

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ПрАТ «ПРИВАТНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД «ЗАПОРІЗЬКИЙ  
ІНСТИТУТ ЕКОНОМІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ»

Кафедра Інформаційних технологій

ДО ЗАХИСТУ ДОПУЩЕНА

Зав.кафедрою \_\_\_\_\_  
д.е.н.,доцент Левицький С.І.  
«\_\_\_» \_\_\_\_\_ 2023 р.

МАГІСТЕРСЬКА ДИПЛОМНА РОБОТА  
РОЗРОБКА СИСТЕМИ АВТОМАТИЗОВАНОГО МОНІТОРИНГУ  
РЕЗУЛЬТАТІВ МЕДИЧНИХ АНАЛІЗІВ

Виконав  
ст. гр. КІ-112м

\_\_\_\_\_  
(підпис)

А.М. Кібітов

Керівник  
к.т.н.

\_\_\_\_\_  
(підпис)

О.А. Хараджян

Запоріжжя  
2023

ПРАТ «ЛВНЗ «ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ  
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ»

Кафедра Інформаційних технологій

ЗАТВЕРДЖУЮ  
Зав. кафедрою

\_\_\_\_\_  
д.е.н., доцент Левицький С.І.  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.

ЗАВДАННЯ  
НА МАГІСТЕРСЬКУ ДИПЛОМНУ РОБОТУ

Студенту гр. КІ – 112м, спеціальності «Комп'ютерна інженерія»

\_\_\_\_\_  
*Кібітов Артем Миколайович*

1. Тема: *Розробка системи автоматизованого моніторингу результатів медичних аналізів.*

затверджена наказом по інституту « \_\_\_\_ » \_\_\_\_\_ 2022 р. № \_\_\_\_\_

2. Термін здачі студентом закінченої роботи: « \_\_\_\_ » \_\_\_\_\_ 2023 р.

3. Перелік питань, що підлягають розробці:

1. Аналіз систем автоматизації в медицині;
2. Аналіз систем охорони здоров'я на основі IoT;
3. Системи медичної автоматизації;
4. Розробка структури системи автоматизованого моніторингу медичних аналізів;
5. Розробка підсистеми відображення даних системи автоматизованого моніторингу;
6. Розробка структури даних програми системи;

7. Розробка програмних модулів системи автоматизованого моніторингу аналізів;
8. Розробка класів і методів для обміну даними;
9. Розробка класів і методів для взаємодії з базою даних;
10. Тестування системи.

#### 4. Календарний графік підготовки кваліфікаційної роботи

№ етапу	Зміст	Терміни виконання	Готовність по графіку %, підпис керівника	Підпис керівника про повну готовність етапу, дата
1	Збір практичного матеріалу за темою кваліфікаційної бакалаврської роботи	04.09.23-17.10.23		
2	I атестація I розділ кваліфікаційної бакалаврської роботи	23.10.23-28.10.23		
3	II атестація II розділ кваліфікаційної бакалаврської роботи	20.11.23-25.11.23		
4	III атестація III розділ кваліфікаційної бакалаврської роботи, висновки та рекомендації, додатки, реферат	18.12.23-23.12.23		
5	Перевірка кваліфікаційної бакалаврської роботи на оригінальність	18.12.23-23.12.23		
6	Доопрацювання кваліфікаційної бакалаврської роботи, підготовка презентації, отримання відгуку керівника і рецензії	25.12.23-06.01.24		
7	Попередній захист кваліфікаційної бакалаврської роботи	08.01.24-13.01.24		
8	Подача кваліфікаційної бакалаврської роботи на кафедру	за 3 дні до захисту		
9	Захист кваліфікаційної бакалаврської роботи	15.01.24-20.01.24		

Дата видачі завдання: 16.01.2023 р.

Керівник кваліфікаційної  
бакалаврської роботи

\_\_\_\_\_

(підпис)

О.А. Хараджян

\_\_\_\_\_

(прізвище та ініціали)

Завдання отримав до виконання

\_\_\_\_\_

(підпис)

А.М. Кібітов

\_\_\_\_\_

(прізвище та ініціали)

## РЕФЕРАТ

Дипломна робота містить 62 стор., 9 рис., 0 таблиць, 3 додаток, 7 використаних джерел.

Об'єкт роботи: системи медичного моніторингу.

Предмет роботи: системи моніторингу результатів медичних аналізів.

Мета роботи: розробити систему автоматизованого моніторингу результатів медичних аналізів.

Задачі роботи: аналіз систем автоматизації в медицині; аналіз систем охорони здоров'я на основі IoT; системи медичної автоматизації; розробка структури системи автоматизованого моніторингу медичних аналізів; розробка підсистеми відображення даних системи автоматизованого моніторингу; розробка структури даних програми системи; розробка програмних модулів системи автоматизованого моніторингу аналізів; розробка класів і методів для обміну даними; розробка класів і методів для взаємодії з базою даних; тестування системи.

ІОТ, ІОМТ, СИСТЕМА МОНІТОРИНГУ, МЕДИЧНІ АНАЛІЗИ,  
ФРЕЙМВОРК QT

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	6
ВСТУП	7
Розділ 1 СИСТЕМИ АВТОМАТИЗАЦІЇ В МЕДИЦИНІ.....	9
1.1. Системи охорони здоров'я на основі IoT .....	9
1.2. Системи медичної автоматизації.....	20
Розділ 2 РОЗРОБКА СИСТЕМИ АВТОМАТИЗОВАНОГО МОНІТОРИНГУ .....	31
2.1. Загальна структура системи .....	31
2.2. Розробка підсистеми відображення даних .....	33
2.3. Структури даних програми .....	39
Розділ 3 ОПИС ПРОГРАМИ АВТОМАТИЗОВАНОГО МОНІТОРИНГУ АНАЛІЗІВ .....	43
3.1. Загальна структура програми... <b>Ошибка! Закладка не определена.</b>	
3.2. Класи і методи для обміну даними.....	49
3.3. Методи для взаємодії з базою даних.....	52
ВИСНОВКИ .....	57
РЕКОМЕНДАЦІЇ .....	60
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	61
ДОДАТКИ .....	79

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,  
СКОРОЧЕНЬ І ТЕРМІНІВ

Слово / словосполучення	Скорочення	Умови використання
E		
Електрокардіограма	ЕКГ	
Електронні медичні записи	ЕМЗ	
C		
Comma-separated values	CSV	
I		
Internet of medical things	IoMT	
Internet of things	IoT	
W		
Wireless body area network	WBAN	
Wireless sensors net	WSN	

## ВСТУП

Система охорони здоров'я є однією з останніх систем, яка використовує комп'ютерні та інформаційні технології у своїй повсякденній діяльності, хоча швидкість адаптації прискорюється. На початку 20 століття, коли заклади медичної допомоги ставили перед собою задачу автоматизувати всі клінічні та управлінські системи, практично ніхто не міг забезпечити інтегроване рішення для всіх інформаційних потреб. Одне із питань полягало в тому, чи варто створювати власну систему чи краще придбати готове рішення, яке не повністю відповідатиме певним потребам. До процесів, що можуть бути автоматизовані можна віднести: створення електронної медичної карти, починаючи з автоматизованих нотаток лікаря та медсестри, архівування даних про відвідування, аналіз і застосування в управлінні процесами, і онлайн система збору даних що забезпечувало аналіз і вдосконалення процесу в реальному часі.

Сучасний стрімкий розвиток інформаційних технологій також спонукає до розвитку усі існуючі галузі, зокрема і галузь медицини. За останні 50 років зросла загальна кількість та точність діагностичних процедур. Кількість аналізів, їх якість та об'єм також зросли. Це все призводить до збільшення паперової роботи. Необхідно проаналізувати та побачити відхилення. А якщо це стосується хронічних хвороб проблема інтерпретації аналізів зростає майже в геометричній прогресії.

Саме тому створення та розвитку систем медичного моніторингу активно розвиваються останні роки.

Об'єкт роботи: системи медичного моніторингу.

Предмет роботи: системи моніторингу результатів медичних аналізів.

Мета роботи: розробити систему автоматизованого моніторингу результатів медичних аналізів.

Задачі роботи:

- аналіз систем автоматизації в медицині;
- аналіз систем охорони здоров'я на основі IoT;
- системи медичної автоматизації;
- розробка структури системи автоматизованого моніторингу медичних аналізів;
- розробка підсистеми відображення даних системи автоматизованого моніторингу;
- розробка структури даних програми системи;
- розробка програмних модулів системи автоматизованого моніторингу аналізів;
- розробка класів і методів для обміну даними;
- розробка класів і методів для взаємодії з базою даних;
- тестування системи.



## РОЗДІЛ 1

### СИСТЕМИ АВТОМАТИЗАЦІЇ В МЕДИЦИНІ

#### 1.1. Системи охорони здоров'я на основі IoT

Системи охорони здоров'я на основі Інтернету речей. Термін «Інтернет речей» (IoT) був винайдений Кевіном Ештоном у 1999 році та стосується даних в Інтернеті, пов'язаних із глобальною архітектурою послуг. IoT є продуктом передових досліджень інформаційно-комунікаційних технологій. Така технологія потенційно може підвищити якість життя міських жителів. Оскільки населення планети зростає з дивовижною швидкістю, а також зростає поширеність хронічних захворювань, зростає попит на розробку економічно ефективних систем охорони здоров'я, які можуть ефективно керувати та надавати широкий спектр медичних послуг, одночасно знижуючи загальні витрати. IoT останнім часом став ключовим напрямком розвитку суспільства, що дозволяє в тому числі вдосконалювати систему моніторингу охорони здоров'я. Система моніторингу охорони здоров'я IoT спрямована на точне відстеження даних про стан здоров'я людей і підключення різних служб і речей у світі через Інтернет для збору, обміну, моніторингу, зберігання та аналізу даних, створених цими речами. Однак IoT – це парадигма, де всі пов'язані фізичні об'єкти в будь-якій інтелектуальній програмі, наприклад «розумне місто», «розумний дім» і «розумна охорона здоров'я», адресуються та керуються віддалено. Діагностика розладів і моніторинг пацієнтів є важливими для надання медичної допомоги, і застосування сенсорних мереж до людського тіла значно допоможе в цьому. Крім того, інформація легко доступна з будь-якої точки світу в будь-який час.

Пацієнти з важкими травмами або з певних регіонів можуть мати труднощі з відвідування лікарні. Тому вони можуть використовувати

відеоконференції для спілкування зі своїми лікарями, щоб покращити своє здоров'я, заощаджуючи гроші та час. Пацієнти можуть використовувати цю технологію для запису стану свого здоров'я на свої телефони. Очікується, що переваги Інтернету речей будуть покращені та призведуть до індивідуального лікування, покращуючи результати пацієнтів, заощаджуючи витрати на управління охороною здоров'я. Системи IoT дозволяють лікарям дистанційно стежити за своїми пацієнтами та ефективніше планувати прийом. Пацієнти також можуть покращити своє медичне обслуговування вдома, щоб зменшити потребу у відвідуваннях лікаря та ймовірність отримання непотрібного або невідповідного лікування в лікарнях чи клініках. З цієї причини якість медичного обслуговування та загальна безпека пацієнтів можуть покращитися, а загальна вартість лікування може знизитися. Незабаром ми матимемо доступ до системи моніторингу здоров'я, якою можна буде користуватися, не виходячи з дому, і оптимізувати процеси в лікарнях. Датчики IoT мають бути щільно розташовані, щоб постійно контролювати тіло та навколишнє середовище. Ці зусилля дозволять відстежувати прогрес лікування хронічних захворювань і реабілітації. У майбутньому віртуальних консультацій для дистанційної медичної допомоги IoT зможе забезпечувати ефективні з'єднання даних з

Системи охорони здоров'я на основі Інтернету речей та їх застосування полегшують життя людей різними способами.

