

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**

**НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра інженерії програмного забезпечення

## **Пояснювальна записка**

до магістерської роботи  
на ступінь вищої освіти магістр

на тему: **«РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПОШУКУ  
ЛІКАРЯ ЗА ПРОФІЛЕМ МОВОЮ JAVA»**

Виконав: студент 7 курсу, групи ППЗМ-71  
спеціальності

121 Інженерії програмного забезпечення

(шифр і назва спеціальності)

Коломійчук В.В.

(прізвище та ініціали)

Керівник Негоденко О.В.

(прізвище та ініціали)

Рецензент \_\_\_\_\_

(прізвище та ініціали)

Нормоконтроль \_\_\_\_\_

(прізвище та ініціали)

# ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

## НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Магістр»

Спеціальність підготовки - 121 «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Інженерії програмного забезпечення

Негоденко О.В.

“ \_\_\_\_\_ ” \_\_\_\_\_ 2021 року

### З А В Д А Н Н Я НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТА

**КОЛОМІЙЧУКА ВЛАДИСЛАВА ВОЛОДИМИРОВИЧА**

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка програмного забезпечення для пошуку лікаря за профілем мовою Java»

Керівник роботи: Негоденко О.В. завідувач кафедри. кандидат технічних наук, доцент  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «12» березня 2021 року №65

2. Строк подання студентом роботи «01» червня 2021 року

3. Вхідні дані до роботи

Науково-технічна література з питань, пов'язаних з програмним забезпеченням, програмуванням на мові Java, та роботою із базою даних;

4. Зміст розрахунково-пояснювальної записки(перелік питань, які потрібно розробити).

4.1 Аналіз проектного рішення

4.2 Аналіз методів діагностування захворювань.

4.3 Реалізація програмного забезпечення.

5. Перелік демонстраційного матеріалу (назва основних слайдів)

1. Мета роботи, об'єкт і предмет досліджень

2. Аналіз Існуючих рішень

3. Задачі проекту

4. Вибір засобів розробки і проектування

5. Експертна система (архітектура)

6. Інтерфейс додатку

7. Висновки

6. Дата видачі завдання «19» квітня 2021

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів бакалаврської Роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	19.04.2021	
2	Опис предметної області	19.04.2021	
3	Постановка задачі	22.04.2021	
4	Вибір алгоритмів шифрування	25.04.2021	
5	Програмна реалізація	02.05.2021	
6	Тестування програмного забезпечення	07.05.2021	
7	Оформлення пояснювальної записки	10.05.2021	
8	Розробка демонстраційних слайдів	16.05.2021	
9	Попередній захист роботи	16.05.2021	
10	Здача роботи в деканат	01.06.2021	

Студент

\_\_\_\_\_ ( підпис )

Коломійчук В.В.

(прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ ( підпис )

Негоденко О.В.

(прізвище та ініціали)





## РЕФЕРАТ

Текстова частина магістерської роботи 50 стр., 16 рис., 7 табл., 54 джерел

ДІАГНОСТИЧНИХ СИСТЕМА, ПОПЕРЕДНІЙ ДІАГНОЗ, ПРОЦЕС ДІАГНОСТУВАННЯ, ЗАПИС ПО СИМПТОМАМ, ПРОЦЕС АВТОМАТИЗАЦІЇ, МЕТОДИ ДІАГНОСТИКИ, СИМПТОМИ ЗАХВОРЮВАННЯ, АЛГОРИТМ, АВТОМАТИЗАЦІЯ

**Об'єкт дослідження:** процес автоматизації діагностування захворювань пацієнтів по симптомам.

**Предмет дослідження:** методи діагностування захворювань за допомогою експертно-діагностичних систем.

**Мета дослідження:** розробка системи, яка за допомогою методів класифікатора виявить одне з можливих захворювань після введення симптомів та організує відповідний візит до лікаря.

## ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ПРОЕКТНОГО РІШЕННЯ .....	10
1.1.Характеристика процесу діяльності.....	10
1.2.Постановка задачі.....	12
1.3.Огляд існуючих аналогів.....	14
1.4.Вибір інформаційного забезпечення.....	15
РОЗДІЛ 2 МЕТОДИ ДІАГНОСТУВАННЯ ЗАХВОРЮВАНЬ .....	21
2.1. Математична постановка задачі .....	21
2.2 Експертна система.....	26
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	33
3.1. Технічне забезпечення.....	33
3.2. Засоби розробки діагностичної системи.....	34
3.3. Архітектура програмного забезпечення .....	35
3.4. Інструкція користувача.....	42
ВИСНОВКИ.....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ .....	59
ДОДАТКИ.....	65

## ВСТУП

**Актуальність дослідження.** Актуальність роботи полягає у тому, що останнім часом все більше послуг, у тому числі пов'язаних з охороною здоров'я, намагаються перевести у цифровий формат, зрозумілий сучасним гаджетам, наприклад нещодавно створена електронна система охорони здоров'я «eHealth» та впровадження електронних рецептів на ліки. Окрім перенесення послуг до простору інформаційних технологій, важливим завданням є підвищити рівень встановлення правильних діагнозів за рахунок автоматизації цього процесу та скоротити час взаємодії лікар – пацієнт, створивши систему встановлення діагнозу на основі симптомів пацієнта.

Комп'ютерні технології знайшли своє місце у повсякденних рішеннях. Не стала винятком і медицина. Експерти впевнені, що лікар витрачає 65 % часу на спілкування з пацієнтом, щоб зібрати анамнез. Медсестра також може проводити анкетування, але це лише перенесе роботу на іншу людину. Творці медичних чат-ботів пропонують якісно новий підхід до вирішення проблеми. Інтелектуальні програми можуть опитувати пацієнта, реєструвати симптоми та скарги і складати попередню картину захворювання. Окрім того, використання автоматизованих систем дозволило би позбавитись від фактору людської помилки при встановленні захворювання, збільшити якість медичного обслуговування та підвищити довіру пацієнтів до лікарів.

**Метою дослідження** є розробка системи, яка за допомогою методів класифікатора виявить одне з можливих захворювань після введення симптомів та організує відповідний візит до лікаря.

Задачі дослідження:

1. вивчити існуючі рішення;
2. розробити систему на мові java, яка реалізує процес запису на прийом до лікаря ;
3. вибрати відповідні алгоритми для реалізації системи;
4. впровадити додаток на мові java, здатний опитати користувача,



визначити загальну картину захворювання, а також записатися на прийом до лікаря.

**Об'єкт дослідження:** процес автоматизації діагностування захворювань пацієнтів по симптомам.

**Предмет дослідження:** методи діагностування захворювань за допомогою експертно-діагностичних систем.

## РОЗДІЛ 1 АНАЛІЗ ПРОЕКТНОГО РІШЕННЯ

### 1.1. Характеристика процесу діяльності

Спочатку комп'ютери призначалися головним чином для виконання складних математичних розрахунків, в даний час домінуючим напрямком є збір та обробка інформації.

Такий перерозподіл основних функцій, які виконують обчислювальні технології, цілком зрозуміло - зменшення вартості комп'ютерів зробило їх доступними для дуже малих підприємств і навіть приватних осіб.

Сьогодні керувати бізнесом без комп'ютера просто немислимо. Комп'ютери давно і міцно увійшли в такі сфери управління, як бухгалтерія, управління складами, асортимент та закупівлі. Однак сучасний бізнес вимагає набагато ширшого використання інформаційних технологій в управлінні підприємством. Прибутковість та розвиток інформаційних технологій пояснюється тим, що сучасний бізнес надзвичайно чутливий до помилок управління. Інтуїції, особистого досвіду керівника та розміру капіталу вже недостатньо, щоб бути першим. Для того, щоб прийняти будь-яке компетентне управлінське рішення в умовах невизначеності та ризику, необхідно постійно контролювати різні аспекти фінансово-господарської діяльності, будь то торгівля, виробництво чи надання будь-яких послуг. Тому сучасний підхід до управління передбачає інвестування в інформаційні технології. І чим більшим буде підприємство, тим серйознішими повинні бути ці інвестиції. Вони є необхідністю - у жорстокій конкурентній боротьбі перемагають лише ті, хто краще оснащений та найбільш ефективно організований.

Автоматизована система запису до лікаря по симптомам хвороби містить дані про лікарів, пацієнтів та зв'язки, необхідні для функціонування поліклініки. База даних дозволяє додавати, змінювати, шукати та видаляти дані, а також переглядати їх.

Майбутня система автоматизує процес первинної діагностики, коли

пацієнт вперше звернеться до медичного закладу. В даний час усі процеси, пов'язані з діагностикою, в основному виконуються лікарями особисто, без використання будь-яких ІТ-систем, що, в свою чергу, має деякі суттєві недоліки:

- фактор людської помилки - скільки лікарів, скільки діагнозів;
- низький рівень довіри пацієнтів до професійних компетенцій лікарів.

Система взаємодії (лікар – пацієнт)

Таблиця 1.1 – Опис функцій застосування

Актор	Функція	Опис функції
1. Пацієнт	1. Реєстрація / Аутентифікація	Гість створює обліковий запис лікаря шляхом введення персональних даних та ключу активації облікового запису
	2. Перегляд статистики	Гість може переглядати статистику використання системи
2. Лікар	1. Встановлення діагнозу пацієнта	За вказаними симптомами пацієнта і/або результатами медичного обстеження лікар отримує список ймовірних захворювань пацієнта
	1.1 Ідентифікація пацієнта	Пошук облікового запису пацієнта у системі
	1.1.1 Пошук пацієнта у системі	Пошук за персональними даними пацієнта
	1.1.2 Реєстрація нового пацієнта у системі	Якщо пацієнт ще не має облікового запису у системі, лікар реєструє пацієнта вводячи його персональні дані до системи

Таблиця 1.1 – Опис функцій застосування

2. Лікар	1.2 Введення інформації про стан здоров'я пацієнта	Лікар вносить симптоми пацієнта і/або результати медичного обстеження
	1.3 Отримання списку ймовірних захворювань пацієнта	Лікар отримує список захворювань, що за вказаними симптомами і/або результатами обстеження є найбільш ймовірними
	2. Оновлення бази знань системи	Лікар вносить до системи данні про нові взаємозв'язки симптомів та захворювань
	2.1 Введення нових симптомів	Лікар вводить назву нового симптому, а також інформацію про його інтенсивність за допомогою функцій належності.
	2.2 Введення нових захворювань	Лікар вводить назву нового захворювання
	2.3 Створення групи правил	Лікар створює групу правил, до якої згодом будуть додані нові правила взаємодії симптомів та захворювань
	2.4. Створення нових правил	Лікар обирає групу правил, до якої треба додати нове правило, потім він обирає зі списку наявних симптомів необхідну комбінацію захворювань та їх рівнів залежності, а у відповідність вказує ймовірність певного захворювання (також із вже наявного списку)

## 1.2. Постановка задачі

Тематична область є частиною фактичної системи, яка є предметом цього дослідження. При проектуванні автоматизованих інформаційних систем предметна область представлена моделями даних з декількома рівнями. Кількість рівнів залежить від складності завдань, що вирішуються, але в кожному випадку вона охоплює концептуальний та логічний рівень.

Предметом цієї роботи є робота служби запису до лікаря по симптомам хвороби, завданням якої є відстеження діагнозів пацієнта на основі симптомів.

У клініці працюють лікарі різних спеціальностей з різною кваліфікацією. Коли лікар починає працювати, фіксуються всі дані: прізвище, ім'я, по батькові, спеціальність, категорія.

Усі пацієнти проходять обов'язкову реєстрацію, під час якої до бази даних вносяться стандартні персональні дані - прізвище, ім'я, по батькові, рік народження.

Всі бесіди пацієнтів із клінікою реєструються, визначається вартість лікування та запам'ятовується дата співбесіди.

Створення експертно-діагностичної системи, основним завданням якої є отримання на основі переліку симптомів пацієнта, встановити найбільш вірогідні захворювання, що відповідають стану здоров'я пацієнта.

