою програмування, що використана при розробці системи є PHP. PHP – це серверна мова, яка дозволяє писати код, що потім може виконуватися на стороні серверу, вона є досить проста в освоєнні та має потужний функціонал для вирішення задач поставлених при розробці технічного завдання.

За створення, таблиць в базі даних відповідає механізм міграцій, що надає фреймворк Laravel. Для того, щоб зробити запис або створити таблицю в базі даних зазвичай пишуть запити мовою SQL, а механізм міграцій дозволяє простим чином в коді створювати таблиці та за допомогою моделі вносити дані в базу даних без прописування відповідних SQL-команд.

На рис. 4.6 можна побачити фрагмент коду, що відповідає за створення таблиці в базі даних.

```
class CreatePatientsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('patients', function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->unsignedBigInteger('patient_id')->nullable();
            $table->string('name',25);
            $table->string('l_name',25);
            $table->string('patronymic',25);
            $table->string('phone_num',15);
            $table->foreign('patient_id')->references('id')->on('users');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('patients');
    }
}
```

Рисунок 4.6 – Створення таблиці «patients»

Малюнок рис. 4.6 зображує два методи, які відповідають за створення та видалення таблиці в БД, перший метод описує структуру майбутньої таблиці, а другий параметри видалення таблиці.

4.2 Програмні засоби для розробки ПЗ

Для створення системи з клієнт-серверною архітектурою необхідна наявність серверу для повної функціональності веб-додатку. Тому для розгортання проекту на сервері було обрано сервер «OpenServe». Даний сервер є

локальним, який встановлюється на персональний комп'ютер користувача. Даний програмний пакет містить в собі інструменти для розробки архітектури бази даних, консоль розробника, різні версії PHP, та можливість налаштування всіх функцій, що пов'язані з сервером.

На малюнку 4.7 можна побачити основні можливості, що надає «OpenServer» з контекстного меню.

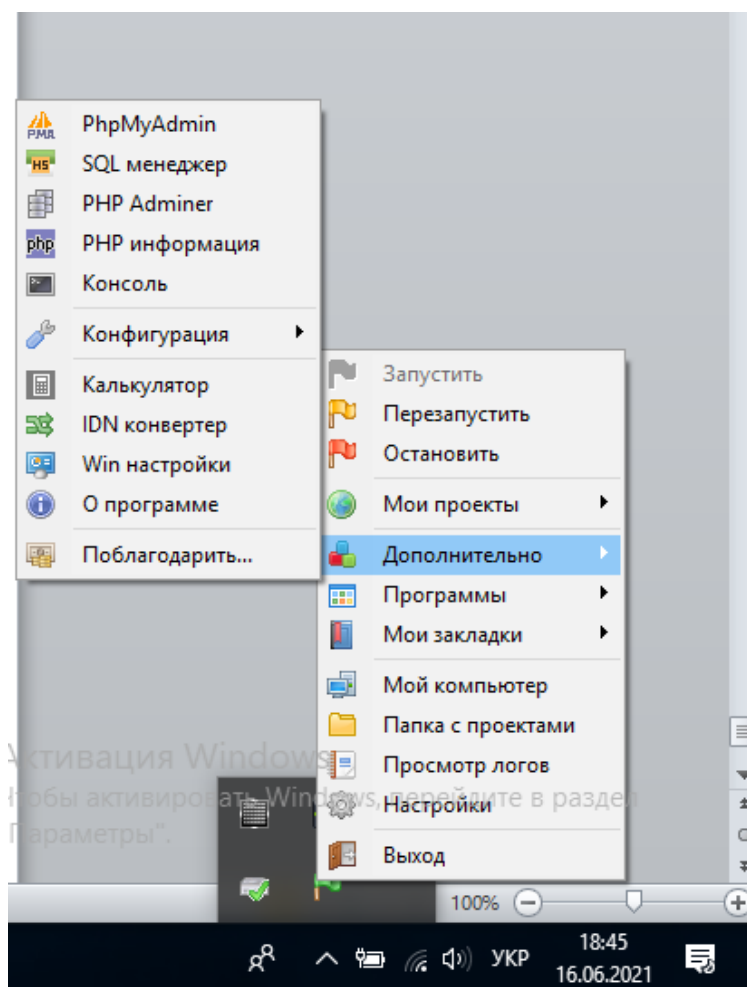


Рисунок 4.7 – Контекстне меню «OpenServer»

На малюнку рис. 4.8 можна побачити веб-інтерфейс для розробки баз даних, який йде в комплекті з «OpenServer» та має назву «phpMyAdmin».

Кафедра інтелектуальних інформаційних систем
Система первинної медичної діагностики на основі нечіткої логіки

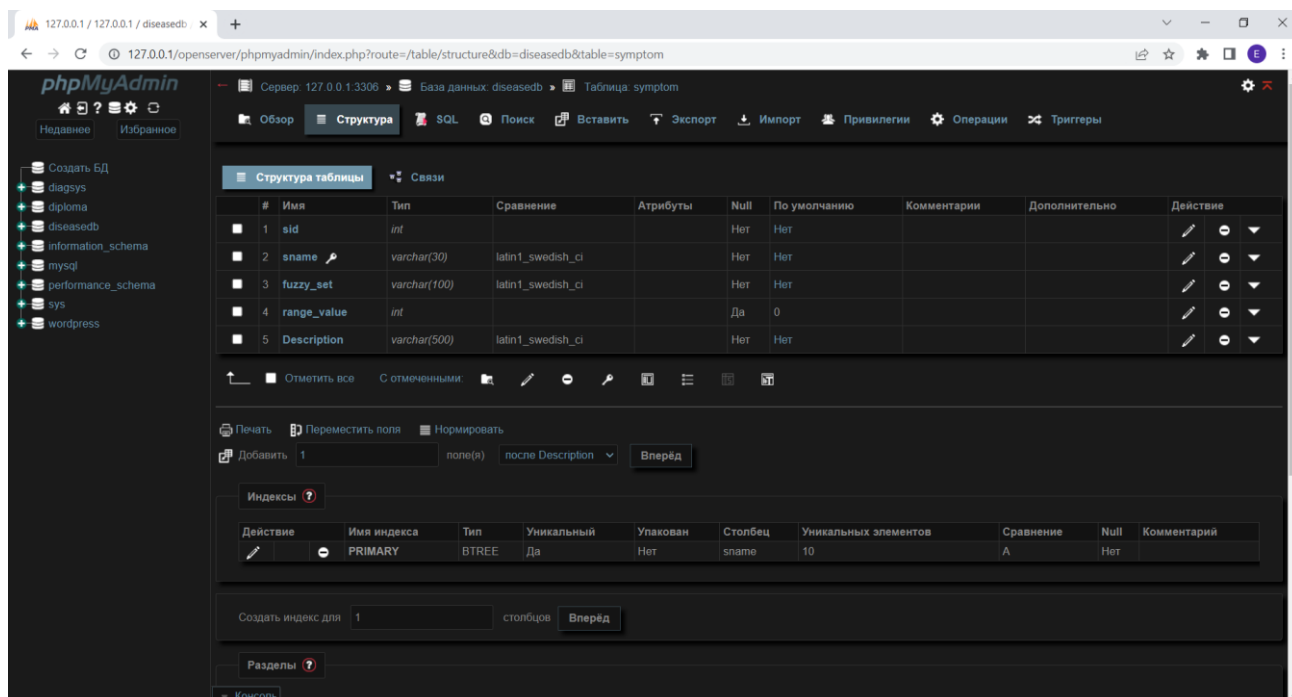


Рисунок 4.8 – Интерфейс адміністрування бази даних

Для написання коду було обране програмне забезпечення «VSCode». На малюнку рис. 4.9 можна побачити головне вікно ПЗ.