Дистанційна медична допомога: бездротові рішення на основі Інтернету речей надають медичну допомогу пацієнтам, а не доставляють пацієнта до медичної допомоги. Дані збираються безпечно за допомогою датчиків на основі Інтернету речей, і дані аналізуються за допомогою алгоритму, перш ніж надаватися медичним працівникам для отримання відповідних рекомендацій.

Моніторинг у режимі реального часу: датчики неінвазивного моніторингу, керовані IoT, збирають повну інформацію. Зберіганням даних керують шлюзи та хмарний аналіз.

Профілактична допомога: системи охорони здоров'я IoT використовують дані датчиків, які допомагають завчасно виявляти надзвичайні ситуації та сповіщають членів родини. Машинне навчання для відстеження тенденцій здоров'я та раннього виявлення аномалій досягається через підхід IoT.

Що стосується того, як програми, пристрої та люди спілкуються один з одним, щоб надавати медичні рішення, ми вийшли на абсолютно новий рівень еволюції. IoT дав нам нову перспективу та інструменти для інтегрованої мережі охорони здоров'я, що значно покращує якість медичної допомоги.

IoT дає можливість автоматизувати системи охорони здоров'я, які раніше вимагали значної кількості часу та залишали місце для помилок через участь людини. Наприклад, щоб контролювати повітряний потік і температуру в операційних, багато лікарень тепер використовують мережеві пристрої.

Існує майже безмежна кількість способів, якими IoT може покращити медичне обслуговування; однак, нижче наведено деякі з ключових переваг:

- зниження вартості догляду;
- зменшення помилки людини;
- усунення обмежень відстані;
- зменшена кількість паперової роботи та ведення документації;
- хронічні захворювання виявляються рано;
- покращення в розподілі та призначенні ліків;
- необхідність швидкої медичної допомоги;
- кращі результати лікування.

Як показано на рис. 1.1 архітектура IoWT та її з'єднань складається з трьох елементів: WBAN, шлюзу, підключеного до Інтернету, та хмари. WBAN – це зовнішній компонент IoWT, який обертається навколо тіла, щоб непомітно збирати дані про здоров'я. WBAN збирає дані від датчиків, які безпосередньо контактують з тілом, або від датчиків у навколишньому

середовищі, які можуть збирати непрямі дані про поведінку людини. WBAN може аналізувати дані або передавати їх для дистанційного аналізу. Крім того, мобільні обчислювальні пристрої, такі як смартфони, планшети та ноутбуки, повинні бути підключені до Інтернету, щоб надсилати дані до потужних обчислювальних ресурсів

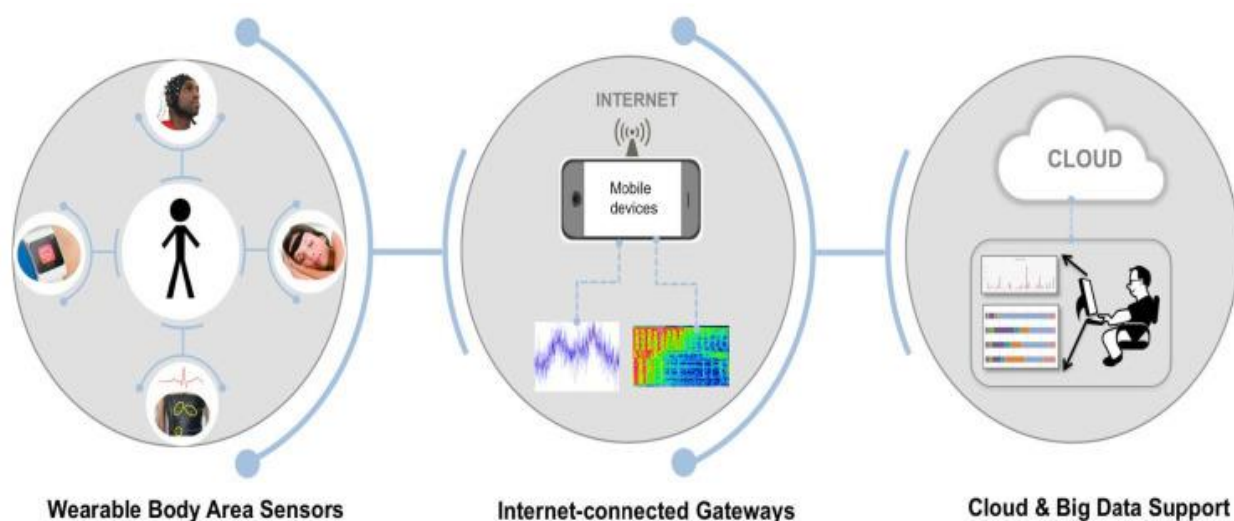


Рис. 1.1. Архітектурні елементи IoWT

У режимі реального часу використовувати переносні пристрої для моніторингу та збереження активності та фізіологічних функцій пацієнтів. Такі пристрої мають один або більше сенсорних вузлів, але кожен сенсорний вузол зазвичай має радіопередавач, низько-швидкісний блок опрацювання та невелику пам'ять. Датчики можуть вимірювати різні фізіологічні параметри та активність, включаючи насиченість киснем, тиск та температуру, електрокардіограму (ЕКГ), електроміографію, пульс і т.д.

Технології Bluetooth, інфрачервоний зв'язок, зв'язок ближнього поля NFC, RFID, Wi-Fi та бездротові трансивери Zigbee можуть підтримувати зв'язок носимих пристроїв зі смартфонами та іншими пристроями. Технологія сприяє лікуванню, полегшуючи дистанційну діагностику та моніторинг.

Сенсорний гаджет, створений Vedaеі, здатний контролювати та аналізувати дії пацієнтів. Технологія IoT, яка вимірює соціальну дистанцію,

може допомогти запобігти захворюванню на COVID-19. Три рівні датчиків IoT, алгоритми машинного навчання та програми для смартфонів використовуються для щоденного моніторингу тиску, насиченості кисню в крові, частоти кашлю та температури. Рамки, окреслені авторами, допомогли користувачам триматися на безпечній відстані від передачі вірусу та часто оновлювати свою інформацію. Також була представлена система моніторингу відстані на основі радіочастот, яка може використовуватися як у приміщенні, так і на вулиці.

Також можна використати підключення до Інтернету речей мережеву систему датчиків, що носяться, для промислових додатків у галузі охорони здоров'я та безпеки на відкритому повітрі. Носимі датчики, які носять працівники, збирають фізіологічні та навколишні дані, які передаються оператору системи та співробітникам для моніторингу та аналізу. Дані, зібрані від кількох працівників, які носять переносні датчики, можуть бути передані через мережу на сховище. Мережа поєднує мережу виявлення медичного сигналу на базі Bluetooth із гетерогенною платформою IoT.

Датчики використовуються не тільки для збору даних; їх також можна використовувати для виконання різноманітних поточних завдань і завдань моніторингу в системах охорони здоров'я на основі IoT. Артеріальний тиск, температура тіла, пульсоксиметрія та рівень глюкози в крові є кількома прикладами гетерогенних переносних сенсорних пристроїв, розроблених для збору біомедичних даних пацієнтів. Належна якість і розвиток цих систем моніторингу охорони здоров'я на основі Інтернету речей безпосередньо пов'язані з надійними даними від датчиків або сенсорних мереж, що вимагає використання передових методів обробки сигналів, об'єднання даних датчиків і аналізу даних. У медичній науці датчики, які вимірюють частоту серцевих скорочень, температуру тіла та інші речі, використовуються для виявлення та діагностики захворювань на самій ранній стадії.

Датчики контролю стану здоров'я можна розділити на контактні (тобто тілесні) і безконтактні (тобто периферійні) (рис. 1.2). Контактні датчики

прикріплюються до тіла для моніторингу фізіологічної поведінки, визначення хімічного рівня та моніторингу, пов'язаного з оптичними вимірюваннями. Контактні датчики також використовуються для моніторингу, пов'язаного з терапією, такі як прийом ліків, стимуляція та надзвичайні ситуації. Безконтактні датчики використовуються для моніторингу факторів, пов'язаних із фітнесом і здоров'ям, поведінки та реабілітації.

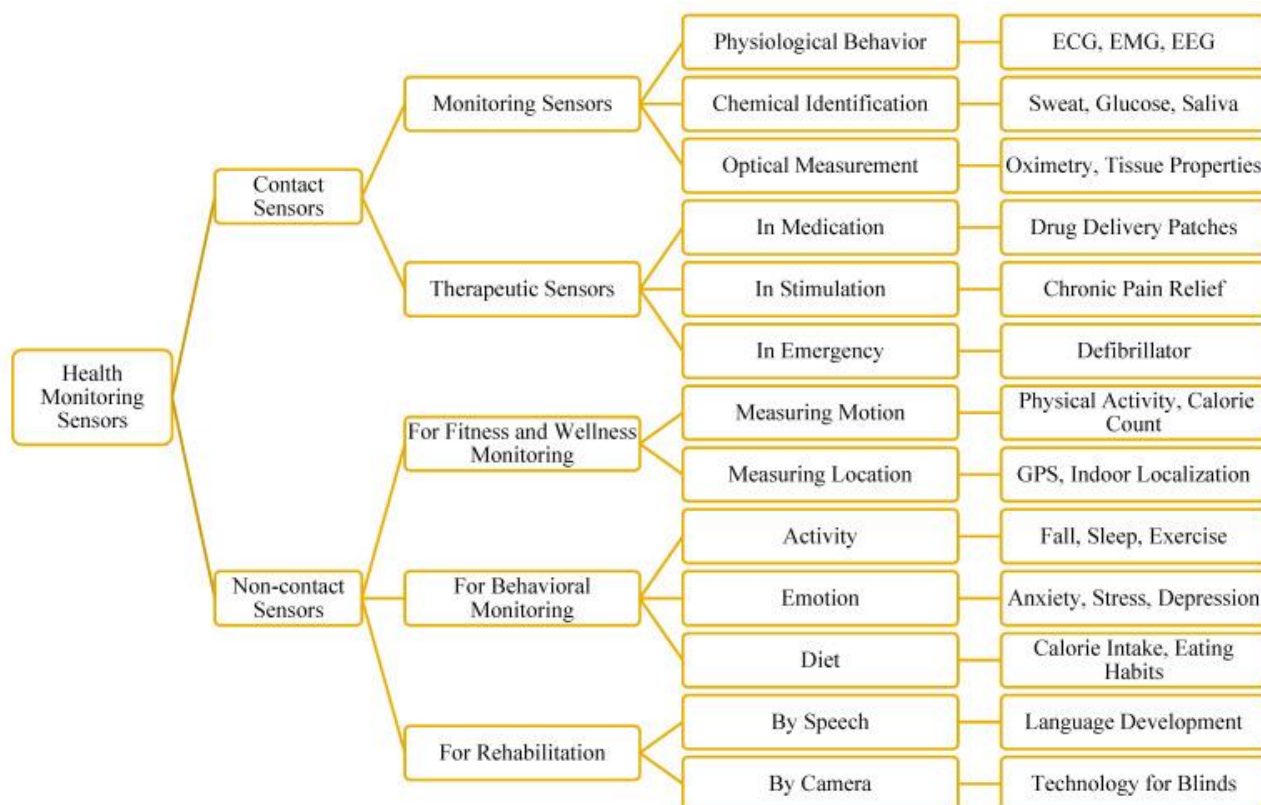


Рис. 1.2. Класифікація датчиків контролю стану здоров'я

Розглянемо найбільш доступні датчики. Визначення частоти серцевих скорочень/Системи серцевого моніторингу/Інсульт.