Цілями розробки є:

- прибрати фактор людської помилки при встановленні діагнозу;
- підвищити відсоток правильно встановлених діагнозів;
- полегшити процес встановлення діагнозу по симптомам;
- підвищити рівень довіри пацієнтів до лікарів.

Для досягнення поставлених цілей необхідно вирішити наступні задачі:

- встановлення захворювання пацієнта за симптомами і/або результатами медичного обстеження;
- реєстрація лікарів у системі;
- реєстрація пацієнтів у системі;
- поповнення бази знань інформацією про нові симптоми та пов'язані з ними хвороби;
- перегляд історії захворювання пацієнта.

### 1.3. Огляд існуючих аналогів

Популярність та широке використання експертних систем є причиною їх постійного розвитку та подальшого впровадження в тих сферах людського життя, де вони досі не використовувались. Експертні системи в медицині, як правило, є частиною технологічного обладнання лабораторій, залишаючись приватною частиною системи і не стаючи державною власністю. Однак вони існують у світі і є відомі експертні системи, творці яких з різних причин вирішили розкрити деякі деталі реалізації розроблених систем.

«MYCIN» - це експертна система, розроблена для потреб медичної діагностики, зокрема, вона призначена для роботи в галузі діагностики та лікування інфекцій крові [20]. Система ставить відповідний діагноз на основі представлених симптомів і рекомендує медичну процедуру у випадку діагностованих інфекцій. Він складається з 455 правил, розроблених за сприяння групи експертів з інфекційних захворювань Стенфордського ВУЗу. Головною його перевагою є використання різних підходів.

Система INTERNIST-I була комп'ютерним діагностичним інструментом, розробленим на початку 1970-х років в Університеті Пітсбурга як навчальний експеримент. Система виявила кілька сотень різних захворювань на рівні кваліфікованого лікаря. Складні діагнози внутрішніх захворювань, виведені системою, встановлюються на основі історії хвороби, симптомів та результатів лабораторних досліджень [10].

Онлайн реєстрація – це швидкий та зручний спосіб самостійно записатися на будь-які події. Для визначення Інтернет ресурсів, що надають можливість реєстрації на прийом до лікаря, було визначено такі існуючі ресурси: система «Дніпромед», портал «Поліклініка», а також портал «Dос.ua». Інтернет портал «Дніпромед» [12] – портал для пошуку спеціалістів відповідної категорії. Після пошуку, буде показаний потрібний центр і лікарня з переліком фахівців. Портал [dnipromed.com](http://dnipromed.com) - спільний проект компанії CDM і КП «Дніпропетровський міський Центр здоров'я» Департаменту охорони здоров'я Дніпропетровської міської ради.

Це 15 унікальна і найбільш повна медична інформаційна система, яка не має аналогів серед спеціалізованих сайтів у Дніпропетровській області. Інформаційна система «Поліклініка» [13] – проєкт Житомирської поліклініки №2, запущений у 2013 році.

За цим проєктом для пацієнта створюється «персональний кабінет», з якого він отримує інформацію про спеціалістів медичного закладу, нагадування про наступні візити та в подальшому щеплення та профілактичні огляди. Пацієнту не треба приходити в поліклініку для отримання талону запису до лікаря, що знімає чергу в реєстратурі.

Лікар отримує інформацію про пацієнтів, які записалися на прийом, зберігає час прийому, що дає можливість більш якісно розмежовувати робочий час працівників та вести статистику. Doc.ua [14] –сервіс пошуку лікаря або лікарні за потрібними критеріями у місті Київ. Портал надає можливість введення критеріїв пошуку та видає результати в межах міста Київ.

#### **1.4. Вибір інформаційного забезпечення**

##### Вхідні дані.

- персональні дані лікарів при реєстрації: прізвище; ім'я; логін; ключ реєстрації; електронна адреса; пароль облікового запису.
- персональні дані пацієнтів: прізвище; ім'я; дата народження; контактний номер телефону
- інформація про новий симптом: назва симптому; границі симптому (числові значення лівої та правої границі); крок показнику симптому (наприклад, крок «0.1» для температури тілота крок «1» для систолічного тиску); трикутні функції належності для кожного рівня захворювання.
- інформація про нове захворювання (назва захворювання);
- інформація про нову групу захворювань (назва групи);
- інформація про нове правило взаємозв'язку симптомів та захворювань:

симптоми та їх рівень належності (інтенсивність); ймовірність захворювання для даної комбінації симптомів.

- інформація для діагностики захворювання: персональні дані пацієнта; область діагностики (група правил); показники симптомів для певної групи правил.

### Вихідні дані.

- Список захворювань, що відповідають зазначеним симптомам. Список захворювань виводиться у відсортованому порядку з відповідними числовими значеннями ймовірності, починаючи з захворювання, що має найбільшу ймовірність наявності у пацієнта.
- Інформація про відвідування та діагностики пацієнта.
- Система виводить усю історію відвідувань пацієнта разом з відповідними показниками симптомів на момент відвідувань, таким чином можливо відслідкувати динаміку самопочуття пацієнта.
- Інформація про групи правил. Для лікарів доступна сторінка, на якій можна ознайомитися з групами правил, а також дізнатися список взаємопов'язаних симптомів та захворювань.
- Сторінка статистики для гостей сервісу. Інформація про кількість лікарів, що зареєстровано, кількість пацієнтів, симптомів, захворювань.



Таблиця 1.2. – Структура та призначення сутностей бази даних

Назва	Поле	Призначення
Symptom – містить інформацію про симптоми	[Id] INT	Унікальний ідентифікатор
	[Name] NVARCHAR(100)	Назва симптому
	[LeftBound] FLOAT	Ліва границя симптому
	[RightBound] FLOAT	Права границя симптому
	[Step] FLOAT	Крок симптому
Disease – містить інформацію про захворювання	[Id] INT	Унікальний ідентифікатор
	[Name] NVARCHAR(100)	Назва захворювання
MfLevel – містить рівні симптомів	[Id] INT	Унікальний ідентифікатор
	[Level] NVARCHAR(100)	Назва рівня інтенсивності
Symptom Mf – містить інформацію про інтенсивність симптомів	[Id] INT	Унікальний ідентифікатор
	[SymptomId] INT	Індекс симптому
	[LeftBound] FLOAT	Ліва границя функції належності
	[Middle] FLOAT	Вершина функції належності
Disease probability – містить інформацію про ймовірність захворювань	[Id] INT	Унікальний ідентифікатор
	[DiseaseId] INT	Індекс захворювання
	[MFLevelId] INT	Індекс рівня інтенсивності
Rules group – групи правил	[Id] INT	Унікальний ідентифікатор
	[Name] NVARCHAR(400)	Назва групи правил

Таблиця 1.2. – Структура та призначення сутностей бази даних

Rule – містить інформацію проправила	[Id] INT	Унікальний ідентифікатор правила
	[Rule_Name] NVARCHAR(400)	Назва правила
	[GroupId] INT	Індекс групи правил
Auth User – містить інформацію пролікарів	[Id] INT	Унікальний ідентифікатор користувача
	[Password] VARCHAR(128)	Пароль
	[Last_login] DATETIME(6)	Дата останнього входу в систему
	[Is_superuser] TINYINT(1)	Чи має права адміністратора (для редагування)
	[Username] VARCHAR(150)	Логін
	[First_name] VARCHAR(30)	Ім'я
	[Last_name] VARCHAR(150)	Прізвище
	[Email] VARCHAR(254)	Електронна адреса користувача
	[Is_active] TINYINT(1)	Чи зараз у системі
[User_Date_joined] DATETIME(6)	Дата реєстрації	
Antecedent – взаємозв'язки між правилами та симптомами	[Id] INT	Унікальний ідентифікатор зв'язку
	[MFId] INT	Індекс рівня інтенсивності симптому
	[RuleId] INT	Індекс правила

Таблиця 1.2. – Структура та призначення сутностей бази даних

Consequent – взаємозв'язки між правилами і захворюваннями	[Id] INT	Унікальний ідентифікатор наслідку
	[DiseaseProbId] INT	Індекс ймовірності захворювання
	[RuleId] INT	Індекс правила
Registration key – список секретних ключів для реєстрації лікарів (для реєстрації необхідно мати ключ, щоб не дати можливості будь-кому вносити зміни до системи)	[Id] INT	Унікальний ідентифікатор реєстраційного ключа
	[RegisterKey] NVARCHAR(100)	Реєстраційний ключ
Patient – містить інформацію про персональні дані пацієнтів	[Id] INT	Унікальний ідентифікатор пацієнта
	[First_name] NVARCHAR(100)	Ім'я
	[Last_name] NVARCHAR(100)	Прізвище
	[Birth_date] DATE	Дата народження пацієнта
	[Phone_number] NVARCHAR(100)	Контактний номер телефону

Таблиця 1.2. – Структура та призначення сутностей бази даних

Patient history – історія відвідувань пацієнтів (необхідно для відстеження лікарем динаміки захворювання)	[Id] INT	Унікальний ідентифікатор історії
	[Patient_id] INT	Індекс пацієнта
	[Doctor_id] INT	Індекс лікаря
	[Check_date] DATE	Дата обстеження
Patient symptoms – історія діагностики пацієнтів	[Id] INT	Унікальний ідентифікатор симптому
	[Patient_check_id] INT	Індекс пацієнта
	[Symptom_id] INT	Індекс симптому
	[Symptom_value] NVARCHAR (100)	Значення симптому

## РОЗДІЛ 2 МЕТОДИ ДІАГНОСТУВАННЯ ЗАХВОРЮВАНЬ

### 2.1. Математична постановка задачі

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  $n \gg m$  [23]. 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  $\langle \text{нежить, помірний} \rangle$ .

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 2.1. [23], щоб представити рівень впевненості у наявності захворювання. Записи в таблицях профілів захворювання базуються на цих нечітких множинах.

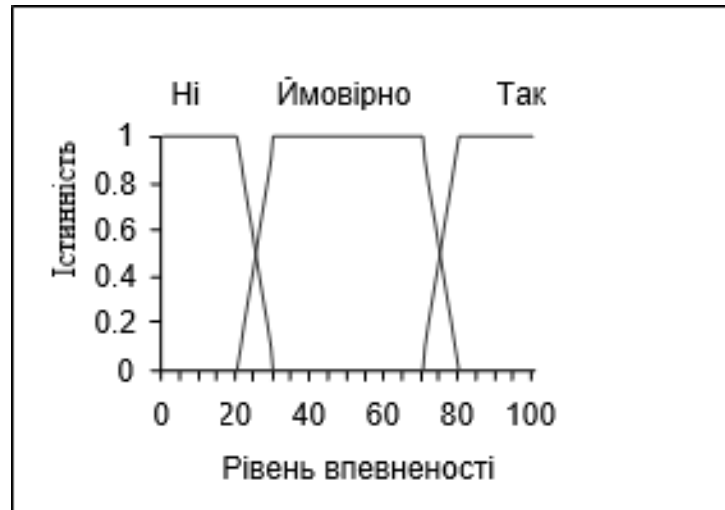


Рисунок 2.1. – Нечіткі множини, що описують впевненість наявності

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

У таблиці 2.2 представлена порожня таблиця для профілю захворювання  $d_x$ . Він вказує п'ять нечітких значень для кожної відповідного симптому  $r_i, i = 1, \dots, k$ .

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 2.2 наведено профіль для Грипу [23], а у таблиці 2.4 наведено приклад запису симптомів пацієнта з відповідними

значеннями нечіткими значеннями інтенсивності.

Таблиця 2.2 – Профіль для Грипу

<b>Рівень належності Симптоми</b>	<b>Дуже низький</b>	<b>Низький</b>	<b>Помірний</b>	<b>Високий</b>	<b>Дуже високий</b>
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
11. Втрата апетиту	Ні	Ймовірно	Так	Так	Так
12. Чхання	Ймовірно	Ймовірно	Так	Так	Так

Таблиця 2.3 – Таблиця симптомів пацієнта

<b>Симптом</b>	<b>Інтенсивність</b>
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. Чхання	Середній

Введемо наступні умовні позначення:

- $s[f]$  - значення рівня належності симптому  $f$ ;
- $r_{ij}$  -  $j$ -й симптом  $i$ -го захворювання;
- $P_{ij}[r_{ij}, v]$  - ймовірність наявності  $i$ -го захворювання, коли симптом  $r_{ij}$  має рівень належності  $v$ ;
- $\delta_{ij}$  - рішення щодо діагнозу  $i$ -го захворювання на основі відповідного симптому  $r_{ij}$ ;
- $k_i$  - загальна кількість симптомів для  $i$ -го захворювання;
- $w_{ij}$  - вага симптому  $r_{ij}$  при діагностиці  $i$ -ї хвороби;
- $\sigma_i$  - загальне рішення діагнозу для  $i$ -ї хвороби.