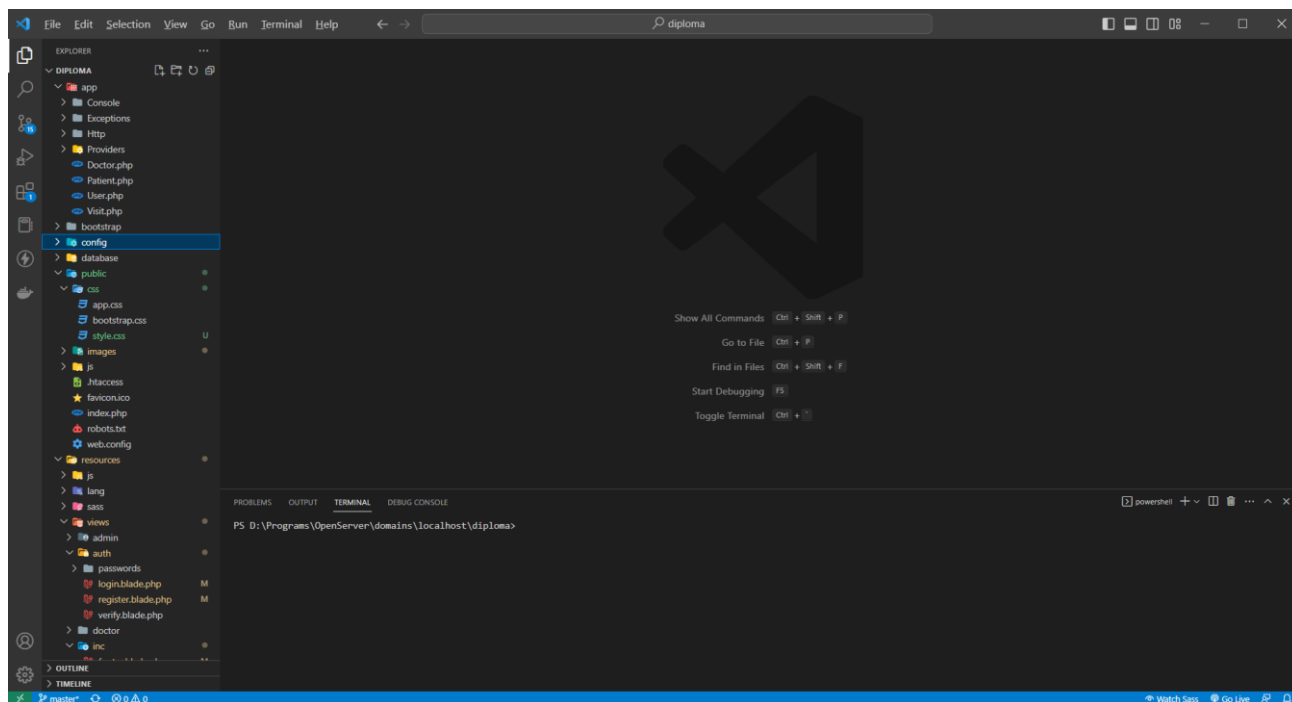


Рисунок 4.9 – Інтерфейс ПЗ «VSCode»

«VSCode» - текстовий редактор, що спрямований на розробку додатків різних напрямків, має великий спектр розширень, які можна завантажити та налаштувати для комфортнішого написання коду. Даний редактор є безкоштовним та легким у використанні. Більшість особливостей аналогічні іншим схожим редакторам для написання коду (Atom, brackets). На рис. 4.10 можна побачити основні особливості ПЗ «VSCode».

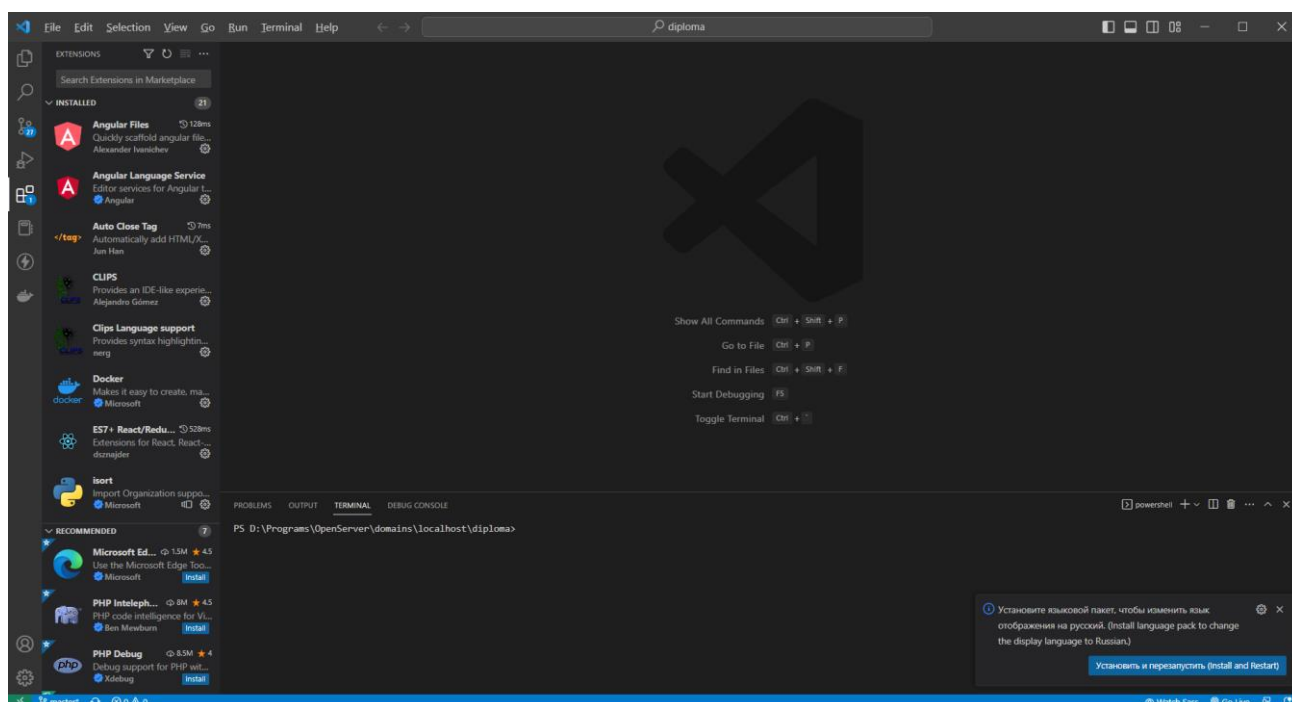


Рисунок 4.10 – Особливості ПЗ «VSCode»

Із особливостей даного текстового редактора можна виділити виділення синтаксису різними кольорами для читабельності коду, ієрархічна структура файлів та папок проекту, набір гарячих клавіш для швидшого написання коду, зручний механізм пошуку розширень.

Також було використане ПЗ, що має назву GitHub. Дане ПЗ було використане для підтримки проекту. Система Git дозволяє відстежувати зміни, що були внесені в код з початку розробки веб-додатку. На рис. 4.12 можна побачити інтерфейс системи GitHub.

