КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 «Комп'ютерні науки»
Освітньо-наукова програма «Управління проектами»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

«Дослідження процесів управління проєктом розробки інформаційної системи для моніторингу здоров'я серця людини»

Студента 2-го курсу групи УП-21

Владислава БЄЛИХ
(ім'я, прізвище)

(підпис студента)

Науковий керівник:

к.е.н.
(науковий ступінь, вчене звання)

Анна КОЛОМІЄЦЬ
(ім'я, прізвище)

(дата) *(підпис)*

Попередній захист:

(Висновок: "До захисту в Екзаменаційній комісії")

Завідувач кафедри технологій управління

(підпис) Віктор МОРОЗОВ *(прізвище, ініціали)* *(дата)*

Київ – 2025

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій

Кафедра технологій управління
Освітній рівень Магістр
Спеціальність 122 Комп'ютерні науки
Освітня програма Управління проектами

ЗАТВЕРДЖУЮ
Завідувач кафедри
професор Морозов В.В.

_____
"26" листопада 2024 року

З А В Д А Н Н Я
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Студент: Бєлих Владислав Юрійович
Група: УП-21

1. Тема кваліфікаційної роботи
«Дослідження процесів управління проєктом розробки інформаційної системи для моніторингу здоров'я серця людини».
Затверджена наказом по від 26 листопада 2024 року № 5.
2. Строк подання студентом готової роботи - "6" травня 2025 р.
3. Цільова установка та вихідні дані до роботи: Дослідження сучасних підходів до управління IT-проєктами, зокрема у сфері цифрової медицини; вивчення нормативних вимог до обробки персональних медичних даних; аналіз методів математичного моделювання в контексті систем кардіомоніторингу; розробка та впровадження інформаційної системи, що інтегрує штучний інтелект і мобільні технології для персоналізованого моніторингу стану серця людини.
4. Зміст роботи: Аналіз існуючих інформаційних систем для моніторингу здоров'я серця; визначення проблемних аспектів і потреб у цифровій медичній сфері; дослідження нормативно-правової бази (GDPR, HIPAA); формування технічних та функціональних вимог до інформаційної системи; створення концептуальної та математичної моделей системи; побудова ієрархічної структури робіт (WBS), структури управління проєктом (OBS), планування ресурсів і бюджету, управління ризиками; розробка архітектури програмного забезпечення; моделювання бази даних; проектування алгоритмів функціонування системи та користувацького інтерфейсу; тестування системи та оцінка її ефективності.
5. Перелік графічного матеріалу: Актуальність і проблематика дослідження; концептуальна модель інформаційної системи моніторингу здоров'я серця;

математична модель аналізу показників серцевої діяльності; алгоритм обробки даних; WBS-структура проєкту; OBS-структура управління; діаграма ризиків; беклог продукту; календарний план реалізації; архітектура інформаційної системи; концептуальна та логічна моделі бази даних; інтерфейс користувача (сторінки мобільного застосунку та вебверсії); результати тестування системи.

6. Календарний план виконання роботи:

№ з/п	Назва частин роботи	План виконання роботи
1	Вивчення літературних джерел з предмету дослідження	01.12.24-10.12.24
2	Аналіз існуючих інформаційних систем моніторингу здоров'я серця	10.12.24-15.12.24
3	Складання розгорнутого плану дипломної роботи	16.12.24-20.12.24
4	Ознайомлення наукового керівника з планом. Внесення змін	21.12.24
5	Підготовка розділу 1	22.12.24-08.01.25
6	Підготовка розділу 2	08.01.25-20.02.25
7	Підготовка розділу 3	20.02.25-20.03.25
8	Підготовка розділу 4	20.03.25-30.03.25
9	Підготовка розділу 5	30.03.25-17.04.25
11	Оформлення кваліфікаційної роботи, перевірка коректності всіх розділів	17.04.25-27.04.25
12	Передача кваліфікаційної роботи науковому керівнику	06.05.25
13	Передача кваліфікаційної роботи рецензенту для рецензування	12.05.25
14	Захист роботи	26.05.25-28.05.25

Дата видачі завдання «26» листопада 2024 р.