Вплив симптому  $r_{ij}$  на рішення стосовно ймовірності діагнозу  $\delta_i$  можна отримати безпосередньо з таблиці-профілю захворювання  $P_{ij}[r_{ij}, v]$  [23]. Тоді:

$$\delta_{ij} = P_{ij}[r_{ij}, s[r_{ij}]] \quad (1)$$

Враховуючи дію всіх симптомів, загальне рішення діагнозу для  $i$ -ї хвороби обчислюється наступним чином:

$$\sigma_i = \left( \sum_{j=1}^{j=k_i} w_{ij} \delta_{ij} \right) / \left( \sum_{j=1}^{j=k_i} w_{ij} \right) \quad (2)$$

Останнім кроком є отримання чіткого значення наявності у пацієнта кожного захворювання з множини  $D$ . Коефіцієнт ваги  $w_{ij}$  вводиться у випадку коли деякі симптоми мають більший або менший вплив при діагностуванні захворювання. Лікар-експерт вважає, що всі симптоми мають однаковий вплив, то коефіцієнт буде  $w_{ij}$  однаковим для всіх симптомів.



У цьому випадку формулу (2) можна спростити наступним чином:

$$\sigma_{ij} = \frac{1}{k} \sum_{i=1}^{j=k_i} \delta_i \quad (3)$$

### Приклад

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

$$k_i = 10, w_{ij} = 1,$$

де  $j = 1, \dots, 10$ .

Використовуючи рівняння (1) для прийняття діагностичних рішень  $\sigma_{ij}$  ( $j = 1, \dots, 10$ ), 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:

$\sigma_i = (7 \text{ «Так»} + 2 \text{ «Ймовірно»} + 1 \text{ «Ні»}) / 10$  Нечітка множина загального рішення  $\sigma_i$  показана на рисунку 2.2.

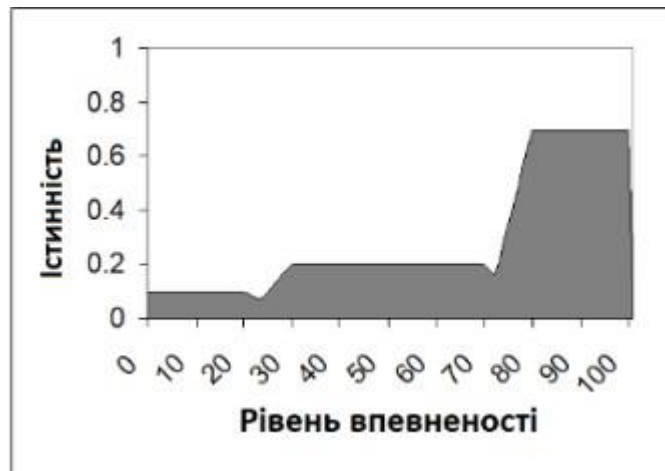


Рисунок 2.2 – Нечітка множина загального рішення  $\sigma_i$

Після отримання загального рішення  $\sigma_i$  необхідно розрахувати чітке значення ймовірності наявності захворювання у пацієта. Існує багато методів дефазифікації, однак найбільш поширеними є центроїд (CENTROID) та максимум (MAXIMUM). Використовуючи метод центроїд маємо:

- $c_i$  - середнє арифметичне (центроїд) нечіткої множини  $\sigma_i$ ;
- $c_{\text{Так}}$  - середнє арифметичне (центроїд) 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:

$$q_i = \left( \frac{c_i}{c_{\text{Так}}} \right) * 100\% = \left( \frac{0.69}{0.87} \right) * 100\% = 79.3\%$$

## 2.2 Експертна система

Експертні системи - це програмні пакети, які збирають досвід фахівців у даній галузі з метою його тиражування (досвіду) для консультацій з менш кваліфікованими користувачами. Розвиток ЄС - це галузь інформатики, яка активно розвивається і спрямована на використання комп'ютерів для обробки інформації в тих галузях науки і техніки, де традиційні моделі математичного моделювання є неадекватними, де важлива семантична та логічна обробка даних, досвід лікарів. Експертні системи досить молоді - перші системи цього типу з'явилися в США в середині 1970-х. В даний час у світі існує кілька тисяч промислових ЕК, які консультують з питань управління складними диспетчерськими пунктами, проведення медичних діагнозів, вирішення проблем з електронними пристроями тощо. На сьогоднішній день простіше перерахувати сфери, де ЕК не використовуються, ніж ті, де вони вже використовуються.

Основною відмінністю ЄС від іншого програмного забезпечення є наявність бази даних, в якій дані зберігаються у вигляді записів мовою знань (MPZ), що дозволяє легко змінювати та доповнювати базу знань у формі, зрозумілій для фахівців. - Програмісти ЄС. У звичайних програмах знання вбудовані в алгоритм,

і лише програміст (автор програми) може це виправити (якщо він пам'ятає, як побудована його програма).

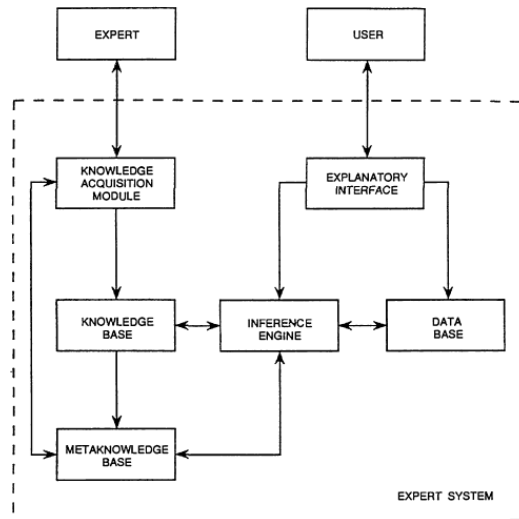
Донедавна саме різні МЗ були центральним питанням розвитку ЄС. В даний час існує кілька десятків мов або моделей представлення знань. Найпопулярнішими з них є: продукти, семантичні мережі, фрейми, обчислення предикатів першого порядку, мови програмування ОО та ін. Для даних моделей існує потрібна математична нотація, розроблені системи обробки даних, що реалізують ці МРЗ. Вибір конкретної моделі залежить від структури знань у даній предметній області. По-перше, необхідно визначити дані структури. Уточнення елементів знань та їх взаємозв'язків відбувається в безпосередньому контакті зі спеціалістами з даної предметної галузі - експертами. Ми називаємо цей процес набуттям знань, а творців ЄС, які набувають та структурують знання, називають інженерами знань.

Ми відразу визначимо ті предметні галузі, де розподіл знань має сенс. Це сфера, де домінують емпіричні знання, де накопичення фактів передують розвитку теорій (медицини, геології, фінансів тощо). Такі впорядковані галузі, як математика, фізика та теоретична механіка, описують свою закономірність на розробленому математичному апараті, що дозволяє моделювати машини за допомогою традиційного алгоритмічного програмування (без підкреслення рівня знань). Знання важливі там, де визначення незрозумілі, поняття змінюються, випадки залежать від багатьох факторів, де існує висока невизначеність, неясна інформація. Загалом, знання - це основні права в даній предметній галузі, які дозволяють людині вирішувати різноманітні проблеми, тобто факти, концепції, відносини, оцінки, правила, евристику (фактичне знання), а також прийняття рішень стратегії в цій галузі (або стратегічні знання).

Це комп'ютерна система, що імітує процес міркувань експерта в певній галузі знань [7, 12]. У більшості випадків експертні системи будуються для розповсюдження досвіду та знань у вирішенні конкретних завдань спеціаліста для людей, які не є експертами в певній галузі інтелектуальної діяльності. Крім того, їх можна створювати для різних процесів, таких як: діагностика,

підтримка рішень, проектування та планування. Типова архітектура експертної системи [13] представлена на блок-схемі на рисунку 2.3

Рис. 2.3. Типова експертна система



Ядро будь-якої експертної системи складається з бази знань, бази даних та системи висновків. Ці три блоки разом з певним інтерфейсом для спілкування з користувачем утворюють мінімальну конфігурацію системи, яку вже можна назвати експертною. База знань містить загальні знання, що стосуються проблемної області (симптоми, захворювання, принципи їх взаємодії). Завдання бази даних - зберігати дані, що стосуються середовища експертної системи (лікарі, пацієнти, історія хвороби тощо). Дані можна отримати за допомогою діалогу між експертною системою та користувачем. Система умовиводу працює з правилами і робить незрозумілі висновки.

Приклади використання експертних систем у медицині:

Одне з найдавніших та найвідоміших медичних випробувань, MYCIN, було обговорено в уривках у попередніх розділах. Тут коротко будуть розглянуті деякі медичні цілі ЕК, які стали класичними та сучасними.

Моніторинг ЕК у післяопераційних хворих

Система VM призначена для спостереження за станом післяопераційних пацієнтів, яким потрібна штучна вентиляція легенів. Система визначає тип респіратор та режим його роботи залежно від розвитку захворювання та реакції

людини на лікувальні дії. Система працює з даними, які потім змінюються з часом (динамічна система). Погляд системи на час обмежений лише її теперішнім та попереднім статусом.

Знання в системі подаються у вигляді наступних типів правил: перехід, ініціалізація, стан та обробка. Система постійно отримує нові показання приладів і реалізує всі його правила.

Правила переходів можуть визначити моменти зміни станів пацієнта, коли необхідно змінити послідовність моніторингу. Момент зміни стану визначається виходами показників за задані межі. Правила ініціалізації встановлюють новий контекст, тобто нові значення, які очікуються.

Після встановлення контексту правила держави визначають нову поведінку системи. Тому показання пристрою постійно змінюються, система змінює контекст лише тоді, коли це достатньо виправдано правилами переходу. Правила лікування визначають необхідні режими роботи вентилятора.

Приклади правил експертної системи ВМ.

Правило 1: Якщо частота серцевих скорочень є прийнятною і частота серцевих скорочень не змінюється більше 20 ударів / хв на протязі 15 хвилин, то артеріальний тиск є прийнятним, а середній артеріальний тиск не змінюється більше 15 мм рт. протягом 15 хвилин, і систолічний артеріальний тиск є прийнятним, гемодинаміка стабільна.

Правило 2: якщо пацієнта переводять з ОБ'ЄМУ в ЦМВ або з АСІСТУ на ЦМВ, очікується, що: допустимий середній тиск повинен бути в межах 75 ... 110, допустимий пульс - 60 ... 110, вміст CO<sub>2</sub> у вихлопі - 28 ... 42.

Правило 1 застосовується до пацієнтів у різних режимах вентиляції (ГУЧНІСТЬ, CMV ASSIST та T-RIESE), а правило 2 застосовується до всіх пацієнтів у режимі контрольованої примусової вентиляції.

Правило 1 ілюструє, як система управління контролює стан пацієнта,

використовуючи терміни, значення яких змінюється залежно від клінічного контексту (наприклад, «прийнятний»).

Правило 2 показує, як система використовує свої очікування в певному контексті, щоб точно визначити значення термінів (наприклад, «прийнятний пульс»).

### ЕК аналіз причин артеріальної гіпертензії

Постановка проблеми. Близько 10% населення планети страждає на гіпертонію. Причини підвищення артеріального тиску дуже різноманітні - понад 30 основних захворювань. Складність діагностики цих захворювань полягає в тому, що вони можуть відноситись до компетенції фахівців у різних галузях медицини: нефрології, ангіології, урології та інших. Тому створення ЕК, яка включала б знання найкращих фахівців у різних галузях медицини, пов'язаних з артеріальною гіпертензією, є суттєвою допомогою у вдосконаленні діагностики цих захворювань.