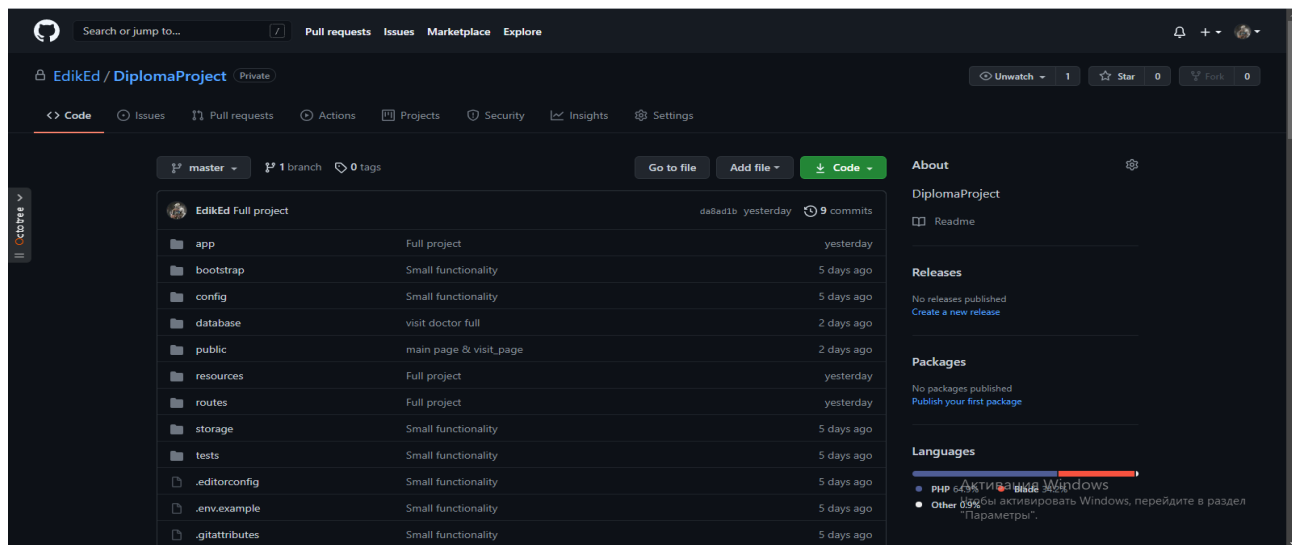


Рисунок 4.12 – Інтерфейс системи GitHub

4.3 Опис програмної реалізації

Перше, що побачить користувач при переході за посиланням, яке веде до даної системи – це стартова сторінка, яка визначена представленням «welcome.blade.php». На рис. 4.13 можна побачити інтерфейс початкової сторінки системи.

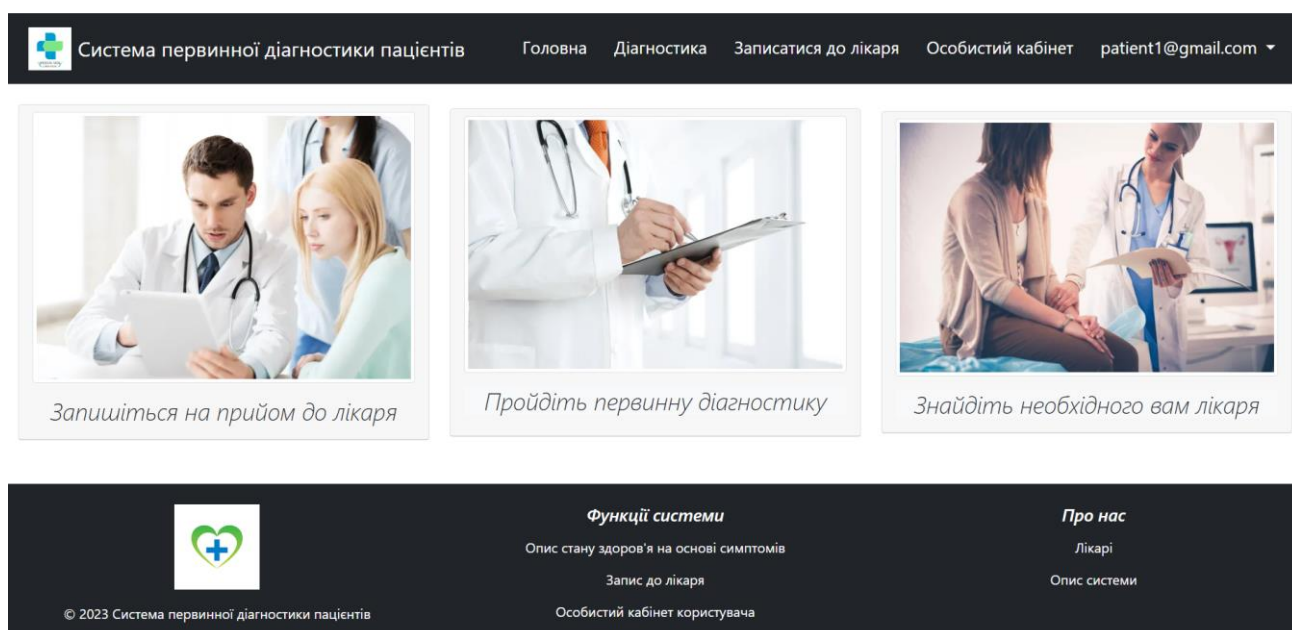


Рисунок 4.13 – Вигляд початкової сторінки системи

Початкова сторінка дає коротку інформацію про систему та її функціонал, а також можливість входу зареєстрованого користувача до свого облікового запису або можливість зареєструватися для нових користувачів системи. Laravel – містить шаблонізатор, який допомагає створювати представлення без надлишковості коду. Інакше кажучи, створивши один раз шаблон, всі подальші представлення, що мають схожу структуру можуть наслідувати її від основного представлення-шаблону.

Механізм маршрутизації відповідає за навігацію користувача по сайту та виведення відповідних представлень. На рис. 4.14 зображено файл з вмістом маршрутів системи.

```
4 Route::get('/', function () {
5     return view('welcome');
6 });
7
8 Auth::routes();
9
10 // Special routes
11 Route::get('/home', 'HomeController@index')->name('home');
12
13 Route::get('/about', 'PageController@about')->name('about');
14
15 Route::get('/visit', 'VisitController@index')->name('visit');
16
17 Route::get('/visit/chooseTime', 'VisitController@chooseTime')->name('visit.time');
18
19 Route::post('/visit/create', 'VisitController@create')->name('visit.create');
20
21 Route::post('/visit/find', 'VisitController@show')->name('visit.find');
22
23
24 //Patients
25 Route::get('/patient/cabinet', 'PatientController@index')->name('patient.cabinet');
26
27 Route::post('/patient/cabinet', 'PatientController@create')->name('patient.create.cabinet');
28
29 Route::get('/patient/visit/destroy', 'PatientController@destroy')->name('patient.destroy');
30
31 // Doctors
32 Route::get('/doctors', 'DoctorsController@index')->name('doctors');
33
34 Route::get('/doctor/cabinet', 'DoctorsController@doctorCab')->name ('doctor.cabinet');
35
36 Route::get('/doctor/destroy/visit', 'DoctorsController@destroy')->name ('doctor.cabinet.destroy');
```

Рисунок 4.14 – Маршрути системи

На малюнках рис. 4.15, рис. 4.16, рис. 4.17, зображено основні структурні розділи користувача-пацієнта, що доступні після авторизації в системі. Всі розділи інтерфейсу представлені відповідними файлами представлення як наприклад «showCabinet.blade.php», «index.blade.php».