Перше застосування датчиків моніторингу здоров'я було через системи моніторингу охорони здоров'я на основі Інтернету речей; вони можуть збирати та вимірювати необхідні дані, надійно передавати ці дані через різні етапи до шлюзу та хмарного сервера, а також виконувати деякі периферійні завдання, щоб забезпечити прийняття рішень із низькою затримкою щодо захворювань, пов'язаних із серцем, і прогнозування. Кілька проектів

передбачають використання технології WSN для безперервного моніторингу серцевих пацієнтів, яким потрібна система моніторингу в реальному часі. WSN має кілька медичних датчиків і пристроїв, які можуть відстежувати артеріальний тиск, температуру тіла, частоту серцевих скорочень і пульс. ЕКГ критичного пацієнта в режимі реального часу також зберігається, щоб пацієнт був на постійному контролі.

Вимірювання температури тіла. Під час пандемії розумні пристрої для моніторингу стану здоров'я на основі Інтернету речей із датчиками для пацієнтів з COVID-19 на основі температури тіла, пульсу та SpO2 були корисними. Через мобільний додаток ці системи можуть вимірювати температуру тіла людини, насичення киснем і частоту пульсу.

Розпізнавання активності. Одне з багатьох застосувань медичних датчиків, які зараз використовуються, це розпізнавання активності. Майже всі фітнес-трекери виконують такий вид розпізнавання. Фітнес-трекери зараз є найпопулярнішими носіями для відстеження активності людини. У фоновому режимі виконується багато припущень, але більшість із них включає високочутливий 3D-акселерометр, який дозволяє датчику визначати прискорення.

Контроль рівня глюкози в крові та концентрації гемоглобіну. Датчики серцевого ритму, монітори глюкози в крові, ендоскопічні капсули та інші пристрої складають Інтернет медичних речей (ІоМТ), які разом створюють систему моніторингу WBSN для діабетиків ІоМТ.

Виявлення та моніторинг частоти дихання. Ми можемо стежити за дихальною системою людського організму кількома способами. Деякі автори використовували складні датчики, які відстежували патерни дихання. Датчик біоімпедансу також може використовуватись.

Моніторинг сну. Ця програма для відстеження сну допомагає користувачеві налаштувати режим сну та підтримувати здоровий життєвий цикл. Для цього використовуються різні датчики. Пристрої, що носяться, часто відстежують частоту серцевих скорочень, частоту пульсу, рівень кисню

в крові і режим дихання, і, беручи до уваги ці вимірювання, вони можуть прийняти обґрунтоване рішення щодо якості сну.

Молекулярна діагностика та клінічна діагностика. Завдяки швидкому та доступному застосуванню в охороні здоров'я зі знизеним ризиком інфікування останні розробки в області біосенсорів для зручної діагностики та імплантованих пристроїв для зручної для пацієнта терапії привернули велику увагу. Швидкий розвиток сенсорних платформ і імплантованих пристроїв із спеціалізованою функціональністю став можливим завдяки використанню нещодавно створених матеріалів у медичне обладнання.

Визначення насичення крові киснем. Поряд з точним постійним моніторингом рівня внутрішньосудинного кисню, важливо контролювати стан серцево-судинної системи пацієнтів після кардіоторакальної хірургії. Існують нові типи даних, такі як насичення киснем, які постійно збираються за допомогою датчиків насичення киснем (SpO<sub>2</sub>) і представляють відсоток насиченого киснем гемоглобіну порівняно із загальною кількістю гемоглобіну в крові; вони стають доступними для широкого кола користувачів.

Таким чином, можна сказати, що датчики моніторингу здоров'я використовуються в різних програмах і можуть використовуватися в майбутньому для аналізу різних захворювань, особливо тих, які більше зосереджені на зборі зразків або даних, моніторингу або оцінці. Ми можемо стверджувати, що щоразу, коли використовується датчик, існує можливість зібрати необхідні дані та отримати бажані результати залежно від. Крім того, об'єднання хмари, географічних інформаційних систем і мобільних пристроїв покращило процес збору та моніторингу даних на основі датчиків, дозволяючи при цьому гнучко віддалено ділитися та спілкуватися.

Огляд процесу проведення медичних аналізів. Застосування інформаційних технологій в системі охорони здоров'я є актуальним напрямком розвитку галузі. Існує безліч різноманітних програмних розробок



– веб-систем, мобільних додатків, онлайн-платформ – як для фахівців, так і для пацієнтів. Особливий інтерес викликають складні інтегровані системи, що об'єднують лікарів, пацієнтів, страхові компанії та інші організації.

Реалізація таких платформ потребує застосування сучасних гнучких архітектурних рішень. Зокрема, поширеним є використання мікросервісного підходу, який дозволяє створити систему, що масштабується та високопродуктивну систему. У літературі описано переваги та можливі труднощі переходу охорони здоров'я на мікросервіси. Загалом, цей напрям є актуальним фокусом досліджень та впроваджень.

Для визначення напрямків розвитку таких систем необхідно з'ясувати які компоненти необхідні для постійного моніторингу.

Розглянемо ключові причини, чому постійний моніторинг здоров'я є актуальним:

- дозволяє відстежувати динаміку поширеності хвороб та захворюваності, прогнозувати тенденції, виявляти ризики для своєчасного реагування;
- надає актуальні дані про стан здоров'я різних груп населення для прийняття обґрунтованих управлінських рішень в сфері охорони здоров'я;
- допомагає оцінювати ефективність лікувально-профілактичних заходів та оптимізувати розподіл ресурсів;
- можливість контролювати динаміку показників для верифікації наукових гіпотез щодо впливу різних факторів на здоров'я;
- забезпечує накопичення великих масивів статистичних даних для аналізу за допомогою сучасних методів штучного інтелекту.

Отже, постійний моніторинг надає цінну актуальну інформацію для ефективного управління системою охорони здоров'я.

Процес медичного обслуговування в Україні називається – диспансеризація. Диспансеризація – це комплекс заходів, спрямованих на

раннє виявлення хронічних захворювань. В Україні диспансеризація населення є обов'язковою і проводиться 1 раз на рік.

Диспансеризація проводиться в декілька етапів. Основні етапи:

- попередній (анкетування та аналіз даних пацієнта);
- огляд сімейним лікарем;
- проведення лабораторних досліджень;
- інструментальні обстеження;
- огляд вузькопрофільними спеціалістами.

За результатами отриманих результатів виявляють групу здоров'я та складають індивідуальні рекомендації для пацієнта.

Дані диспансеризації заносять до цифрової системи для подальшого аналізу та моніторингу.

Отже, диспансеризація – це важливий профілактичний захід для відстеження стану здоров'я населення та раннього виявлення можливих проблем.

Процес проведення медичних аналізів починається зі звернення пацієнта до лікаря та оформлення направлення на необхідні дослідження. Направлення фіксується в інформаційній системі медичної установи.

Пацієнт проходить взяття біоматеріалу (кров, сеча тощо) для проведення лабораторних досліджень. Біоматеріал реєструється в лабораторній інформаційній системі та надходить на аналіз.

В залежності від типу досліджень, проводяться відповідні аналізи:

- біохімічні – визначають вміст речовин в біологічних рідинах;
- гематологічні – дослідження показників крові;
- імунологічні – визначення імунного статусу;
- мікробіологічні – виявлення мікроорганізмів;
- патогістологічні – вивчення тканин організму;
- тощо.

Результати аналізів заносяться в лабораторну інформаційну систему, звідки передаються в єдину медичну інформаційну систему медичного закладу.

На основі отриманих результатів лікар встановлює або уточнює діагноз та призначає подальше лікування.

На разі в Україні для тих пацієнтів, що підписали декларацію з лікарем первинної медичної допомоги мають право на певний перелік безкоштовних лабораторних та діагностичних досліджень. До лабораторних досліджень можна віднести:

- загальний аналіз крові;
- загальний аналіз сечі;
- рівень глюкози у крові;
- загальний рівень холестерину;
- швидкі тести (на вагітність; на тропонін; на ВІЛ; на вірусні гепатити та на антиген SARS-CoV-2).

До діагностичних або інструментальних досліджень, які проводяться для поглибленої діагностики стану здоров'я та функціонування внутрішніх органів, можна віднести.

- ЕКГ – для діагностики серцево-судинних захворювань;
- флюорографія або рентген грудної клітки – перевірка стану легенів.
- УЗД органів черевної порожнини – печінка, нирки, підшлункова залоза.
- колоноскопія – дослідження слизової оболонки товстої кишки.

За результатами лікар визначає необхідність подальшого спостереження або лікування.

Даний процес супроводжується значним документообігом, може вимагати узгодження даних між різними підрозділами, а також потребує ретельного контролю та аналізу отриманих результатів.

## 1.2. Системи медичної автоматизації

Медична автоматизація – це застосування автоматизованих технологій і систем для оптимізації та вдосконалення процесів у сфері надання медичної допомоги. Її мета – зменшити ручне навантаження, мінімізувати людські помилки та оптимізувати ефективність у різних сферах медицини за допомогою автоматизації.

Медична автоматизація підвищила ефективність моніторингу та подальшого спостереження за пацієнтами. За допомогою підключених пристроїв і датчиків можна збирати дані в режимі реального часу про стан здоров'я пацієнтів, зокрема частоту серцевих скорочень, артеріальний тиск і рівень глюкози.

Ці дані можуть автоматично передаватись медичним працівникам, що дозволяє їм уважно стежити за прогресом пацієнтів і приймати рішення на основі актуальної інформації.

Є два ключових аспекти, які становлять значну частину процесів у медичній автоматизації.

Автоматизація адміністративних процесів у медичній сфері. Одним із ключових аспектів медичної автоматизації є автоматизація адміністративних процесів. Це включає в себе впровадження спеціалізованих систем і програмного забезпечення, які забезпечують ефективне управління медичною документацією, планування призначень, відстеження рахунків і платежів, а також інші адміністративні завдання.

Завдяки автоматизації цих завдань навантаження на медичний персонал зменшується, а доступні ресурси оптимізуються.

Наступним аспектом медичної автоматизації – це автоматизація діагностики та лікування.

Завдяки прогресу в галузі штучного інтелекту та машинного навчання були розроблені алгоритми та системи, які можуть швидко й точно аналізувати великі обсяги медичних даних.

Це дозволяє оптимізувати діагностичні процеси, визначати закономірності та аномалії в результатах медичних тестів і надавати точніші та персоналізовані рекомендації щодо лікування.

Медична автоматизація пропонує ряд суттєвих переваг, які позитивно впливають на ефективність і якість медичного обслуговування. Ось деякі з основних переваг:

**Підвищення ефективності:** медична автоматизація прискорює та спрощує завдання, які раніше вимагали багато часу та ручних зусиль, що створює більшу продуктивність та ефективність для медичних працівників.

**Зменшення кількості помилок:** замінивши завдання, які часто виконуються вручну, як-от введення даних або інтерпретація результатів, автоматизованими процесами, медичні працівники можуть зменшити ризик помилок і підвищити точність результатів.

**Швидкий і легкий доступ до інформації:** централізація та більш ефективне керування медичними даними та інформацією про пацієнтів забезпечує швидкий і безпечний доступ до відповідної інформації.