Система MODIS призначена для діагностики різних симптоматичних форм артеріальної гіпертензії. Система MODIS може застосовуватися як в загальних поліклініках, так і в спеціалізованих клініках. Звичайно, рівень інформації про пацієнта, доступний у цих закладах, різниться. У клініці медичної особи доступна більш загальна інформація: скарги пацієнта, дані зовнішнього огляду, історія пацієнта та результати усіх оглядів. Тому, виходячи із загальної інформації, ЕК повинна більш точно визначити коло можливих захворювань та вибрати найбільш імовірні. ЕК дає рекомендації щодо того, до яких спеціалістів звертатися до пацієнта та направляти їх на спеціалістичні огляди.

Коли ЕК використовується у спеціалізованих лікарнях, доступна детальніша інформація, тобто дані спеціальних обстежень (наприклад, аортографія чи внутрішньовенна урографія). У цьому випадку ЕК повинен

надати більш детальний аналіз. Основними поняттями, з якими експерт працює в цій системі, є хвороби, їх форми та групи, симптоми, лабораторні дані тощо. Для їх опису використовуються три типи знань: описові знання, що використовуються експертом для опису взаємозв'язку між визначеннями. Наприклад, додаткова інформація експертів пов'язана з поняттям систолічного артеріального тиску. Наприклад, він розуміє, що систолічний тиск може бути в межах 50-350, а також, що систолічний тиск завжди більший, ніж діастолічний тощо;

Знання того, як експерт робить нові твердження на основі аналізу певної інформації. Система використовує усі правила для постановки діагнозу. Типовим прикладом такого типу знань є наступне діагностичне правило: «Якщо у молодого пацієнта спостерігаються пароксизми тиску, а результати гормональних аналізів крові виявляють більш ніж удвічі більше вмісту катехоламінів, пацієнт, швидше за все, страждає феохромоцитом»; знання того, як фахівець обирає найбільш перспективні шляхи пошуку. Ці знання, це список правил, що дозволяють експерту не брати до уваги менш імовірні гіпотези, а лише найбільш вірогідні. Це знання належить до метарівню, оскільки воно задає певний зразок використання знань з першого та другого рівнів. Приклад такого правила: «Якщо у пацієнта спостерігаються суттєві зміни в тестах сечі, перше, на що слід підозрювати, - це нефрологічна природа гіпертонії».

Експертні системи - це діагностичні програми високого рівня, які належать до інтелектуальних інформаційних систем, заснованих на знаннях. Галузь інформатики, яка вивчає інтелектуальні системи, називається штучним інтелектом. Експертна система, крім бази даних, включає базу знань та розгалужену систему програм для її обробки. Що слід розуміти під базою знань? База знань (з англ. *Isholou ^ eVaze*) - це спеціальний тип бази даних, призначений для управління знаннями (метаданими), тобто збору, зберігання, пошуку та розповсюдження знань.

Найважливішим параметром бази знань є якість знань, які вона містить. Найкращі бази знань містять новітні наукові знання, вдосконалені системи пошуку

інформації, мають ретельно продуману структуру та формат знань. Медичні знання - це інформація, яка може знадобитися лікарю.

Зокрема, це може бути інформація про стан здоров'я пацієнта, конкретну галузь медицини, ситуації, проблеми, правила їх вирішення, а також конкретні умови, що визначають вибір цих правил.

Медичні знання відрізняються від медичних даних більшою складністю, абстрактністю, повнотою та універсальністю опису певної галузі медицини. Слід пам'ятати, що з точки зору інформатики медицина є неспецифічною наукою, тому створення баз знань у медицині є досить трудомістким процесом. У реальному житті кількість ситуацій та «діагностичних правил» іноді перевищує знання, вбудовані в систему.

Експертні системи, як правило, здатні робити дедуктивні висновки (від загальних до конкретних) на основі неповних, нечітких та суперечливих знань, отриманих від професійних експертів. Якщо програми, що працюють з базами даних, можуть відповісти на питання лише за умови, що воно існує в базі даних, тоді експертна система відповідає на це питання, роблячи логічний висновок в умовах невизначеності або неповного знання.

Іншими словами, це допускає неточностей. Більше того, необхідною умовою функціонування експертної системи є її здатність пояснювати причини конкретного застосування: вона аналізує ситуацію та дає рекомендації щодо вирішення проблеми.



## РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 3.1. Технічне забезпечення

Діагностична система призначена для визначення діагнозів на базі різних медичних установ, таких як: обласні клініки, провінційні лікарні та спеціалізовані діагностичні центри. Крім того, систему можна встановити в приватних лабораторіях для автоматичної обробки нещодавно отриманих результатів тестування з метою якнайшвидшого інформування пацієнтів про критичні параметри показників стану здоров'я.

У зв'язку з тим, що експертно-діагностичну систему можна встановлювати на широкому діапазоні комп'ютерів з різними технічними параметрами, однією з необхідних умов є мінімізація системних ресурсів, необхідних для комп'ютера користувача. Для коректної роботи розробленої системи достатньо наступних особливостей:

а) комп'ютер із наступною конфігурацією:

- 1) тактова частота процесора 1 ГГц і більше;
- 2) обсяг оперативної пам'яті не менше 128 МБ;
- 3) веб-браузер із підтримкою JavaScript;
- 4) інші компоненти, такі як простір на жорсткому диску або графічна пам'ять, можуть мати будь-які параметри, оскільки вони суттєво не впливають на роботу програми;
- 5) стабільне з'єднання з Інтернетом.

б) рекомендована конфігурація комп'ютера:

- 1) тактова частота процесора не нижче 1,3 ГГц;
- 2) обсяг оперативної пам'яті не менше 256 МБ;
- 3) веб-браузер із підтримкою JavaScript;
- 4) інші компоненти, такі як простір на жорсткому диску або графічна пам'ять, можуть мати будь-які параметри, оскільки вони суттєво не впливають на роботу програми;

- 5) наявність швидкого, стабільного з'єднання з Інтернетом.
- в) комп'ютерна периферія, що включає:
  - 1) монітор;
  - 2) миша;
  - 3) клавіатура;
  - 4) принтер.

### **3.2. Засоби розробки діагностичної системи**

Інтегроване середовище розробки Visual Studio Code було використано для створення експертної діагностичної системи FDetec. Visual Studio Code - це потужний міжплатформений редактор програмного коду, який працює як мінімалістичний текстовий редактор в операційних системах Windows, macOS та Linux. VS Code здатний працювати з JavaScript та TypeScript «нестандартно» і має велику кількість розширень для роботи з різними мовами програмування (C ++, C #, Java, Python) [24].

Основною мовою програмування є Python - універсальна, сучасна мова програмування високого рівня, переваги якої включають, серед іншого висока ефективність програмних рішень та структурованого коду [25].

Синтаксис Python максимально спрощений, що дозволяє фахівцям з різним досвідом швидко створювати програмне забезпечення. Ядро має дуже зручну структуру, а широкий спектр вбудованих бібліотек дозволяє використовувати багато корисних функцій та можливостей. Python можна використовувати для написання настільних програм, а також для створення WEB-сервісів.

Технологією створення системи є Django - веб-фреймворк високого рівня на мові програмування Python, що дозволяє швидко створювати гнучкі веб-сервіси [26]. Django розроблений таким чином, що значна кількість компонентів, необхідних для розробки, вже вбудована в проект - це робиться

таким чином, що розробники зосереджуються лише на розробці власних алгоритмів та механізмів, незалежно від на тривіальні та основні компоненти. Це безкоштовний фреймворк з відкритим кодом, має широке та активне співтовариство, хорошу документацію.

Для зберігання даних було обрано MySQL Community Edition - реляційну базу даних на рівні підприємства, яка може працювати навіть за великих навантажень [27]. Його перевагами є повна відповідність стандартам SQL, добре підготовлена документація для розробників, а також велика спільнота, де можна знайти рішення будь-якої проблемної ситуації.

Доступ до бази даних забезпечує Django ORM, головною особливістю якого є відображення записів бази даних до об'єктів конкретних класів додатків і навпаки. Це дуже зручний механізм доступу та роботи з даними в умовах складності об'єктів та їх частоті зміни. Серед переваг - гнучка робота із записами в базі даних, незалежно від бази даних. Bootstrap був використаний для створення користувальницького інтерфейсу - фреймворку з відкритим кодом для створення веб-сайтів із використанням HTML, CSS та JS [28, 29]. Однією з особливостей Bootstrap є наявність безкоштовних наборів стилів для створення зовнішнього вигляду веб-сайтів, що стає дуже важливим і переконливим критерієм вибору технології для творців веб-сайтів, які орієнтуються не на зовнішній вигляд, а на функціональність веб-сайту. Набір інструментів, що використовується Bootstrap, характеризується високою швидкістю розробки. Усі використовувані технології та інструменти були обрані головним чином на основі сумісності для роботи у веб-додатку, а також повноти документації користувача та великої спільноти користувачів.

### **3.3. Архітектура програмного забезпечення**

У цій главі описується структурна схема послідовності для клініциста для діагностики хвороби пацієнта від моменту входу в систему дотримати ймовірний перелік захворювань із відповідною ймовірністю. На схемі є шість

класів:

- «Авторизатор (лікар)». Клас лікаря. Серед можливих застосувань у сценарії діагностики пацієнта є такі дії: автентифікація, створення симптомів, захворювань, правил та діагностика пацієнтів.
- «Модуль перегляду». Модуль, який обробляє запит HTTP для запуску діагностики. Після перевірки параметрів запиту (наявність значень симптомів) він викликає методи з інших модулів для подальшої обробки запитів. Точка входу запиту до системи.
- «Діагностичний модуль». Модуль, що містить методи та алгоритми діагностики потенційних захворювань пацієнта.
- Msgstr «Модуль CreateRule». Модуль, головним завданням якого є пошук взаємозв'язку між симптомами та захворюваннями у формі правил.
- «БД». Об'єкт бази даних.

#### Структурна схема класів програмного забезпечення

Схема реалізації експертної діагностичної системи. Набір класів еквівалентний набору сутності бази даних, що полегшує обробку даних: Django ORM зчитує дані з бази даних MySQL і використовує механізм відображення для перетворення записи із бази даних до набору об'єктів, які можна обробити, відфільтрувати та відсортувати за допомогою синтаксису Python.

Доступні наступні класи:

- «RegistrationKey» - модель, що містить значення секретного ключа для реєстрації лікарів;
- «Антецедент» - клас, який посилається на правило та рівень приналежності симптому, що є частиною цього правила (так звана причина);
- «Наслідковий» - тип, що посилається на правило та ймовірність захворювання, що є наслідком правила (так званий наслідок правила);

- «Симптоми пацієнта» - модель, що описує значення даного симптому пацієнта під час його візиту до лікаря;
- «SymptomMf» - модель, яка описує симптоми, використовуючи кілька функцій членства в залежності від інтенсивності симптому (наприклад, низький, нормальний та високий кров'яний тиск);
- «Правило» - клас, що описує взаємодію інтенсивності симптому з ймовірністю даного захворювання (наприклад, як низький артеріальний тиск впливає на ймовірність серцевої недостатності);
- «DiseaseProbability» - тип, який пов'язує захворювання з варіантами ймовірності (наприклад, низькою, середньою або високою ймовірністю);
- «Хвороба» - модель, що описує хворобу за назвою;
- «RuleGroup» - клас, що поєднує правила взаємодії симптомів та захворювань із загальним напрямком досліджень;
- «MfLevel» - тип, що являє собою набір можливих значень інтенсивності симптомів;
- «Симптом» - модель, що описує симптом за назвою, числовим значенням нижньої та верхньої меж симптому, а також ступенем симптому;
- «PatientHistory» - клас, що містить дані про відвідування пацієнтом лікаря за датою та ідентифікатором лікаря;
- «Пацієнт» - тип, що містить особисті дані пацієнтів (ім'я та прізвище, дата народження та контактний телефон);
- AuthUser - модель, що описує рахунок лікаря. Там ви можете знайти багато інформації про персональні дані та здатність лікаря вносити зміни до стану системи.