Система первинної діагностики пацієнтів Головна Діагностика Записатися до лікаря Особистий кабінет new@patient ▾

Дані користувача/пацієнта

Ім'я
Введіть ім'я

По батькові
Введіть ім'я по батькові

Прізвище
Введіть прізвище

Номер телефону
Введіть номер телефону

[Створити кабінет](#)


Функції системи Про нас
Опис стану здоров'я на основі симптомів Лікарі

Рисунок 4.15 – Створення особистого кабінету для нового користувача

На рис. 4.15 зображена форма для отримання даних від користувача, який щойно зареєструвався в системі. Як видно із зображення дана форма складається із текстових полів та кнопки для відправки отриманих даних.

Система первинної діагностики пацієнтів Головна Діагностика Записатися до лікаря Особистий кабінет patient1@gmail.com ▾

Дані пацієнта



Прізвище: Коваленко
Ім'я: Микола
По батькові: Миколайович
Телефон: +380682608833

Записи на прийом до лікаря

| Ім'я | По батькові | Спеціальність | Час | Дата | Адреса | |
|-----------|-------------|---------------|-------|----------|--------------------------------------|---------------------------|
| Сергій | Іванович | Травматолог | 12:00 | 21-06-21 | 54020, м.Миколаїв, вул.Корабелів 14В | Скасувати |
| Валентина | Миколаївна | Психотерапевт | 13:00 | 18-06-21 | 54029, м. Миколаїв, вул. Рюміна, 5 | Скасувати |
| Микола | Іванович | Хірург | 13:00 | 21-06-21 | м. Миколаїв, вул. Корабелів, 14 | Скасувати |

Функції системи Про нас

Рисунок 4.16 – Вигляд особистого кабінету користувача з записами

Даний розділ має такі структурні компоненти як, карткове представлення інформації пацієнта та таблиця всіх записів до лікарів. В таблиці містяться кнопки що реалізують функціонал скасування запису. Всі дані які виведені у представленні взяті із бази даних.

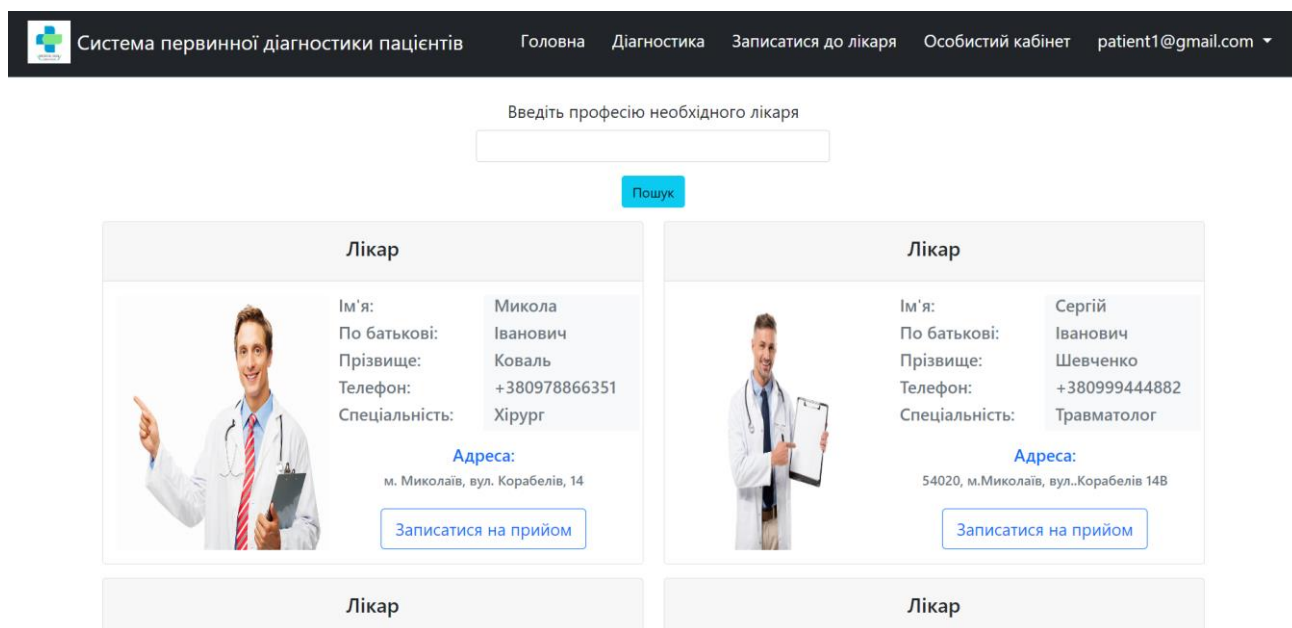


Рисунок 4.17 – Розділ, що відповідає за запис пацієнта до лікаря

На малюнку рис. 4.17 зображено інтерфейс, що відповідає за пошук необхідних лікарів та запис пацієнтів на прийом. Даний інтерфейс реалізований у вигляді представлення інформації про лікарів в картках. Під блоком навігації сайту створене поле для пошуку необхідного лікаря за ім'ям. Кнопка, що відповідає за запис до лікаря перенаправляє користувача до сторінки з вибором часу та дати прийому. На малюнку рис. 4.18 можна побачити результат дії пошуку.


Система первинної діагностики пацієнтів Головна Діагностика Записатися до лікаря Особистий кабінет patient1@gmail.com

Введіть професію необхідного лікаря

Травматолог

Пошук

Лікар



Ім'я: Сергій
По батькові: Іванович
Прізвище: Шевченко
Телефон: +380999444882
Спеціальність: Травматолог

Адреса:
54020, м.Миколаїв, вул..Корабелів 14В

Записатися на прийом

Рисунок 4.18 – Пошук необхідного лікаря

На рис. 4.19 можна побачити меню вибору дати та часу для запису на прийом до лікаря.


Система первинної діагностики пацієнтів Головна Діагностика Записатися до лікаря Особистий кабінет patient1@gmail.com

Оберіть час прийому на найближчий тиждень

Дата: 17-02-23 Дата: 20-02-23 Дата: 21-02-23 Дата: 22-02-23 Дата: 23-02-23

09:00 09:00 09:00 09:00 09:00
10:00 10:00 10:00 10:00 10:00
11:00 11:00 11:00 11:00 11:00
12:00 12:00 12:00 12:00 12:00
13:00 13:00 13:00 13:00 13:00
14:00 14:00 14:00 14:00 14:00
15:00 15:00 15:00 15:00 15:00

Записатися на прийом до лікаря

 **Функції системи**
Опис стану здоров'я на основі симптомів
Запис до лікаря

Про нас
Лікарі
Опис системи

Рисунок 4.19 – Вибір часу та дати для запису на найближчий тиждень

Малюнок рис. 4.19 зображує досить простий механізм запису до лікаря на найближчий тиждень. Користувач системи, що авторизований як пацієнт, має можливість обрати вільний час запису. Якщо той час на який пацієнт бажав записатися відсутній у списку то це говорить про те, що дехто із користувачів системи вже зробив запис на цей час. На малюнку рис. 4.19 можна побачити відсутність деяких часів прийому. Малюнок рис. 4.20 зображує розділ де знаходяться основні етапи проходження діагностування.