**Покращена якість медичної допомоги:** завдяки автоматизації вивільняються ресурси та час, щоб медичні працівники могли зосередитися на наданні більш якісної та більш персоналізованої медичної допомоги.

**Покращена безпека та відповідність нормативним вимогам:** автоматизовані системи можуть гарантувати конфіденційність і конфіденційність даних пацієнтів, а також відповідність нормам і встановленим стандартам безпеки.

**Краща координація та зв'язок:** автоматизація може полегшити зв'язок та координацію між різними учасниками системи охорони здоров'я, такими як лікарі, спеціалісти, лабораторії та аптеки.

Використовуючи автоматизовані технології та системи, спосіб надання медичної допомоги можна трансформувати, спрощуючи процеси, підвищуючи безпеку та покращуючи досвід як для медичних працівників, так і для пацієнтів.

Автоматизацію медичного управління можна досягти за допомогою багатьох різних інструментів і систем.

Електронні медичні записи – збір та збереження інформації про пацієнтів в цифровому форматі. Дозволяє лікарям мати швидкий доступ до медичних даних пацієнта. Сюди можна віднести, зокрема електронну історію хвороби. Впровадження електронної системи історії хвороби дозволяє керувати медичними записами кожного пацієнта в цифровому вигляді. Це включає демографічні дані, історію хвороби, результати обстежень, попередні лікування тощо. Завдяки електронній історії хвороби вам більше не потрібні паперові записи, ви можете отримати доступ до інформації та оновлювати її швидше, що, у свою чергу, полегшує прийняття клінічних рішень.

Електронний документообіг – автоматизація створення, управління доступом та поширення медичних документів в електронному вигляді.

Електронний документообіг в сфері охорони здоров'я – це організація роботи з документами в електронному вигляді, що включає:

Створення медичних документів (амбулаторні карти, звіти, направлення, рецепти і т.д.) безпосередньо в цифровому форматі за допомогою програмного забезпечення.

Формування потоків електронних медичних документів між структурними підрозділами медзакладу чи різними установами.

Зберігання документів в електронних архівах і базах даних з управлінням прав доступу.

Отримання та інтеграція зовнішніх е-документів (результати аналізів, висновки тощо).

Це дозволяє підвищити ефективність документообігу, скоротити витрати часу персоналу та поліпшити якість медичного обслуговування.

Мобільні додатки – для віддаленого моніторингу пацієнтів, комунікації лікар/пацієнт, запису на прийом тощо. Ось основні можливості мобільних додатків, що застосовуються в сфері охорони здоров'я:

- збір даних про стан здоров'я – запис показників від медичних девайсів, результатів аналізів, симптомів тощо;
- ведення мобільної електронної амбулаторної карти з вільним доступом – перегляд попередніх діагнозів, призначень, аналізів;
- відеоконсультації з лікарем в режимі реального часу з будь-якої точки;
- запис на прийом до лікаря або на процедури медзакладу онлайн;
- обмін повідомлення з лікарем у зручний час;
- отримання нагадувань про прийом ліків, профілактичні заходи тощо.

Такі технології суттєво покращують доступність медичної допомоги та якість обслуговування пацієнтів.

Автоматизовані системи для лабораторій та аптек – облік та управління ліками, тестами, результатами аналізів.

Автоматизовані системи дозволяють підвищити ефективність роботи лабораторій та аптек в сфері охорони здоров'я завдяки наступним можливостям:

Для лабораторій:

- автоматизація процесів реєстрації матеріалу та замовлень на дослідження;
- автоматична обробка і збереження результатів аналізів;
- формування електронних протоколів та інтеграція з медичними інформаційними системами;
- підтримка прийняття рішень на основі накопичених лабораторних даних.

Для аптек:

- оформлення рецептів, облік лікарських засобів;
- контроль термінів придатності та запасів медикаментів;

- інтеграція з системами електронних рецептів та складського обліку.

Такі системи дозволяють уникати помилок, прискорюють рутинні процеси, а також надають дані для аналітики.

Бізнес-аналітика в охороні здоров'я – аналіз великих даних для підтримки прийняття управлінських рішень. Основні напрямки застосування:

Аналіз ефективності лікувальних процедур, медичних програм на основі статистичних даних та виявлення оптимальних рішень.

Прогнозування попиту на послуги за профілями, планування необхідних потужностей клінік. Виявлення шахрайств та неналежного використання коштів за рахунок перехресного аналізу даних. Аналіз навантаження, якості та ефективності роботи медичних працівників. Оцінка ризиків розвитку захворювань у конкретних груп пацієнтів для їх запобігання.

Застосування подібної аналітики допомагає приймати більш ефективні управлінські рішення в системі охорони здоров'я. Такі інструменти значно підвищують ефективність охорони здоров'я за рахунок автоматизації рутинних процесів.

Розумне планування. Розумна система планування автоматизує процес планування прийому до лікаря. Це дозволяє пацієнтам швидко та легко планувати свої зустрічі, вибираючи доступний час залежно від наявності медичного працівника. Це зменшує час і ризик помилок, пов'язаних із записом на прийом вручну, оптимізує час лікарів і покращує враження пацієнтів.

Динамічне планування черги пацієнтів на прийом до лікаря на основі аналізу медичних карток та терміновості призначень. Автоматизоване планування розкладу для пацієнтів з хронічними захворюваннями з нагадуваннями про прийом ліків, процедури тощо.

Smart-нагадування пацієнтам про заплановані аналізи, огляди, диспансеризацію за графіком.



Інтелектуальне планування закупівель та постачання ліків, витратних матеріалів, обладнання медзакладами на основі аналізу великих даних. Розумне планування можна використовувати також для організації роботи медичного персоналу та оптимізація графіків з урахуванням показників ефективності, навантаження, відпусток.

Організація залів очікування. Кімнати очікування можуть бути автоматизовані за допомогою систем керування чергою та виклику пацієнтів.

Ці системи сповіщають пацієнтів, коли настає їхня черга, і інформують їх про затримки або зміни в розкладі. Крім того, вони також можуть включати екрани або пристрої, які показують порядок, мінімізуючи непотрібне очікування та покращуючи допомогу лікаря.

Медичну автоматизацію можна ще більше вдосконалити за допомогою систем планування ресурсів підприємства, систем управління взаємовідносинами з клієнтами і систем управління бізнес-процесами.

Інтеграція зовнішніх електронних медичних документів в інформаційні системи охорони здоров'я дозволяє автоматизувати такі процеси:

- отримання результатів лабораторних та інструментальних обстежень пацієнтів з централізованих баз даних медичних Закладів
- завантаження висновків консультацій лікарів-фахівців та результатів консилиумів для інтеграції з електронною медичною картою пацієнта;
- підключення до електронних реєстрів хворих на окремі захворювання (онкологічні, рідкісні тощо);
- отримання інформації про проведені раніше обстеження чи лікування в інших медичних закладах.

Інтеграція лабораторних інформаційних систем медзакладів з єдиною електронною системою охорони здоров'я дозволяє автоматизувати процеси.

Після проведення аналізів, результати в цифровому вигляді надходять до центрального сховища.

Лікар при запиті на отримання даних пацієнта може переглянути і завантажити результати будь-яких попередніх лабораторних досліджень, проведених в акредитованих центрах.

Результати автоматично інтегруються з електронною медичною картою пацієнта.

Лікар отримує оперативний доступ до критично важливих даних для встановлення діагнозу та моніторингу стану пацієнта.

Інформація також використовується для аналізу якості та ефективності надання послуг закладами.

Така інтегрована система дозволяє підвищити якість медичного обслуговування.

Один із варіантів структури інформаційної системи для автоматизації моніторингу результатів медичних аналізів.

База даних:

- персональні дані пацієнтів;
- дані про проведені аналізи (вид аналізу, дата, показники);
- референтні значення показників;
- дані про лікарів та медперсонал.

Інтеграційний модуль:

- імпорт результатів аналізів з лабораторними інтегрованими системи;
- експорт аналітичних звітів.

Аналітичний модуль:

- порівняння результатів з референтними значеннями;
- розрахунок аналітичних показників;
- побудова графіків динаміки результатів;
- формування звітів.

Модуль безпеки:

- авторизація і розмежування доступу до даних;
- захист персональних даних.

В таких системах можна запланувати також і мобільний додаток для доступу лікарів який збільшить швидкість доступу до результатів.

Аналітичний модуль в системі моніторингу результатів медичних аналізів виконує такі основні функції:

Обробка та агрегація даних:

- виявлення відхилень та похибок в результатах аналізів;
- розрахунок середніх, медіан та інших статистичних показників;
- агрегація результатів по періодах спостереження.

Візуалізація динаміки:

- графіки зміни показників в часі для кожного пацієнта;
- порівняння з нормативними значеннями;
- тренди статистичних показників.

Виявлення відхилень:

- попередження про критичні зміни у стані пацієнта;
- сигналізація щодо виходу показників за допустимі межі;
- прогнозування розвитку патологічних процесів.

Автоматизована система управління результатами аналізів може підвищити задоволеність пацієнтів від передачі результатів тестів, замовлених їхнім постачальником первинної медичної допомоги, і може підвищити задоволеність пацієнтів від передачі інформації щодо їх стану та планів лікування.

Комунікація між пацієнтом і лікарем важлива для високоякісного медичного обслуговування і може покращити результати лікування пацієнтів. Обговорення між пацієнтом і лікарем результатів аналізів є невід'ємною частиною тривалого медичного обслуговування. В амбулаторних умовах повідомлення про результати аналізів, яке часто відбувається між клінічними зустрічами, є важливою частиною діагностичного та терапевтичного плану. Крім того, неможливість повідомити пацієнтам результати діагностичних тестів і плани подальшого спостереження становить важливу проблему безпеки пацієнтів, оскільки численні дослідження показали, що погане

повідомлення результатів тестів пов'язане із затримками в лікуванні та втраченими можливостями подальшого спостереження. Дослідження показали, що лікарі і пацієнти стурбовані цими проблемами.

Існує ряд перешкод для ефективного інформування лікарів про результати аналізів пацієнтів. Методи спілкування між лікарями значно відрізняються, і це пов'язано з пропуском результатів тесту та затримкою зв'язку. Крім того, очікування пацієнтів щодо повідомлення результатів дуже різняться, і методи, які використовують лікарі, можуть не відповідати їм. Загалом пацієнти віддавали перевагу телефонному сповіщенню над звичайною поштою, і багато хто вважав електронне сповіщення незручним через проблеми безпеки. Інше опитування показало, що пацієнти зазвичай не обговорюють зі своїми лікарями свої переваги щодо сповіщення про результати тесту.

Незважаючи на ці проблеми, недоліки якості в управлінні результатами амбулаторних аналізів підтверджують необхідність подальшої оцінки та перегляду систем, які лікарі використовують для виконання цих завдань. Все більше практик впроваджують електронні медичні записи (ЕМЗ) для покращення догляду за пацієнтами. Хоча додатки для перегляду результатів аналізів доступні в більшості комерційних і домашніх програмних засобах, зазвичай вони вимагають від лікарів окремо відстежувати свої тести, що очікують на розгляд, і індивідуально перевіряти медичну карту пацієнта на предмет результатів. Зручніше було б запровадити автоматизовані системи сповіщення лікарів про результати аналізів, але ці системи зазвичай розгортаються лише в стаціонарних умовах для критично ненормальних результатів. Потік подальшої роботи в амбулаторних умовах відрізняється, оскільки пацієнти не перебувають у контрольованому середовищі, що створює перешкоди для спілкування.