Керівник роботи доцент кафедри ТУ, к.е.н. Анна КОЛОМІЄЦЬ

_____

(підпис)

Завдання прийняв до виконання студент групи УП-21

БЄЛИХ Владислав

(ім'я, прізвище)

_____

(підпис)

АНОТАЦІЯ

кваліфікаційної роботи магістра на тему

«Дослідження процесів управління проєктом розробки персоналізованої інформаційної системи моніторингу здоров'я серця людини»

Студент: Бєлих Владислав Сергійович

Науковий керівник: Коломієць Анна Степанівна

Рік захисту – 2025

Метою кваліфікаційної роботи є розробка та впровадження персоналізованої інформаційної системи для моніторингу стану серця людини, яка поєднує сучасні підходи до управління IT-проєктами з інструментами штучного інтелекту для підвищення ефективності ранньої діагностики та профілактики серцево-судинних захворювань.

Ціль проєкту – створити багатофункціональну інформаційну систему з інтеграцією мобільних пристроїв і хмарних технологій, яка забезпечує збір, аналіз та візуалізацію медичних показників у режимі реального часу, сприяє підвищенню якості медичних послуг та впровадженню цифрових інновацій у галузі охорони здоров'я.

Наукова новизна дослідження полягає у поєднанні сучасних методів управління IT-проєктами (WBS, управління ризиками, бюджетування) з технологіями машинного навчання для створення персоналізованої медичної системи, здатної автоматизувати аналіз стану серця людини та генерувати оперативні рекомендації для медичного персоналу.

Практичне значення отриманих результатів:

- створена інформаційна система може бути впроваджена в медичних установах для персоналізованого моніторингу пацієнтів у режимі реального часу;
- інтеграція мобільних застосунків і хмарних платформ забезпечує зручний доступ до даних для лікарів і користувачів;

- запропоновані технологічні та управлінські рішення можуть бути адаптовані до інших проєктів цифрової медицини.

Кваліфікаційна робота складається з анотації, вступу, основної частини, що включає п’ять розділів, висновків, переліку використаних джерел та додатків.
У першому розділі проведено аналіз сучасних інформаційних систем для моніторингу здоров’я серця, визначено проблематику та технічні вимоги.
У другому розділі сформовано концептуальну та математичну модель системи, виконано контент-аналіз нормативних вимог.
У третьому розділі обґрунтовано управлінську структуру, визначено ресурси, графік реалізації, побудовано WBS, розраховано бюджет та ідентифіковано ризики.
У четвертому розділі реалізовано архітектуру програмного забезпечення, проєкт бази даних, алгоритм функціонування та інтерфейс користувача.
У п’ятому розділі здійснено тестування системи, оцінено її ефективність, відповідність вимогам безпеки та можливості впровадження.
У висновках узагальнено результати дослідження та наведено перспективи подальшого розвитку.

Робота містить **109 сторінок основного тексту**, **30 рисунків**, **18 таблиць**. Додатки складають **4 сторінки**.

Ключові слова: інформаційна система, моніторинг здоров’я, серцево-судинні захворювання, кардіомоніторинг, управління IT-проєктом, WBS, база даних, алгоритм, штучний інтелект, безпека, тестування.

ЗМІСТ

ВСТУП

Сучасна медицина активно використовує інформаційні технології для покращення діагностики, лікування та профілактики захворювань. Особливо актуальним є питання впровадження персоналізованих систем моніторингу здоров’я серця, які дозволяють здійснювати безперервний контроль стану пацієнта в режимі реального часу. Це набуває особливої значущості в контексті зростаючої кількості серцево-судинних захворювань, які залишаються однією з провідних причин смертності у світі.

Актуальність дослідження зумовлена необхідністю вдосконалення методів моніторингу здоров'я серця, що дозволить підвищити ефективність діагностики, своєчасність лікування та профілактики серцево-судинних захворювань. Впровадження персоналізованих інформаційних систем відслідковування здоров’я серця може значно покращити якість медичної допомоги, забезпечити зручність для пацієнтів та зменшити навантаження на медичний персонал.