The screenshot displays a web application interface for a primary patient diagnostic system. The top navigation bar includes the system name, a home button, and links for 'Діагностика', 'Записатися до лікаря', 'Особистий кабінет', and a user email 'patient1@gmail.com'. The main content area is divided into two columns:

- Left Column: 'Зазначте ваші симптоми (Для діагностики)'**
 - Category: **Грип, Простуда, ОРЗ, Запалення легенів**
 - Items:
 - лихоманка (Діапазон:[1.0,3.0] з кроком 1.0)
 - біль у горлі (Діапазон:[1.0,0.5] з кроком 1.0)
 - кашель (Діапазон:[1.0,3.0] з кроком 1.0)
 - нежить (Діапазон:[1.0,3.0] з кроком 1.0)
 - головний біль (Діапазон:[1.0,4.0] з кроком 1.0)
 - м'язовий біль (Діапазон:[5.0,20.0] з кроком 1.0)
 - слабкість (Діапазон:[1.0,3.0] з кроком 1.0)
 - біль у грудях (Діапазон:[1.0,4.0] з кроком 1.0)
 - Нудота (Діапазон:[5.0,20.0] з кроком 1.0)
- Right Column: 'Зазначте ваші симптоми (Для отримання рекомендації)'**
 - Category: **Стоматолог**
 - Реакція на гаряче, солодке, холодне
 - Зубний біль
 - Кровоточивість ясен
 - Category: **Окуліст (Офтальмолог)**
 - Короткозорість
 - Далекозорість
 - Біль в очах
 - Category: **Психотерапевт**
 - Вас все дратує
 - Ви занадто багато працюєте та втомлюєтесь
 - Багато ризикових вчинків та нервових випадків
 - Category: **Кардіолог**
 - Біль у грудній клітці
 - Задишка та раптові напади задухи ночами
 - Відчуття перебоїв у роботі серця
 - Category: **Дієтолог**
 - Нестача або надлишок ваги
 - Постійне відчуття голоду
 - Різка зміна харчових уподобань

Рисунок 4.20 – Діагностування

Система також передбачає окремий функціонал для всіх лікарів системи. Кожний лікар системи має свій особистий кабінет з переліком зазначених записів на прийом. На рис. 4.21 можна побачити особистий кабінет лікаря із записами пацієнтів на прийом до нього. Також можна переглянути історію пацієнтів, а саме результати діагностування, та перелік скарг та симптомів.

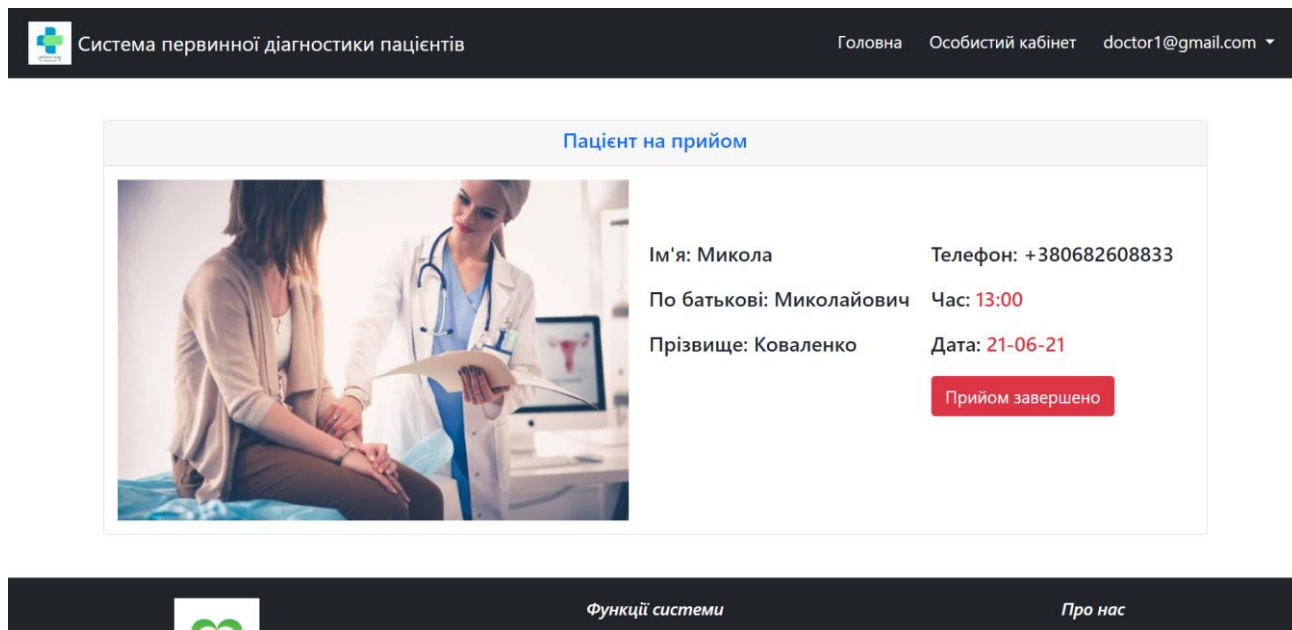


Рисунок 4.21 – Особистий кабінет лікаря

Всі дані які є в особистому кабінеті лікаря беруться із таблиці «visits». рис. 4.21 зображує карткову інформацію, яка відображує дані про прийом та кнопку що відповідає за завершення прийом та видаленням запису про прийом в БД. Відповідно прав до доступу користувачів можна побачити зміни у блоці навігації.

4.4 Керівництво користувача

Розглянемо три окремих керівництва для користувачів залежно від прав доступу до системи. Дані керівництва передбачені для користувачів системи із правами користувача, із правами лікаря та адміністратора сайту.

Процес авторизації для всіх користувачів однаковий. В залежності від результату авторизації система сама надає необхідний функціонал користувачу.

Сторінка входу до системи зображена на рис. 4.22.

The screenshot shows the login interface of a web application. At the top, there is a dark header with the application logo and name 'Система первинної діагностики пацієнтів' on the left, and 'Вхід' and 'Реєстрація' on the right. The main content area is titled 'Вхід до системи' and contains a login form with fields for 'Пошта E-mail' (containing 'edikergart2017@gmail.com') and 'Пароль' (masked with dots). Below the password field is a checkbox for 'Запам'ятати' and a blue 'Вхід' button. A link 'Забули пароль?' is also present. At the bottom, there is a dark footer with a logo on the left, a central menu titled 'Функції системи' with links for 'Опис стану здоров'я на основі симптомів', 'Запис до лікаря', and 'Особистий кабінет користувача', and a right-side menu titled 'Про нас' with links for 'Лікарі' and 'Опис системи'. Copyright information '© 2023 Система первинної діагностики пацієнтів' is visible in the bottom left.

Рисунок 4.22 – Сторінка авторизації користувачів

Після авторизації користувач потрапляє на головну сторінку де може побачити коротку інструкцію користування системою. На малюнку рис. 4.23 зображено головну сторінку.

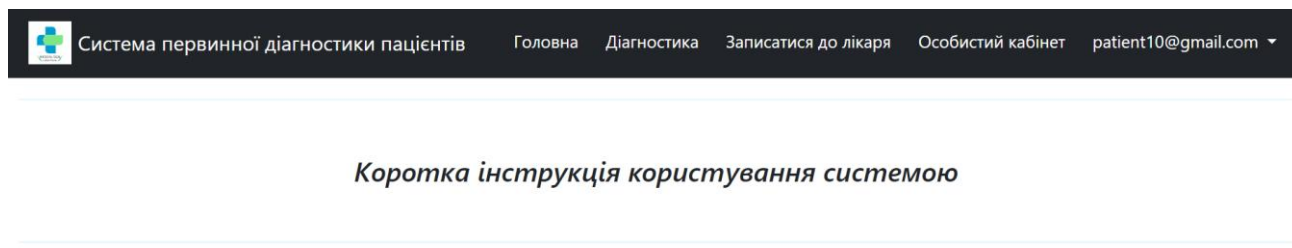


Рисунок 4.23 – Головна сторінка системи

4.4.1 Керівництво для користувача з правами доступу «user»

Після авторизації створюємо особистий кабінет рис. 4.24.