Розглянемо приклад лікарні Partners HealthCare System, Inc, яка великою лікарнею в Массачусетсі. Вона складається з кількох академічних і

громадських лікарень, у тому числі Brigham and Women's Hospital, Massachusetts General Hospital та інших лікарень і амбулаторних практик.

Основним амбулаторним EM3 у Partners HealthCare System, Inc, є Longitudinal Medical Record, програма внутрішньої розробки на основі веб-інтерфейсу, яка забезпечує функціональні можливості планування лікування та призначення ліків, формування та отримання медичних списків. Longitudinal Medical Record також включає можливість пошуку та перегляду результатів лабораторних досліджень, які були виконані в Partners HealthCare System, Inc.

Partners HealthCare System, Inc, розробила автоматизовану систему сповіщення про результати тестування, відому як «Менеджер результатів», яка вбудована в Longitudinal Medical Record. Вона забезпечує відстеження всіх результатів аналізів, пов'язаних із лікарем, який призначає.

Підсумкова сторінка, яка постійно оновлюється в міру публікації нових результатів лабораторіями, надає стисло інформацію для кожного пацієнта, включаючи дату контрольного візиту, ім'я пацієнта та номер медичної карти, тип результату тесту та те, чи містить результат тесту відхилення. Лікарі можуть швидко створювати та документувати листи-повідомлення пацієнтові, електронні листи та вставляти в листи кількісні результати. Також доступні демографічна та контактна інформація пацієнта, сторінка з детальною інформацією про результати, на якій відображаються результати кожного лабораторного дослідження з референтними діапазонами, примітка про відвідування та відстеження підтвердження перегляду результатів лікарем.

Лікарі в контрольній групі дослідження (як до, так і після експерименту) практикували управління результатами аналізів, відстежуючи статус своїх замовлень і результатами вручну та окремо, і кожен лікар мав власний підхід до відстеження. Це часто вимагало повторних перевірок історії пацієнта від лабораторії в очікуванні результату. Лікарі також були

повідомлені про нові результати тестів за допомогою паперових роздруківок, створених тестовими установами.

Основним результатом було задоволення пацієнтів загальною інформацією про результати тесту. Вторинними результатами були задоволеність інформацією про умови та лікування, які надавалися пацієнтам, а також результати їхніх тестів, задоволеність навичками слухання свого лікаря та задоволеність загальним спілкуванням свого лікаря.

Загалом це дослідження показало, що автоматизована система керування, яка забезпечує централізоване відстеження результатів тестування та полегшує контакт з пацієнтами, підвищила загальну задоволеність повідомленням результатів тестування. Поліпшення вторинного результату, задоволеності пацієнтів отриманням інформації щодо умов і лікування, пов'язаних з тестами, свідчить про те, що цей фактор мав прямий вплив на загальну задоволеність пацієнтів повідомленням результатів тестів.

## РОЗДІЛ 2

### РОЗРОБКА СИСТЕМИ АВТОМАТИЗОВАНОГО МОНІТОРИНГУ

#### 2.1. Загальна структура системи

В інтерфейсі програми передбачається можливість:

- отримати дані про аналізи з файлу;
- відобразити дані про пацієнта;
- відобразити дані про аналізи;
- додати дані в базу даних;
- відобразити історію зміни аналізів одного пацієнта в часі.

Програма розроблена з використанням фреймворку QT, що дозволяє уніфікувати програмний інтерфейс модулів програми. Програма складається з наступних файлів:

AnalizMed.pro  
database\_module.cpp  
database\_module.h  
file\_csv.cpp  
file\_csv.h  
main.cpp  
mainwindow.cpp  
mainwindow.h  
mainwindow.ui

Графічний інтерфейс програми розроблено за допомогою QtDesigner, макет якого представлено на рис. 2.1.

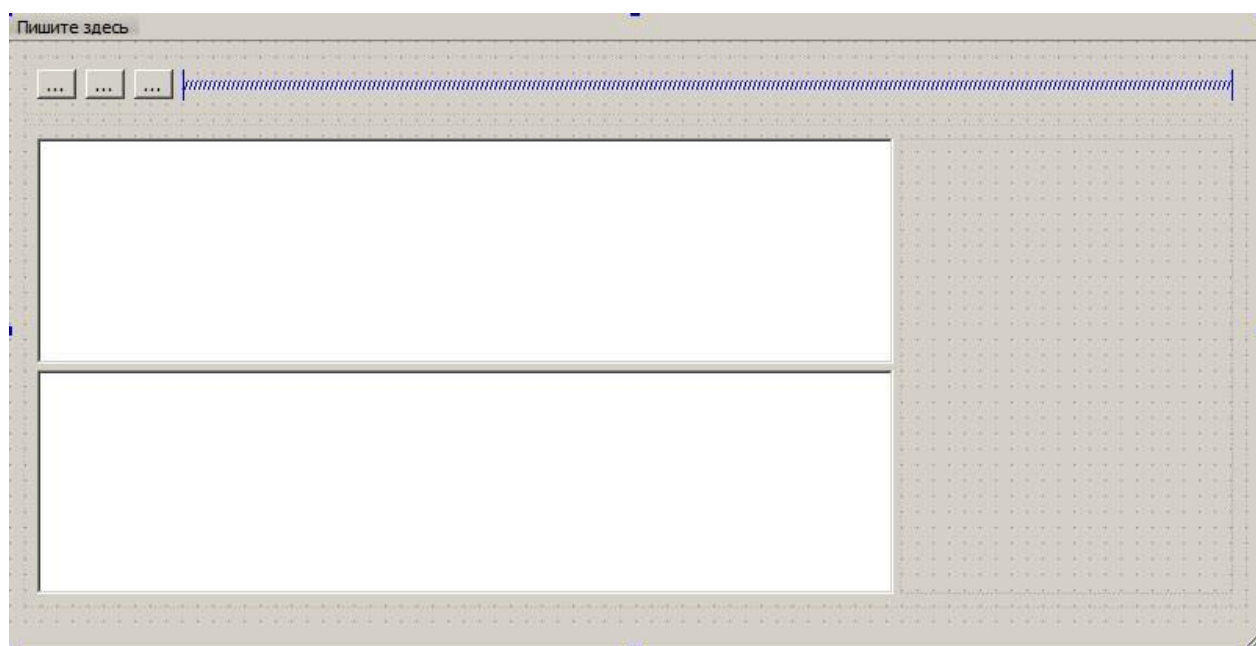


Рис. 2.1. Макет графічного інтерфейсу програми

Для взаємодії з користувачем використано наступні класи Qt:

- QWidget – використовується як контейнер;
- QPushButton – використовується для кнопок;
- QTextEdit – використовується для відображення текстової інформації;
- QTableWidgetItem – використовується для відображення таблиць;
- QChart – використовується для відображення графіків.

Дерево компонентів графічного інтерфейсу представлено на рис. 2.2. Так як в QtDesigner відсутні компоненти для сторонніх віджетів, то для візуалізації використовуємо QtDesigner, як контейнер, а потім в конструкторі відповідного класу розміщуємо в ньому об'єкт діаграм QChart.



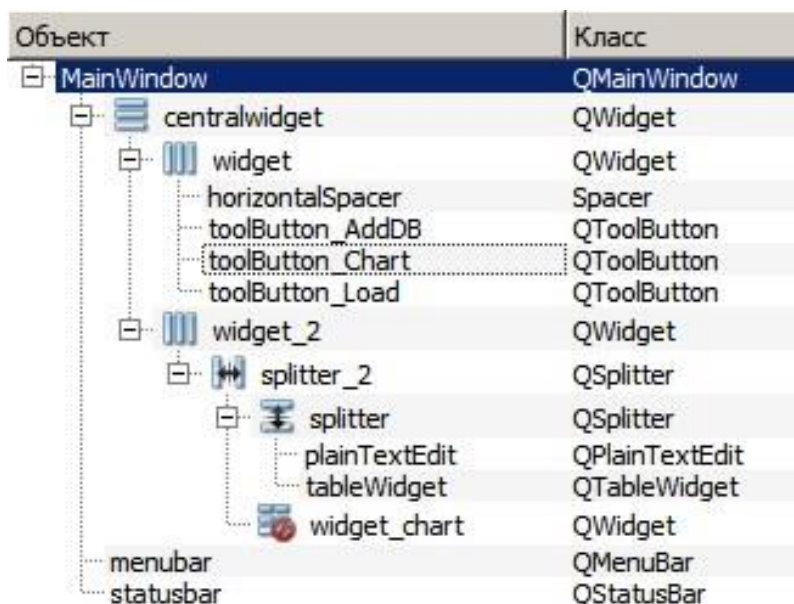


Рис. 2.2. Компоненти графічного інтерфейсу

## 2.2. Розробка підсистеми відображення даних

Клас `QChart` керує графічним представленням різних типів серій та інших пов'язаних з діаграмами об'єктів, таких як легенда та осі. `QChart` — це віджет `QGraphicsWidget`, який можна показати в `QGraphicsScene`. Простішим рішенням є відображення діаграми в макеті за допомогою зручного класу `QChartView` замість `QChart`. У QML діаграми відображаються за допомогою типу `ChartView`.

Деякі компоненти діаграми також можна представити як полярні діаграми за допомогою класу `QPolarChart`, який є спеціалізацією класу `QChart`, або типу `PolarChartView` QML, який є спеціалізацією типу `ChartView`.

Зовнішній вигляд діаграм можна налаштувати, використовуючи теми, змінюючи кольори та властивості, приховуючи компоненти діаграм або анімуючи діаграми.

Відповідачі моделей дозволяють використовувати модель даних, отриману від класу `QAbstractItemModel`, як джерело даних для діаграми. Картографи моделей можуть бути горизонтальними або вертикальними.

Модуль Qt Charts надає такі типи діаграм:

- Line and spline charts – лінійні та сплайнові діаграми;
- Area and scatter charts – точкові та площинні діаграми;
- Bar charts – стовпчасті діаграми;
- Pie charts – кругові діаграми;
- Box-and-whiskers charts;
- Candlestick charts – свічкові діаграми;
- Polar charts – полярні діаграми.

Кожен тип діаграми представлено похідним класом `QAbstractSeries` або похідним типом `AbstractSeries` у QML. Діаграми створюються за допомогою екземпляра класу серії та додавання його до екземпляра `QChart` або `ChartView`.

Розглянемо даграми, які будуть використані в проекті.

Лінійні та сплайнові діаграми. Лінійні та сплайнові діаграми представляють дані як ряд точок даних, з'єднаних лініями. У лінійній діаграмі точки даних з'єднані прямими лініями, тоді як у сплайн-діаграмі вони з'єднані сплайном. Сплайн малюється за допомогою `QPainterPath` (рис. 2.).



Рис 2.1. Лінійні та сплайнові діаграми QLineSeries

Лінійна діаграма реалізується за допомогою класу QLineSeries або типу QML LineSeries.

Сплайн-діаграма реалізується за допомогою класу QSplineSeries, який успадковує QLineSeries, або типу SplineSeries, який успадковує LineSeries.

Точкова та площинна діаграма. Областьні діаграми представляють дані як область, обмежену двома лініями, тоді як точкові діаграми представляють дані як набір точок.



Рис 2.2. Точкова та площинна діаграма QAreaSeries

Діаграма площ реалізується за допомогою класу QAreaSeries або типу AreaSeries QML. За замовчуванням вісь x використовується як одна межа, а QLineSeries або LineSeries як інша. Однак можна використовувати QLineSeries або LineSeries як обидві границі.