### Специфікація функцій

Оскільки мова програмування Python є багато парадигматичною (можуть бути використані об'єктно-орієнтований, процедурний та функціональний підходи), основні методи обробки інформації та алгоритми можна знайти у модулі, кожен в окремому файлі. Кожен файл модуля містить методи, які часто

відповідають за відповідні завдання. Під час розробки експертно-діагностичної системи «FDetect» було створено кілька модулів, кожен з яких містить функціонал, що дозволяє виконувати певний спектр завдань, а саме:

- модуль «Діагностування», який відповідає за лікування симптомів пацієнта, пошук симптомів, пов'язаних із захворюваннями, отримання результатів діагностики та пошук у базі даних інформації про попередні відвідування пацієнта;
- модуль «CreateRule» відповідає за побудову правил, що описують взаємодію симптомів та захворювань, а також за створення нових правил у системі після оновлення бази знань лікарем (додавання нових захворювань, симптомів або правил);
- модуль «GroupManagement» містить алгоритми роботи з групами правил, а саме: створення нових груп правил, додавання нових правил до груп правил, надання інформації про симптоми та захворювання, що виникають у даній групі правил;
- Модуль «Views» призначений для обробки запитів, надісланих браузером на сервер. Методи цього модуля перевіряють запити, і якщо всі критерії задовольняються, методи повертають запитувану інформацію або змінюють стан системи відповідно до запиту.

У таблиці 3.1 наведено детальний опис модулів програмного забезпечення

Таблиця 3.1 – Опис модулів програмного забезпечення

Назва модулю	Назва функції	Призначення
Create Rule module	addRules (symptoms,diseases, group)	Параметри: симптоми, захворювання, група правил. Метод зберігає нове правило до групи правил (метод-обгортка, що викликає інші методи для виконання необхідних операцій).
	Validate Symptoms And Diseases(symptoms, dis eases, group)	Параметри: симптоми, захворювання, група правил. Перевіряє чи наявні дані про всі симптоми, що входять до певної групи правил для діагностики.
	Save Antecedents(rule, li st)	Параметри: правило, список зв'язків між правилами та інтенсивністю симптомів.
	Save Consequents(rule, li st)	Параметри: правило, список зв'язків між правилами та ймовірністю захворювань.
	Get New RuleInstance (group)	Параметри: назва групи. Створює нове правило в існуючій групі правил.
	Parse And GetModels (symptoms, diseases)	Параметри: назви симптомів та захворювань. Дістає з бази даних інформацію про симптоми та захворювання, шукає їх у моделі.
Diagnosing module	calculate(str, group)	Параметри: список симптомів + їх чисельні значення інтенсивності, назва групи діагностики. Клас обгортка, що викликає необхідні методи для діагностики ймовірних захворювань.
	getRules(inputs, outputs, symptoms, group)	Параметри: значення симптомів, список захворювань, назва групи. Знаходить правила з бази знань за допомогою симптомів та захворювань.
	getAntecedents(inputs, ruleNames, mfsList)	Параметри: значення симптомів, назва правил, список інтенсивності симптомів. Створення взаємозв'язків між симптомами та правилами.

Diagnosing module	getConsequents(outputs, ruleNames, disease_probabilities)	Параметри: список захворювань, назваправил, ймовірність захворювань. Створення взаємозв'язків між захворюваннями та правилами.
	createRules(antArr, con Arr)	Параметри: списки зв'язків між правилами та симптомами і захворюваннями. Створення правил за допомогою взаємозв'язків між симптомами та захворюваннями.
	getDiseaseId(groupId)	Параметри: ідентифікатор групи правил. Пошук захворювань, що пов'язані з групою правил.
	defineOutput(disease)	Параметри: назва захворювання. Створення даних про можливі варіанти ймовірності захворювань.
	getVisitInfo(visits)	Параметри: список номерів візитів пацієнта лікарні. Пошук інформації про попередні візитипацієнта.
	getAllSymptomsForGroups()	Параметри: відсутні Пошук симптомів , що пов'язані з кожноюгрупою правил
Group Management	getSymptomsForGroup(array)	Параметри: список груп правил. Пошук симптомів для певних груп правил.
Views Module	home(request)	Http-запит.Повертає головну сторінку сайту.
	stats(request)	Http-запит.Повертає сторінку статистики.
	patients(request)	Http-запит.Якщо користувач аутентифікований, повертає список пацієнтів.



	addpatient(request)	Http-запит. Повертає сторінку створення пацієнта або створює нового пацієнта.
Views module	diagnosing(request)	Http-запит. Повертає сторінку діагностики або проводить діагностику за вказаними симптомами.
	patient_info(request, num=0)	Http-запит . Повертає сторінку з інформацією про пацієнта або додає нові дані про пацієнта.
	createSymptom(request)	Http-запит . Якщо користувач аутентифікований повертає сторінку створення симптому або створює новий симптом.
	createDisease(request)	Http-запит . Якщо користувач аутентифікований повертає сторінку створення захворювання або створює нове захворювання.
	createRule(request)	Http-запит . Якщо користувач аутентифікований повертає сторінку створення правила або створює нове правило.
	manageGroup(request)	Http-запит . Повертає сторінку управління групами правил або створює нову групу правил.
	register(request)	Http-запит . Повертає сторінку реєстрації або реєструє користувача.
	profile(request)	Http-запит . Повертає сторінку редагування облікового запису або оновлює його.

### 3.4. Інструкція користувача

Робота із застосуванням розпочинається з завантаження головної сторінки web-сайту, що зображена на рисунку 3.1.

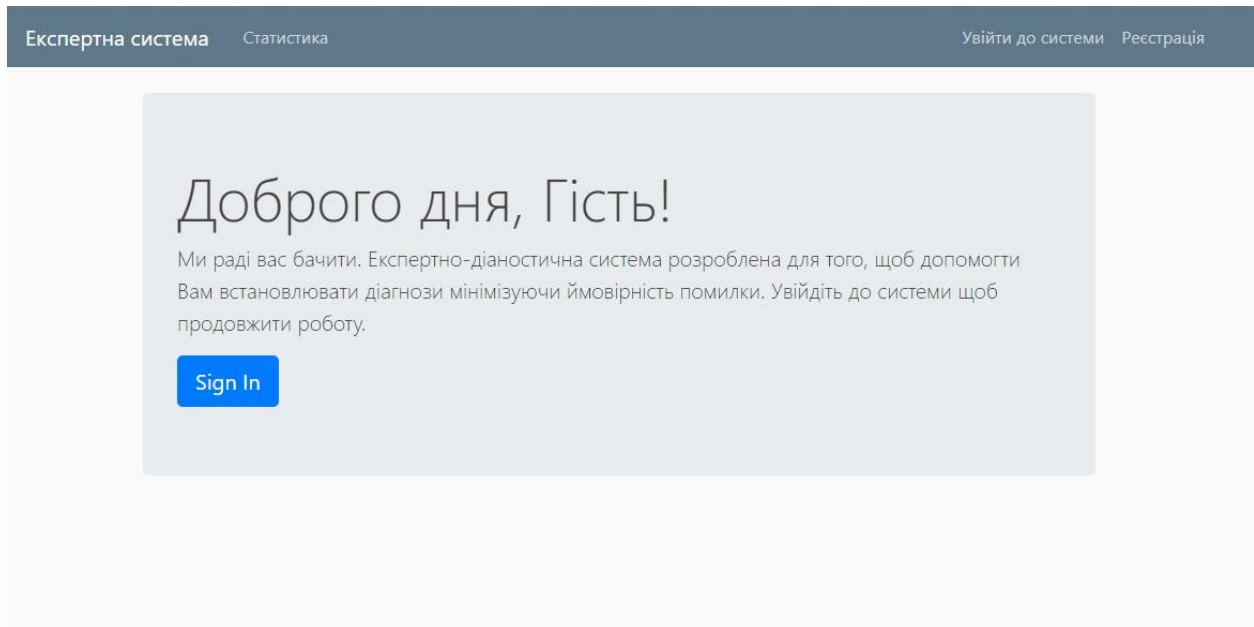


Рисунок 3.1 – Головна сторінка

Гість має наступні варіанти продовження роботи з системою:

- вхід (аутентифікація);
- реєстрація рахунку лікаря після пред'явлення реєстраційного ключа;
- перегляд статистики.

## Аутентифікація

Після натискання кнопки «Увійти» у верхньому правому куті сторінки користувач переходить на сторінку автентифікації (рис. 3.2), де він повинен ввести логін та пароль, а потім натиснути «Вхід».

Експертна система    Статистика    Увійти до системи    Реєстрація

Log In

Username\*

vlad\_vlad

Password\*

.....

Login

Need an account? [Sign Up Now](#)

Рисунок 3.2 – Сторінка аутентифікації

Якщо введені дані вірні, система перейде на головну сторінку системи (для аутентифікованих користувачів, рис. 3.3).

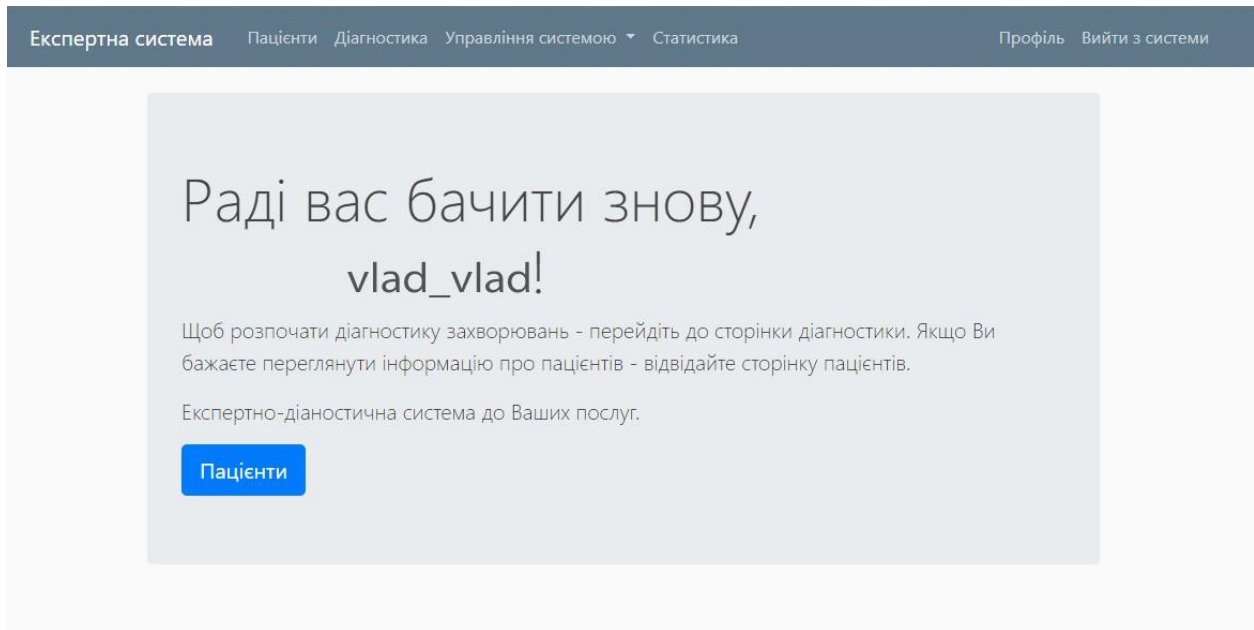


Рисунок 3.3 – Головна сторінка (для аутентифікованих користувачів)

## Views Module Реєстрація

Після натискання кнопки «Зареєструватись» у верхньому правому куті сторінки, користувач переходить на сторінку реєстрації (рис. 3.4), де він повинен ввести свої персональні дані, а також реєстраційний ключ, який надається лише кваліфікованим лікарям зі знаннями, необхідними для роботи з системою. Якщо введені дані вірні, система реєструє рахунок і вводить його в систему.