Кафедра інтелектуальних інформаційних систем
Система первинної медичної діагностики на основі нечіткої логіки

Система первинної діагностики пацієнтів Головна Діагностика Записатися до лікаря Особистий кабінет patient10@gmail.com

Дані користувача/пацієнта

Ім'я
Едуард

По батькові
Володимирович

Прізвище
Ергарт

Номер телефону
+421950442867

[Створити кабінет](#)

Функції системи Про нас
Опис стану здоров'я на основі симптомів Лікарі

Рисунок 4.24 – Створення особистого кабінету

Перегляд особистого кабінету рис. 4.25.

Система первинної діагностики пацієнтів Головна Діагностика Записатися до лікаря Особистий кабінет patient1@gmail.com

Дані пацієнта

Прізвище: Коваленко
Ім'я: Микола
По батькові: Миколайович
Телефон: +380682608833

Записи на прийом до лікаря

| Ім'я | По батькові | Спеціальність | Час | Дата | Адреса | |
|-----------|-------------|---------------|-------|----------|--------------------------------------|---------------------------|
| Сергій | Іванович | Травматолог | 12:00 | 21-06-21 | 54020, м.Миколаїв, вул.Корабелів 14В | Скасувати |
| Валентина | Миколаївна | Психотерапевт | 13:00 | 18-06-21 | 54029, м. Миколаїв, вул. Рюміна, 5 | Скасувати |
| Микола | Іванович | Хірург | 13:00 | 21-06-21 | м. Миколаїв, вул. Корабелів, 14 | Скасувати |

Функції системи Про нас

Рисунок 4.25 – Особистий кабінет

Проходження діагностики рис. 4.26 та рис. 4.27.

Кафедра інтелектуальних інформаційних систем
Система первинної медичної діагностики на основі нечіткої логіки

Система первинної діагностики пацієнтів
Головна
Діагностика
Записатися до лікаря
Особистий кабінет
patient1@gmail.com ▾

Зазначте ваші симптоми (Для діагностики)

Грип, Простуда, ОРЗ, Запалення легенів

лихоманка
(Діапазон:[1.0,3.0] з кроком 1.0)

біль у горлі
3

кашель
2

нежить
2

головний біль
3

м'язовий біль
(Діапазон:[5.0,20.0] з кроком 1.0)

слабкість
2

біль у грудях
(Діапазон:[1.0,4.0] з кроком 1.0)

Нудота

Зазначте ваші симптоми (Для отримання рекомендації)

Стоматолог

Реакція на гаряче, солодке, холодне

Зубний біль

Кровоточивість ясен

Окуліст (Офтальмолог)

Короткозорість

Далекозорість

Біль в очах

Психотерапевт

Вас все дратує

Ви занадто багато працюєте та втомлюєтесь

Багато ризикових вчинків та нервових випадків

Кардіолог

Біль у грудній клітці

Задишка та раптові напади задухи ночами

Відчуття перебоїв у роботі серця

Дієтолог

Нестача або надлишок ваги

Постійне відчуття голоду

Грип : 0.6454010301692421; ОРЗ : 0.6454010301692421; Запалення легенів : 0.5249433106575964; Ангіна : 0.4750566893424036;

Рисунок 4.26 – Результати проходження діагностування

Система первинної діагностики пацієнтів
Головна
Діагностика
Записатися до лікаря
Особистий кабінет
patient1@gmail.com ▾

Введіть професію необхідного лікаря

Пошук

Лікар

Ім'я: Микола

По батькові: Іванович

Прізвище: Коваль

Телефон: +380978866351

Спеціальність: Хірург

Адреса:

м. Миколаїв, вул. Корабелів, 14

Записатися на прийом

Лікар

Ім'я: Сергій

По батькові: Іванович

Прізвище: Шевченко

Телефон: +380999444882

Спеціальність: Травматолог

Адреса:

54020, м.Миколаїв, вул.Корабелів 14В

Записатися на прийом

Лікар

Лікар

Рисунок 4.27 – Запис на прийом за рекомендацією

Запис на прийом та його результати рис. 4.28 та рис. 4.29.

Система первинної діагностики пацієнтів Головна Діагностика Записатися до лікаря Особистий кабінет patient1@gmail.com

Оберіть час прийому на найближчий тиждень

○ Дата: 17-02-23 ○ Дата: 20-02-23 ● Дата: 21-02-23 ○ Дата: 22-02-23 ○ Дата: 23-02-23

○ 09:00 ○ 09:00 ○ 09:00 ○ 09:00 ○ 09:00

○ 10:00 ○ 10:00 ○ 10:00 ○ 10:00 ○ 10:00

○ 11:00 ○ 11:00 ● 11:00 ○ 11:00 ○ 11:00

○ 12:00 ○ 12:00 ○ 12:00 ○ 12:00 ○ 12:00

○ 13:00 ○ 13:00 ○ 13:00 ○ 13:00 ○ 13:00

○ 14:00 ○ 14:00 ○ 14:00 ○ 14:00 ○ 14:00

○ 15:00 ○ 15:00 ○ 15:00 ○ 15:00 ○ 15:00

Записатися на прийом до лікаря

Функції системи Про нас

Опис стану здоров'я на основі симптомів Лікаря

Запис до лікаря Опис системи

Рисунок 4.28 – Запис на прийом

Система первинної діагностики пацієнтів Головна Діагностика Записатися до лікаря Особистий кабінет patient1@gmail.com

Дані пацієнта

Прізвище: Коваленко

Ім'я: Микола

По батькові: Миколайович

Телефон: +380682608833

Записи на прийом до лікаря

| Ім'я | По батькові | Спеціальність | Час | Дата | Адреса | |
|-----------|-------------|---------------|-------|----------|--------------------------------------|-----------|
| Сергій | Іванович | Травматолог | 12:00 | 21-06-21 | 54020, м.Миколаїв, вул.Корабелів 14В | Скасувати |
| Валентина | Миколаївна | Психотерапевт | 13:00 | 18-06-21 | 54029, м. Миколаїв, вул. Рюміна, 5 | Скасувати |
| Микола | Іванович | Хірург | 13:00 | 21-06-21 | м. Миколаїв, вул. Корабелів, 14 | Скасувати |

Функції системи Про нас

Рисунок 4.29 – Результат запису на прийом

4.4.2 Керівництво для користувача з правами доступу «doctor»