Точкова діаграма реалізується за допомогою класу QScatterSeries або типу QML ScatterSeries.

Стовпчасті діаграми. Гістограма представляє дані у вигляді горизонтальних або вертикальних стовпчиків, згрупованих за категоріями. Клас `QBarSet` і тип `QML BarSet` представляють один набір стовпчиків на гістограмі. Клас `QAbstractBarSeries` є абстрактним батьківським класом для всіх класів рядів `Bar`, а тип `AbstractBarSeries` є батьківським типом типів рядів `Bar`. Тип ряду визначає спосіб представлення даних.

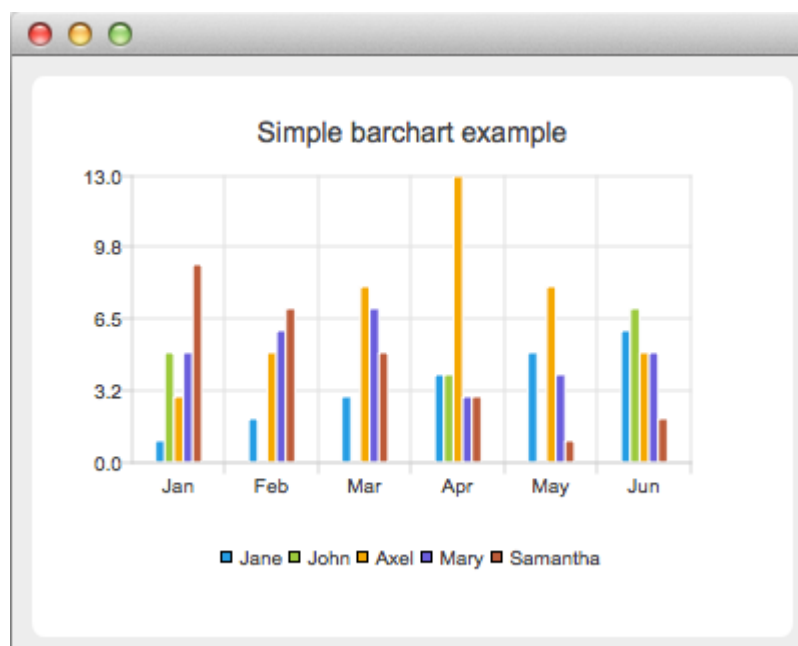


Рис. 2.3. Стовпчасті діаграми `QBarSeries`.

Клас `QBarSeries` і тип `QML BarSeries` представляють дані у вигляді вертикальних смуг, згрупованих за категоріями. Подібним чином клас `QHorizontalBarSeries` і тип `QML HorizontalBarSeries` представляють дані у вигляді горизонтальних смуг.

Клас `QStackedBarSeries` і тип `StackedBarSeries` представляють ряд даних у вигляді вертикально складених стовпчиків, з одним стовпчиком на категорію. Відповідний горизонтальний клас `QHorizontalStackedBarSeries` і тип `HorizontalStackedBarSeries`.

Клас `QPercentBarSeries` і тип `QML PercentBarSeries` представляють серію категоризованих даних як відсоток кожної категорії. Відповідним

горизонтальним класом і типом є `QHorizontalPercentBarSeries` і `HorizontalPercentBarSeries` відповідно.

Секторні діаграми. Кругові діаграми представляють дані як кругову діаграму, яка складається з секторів. Сектор реалізується за допомогою класу `QPieSeries` або типу `QML PieSeries`, а сектори додаються за допомогою класу `QPieSlice` або типу `PieSlice QML`.

Пиріг можна перетворити на пончик, вказавши розмір отвору від 0,0 до 1,0.

Діаграми `Box-and-whiskers`. Коробчаті діаграми представляють дані як квартилі, розширені вусами, які показують мінливість значень. Елементи в графічних рядах згруповані за категоріями, подібно до наборів стовпців у рядах. Для кожного елемента «коробка з вусами» вказані нижній екстремум, нижній квартиль, медіана, верхній квартиль і верхній екстремум.

Коробчату діаграму реалізовано за допомогою класів `QBoxPlotSeries` і `QBoxSet` або типів `QML BoxPlotSeries` і `BoxSet`.

Полярні діаграми представляють дані у вигляді кругового графіка, де розміщення даних базується на куті та відстані від центру графіка, полюса.

Клас `QPolarChart` є спеціалізацією класу `QChart`. Він підтримує лінії, сплайни, площі та розкид, а також усі підтримувані ними типи осей. Вісь можна використовувати як радіальну або кутову вісь. У `QML` відповідним типом є `PolarChartView`.

`Qt Charts` підтримує такі типи осей:

- `Value axis` – вісь значень;
- `Category axis` – вісь категорій;
- `Bar category axis` – вісь стовпчастих категорій;
- `Date-time axis` – вісь часу;
- `Logarithmic value axis` – вісь логарифмічних значень.

Вісь можна налаштувати для відображення лінії з галочками, лініями сітки та відтінками. Значення на осі відображаються в місцях позначок. Усі типи осей є спеціалізаціями класу `QAbstractAxis` або типу `QML AbstractAxis`.

Вісь значень додає реальні значення до осі діаграми. Він реалізований за допомогою класу `QValueAxis` або типу `QML ValueAxis`.

Вісь категорії реалізується за допомогою класу `QCategoryAxis` або типу `QML CategoryAxis`. Він має іменовані діапазони та регульовану ширину діапазонів.

Вісь категорій стовпця подібна до осі категорій, але ширина діапазону однакова для всіх діапазонів. Вісь категорії барів реалізується за допомогою класу `QBarCategoryAxis` або типу `QML BarCategoryAxis`.

Вісь дати й часу додає дати й час до осі діаграми. Він реалізований за допомогою класу `QDateTimeAxis` або типу `QML DateTimeAxis`.

Логарифмічна вісь додає логарифмічний масштаб до осі діаграми. Логарифмічна шкала — це нелінійна шкала, яка базується на порядку величини, так що кожна позначка на осі є попередньою позначкою, помноженою на значення. Логарифмічна вісь реалізується за допомогою класу `QLogValueAxis` або типу `QML LogValueAxis`.

Для однієї діаграми можна визначити кілька осей. Осі можна розмістити вниз, вгору, ліворуч або праворуч від діаграми. Далі сокири можуть бути різних типів.

Легенда — це графічний об'єкт, який відображає легенду діаграми. Об'єкти легенди не можна створювати або видаляти, але на них можна посилатися через клас `QChart` або тип `QML ChartView`. Стан легенди оновлюється за допомогою `QChart` або `ChartView`, коли ряд змінюється.

Легенду можна розташувати під або над діаграмою, а також ліворуч чи праворуч від неї. За замовчуванням легенда додається до діаграми, але її можна від'єднати від окремого графічного елемента, який можна вільно переміщувати.

З легенди можна приховати як окремі маркери, так і всю легенду.

Маркери легенди можна змінювати за допомогою базового класу `QLegendMarker` і підкласів для кожного типу серії: `QAreaLegendMarker`,

QBarLegendMarker, QBoxPlotLegendMarker, QCandlestickLegendMarker і QXYLegendMarker.

Деталізація даних. Ефекти деталізації можна застосувати, наприклад, до гістограм або кругових діаграм. Коли користувачі вибирають елемент у діаграмі, відображається більш детальний вигляд елемента. Це реалізується шляхом видалення першої серії та додавання іншої.

Масштабування та прокручування. Користувачі можуть використовувати клавіатуру для масштабування та прокручування. Вони можуть прокручувати діаграми за допомогою клавіш зі стрілками та збільшувати або зменшувати діаграми за допомогою клавіш «плюс» і «мінус». Крім того, QRubberBand можна використовувати для вибору області для збільшення. На сенсорних пристроях жести можна використовувати для панорамування та масштабування.

Можнає підключити слоти до сигналів, які надходять, коли кінцеві користувачі натискають елементи на діаграмах або наводять на них курсор миші. Це дає змогу додавати до діаграм такі елементи, як виноски.

### 2.3. Структури даних програми

Для представлення даних аналізів використовується два файли:

- загальні дані про пацієнта та аналіз;
- безпосередньо дані про аналізи.

Файли мають різну структуру даних і їх доцільно зберігати у форматі CSV значень, розділених комами.

Формат значень, розділених комами (CSV) використовувався для обміну та перетворення даних між різними програмами електронних таблиць протягом досить тривалого часу.

Визначення формату CSV. Хоча існують різні специфікації та реалізації формату CSV формальної специфікації не існує, що дозволяє широкий спектр

інтерпретацій CSV-файлів. Розглянемо формат, якого дотримується більшість реалізацій.

Кожен запис розташовується в окремому рядку, розділеному символом CRLF. Наприклад:

```
nn1,nn2,nn3 CRLF
```

```
aa1,aa2,aa3 CRLF
```

Останній запис може не мати кінцевого символу CRLF. Наприклад:

```
nn1,nn2,nn3 CRLF
```

```
aa1,aa2,aa3
```

В деяких випадках присутній необов'язковий рядок заголовка як перший рядок файлу з тим самим форматом, що й рядки звичайного запису. Цей рядок містить назви, що відповідають полям у файлі, і має таку саму кількість полів, як і записи в файлі

```
field_name1,field_name2,field_name3 CRLF
```

```
nn1,nn2,nn3 CRLF
```

```
aa1,aa2,aa3 CRLF
```

У заголовку та в кожному записі може бути одне або декілька полів, розділених комами. В кожному рядку має бути однакова кількість полів. Пробіли не ігноруються, і є частиною поля. В останньому полі запису не повинна стояти кома.

```
nn1,nn2,nn3
```

Поле може бути взяте в подвійні лапки.



```
"nn1", "nn2", "nn3" CRLF
aa1, aa2, aa3 CRLF
```

Поля, що містять розриви рядків (CRLF), подвійні лапки та коми, повинні бути взяті в подвійні лапки.

```
"aaa1", "b CRLF
bb2", "ccc3" CRLF
aaa, ttt, rrr
```

Якщо подвійні лапки використовуються для включення полів, тоді подвійні лапки, що з'являється всередині поля, потрібно екранувати, іншими подвійними лапками.

```
"aaa1", "b""bb2", "ccc3"
```

Загальним використанням CSV є US-ASCII, але інші набори символів, визначені для «текстового» дерева, можуть використовуватися разом із параметром «charset».

Параметр «заголовок» вказує на наявність або відсутність рядка заголовка.

Відповідно до пункту 4.1.1. RFC 2046, цей тип медіаданих використовує CRLF для позначення розривів рядків. Однак розробники можуть використовувати інші значення.

Міркування сумісності. Через відсутність єдиної специфікації існують значні відмінності між реалізаціями. Реалізації, які вирішують не використовувати додатковий параметр "заголовок", повинні прийняти власне рішення щодо того, відсутній заголовок чи присутній.

У той час як існують численні приватні специфікації для різних програм і систем, єдиної «основної» специфікації для цього формату не існує.

Нижче наведено приклад структури файлів даних.

### Файл загальних даних 1.head.csv

```
Name, Date_BD, Date_An  
Пацієнт, Дата_народження, Дата_Аналізу  
"Іванов А.В.", 19970423, 20230713
```

### Файл безпосередньо результатів аналізів 1.data.csv

```
#, Название, Ответ, Норма min, Норма max, ед  
0, АЧТЧ, 22.1, 23, 39, сек  
1, Протромбіновий час, 10.4, 8, 15, сек  
2, Протромбінове відношення, 0.81, 0.8, 1.3,  
3, МНВ (INR), 0.82, 0, 1.0,  
4, Фібриноген А, 2.7, 2, 4, г/л  
5, РФМК, 4.0, 0, 4.5, мг%
```

Така структура даних дозволяє використовувати прості засоби для конвертації результатів з різних типів файлів.