Експертна система    Статистика    Увійти до системи    Реєстрація

### Registration

Ім'я\*

Прізвище\*

Ключ реєстрації\*

Username\*

Required. 150 characters or fewer. Letters, digits and @/./+/\_ only.

Email\*

Рисунок 3.4 – Сторінка реєстрації користувача

## Персональна сторінка користувача

В кабінеті користувача (рис. 3.5), який ви отримаєте після натискання кнопки «Профіль», ви можете вносити зміни до своїх особистих даних, включаючи оновлення пароля.

Експертна система   Пацієнти   Діагностика   Управління системою ▾   Статистика   Профіль   Вийти з системи

### Мій профіль

Ім'я\*

Прізвище\*

Username\*

Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

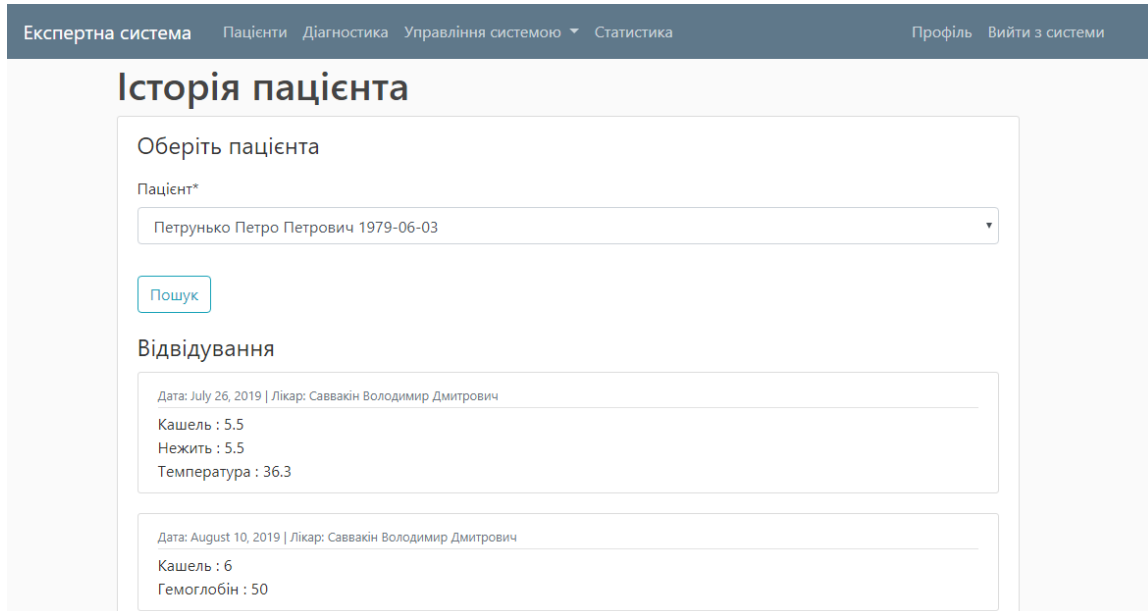
Електронна адреса\*

Оновити

Рисунок 3.5 – Особистий кабінет користувача

## Сторінка пацієнта

На веб-сайті пацієнта ви можете переглянути його персональні дані, а також історію відвідувань лікаря з відповідною датою та станом здоров'я на даний момент (рис. 3.6).



Експертна система Пацієнти Діагностика Управління системою Статистика Профіль Вийти з системи

### Історія пацієнта

Оберіть пацієнта

Пацієнт\*

Петрунько Петро Петрович 1979-06-03

Пошук

#### Відвідування

Дата: July 26, 2019 | Лікар: Саввакін Володимир Дмитрович

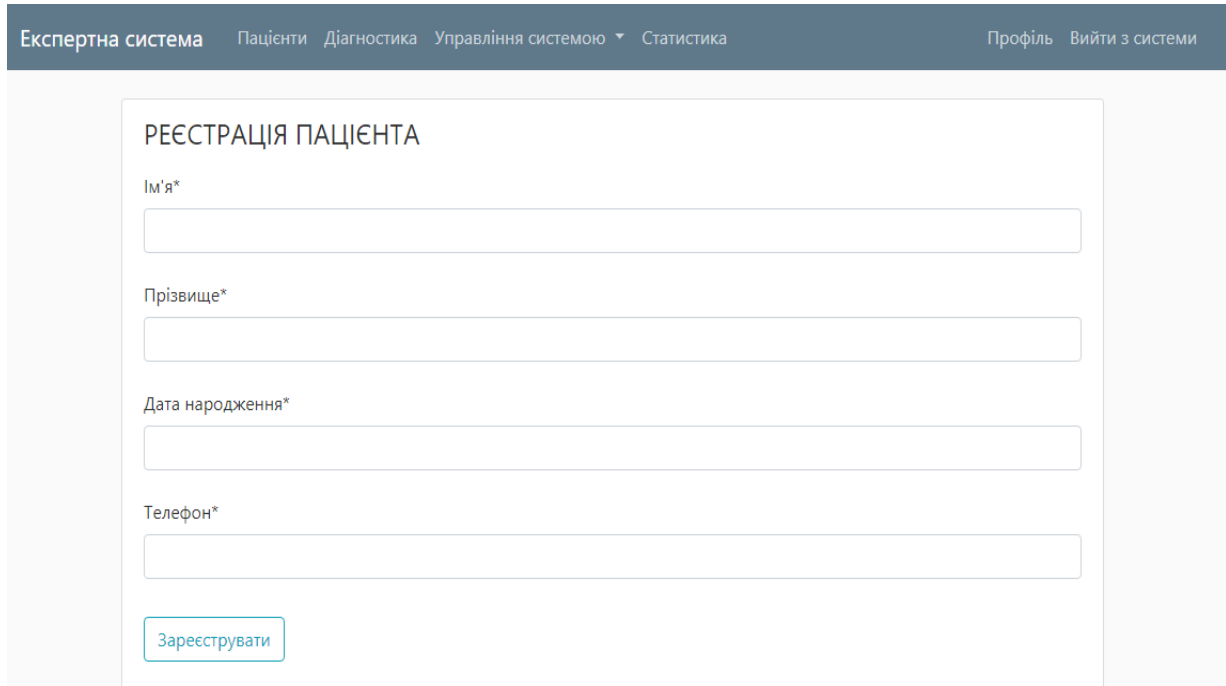
Кашель : 5.5  
Нежить : 5.5  
Температура : 36.3

Дата: August 10, 2019 | Лікар: Саввакін Володимир Дмитрович

Кашель : 6  
Гемоглобін : 50

Рисунок 3.6 – Сторінка пацієнта

Крім того, ви можете натиснути кнопку «Додати пацієнта» та створити новий обліковий запис пацієнта, якщо він вперше потрапляє до лікарні, обладнаної спеціалізованою діагностичною системою. Для цього введіть його персональні дані та натисніть «Зареєструвати» (рис. 3.7).



The image shows a web interface for patient registration. At the top, there is a dark blue navigation bar with the text 'Експертна система' on the left and 'Профіль Вийти з системи' on the right. In the center of the navigation bar are the menu items 'Пацієнти', 'Діагностика', 'Управління системою', and 'Статистика'. Below the navigation bar is a white form titled 'РЕЄСТРАЦІЯ ПАЦІЄНТА'. The form contains four input fields: 'Ім'я\*' (Name), 'Прізвище\*' (Surname), 'Дата народження\*' (Date of birth), and 'Телефон\*' (Phone). Each field is represented by a white rectangular box with a thin border. Below the input fields is a blue button with the text 'Зареєструвати' (Register).

Рисунок 3.7 – Реєстрація пацієнта



## Процес діагностики

Процес діагностики потенційних захворювань пацієнта (рис. 3.8) починається з вибору пацієнта для діагностики. Після включення пацієнта до списку лікар повинен вибрати область діагностики відповідно до заявлених пацієнтом симптомів, наприклад, симптомів застуди (рис. 3.9).

Експертна система   Пацієнти   Діагностика   Управління системою   Статистика   Профіль   Вийти з системи

### Сторінка діагностики

Крок 1: Оберіть пацієнта

Пацієнт\*

-----

Крок 2: Оберіть групу правил

Застуда

Захворювання шлункового тракту

Проблеми з зором

Сердцеві захворювання

Рисунок 3.8 – Сторінка діагностування (вибір пацієнту та області діагностування)

Після заповнення полів (симптомів) та натискання кнопки «Надіслати» на екрані відобразатиметься інформація про перелік захворювань разом із відповідною ймовірністю виникнення кожного з них (рис. 3.10).

Крок 2: Оберіть групу правил

Рисунок 3.9 – Сторінка діагностування (внесення симптомів)  
Управління базою даних про симптоми та захворювання

Рисунок 3.10 – Сторінка діагностування (отримання результатів)

Існує декілька способів управління базою даних про симптоми та захворювання. Створюйте нові симптоми. Створення симптому (рис. 3.11) полягає у введенні його назви та показників. Показниками є його межі (наприклад, 35 і 42 для температури тіла), рівні членства (низький, середній, високий симптом) та ступінь (0,1 для температури). Додавання нових хвороб.

Вводиться назва захворювання (рисунок 3.12).

Додавання нових правил взаємодії симптомів та захворювань. Вводяться симптоми та відповідні рівні симптомів, а також пов'язаний набір симптомів захворювання та їх вірогідність (рис. 3.13);

Створіть набір правил, що поєднує правила для певної діагностичної області, наприклад, серцевих захворювань. Введіть назву групи правил та симптоми та захворювання, яким вона належить (рис. 3.14).

Рисунок 3.11 – Сторінка створення симптому

Рисунок 3.12 – Сторінка створення захворювання

Створюйте нові симптоми.

Рисунок 3.13 – Сторінка створення правила

Рисунок 3.14 – Сторінка створення групи правил

## Перегляд статистики

Сторінка статистики відображає кількість лікарів, зареєстрованих у системі, а також кількість пацієнтів, групи правил, норм, симптомів та захворювань (рис. 3.15).

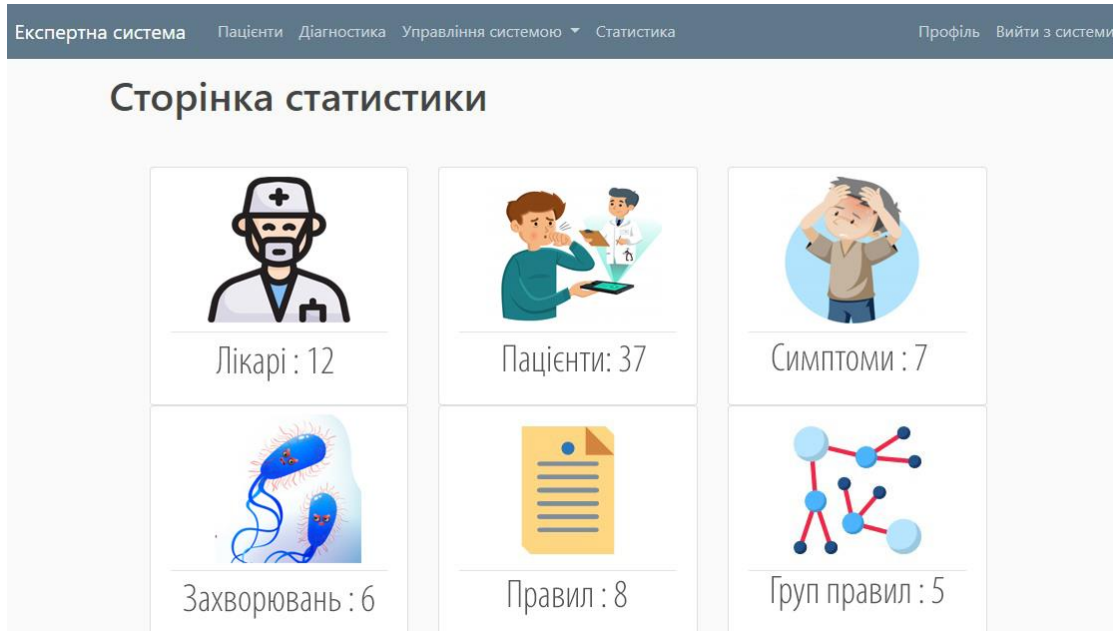


Рисунок 3.15 – Сторінка статистики

Нижче представлені зміст та ідеї проекту, можливі напрями його використання та потенційна вигода для користувачів. Інформація про перелічені пункти подана у таблиці 3.16