Після авторизації перехід в особистий кабінет рис. 4.30

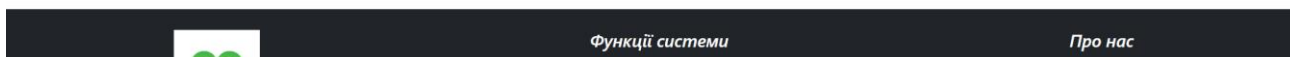
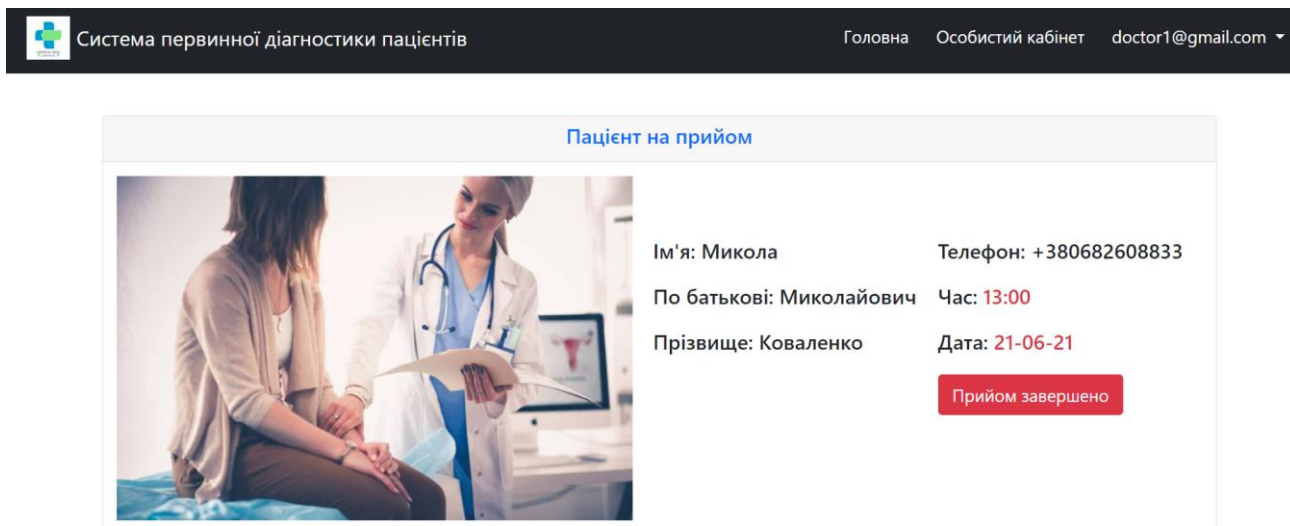


Рисунок 4.30 – Записаний прийом від користувача

4.4.3 Керівництво для користувача з правами доступу «admin»

Основні можливості адміністратора рис. 4.31

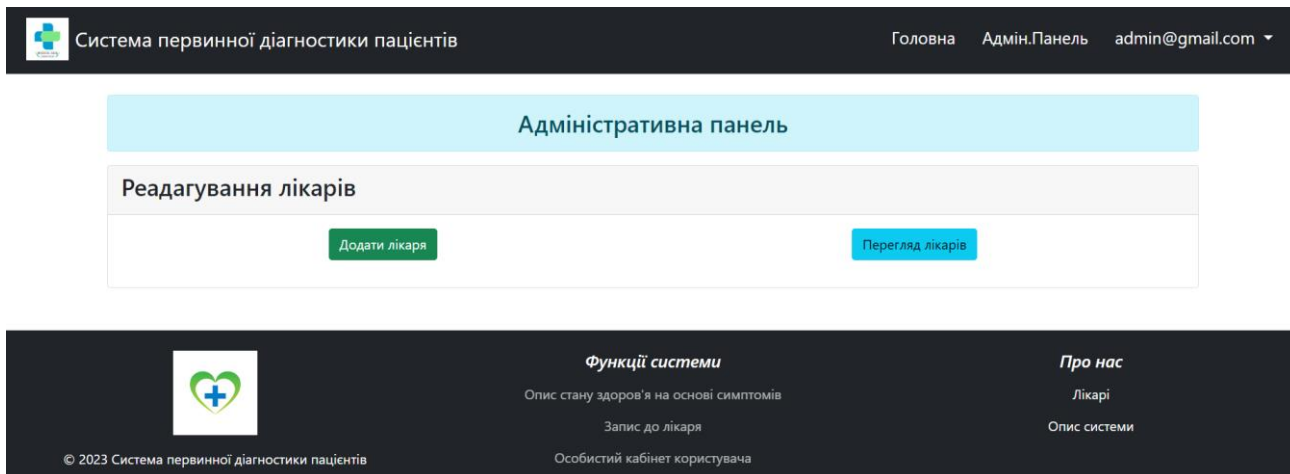


Рисунок 4.31 – Можливості адміністратора

Додавання лікаря до системи рис. 4.32.

The screenshot shows two forms for adding a new doctor. The first form, titled 'Дані для входу до облікового запису' (Login data), contains two input fields: 'Пошта E-mail' (E-mail address) and 'Пароль' (Password). The second form, titled 'Дані лікаря' (Doctor data), contains four input fields: 'Ім'я' (Name), 'По батькові' (Patronymic), 'Прізвище' (Surname), and 'Професія' (Profession).

Рисунок 4.32 – Додавання нових лікарів

Перегляд даних лікарів та видалення облікових записів лікарів рис. 4.33

The screenshot shows a table titled 'Контактна інформація лікарів' (Doctor contact information). The table has seven columns: 'Прізвище' (Surname), 'Ім'я' (Name), 'По батькові' (Patronymic), 'Професія' (Profession), 'Номер телефону' (Phone number), 'Пошта' (E-mail), and 'Адреса' (Address). Each row represents a doctor, and there is a red 'Видалити лікаря' (Delete doctor) button at the end of each row.

| Прізвище | Ім'я | По батькові | Професія | Номер телефону | Пошта | Адреса | |
|-----------|-----------|-------------|---------------|----------------|-------------------|--|-----------------|
| Микола | Коваль | Іванович | Хірург | +380978866351 | doctor1@gmail.com | м. Миколаїв, вул. Корабелів, 14 | Видалити лікаря |
| Сергій | Шевченко | Іванович | Травматолог | +380999444882 | doctor2@gmail.com | 54020, м.Миколаїв, вул.Корабелів 14В | Видалити лікаря |
| Марія | Олійник | Іванівна | Дієтолог | +380668284797 | doctor3@gmail.com | вул. Адмірала Макарова, 1, місто Миколаїв | Видалити лікаря |
| Ольга | Кравченко | Іванівна | Кардіолог | +380668196672 | doctor4@gmail.com | 54058, м. Миколаїв, вул. Київська, 1 | Видалити лікаря |
| Валентина | Мороз | Миколаївна | Психотерапевт | +380676318249 | doctor5@gmail.com | 54029, м. Миколаїв, вул. Рюміна, 5 | Видалити лікаря |
| Тетяна | Ткаченко | Іванівна | Окуліст | +380935874948 | doctor6@gmail.com | 54000 м.Миколаїв, вул. Космонавтів, буд.97 | Видалити лікаря |
| Анатолій | Шевченко | Іванович | Стоматолог | +380672998809 | doctor7@gmail.com | 54020, м.Миколаїв, вул.Корабелів 14В | Видалити лікаря |
| Микола | Шевченко | Петрович | Стоматолог | +380961788739 | doctor8@gmail.com | 54000 м.Миколаїв, вул. Космонавтів, буд.97 | Видалити лікаря |

Рисунок 4.33 – Видалення особистого запису лікарів та перегляд всіх лікарів

Висновки до розділу 4

В даному розділі було обґрунтовано вибір технологій для розробки веб-системи та вибір необхідного програмно забезпечення для більш комфортної

розробки проекту. Детально описано основні функціональні можливості системи, наведено приклади та фрагменти коду, що відповідають за різні функціональні можливості системи. Наведено приклади інтерфейсу різних розділів системи та їх функціональну складову. Детально описано керівництва для користувачів відповідно до їх прав доступу. Надано детальний опис компонентів форм та основних можливостей системи.