## РОЗДІЛ 3

### ОПИС ПРОГРАМИ АВТОМАТИЗОВАНОГО МОНІТОРИНГУ АНАЛІЗІВ

#### 3.1. Програмна реалізація інтерфейсу

Розглянемо окремі класи та методи програми.

Головне вікно програми реалізується класом `MainWindow`. В конструкторі класу виконується ініціалізація графічного інтерфейсу

```
MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
```

Також в конструкторі виконується ініціалізація компонентів таблиць та графіків

```
    build_Table();
    build_Chart();
}
```

Деструктор класу, відповідно, видаляє компоненти графічного інтерфейсу та вивільняє пам'ять

```
MainWindow::~MainWindow()
{
    delete ui;
```

```
}
```

Для відображення даних конкретного аналізу використовується таблиця. Ініціалізація таблиці виконується наступним чином

```
void MainWindow::build_Table()
{
    ui->tableWidget->setRowCount(10);
    ui->tableWidget->setColumnCount(10);
    ui->tableWidget->horizontalHeader()->setVisible(true);
    QStringList v_label;
    long i;
    for(i=0; i<10; i++)
    {
        v_label.append( QString("N %1").arg(i) );
    }
    ui->tableWidget->setVerticalHeaderLabels(v_label);
}
```

### Заповнення заголовків таблиці

```
for(i=0; i<10; i++)
{
    ui->tableWidget->setItem(i,0, new QTableWidgetItem(
QString("%1").arg( i*i ) ) );
}
}
```

Для відображення графіків використовується об'єкт класу QChart. Для створення графіку спочатку необхідно сформувати об'єкти рядів даних QLineSeries.

```
void MainWindow::build_Chart()
```

```

{
    QString ser_name[CNT_SERIES];
    ser_name[0]="E1";
    ser_name[1]="E2";
    ser_name[2]="E3";

    double ser_max[CNT_SERIES];
    ser_max[0] = 10;
    ser_max[1] = 10;
    ser_max[2] = 10;

    QLineSeries *series[CNT_SERIES];

    int i, j;
    for(i=0; i<CNT_SERIES; i++)
        series[i] = new QLineSeries();

```

**Потім необхідно заповнити об'єкт даними**

```

    QVector< QVector<double>> day_data;
    for (i=0;i<20;i++)
    {
        QVector<double> arr;
        for (j=0;j<CNT_SERIES;j++)
            arr.append(0);
        day_data.append(arr);
    }
    for (i=0;i<day_data.size();i++)
    {
        for(j=0; j<CNT_SERIES; j++)
            series[j]->append(i, day_data[i][j]);
    }
    for(i=0; i<CNT_SERIES; i++)
    {
        series[i]->setName(ser_name[i]);
    }

```

```
}
```

Відображення координатних вісей виконується за допомогою класу `QValueAxis`.

```
QValueAxis *axisY[CNT_SERIES];
chart_gr = new QChart;
chart_gr->setTitle("QT Charts example");
QValueAxis *axisX = new QValueAxis;
axisX->setTickCount(10);
axisX->setLabelFormat("%i");
axisX->setTitleText("Day");
chart_gr->addAxis(axisX, Qt::AlignBottom);
```

Для сформованих вісей додаємо відмітки

```
for(i=0; i<CNT_SERIES; i++)
{
    chart_gr->addSeries(series[i]);
    axisY[i] = new QValueAxis;
    axisY[i]->setTickCount(10);
    axisY[i]->setMin(0);
    axisY[i]->setLabelFormat("%.2e");
    if(i != 0) axisY[i]->setVisible(false);
    chart_gr->addAxis(axisY[i], Qt::AlignLeft);
    series[i]->attachAxis(axisX);
    series[i]->attachAxis(axisY[i]);
    axisY[i]->setMax(ser_max[i]);
}
```

Включаємо відображення легенди графіка

```
chart_gr->legend()->setVisible(true);
```

```
chart_gr->legend()->update();
```

### Розміщуємо графік у віджеті для відображення

```
QChartView *chartView = new QChartView(chart_gr);
chartView->setRenderHint(QPainter::Antialiasing);
chartView->setRubberBand(
QChartView::HorizontalRubberBand );
QVBoxLayout *layout = new QVBoxLayout;
layout->addWidget(chartView);
ui->widget_chart->setLayout(layout);
}
```

Обробник кнопки завантаження файлу з аналізами має наступну структуру. Спочатку виконується створення стандартного діалогового вікна для відкриття файлу

```
void MainWindow::on_toolButton_Load_clicked()
{
    QString fileName;
    fileName = QFileDialog::getOpenFileName(this,
        tr("Open File"), "..", tr("Files
(*.head.csv)"));
}
```

Після зчитування імені файлу виконується перетворення його імені

```
QString name;
int idx = fileName.lastIndexOf(".head.csv");
if( idx >=0)
{
    name = fileName.left(idx);
}
```

Виконується зчитування даних з файлу опису аналізів і відображення в елементі plainTextEdit

```

File_CSV csv_file;
if( csv_file.Load(name+".head.csv") ==0 )
{
    return;
}
QString str;
long i, j;
if(csv_file.res.size()>1)
{
    for(i=0; i<csv_file.res[0].size(); i++)
    {
        str += csv_file.res[1][i] + " : " +
csv_file.res[2][i];
        str += '\n';
    }
}
ui->plainTextEdit->setPlainText(str);

```

Після відображення основних даних зчитується файл з даними аналізів і його результати відображаються у таблиці

```

csv_file.Clear();
if( csv_file.Load(name+".data.csv") ==0 )
{
    return;
}
build_Table(csv_file.res);
}

```



### 3.2. Класи і методи для обміну даними

Клас для зчитування даних з файлу має наступну структуру

```
class File_CSV
{
    bool readCSVRow (QTextStream &in, QStringList *row);
public:
    File_CSV(){};
    int Load(QString filename);
    QVector<QStringList> res;
};
```

Метод `File_CSV::Load(QString filename)` забезпечує зчитування файлу та синтаксичний аналіз його структури

```
int File_CSV::Load(QString filename)
{
    res.clear();
```

Відкриття файлу з ім'ям `filename`

```
    QFile csv(filename);
    csv.open(QFile::ReadOnly | QFile::Text);
    if( !csv.isOpen() )
        return 0;
    QTextStream in(&csv);
    QStringList row;
```

## Зчитування рядків з файлу та аналіз рядка методом readCSVRow

```

while (readCSVRow(in, &row))
    res.append(row);
return 1;
}

```

### Метод аналізу рядка має наступний вигляд

```

bool File_CSV::readCSVRow (QTextStream &in, QStringList
*row)
{

```

### Структура масок автомату станів

-1: успішний кінець файлу

-2: кінець файлу всередині лапок

```

static const int delta[][5] = {
    { 1, 2, -1, 0, -1 }, // 0: розбір (store
char)
    { 1, 2, -1, 0, -1 }, // 1: розбір (store
column)
    { 3, 4, 3, 3, -2 }, // 2: вхід у лапки
(no-op)
    { 3, 4, 3, 3, -2 }, // 3: розбір всередині
лапок (store char)
    { 1, 3, -1, 0, -1 }, // 4: вихід з лапок
(no-op)
};

```

### Початок обробки та контроль кінця файлу

```
row->clear();  
if (in.atEnd())  
    return false;
```

### Аналіз символів рядка та перехід між станами атомата

```
int state = 0, t;  
char ch=0;  
QString cell;  
while (state >= 0) {  
    if (in.atEnd())  
        t = 4;  
    else {  
        in >> ch;  
        if (ch == ',') t = 0;  
        else if (ch == '\"') t = 1;  
        else if (ch == '\\n') t = 2;  
        else t = 3;  
    }  
    state = delta[state][t];  
    switch (state) {  
    case 0:  
    case 3:  
        cell += ch;  
        break;  
    case -1:  
    case 1:  
        row->append(cell);  
        cell = "";  
        break;  
    }  
}
```

Обробка стану незавершеного файлу, що призводить до генерації виключення

```

        if (state == -2)
            throw QString("End-of-file found while inside
quotes.");
        return true;
    }

```

### 3.3. Методи для взаємодії з базою даних

Для зберігання даних аналізів використовується вбудована СУБД SQLite.

SQLite — це вбудована бібліотека, яка реалізує самодостатній безсерверний механізм транзакційної бази даних SQL без конфігурації. На відміну від більшості інших баз даних SQL, SQLite не має окремого серверного процесу. SQLite читає та записує безпосередньо у звичайні дискові файли. Повна база даних SQL міститься в одному дисковому файлі. Формат файлу бази даних є крос-платформним.

Фреймворк QT забезпечує підтримку SQLite.

Розглянемо клас для зберігання даних з результатами аналізів. Опис класу має наступний вид.

```

class database_module
{
    QSqlDatabase db;
public:
    database_module();
    ~database_module();

```

```

        void openDB(QString &file);
        int    addRecord(QString name,   QString date,   QString
dateTest,   QString data);
    };

```

### Конструктор та деструктор класу тривіальні

```

database_module::database_module()
{}
database_module::~~database_module()
{
    db.close();
}

```

Відкриття бази даних виконується методом openDB. Цей метод також забезпечує створення таблиць бази даних при їх відсутності.

```

void database_module::openDB(QString &file)
{

```

### Створення об'єкту для бази даних і відкриття бази

```

        db = QSqlDatabase::addDatabase("SQLITE");
        db.setDatabaseName(file);
        if( !db.open() )
        {
            QMessageBox::critical(0,"Error",   QString("Can't open
DB %1")
                                .arg( db.lastError().text() ));
            return;
        }

```

## Створення таблиці, якщо вона відсутня

```

QString str;
 QSqlQuery query;
  str="CREATE TABLE IF NOT EXISTS \"BaseData\" (\"id\"
INTEGER NOT NULL,\"Name\"      varchar(256)          NOT
NULL,\"DateTest\"  varchar(48) NOT NULL, ";
  str += "\"Date\" varchar(48)      NOT      NULL,\"Analyze\"
varchar(2048),PRIMARY KEY(\"id\" AUTOINCREMENT) )";

```

## Виконання запиту на створення таблиці

```

if( !query.exec(str) )
{
    QMessageBox::critical( nullptr , "Error",
        QString("Can't create table BaseData:
%1").arg(query.lastError().text()));
    return;
}
}

```

## Додавання даних у таблицю виконується методом addRecord

```

int  database_module::addRecord(QString name, QString date,
QString dateTest, QString data)
{
    QSqlQuery query;
    QString str;
    str ="insert into BaseData";
    str += "(Name, DateTest, Date, Analyze)";
    str+=
        QString("
values(
\"%1\", \"%2\", \"%3\", \"%4\"") .arg(name) .arg(date) .arg(dateTest) .
arg(data);

```

```
if( !query.exec(str) )
{
    QMessageBox::critical(nullptr,"Error",
                          QString("BaseData Error %1 \n
%2").arg(query.lastError().text()).arg(str) );
    return 0;
}
return 1;
}
```