Таблиця 3.2 – Опис ідеї проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Експертно-діагностична система, основним призначенням якої є за результатами медичного	1. Діагностична система	Пришвидшення процесу встановлення діагнозу пацієнтом, нівелювання факторів людської помилки та можливої низької компетенції лікаря

<p>обстеження отримати список із декількох захворювань, які найбільше відповідають вказаним симптомам. Використання автоматизованих систем дозволило би видалити людську помилку при визначенні діагнозу.</p>	2. База даних пацієнтів зі збереженням історії хвороби	Автоматизація процесу ведення історії хвороби пацієнта, а також збереження інформації про попередні візити до лікаря та тодішній стан самопочуття
	3. База знань, що може доповнюватися новими даними про симптоми та хвороби	Так як інноваційні медичні дослідження постійно відкривають нові факти про симптоми та хвороби, що їх продукують, необхідно завчасно оновлювати дані про хвороби для коректного встановлення діагнозу
	4. Система-помічник, котра допомагає лікарю затвердити діагноз	Пацієнт може бути спокійним приходячи до лікаря, адже він знає, що попередній встановлений лікарем діагноз буде перевірений автоматизованою системою та відкоректований у разі необхідності. Компетенція лікаря більше не буде загрозою для пацієнта.
	5. Система раннього діагностування на базі приватних медичних лабораторій	Приватні лабораторії одразу після дослідження можуть занести результати в систему та дізнатися, чи є висока ймовірність певного захворювання, що передбачає невідкладне лікування. У разі виникнення такої ситуації замовник дослідження одразу отримує сповіщення.

Далі наведені потенційні групи клієнтів, їх характеристики, а також сформований орієнтовний перелік вимог до товару для кожної групи у таблиці 3.3.

Таблиця 3.3 – Характеристика потенційних клієнтів проекту

№ п / п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів дотовару
1.	Діагностування захворювань та встановлення попереднього діагнозу	Лікарі, співробітники медичних закладів	Для лікарів важливим є своєчасне та точне встановлення діагнозу пацієнтів для визначення коректного лікування відповідно до встановленого діагнозу. Окрім цього, важливим є репутація як окремо лікаря, так і медичного закладу в цілому, тому наявність експертно-діагностичної системи зможе суттєво підвищити рівень довіри суспільства до медичного закладу	Коректність встановленого діагнозу Швидкість встановлення діагнозу Можливість оновлювати базу знань
		Пацієнти лікарень	Підчас поганого самопочуття більшість людей одразу не звертаються до медичних закладів із багатьох причин, однією з яких є невпевненість у кваліфікації лікарів. Наявність експертно-діагностичної системи в користуванні лікаря, до якого звертається пацієнт, може вплинути на рішення пацієнти звернутися до лікарні раніше, а як наслідок – коректно встановити діагноз та вчасно розпочати лікування	Коректність встановленого діагнозу Швидкість встановлення діагнозу Можливість оновлювати базу знань

Таблиця 3.3 – Характеристика потенційних клієнтів проекту

2.	Збереження історії хвороби пацієнтів	Лікарі, співробітники медичних закладів	Для лікарів не є можливим гарантувати збереження усіх документів пацієнтів: результатів аналізів, стану самопочуття за попередні відвідування лікаря, історію хвороби. Бажаним є перехід до електронного документообігу	Можливість за короткий час переглянути історію попередніх відвідувань пацієнтів та стану тодішнього здоров'я
		Пацієнти лікарень	Кожного разу пацієнти, що звертаються до лікаря, беруть з собою всі можливі папери, які мають відношення до стану здоров'я. Часто ці папери губляться, а з ними втрачається і історія хвороби пацієнта, відновити яку стає або дуже складно, або неможливо. Збереження всієї інформації про пацієнта в системі допоможе позбавити людей від паперових медичних карток та перенести всі дані до електронного сховища, з якого лікар за лічені хвилини зможе дістати всю необхідну інформацію	Позбавлення від необхідності тримати «на руках» всі документи, що засвідчують стан здоров'я за минулі візити до лікарів



## ВИСНОВКИ

В процесі написання магістерської роботи були отримані результати, що поєднують теоретичні дослідження систем нечіткого логічного висновку з їх практичним застосуванням у задачах діагностики потенційних захворювань пацієнтів. Розроблена експертно-діагностична система, алгоритм діагностики якої базується на правилах нечіткого висновку.

Запропонований підхід дозволяє суттєво знизити рівень залежності правильності обраного методу лікування пацієнта від професійних компетенцій лікарів. Більше того, використання автоматичних систем усуне фактор людської помилки в діагностиці. Позбавлення від вищезазначених недоліків підвищить якість медичної допомоги та скоротить час первинної діагностики.

Основним функціональним рішенням цієї системи є діагностика потенційних захворювань пацієнта, яка полягає у первинному виявленні пацієнта, а потім введенні симптомів пацієнта в систему.

Результатом діагностичної операції є перелік захворювань, пов'язаних з цими симптомами, та ймовірність їх виникнення. Окрім функції діагностики, важливим завданням експертних систем є надання оператору необхідності змінювати базу знань системи зміни програмного коду.

Зручною функцією є збереження історії відвідувань пацієнтів та надання користувачам статистичної інформації про систему.

Для реалізації експертно-діагностичної системи була використана технологія Django Web Framework: серверна частина програми була написана на Python в тандемі з базою даних MySQL, а користувальницький інтерфейс був реалізований за допомогою HTML, CSS та JavaScript (jQuery + Select2). Розроблений інтерфейс програми є адаптивним, тобто можна буде зручно користуватися додатком незалежно від діагоналі екрана комп'ютера і навіть телефону чи планшета.

Альтернативним додатком є використання експертних діагностичних систем на базі державних або приватних медичних лабораторій, останні з яких

дуже популярні серед громадян. Вже сьогодні лабораторії надають не тільки чисельну інформацію про рівень того чи іншого компонента в організмі людини, але й проводять побіжний аналіз результатів.

Розроблена система може доповнити обладнання, доступне в лабораторіях, а у випадку критичних результатів якомога швидше повідомити клієнта електронною поштою про необхідність звернутися до лікаря.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Науковий журнал «Проблеми програмування», №2, 2012. С. 443 - 450.
2. Провотар О.І. Про нові методи опису невизначених величин / Провотар О.І., Лапко О.В. //Проблеми програмування. – №4. – 2012. С. 35-43.
3. Bellman R.E. Decision-Making in a Fuzzy Environment./ Bellman R.E., Zadeh L.A. // Management Science. – 1970. – Vol.17, №4. – p.141-164.
4. Provotar A. I. Fuzzy inference systems and their applications /A. I. Provotar, A. V. Lapko, A. A. Provotar //Cybernetics and Systems Analysis. – 2013. – Vol. 49 – p. 517-525.
5. Sugeno M. The Theory of Fuzzy Integrals and Its Applications /Sugeno M. // PhD thesis, Tokyo Institute of Technology, Tokyo. – 1974.
6. Нечітка логіка [Електронний ресурс]. – Режим доступу: <http://www.victoria.lviv.ua/html/oio/html/theme11.htm> – Назва з екрана.
7. Список Інтернет-ресурсів по теорії нечетких множин [Електронний ресурс].  
– Режим доступу: <http://matlab.exponenta.ru/fuzzylogic/links/default.php>.
8. Rutkowski, L. Metody i Techniki Sztucznej Inteligencji (in Polish). Wydawnictwo Naukowe PWN, Warszawa, 2009. 452 s.
9. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы  
  
/Рутковская Д., Пилиньский М., Рутковский Л.; Москва: Телеком, 2006. 382 с.
10. Zadeh L.A. Fuzzy Sets/ Zadeh L.A. // Information and Control. 1965. Vol.8 p. 338- 353
11. Hodge R. Key Terms in Fuzzy Logic Deep Roots and New Understanding /Hodge R.  
  
//University of Western Sydney, Australia (Private Communications). – 2000.

12. Zadeh L.A. From Computing with Numbers to Computing with Words – From Manipulation of Measurements to Manipulation of Perceptions /Zadeh L.A., P.P. Wang // Computing With Words. New York: Wiley and Sons, – 2001.
13. Zadeh L.A. Fuzzy sets as a basis for a theory of possibility /Zadeh L.A. //Fuzzy Sets Systems. – 1978. – Vol. 1.
14. Zadeh L.A. Fuzzy Sets / Zadeh L.A. // Information and Control. – 1965. – Vol.8.
15. Zadeh L.A. Probability Measures of Fuzzy Events /Zadeh L.A. //Journal of Mathematical Analysis and Applications. – 1968. – Vol.10. – p. 421-427.
16. James L. Crowley // Intelligent Systems: Reasoning and Recognition. [Электронный ресурс]. – Режим доступа: <http://www-prima.imag.fr/Prima/Homepages/jlc/Courses/2011/ENSI2.SIRR/ENSI2.SIRR.S2.pdf>.
17. Randolph A. Miller / Harry E. Pople, Jr. / Jack D. Myers // An Experimental Computer-Based Diagnostic Consultant for General Internal Medicine [Электронный ресурс]. – Режим доступа: <http://people.dbmi.columbia.edu/~ehs7001/Clancey-Shortliffe-1984/Ch8.pdf>.
18. V.S. Jadhav // Review of Application of Expert Systems in the Medicine [Электронный ресурс].–Режим доступа: [http://nci2tm.sinhgad.edu/NCIT2M2014\\_P/ data/NCI2TM\\_31.pdf](http://nci2tm.sinhgad.edu/NCIT2M2014_P/ data/NCI2TM_31.pdf).
19. Mohamed A. Madkour / Mohamed Roushdy // Methodology for Medical Diagnosis based on Fuzzy Logic. [Электронный ресурс]. – Режим доступа: [https://www.kau.edu.sa/Files/0052366/Researches/29055\\_Med%20Fuzzy%20ES.doc](https://www.kau.edu.sa/Files/0052366/Researches/29055_Med%20Fuzzy%20ES.doc).
20. IDE Visual Studio Code // Режим доступа: <https://code.visualstudio.com/>
21. Django Web Framework Documentation // Режим доступа: <https://www.djangoproject.com/>
22. MySQL RDBMS // Режим доступа: <https://www.mysql.com/>
23. Искусственный интеллект и медицина: онлайн чат-бот [Электронный

- ресурс]. — 2018. — URL: <http://medmarketing.ua/blog/meditsinskie-chat-boty-umnye-pomoshchniki-vrachey/> (дата обращения: 25.02.2021).
24. Розыходжаева Д.А. Сравнительный анализ моделей принятия решений в вопросах диагностики заболеваний // Проблемы Науки — 2017 — №15
25. [Электронный ресурс] — URL: <https://cyberleninka.ru/article/n/sravnitelnyy-analiz-modeley-prinyatiya-resheniy-v-voprosah-diagnostiki-zabolevaniy> (дата обращения: 14.02.2021).
26. Шепетухина Я. И. Экспертная система диагностики легочных заболеваний // Вестник НТУ ХПИ — 2005 — №46. [Электронный ресурс] — URL: <https://cyberleninka.ru/article/n/ekspertnaya-sistema-diagnostiki-legochnyh-zabolevaniy> (дата обращения: 14.02.2021).
27. Bootstrap Web Framework // Режим доступа: <https://getbootstrap.com/>
28. Веб-сайт з підручниками по HTML, CSS та JS [Електронний ресурс] // Режим доступу: <https://www.w3schools.com/>;
29. В.Д. Саввакін, О.І. Провотар / Експертно-діагностична система на основі правил нечіткого логічного виведення // Матеріали ІІІ всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2019) – м. Київ.: НТУУ «КПІ ім. Ігоря Сікорського», 20-22 листопада 2019 р. – С. 32-35.
30. Білоконна К.В., Войтко В.В., Бевз С.В. Комп'ютерна програма "Комп'ютерна програма динамічної ідентифікації елементів керування шляхом моніторингу процесів візуалізації результатів роботи" // Свідоцтво про реєстрацію авторського права на твір № 54949, від 23.05.2014.- 2014.- Свідоцтва про державну реєстрацію авторського права на твір.
31. Білоконна К.В., Зінчук В.С., Войтко В.В. Комп'ютерна програма "Програмний модуль розпізнавання тексту із зображення" // Свідоцтво про реєстрацію авторського права на твір №5998, від 05.01.2015.-2015.- Свідоцтво про державну реєстрацію авторського права на твір.