ВИСНОВКИ

В ході виконання роботи було досягнуто підвищення ефективності процесу діагностування ймовірних захворювань пацієнтів за рахунок побудови діагностичної системи, що містить актуальну базу знань взаємозв'язків між симптомами і захворюваннями.

Проаналізовано основні аспекти діагностування пацієнтів в реальному часі та дистанційно. Проаналізовано останні публікації та аналоги, які мають відношення до теми МКР. Визначено чіткий план розробки системи та основні задачі, що стосуються її розробки.

Сформульовано список технологій, які найбільш ефективно підходять для розробки системи первинної діагностики на основі нечіткої логіки. Досліджено можливі моделі системи та її компоненти з теоретичної точки зору. Проаналізовано основні компоненти нечітких експертних систем та підходи реалізації систем медичної діагностики на основі нечіткої логіки. Проаналізовано та розроблено основні критерій та вимоги до розроблюваного ПЗ відповідно до технічного завдання.

Створено систему первинної медичної діагностики на основі нечіткої логіки за допомогою обраних технологій та згідно поставленого технічного завдання.

Отже згідно з поставленим завданням та метою виконано усі пункти. Проведений детальний аналіз усіх публікацій, що стосуються теми МКР, виділено найбільш вагомі та ефективні технології для побудови системи первинної діагностики. Проаналізовані можливі моделі побудови веб-додатку з описаним функціоналом. Розроблено структуру зв'язків даних для системи первинної медичної діагностики на основі нечіткої логіки.

З огляду на проведені дослідження основної мети та функцій охорони праці, можна зробити висновок, що основною метою охорони праці є забезпечення безпечних та здорових умов праці для працівників на робочому місці.

Основні функції охорони праці полягають у забезпеченні безпечних умов праці, запобіганні травмам та професійним захворюванням, а також у виявленні та вирішенні проблем, що можуть негативно впливати на здоров'я працівників. Інші важливі функції охорони праці включають визначення та оцінку ризиків, пов'язаних з виконанням роботи, розробку та впровадження програм попередження нещасних випадків на роботі та профілактики захворювань, а також навчання та підготовку працівників з питань охорони праці.

У підсумку, розділ охорони праці є важливим компонентом будь-якої діяльності, пов'язаної з працею, оскільки він допомагає забезпечити безпеку та здоров'я працівників на робочому місці, а також знижує ризик нещасних випадків на роботі та професійних захворювань.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is REST? [Електронний ресурс]. — Режим доступу: <https://inlnk.ru/bBzyQ>. (дата звернення: 30.05.2021)
2. Markus Egger — MVVM Survival Guide for Enterprise Architectures in Silverlight and WPF [Електронний ресурс]. — 2012. — Режим доступу: <https://www.packtpub.com/application-development/mvvm-survival-guide-enterprise-architectures-silverlight-and-wpf>. (дата звернення 20.05.2021)
3. Кейт Джонс. DOM Scripting: Web Design with JavaScript and the Document Object Model. / К. Джонс — Перше, 2005. — 368 с.
4. Anthony Gore — Full-Stack Vue.js 2 and Laravel 5 [Електронний ресурс]. — 2015. — Режим доступу: <https://bit.ly/2OEODzR>. (дата звернення: 24.04.2021)
5. Martin Fowler — GUI Architectures. Часть 1 [Електронний ресурс]. — 2009. — Режим доступу: <https://bit.ly/2CvCk1e>. (дата звернення: 30.05.2021)
6. Vue.js Material Component Framework — Vuetify.js [Електронний ресурс]. — 2016. — Режим доступу: <https://vuetifyjs.com> (дата звернення: 26.05.2021)
7. Дмитрий Котеров, Алексей Костарев РНР. В подлиннике. / Д. Котеров, А. Костарев — Спб.: «БХВ-Петербург», 2005. — 1120 с.
8. Колисниченко Д. Н. Самоучитель РНР 5 / Д. Н. Колисниченко — Спб.: Наука и Техника, 2007. — 640 с.
9. Кузнецов Максим, Симдянов Игорь РНР 5/6. / М. Кузнецов, И. Симдянов — Спб.: «БХВ-Петербург», 2009. — 1024 с.
10. Кузнецов Максим, Симдянов Игорь Объектно-ориентированное программирование на РНР. / М. Кузнецов, И. Симдянов — Спб.: «БХВ-Петербург», 2007. — 608 с.
11. Л. Аткинсон, З. Сураскін. РНР5. Библиотека профессионалу. / Л. Аткинсон, З. Сураскін — М. : «Вільямс», 2006 — 543 с.
12. Скотт Хокінс. Адміністрування веб-сервера Apache і керівництво по електронній комерції. / С. Хокінс. — М.: «Вільямс», 2001. — 336 с.

13. Хокинс С. Администрирование Web-сервера Apache / С. Хокинс — М.: Вильямс, 2001. — 336 с.
14. Цілі сталого розвитку [Електронний ресурс]. – Режим доступу: <https://inlnk.ru/bxNLe> (дата звернення: 30.05.2021)
15. Роберт Шелдон, Джофрей Мойе MySQL: базовий курс Beginning MySQL. / Р. Шелдон, Д. Мойе — М.: «Диалектика» 2007. — 880 с.
16. MySQL. Довідник. MySQL АВ. — М: «Вільямс», 2006 — 521 с.
17. Кузнецов Максим, Симдянов Игорь MySQL на примерах. / М. Кузнецов, И. Симдянов — Спб.: «БХВ-Петербург», 2008. — 952 с.
18. Леон А. Г. PHP 5. Библиотека профессионала / А. Г. Леон — М. : Вильямс, 2006. — 944 с.
19. Сталий розвиток [Електронний ресурс]. — Режим доступу: <https://www.zhiva-planeta.org.ua/diyalnist/staluy-rozvutok.html>. (дата звернення: 30.05.2021)
20. Рухляда В. С., Карпенко М. Ю. Розробка персонального WEB-сайту. / В. С. Рухляда, М. Ю. Карпенко // Харків: ХНАМГ. – 2008.
21. Томсон Л., Веллінг Л. Розробка Web-додатків на PHP і MySQL. ДияСофтЮП, 2013. - 672с
22. Open Server URL: <https://ospanel.io/> (дата звернення 12.05.2021)
23. Дэвид Макфарланд. Большая книга CSS3 – CSS3. СПб.: Питер, 2016 – 251 с.
24. Бер Бибо, Иегуда Кац. jQuery. Подробное руководство по продвинутому JavaScript – СПб:Питер.2011. – 258 с.
25. Гетия И.Г., Леонтьева И.Н., Шумилин В.К. Методические указания по проведению занятия по дисциплине «Безопасность жизнедеятельности» на тему: «Определение интегральной бальной оценки тяжести труда на рабочем месте» . – М.: МГАПИ, 2002 . – 22 с.

26. Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу //Охорона праці. – 2001. –№ 12. – С. 12-20.

27. НПАОП 0.00-1.28-10. Правила охорони праці під час експлуатації електронно-обчислювальних машин.

28. Гукин Д. – FrontPage для “чайников”. – К.,1996.

29. Пасько В. FrontPage и сайтостроительство. – К., 2000.

30. Кузнєцов М., Симдянов І. Голишев С. РНР 5. Практика створення Web-сайтів. – К., 2007.

Кафедра інтелектуальних інформаційних систем
Система первинної медичної діагностики на основі нечіткої логіки