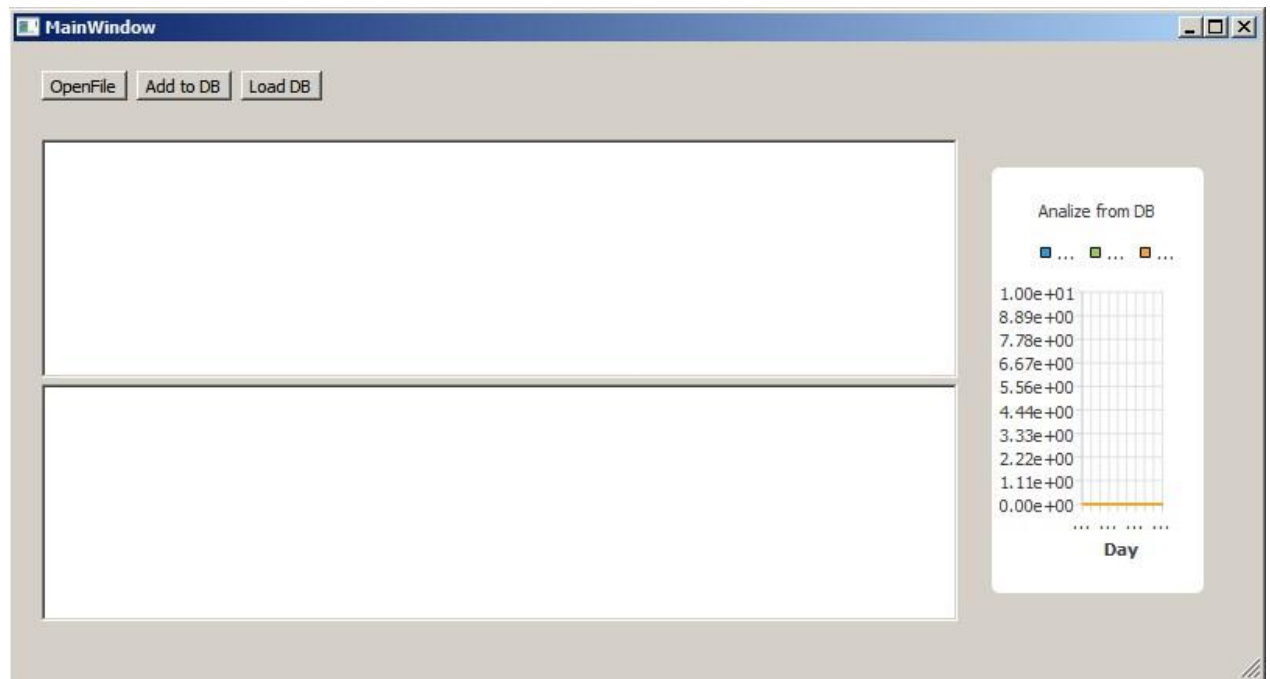


Рис. 3.1 Вигляд головного вікна програми

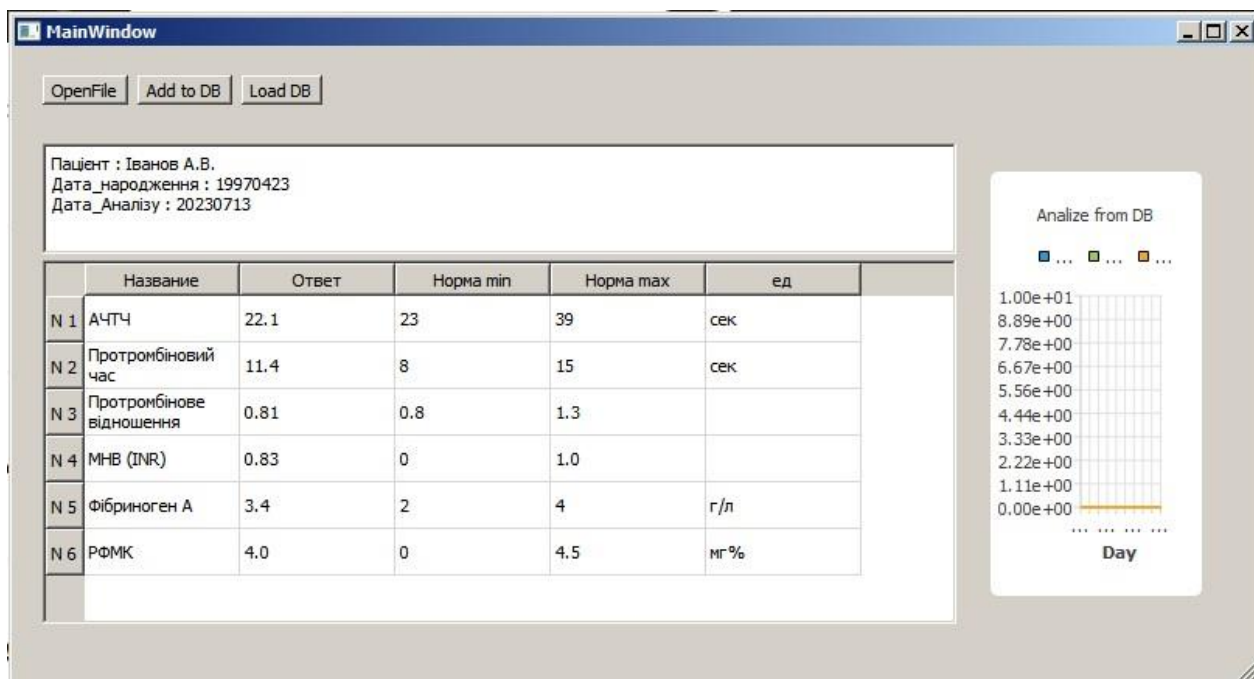


Рис. 3.2. Вигляд завантажених даних у програму

Розроблена програма забезпечує зрозумілий інтерфейс для зчитування файлів з аналізами.



## ВИСНОВКИ

Сучасні інформаційні технології відіграють важливу роль в розвитку медичної галузі, прискорюючи процеси, покращуючи якість надання медичних послуг та сприяючи науковому прогресу. Електронні системи управління лікарнями дозволяють замінити традиційні паперові медичні записи цифровими, що полегшує доступ до інформації, її обмін та забезпечення безпеки даних. Електронні медичні картки забезпечують повний огляд медичної історії пацієнта, що полегшує координацію медичної допомоги.

За допомогою ІТ впроваджуються рішення для проведення консультацій та обстежень на відстані, що особливо важливо для пацієнтів, які знаходяться в зоні віддалених або важкодоступних місць. Це також дозволяє ефективно взаємодіяти між лікарями та пацієнтами без необхідності фізично знаходитися в одному місці.

Сучасні ІТ дозволяють опрацьовувати великі обсяги медичних даних та використовувати штучний інтелект для аналізу зображень, діагностики, прогнозування захворювань та підбору індивідуалізованих методів лікування.

Розвиток мобільних технологій дозволяє створювати додатки, які допомагають пацієнтам ведення особистого здоров'я, нагадують про прийом ліків, забезпечують взаємодію з медичним персоналом та ведуть моніторинг показників здоров'я.

Важливою частиною впровадження ІТ в медицину це використання сенсорів та з'єднання медичних пристроїв через Інтернет розширює можливості моніторингу пацієнтів у реальному часі та автоматизації збору медичних даних.

Проте необхідно пам'ятати про проблему захисту персональних даних пацієнтів.

Загалом, сучасні ІТ відіграють значущу роль у вдосконаленні всіх аспектів медичної галузі, роблячи її більш ефективною, доступною та інноваційною.

Розробка системи автоматизованого моніторингу результатів медичних аналізів є важливим етапом в удосконаленні сфери охорони здоров'я. Такі системи можуть значно полегшити роботу медичного персоналу та покращити точність та швидкість обробки аналізів. Розроблена система передбачала наступні етапи в роботу: розробку структури системи автоматизованого моніторингу медичних аналізів; розробку підсистеми відображення даних системи автоматизованого моніторингу; розробку структури даних програми системи; розробку програмних модулів системи автоматизованого моніторингу аналізів; розробку класів і методів для обміну даними; розробка класів і методів для взаємодії з базою даних; тестування системи.

Розроблена система дозволяє автоматизувати процес обробки медичних аналізів, що робить їх обробку швидшою та ефективнішою. Це важливо для оперативного прийняття рішень у лікуванні пацієнтів.

Застосування розробленої програми дозволяє уникнути помилок, які можуть виникнути при ручній обробці аналізів. Це підвищує точність результатів і дозволяє швидше виявляти патології та визначати оптимальний план лікування.

Система може виконувати рутинні завдання, які раніше вимагали б великої кількості робочого часу від медичного персоналу, що дозволяє зосередитися на більш важливих аспектах надання медичної допомоги.

Програма представлена в роботі може забезпечувати постійний моніторинг показників здоров'я пацієнтів, що сприяє ранньому виявленню проблем та дозволяє запобігти їх подальшому розвитку.

Забезпечення сумісності з існуючими системами електронних медичних записів дозволяє покращити обмін даними та забезпечити єдиною точкою доступу до інформації для медичного персоналу.

Розроблений засіб передбачає захист конфіденційності даних. Важливим аспектом розробки є впровадження заходів забезпечення конфіденційності даних пацієнтів та відповідність вимогам стандартів безпеки в області охорони здоров'я.

Таким чином, розроблена системи автоматизованого моніторингу результатів медичних аналізів має великий потенціал у полі удосконалення медичної практики та підвищення якості надання медичних послуг.

## РЕКОМЕНДАЦІЇ

В подальшому вдосконаленні системи потрібно акцентувати увагу на захисті особистих даних пацієнтів, враховуючи високі стандарти безпеки, для забезпечення довіри та лояльності користувачів.

Також необхідно приділити увагу інтегруванню з існуючими платформами та забезпечувати постійну підтримку користувачів для зручного та ефективного використання.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. QChart Class [Електронний ресурс]. – Режим доступу: [www. URL: https://doc.qt.io/qt-6/qchart.html](http://www.qt.io/doc/qt-6/qchart.html) (дата звернення: 14.10.23)
2. Qt 6 QML Jürgen Bocklage-Ryannel, Cyril Lorquet, Johan Thelin [Електронний ресурс]. – Режим доступу: [www. URL: https://www.qt.io/hubfs/\\_website/QML%20Book/qt6book-with-frontpage.pdf](http://www.qt.io/hubfs/_website/QML%20Book/qt6book-with-frontpage.pdf) (дата звернення: 12.09.23)
3. Qt and C++ [Електронний ресурс]. – Режим доступу: [www. URL: https://www.qt.io/product/qt6/qml-book/ch17-qtcpp-qtcpp](http://www.qt.io/product/qt6/qml-book/ch17-qtcpp-qtcpp) (дата звернення: 10.09.23)
4. Qt SQL [Електронний ресурс]. – Режим доступу: [www. URL: https://doc.qt.io/qt-6/qtsql-index.html](http://www.qt.io/doc/qt-6/qtsql-index.html) (дата звернення: 10.10.23)
5. SQL Database Drivers [Електронний ресурс]. – Режим доступу: [www. URL: https://doc.qt.io/qt-6/sql-driver.html](http://www.qt.io/doc/qt-6/sql-driver.html) (дата звернення: 15.09.23)
6. What Is SQLite? [Електронний ресурс]. – Режим доступу: [www. URL: https://www.sqlite.org/index.html](http://www.sqlite.org/index.html) (дата звернення: 20.10.23)
7. О. М. Ключко Медична інформаційна система моніторингу стану здоров'я населення із захистом персональних даних / Ключко О. М. // Медична інформатика та інженерія. 2020, № 1, С. 17-28. DOI: <https://doi.org/10.11603/mie.1996-1960.2020.1.11126>

ДОДАТКИ

## ДОДАТОК А

Вихідний код програми ledaudio.ino