32. Білоконна К.В. Новітні технології розробки web-додатків // В. В. Войтко, К. В. Білоконна. – Збірник матеріалів Міжнародної науково-практичної Інтернет-конференції «Інноваційні та інформаційні технології в бізнесі та освіті» – Вінниця 2015;
33. Белоконная К.В. Рынок мобильных гаджетов // К. В. Белоконная, А.Н. Романюк, О.А. Сивец. – Збірник матеріалів Всеукраїнської науковопрактичної Інтернет-конференції 2013 року «Електронні інформаційні 83 ресурси в освіті і науці: створення, використання, доступ» – Вінниця 2013. – С. 14-16.
34. Білоконна К.В. Принципи реалізації ігрових програм // К. В. Білоконна – Електронне наукове видання матеріалів конференції «XLII Регіональна науково-технічна конференція професорсько-викладацького складу, співробітників та студентів університету з участю працівників науководослідних організацій та інженерно-технічних працівників підприємств м. Вінниці та області» – [conf.vntu.edu.ua/alvntu/2013/initki/txt/bilokonna.pdf](http://conf.vntu.edu.ua/alvntu/2013/initki/txt/bilokonna.pdf).
35. Білоконна К.В. Активізація навчального процесу з використанням концепції GBL // С.В. Бевз, В.В. Войтко, С.М. Бурбело, К.В. Білоконна - Міжнародний науково-технічний журнал «Вимірювальна та обчислювальна техніка в технологічних процесах» – Хмельницький 2014. – № 4. – С. 236-240.
36. Білоконна К.В. Технологія навчального процесу з використанням концепції GBL // Бевз С.В., Войтко В.В., Бурбело С.М., Білоконна К.В. - Збірник матеріалів Міжнародної науково-практичної Інтернет-конференції «Електронні інформаційні ресурси: створення, використання, доступ» – Вінниця 2014. – С. 54-62.
37. Портал «Дніпромед». [Електронний ресурс]. – Режим доступу: <http://dnipromed.com/> 13.Портал «Поліклініка» [Електронний ресурс]. – Режим доступу: <https://clinica2.munis.com.ua/>
38. Портал «Дос.уа». [Електронний ресурс]. – Режим доступу: <https://doc.ua/>

39. Аналіз та обґрунтування вибору засобів реалізації інформаційних веб-сайтів . [ електронний ресурс] / М. С. Федорова. – Режим доступу до статті: <http://conf.vstu.vinnica.ua/allvntu/2012/initki/pz.php>.
40. XML. [Електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/XML>
41. Microsoft Visual Studio.[Електронний ресурс]. – Режим доступу: [https://ru.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio)
42. Рухляда В. С., Карпенко М. Ю. Розробка персонального WEB-сайту. / В. С. Рухляда, М. Ю. Карпенко // Харків: ХНАМГ. – 2008. 84
43. ASP.NET MVC. [Електронний ресурс]. – Режим доступу: <http://www.asp.net/mvc>
44. Kurtz Jamie, Wortman Brian. ASP.NET Web API 2: Building a REST Service from Start to Finish. / Jamie Kurtz, Brian Wortman // ISBN13: 978-1-484201-10-7
45. CSS-практикум, часть 1: назначение и возможности CSS. [Електронний ресурс]. – Режим доступу: <http://site.nic.ru/content/view/114/37>.
46. Романюк О. Н. Веб-дизайн і комп'ютерна графіка: навчальний посібник / О. Н. Романюк, Д. І. Кательников, О. П. Косо́вєць. – В: ВНТУ, 2007. – 147 с.
47. Тестування програмного забезпечення. [Електронний ресурс] / Wikipedia. – Режим доступу: [http://en.wikipedia.org/wiki/ Software\\_testing](http://en.wikipedia.org/wiki/Software_testing).
48. Локалізація програмного забезпечення [Електронний ресурс] / Wikipedia. – Режим доступу статті: [http://en.wikipedia.org/wiki/ Localization](http://en.wikipedia.org/wiki/Localization).
49. Тестування веб-сайтів. [Електронний ресурс]. – Режим доступу: <http://arhivstatey.pp.ua/index.php?newsid=26947>.
50. Грабовецький Б. Є. Економіка підприємства. / Б. Є. Грабовецький, Т.М. Пілявоз – Вінниця: ВНТУ, 2009 – 248 с.
51. Методичні рекомендації з комерціалізації розробок, створених в результаті науково-технічної діяльності – К.: Наказ Державного комітету України з питань науки, інновації та інформатики (Лист № 1/06-4-97 від

13.09.2010 р.).

52. Козловський В.О., Лесько О. Й. Бізнес-планування: Навчальний посібник / В. О. Козловський, О. Й. Лесько [2-е вид., доп. Та переробл.] – Вінниця: УНІВЕРСУМ-Вінниця, ВНТУ, 2008. – 241
53. Козловський В.О. Техніко-економічні обґрунтування та економічні розрахунки в дипломних проектах та роботах. Навчальний посібник / В.О. Козловський – Вінниця: ВНТУ, 2003. – 75
54. Козловський В.О. Економіка, організація виробництва та менеджмент в дипломних роботах. Навчальний посібник / В.О. Козловський – Вінниця: ВНТУ, 2004. – 94 с.



## ДОДАТОК А ГРАФІЧНИЙ МАТЕРІАЛ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ/НАВЧАЛЬНО  
НАУКОВИЙ ІНСТИТУТ ЗАОЧНОГО ТА ДИСТАНЦІЙНОГО  
НАВЧАННЯ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Магістерська робота на тему:  
«Розробка програмного забезпечення для пошуку лікаря за  
профілем мовою Java»

Виконує:

студент 7 курсу, групи ППЗМ-71  
за спеціальністю Програмна Інженерія  
Коломійчук В.В.

Керівник:

Негоденко О.В., ст. завідувач кафедри.  
кандидат технічних наук, доцент

### МЕТА РОБОТИ, ОБ'ЄКТ І ПРЕДМЕТ ДОСЛІДЖЕНЬ

- **Мета роботи** – розробка системи, яка за допомогою методів класифікатора виявить одне з можливих захворювань після введення симптомів та організує відповідний візит до лікаря
- **Об'єкт досліджень** – процес автоматизації діагностування захворювань пацієнтів по симптомам
- **Предмет досліджень** – методи діагностування захворювань за допомогою експертно-діагностичних систем

## АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

- Система INTERNIST-I
- Портал «Дніпромед»
- Інформаційна система «Поліклініка»
- Портал «Doc.ua»
- Електронна система «eHealth»

3

## ЗАДАЧІ ПРОЕКТУ

**Для досягнення поставленої мети необхідно вирішити такі завдання:**

- Провести аналіз і зробити загальну характеристику предметної області
- Провести аналіз існуючих рішень
- Розробити вимоги до додатку
- Дослідити технології проектування
- Провести вибір засобів розробки і проектування
- Спроекувати структуру бази даних та вибрати відповідні алгоритми для реалізації системи
- Реалізувати додаток

4

## ВИБІР ЗАСОБІВ РОЗРОБКИ І ПРОЕКТУВАННЯ

✓ Java



✓ IntelliJ IDEA



✓ Django Web Framework



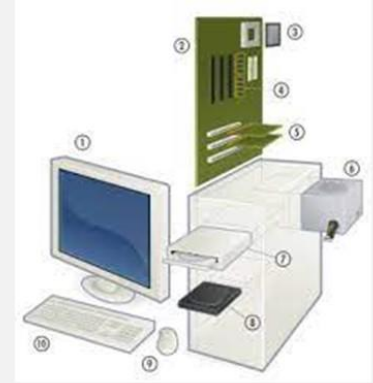
✓ Hibernate



✓ HTML/CSS/JavaScript (jQuery + Select2)

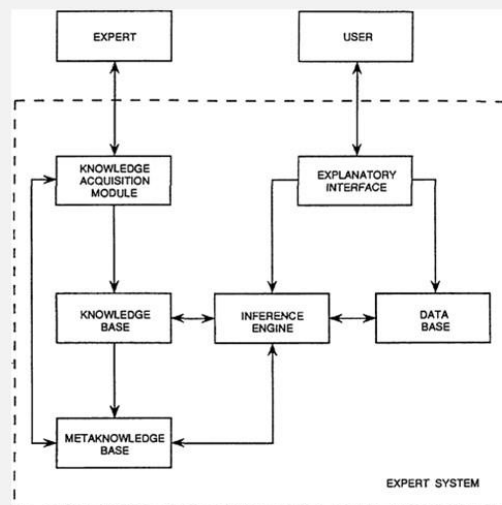


✓ MySQL



5

## ЕКСПЕРТНА СИСТЕМА (АРХІТЕКТУРА)



6

## ІНТЕРФЕЙС ДОДАТКУ

Експертна система | Пацієнти | Діагностика | Управління системою | Статистика | Профіль | Вийти з системи

### РЕЄСТРАЦІЯ ПАЦІЄНТА

Ім'я\*

Прізвище\*

Дата народження\*

Захворіння\*

Зареєструвати

Експертна система | Пацієнти | Діагностика | Управління системою | Статистика | Профіль | Вийти з системи

### Історія пацієнта

Оберіть пацієнта

Пацієнт\*

Петрушич Петро Петрович 1979-08-01

Почати

#### Відвідування

Дата відвідування: 08.08.2019 | Назв. Сесії: Водяникова Діагностика

Кількість: 155

Температура: 38,3

Дата відвідування: 08.08.2019 | Назв. Сесії: Водяникова Діагностика

Кількість: 8

Температура: 38

Експертна система | Пацієнти | Діагностика | Управління системою | Статистика | Профіль | Вийти з системи

### Сторінка діагностики

Крок 1: Оберіть пацієнта

Пацієнт\*

Крок 2: Оберіть групу правил

Застуда

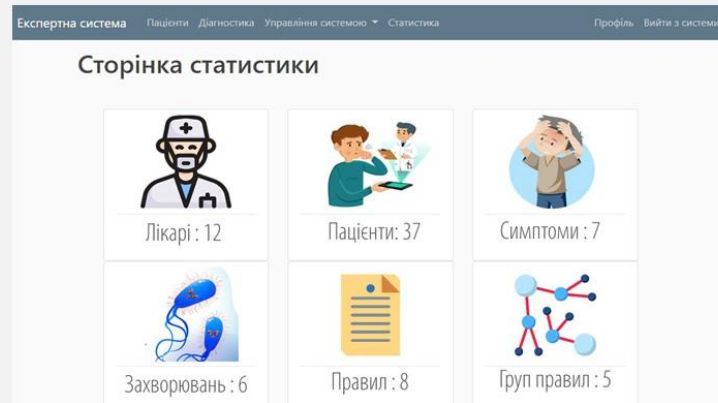
Захворювання шлункового тракту

Проблеми з зором

Серцеві захворювання

7

## ІНТЕРФЕЙС ДОДАТКУ



8

## ВИСНОВКИ

Для досягнення мети проекту було вирішено такі задачі:

- Проведено аналіз існуючих рішень і зроблена загальна характеристика предметної області
- Досліджено технології проектування і розроблені вимоги до додатку
- Проведений вибір засобів розробки і проектування
- Спроектовано структуру бази даних
- Реалізовано додаток

**Новий додаток дозволяє суттєво знизити рівень залежності правильності обраного методу лікування пацієнта від професійних компетенцій лікарів, використання автоматичних систем усуне фактор людської помилки в діагностиці, підвищить якість медичної допомоги та скоротить час первинної діагностики**