Мета дослідження полягає у розробці персоналізованої інформаційної системи для моніторингу стану серцево-судинної системи, яка включає програмний застосунок та інтегровані носимі пристрої для безперервного збору та аналізу медичних показників. За рахунок використання сучасних технологій штучного інтелекту, мобільних платформ та алгоритмів аналізу великих даних буде забезпечено точну діагностику, оперативне реагування на критичні стани пацієнта та своєчасну профілактику серцево-судинних захворювань. В результаті впровадження цієї системи пацієнти отримають зручний інструмент для дистанційного контролю свого стану здоров'я, а медичні працівники — ефективний інструмент для моніторингу, оцінки стану та прийняття обґрунтованих рішень щодо лікування пацієнтів із серцево-судинними захворюваннями.

Предмет дослідження - процеси управління проєктом розробки персоналізованої інформаційної системи для кардіомоніторингу, зокрема

управління часом та вартістю, управління ризиками, а також процеси формування та побудови ефективної команди проєкту.

Об'єкт дослідження – процеси впровадження інформаційних технологій для моніторингу здоров’я серця людини в систему управління медичними установами, спрямовані на покращення якості діагностики, лікування та профілактики серцево-судинних захворювань.

Методи дослідження: системний аналіз для виявлення проблем у сфері кардіомоніторингу, SWOT-аналіз для оцінки ринку інформаційних систем, контент-аналіз нормативних документів (GDPR, HIPAA), концептуальне моделювання архітектури програмного забезпечення, математичне моделювання для оцінки стану серцево-судинної системи, об’єктно-орієнтоване проєктування бази даних, а також методи тестування та валідації інформаційної системи в умовах імітованого середовища.

Наукова новизна роботи полягає в розробці комплексного підходу до управління проєктом створення персоналізованої інформаційної системи моніторингу здоров'я серця людини з інтеграцією сучасних методів управління часом, вартістю та ризиками проєкту. Удосконалено процеси формування ефективної команди проєкту для забезпечення високої продуктивності та якості розробки системи. Отримало подальший розвиток застосування штучного інтелекту та алгоритмів аналізу медичних даних у системах моніторингу серцево-судинних захворювань, що дозволяє підвищити точність діагностики та ефективність профілактики..

Практичне значення отриманих результатів:

- розроблена інформаційна система може бути впроваджена в медичних установах для персоналізованого моніторингу стану серця пацієнтів у режимі реального часу;
- інтеграція з мобільними пристроями та хмарними сервісами забезпечує зручний доступ до даних для лікарів і пацієнтів, що сприяє оперативному прийняттю рішень щодо лікування;

- запропонований підхід до поєднання методів управління IT-проєктами та технологій штучного інтелекту може бути використаний в інших проєктах цифрової медицини з подібною структурою даних і функціональністю.

Апробація результатів*:*

2-га міжнародна науково-практична конференція «Інформаційні системи та технології: результати і перспективи» (IST 2025), 5 березня 2025 р.,с. 182-185. Київ, Україна.

РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ ІНФОРМАЦІЙНИХ СИСТЕМ ВІДСЛІДКОВУВАННЯ ЗДОРОВ'Я СЕРЦЯ

1.1 Аналіз ринку інформаційних систем для моніторингу здоров'я серця

На сьогоднішній день ринок інформаційних систем для моніторингу здоров'я серця характеризується низкою важливих тенденцій:

1. Поширення носимих пристроїв: за останні роки значно зросла популярність портативних девайсів [31], які вимірюють показники серцевого здоров'я, що відкриває нові можливості для моніторингу та аналізу даних в реальному часі.

2. Розвиток телемедицини: технології телемедицини постійно удосконалюються [16, 18], що робить віддалене спостереження за серцевим станом пацієнтів більш ефективним та доступним.

3. Інтеграція з мобільними додатками: завдяки постійному розвитку мобільних технологій, інформаційні системи для моніторингу серцевого здоров'я стають все більш інтегрованими з мобільними додатками [9], що забезпечує зручність та доступність для користувачів.

За останні кілька років спостерігається стійкий ріст ринку інформаційних систем для моніторингу здоров'я серця. За даними аналітичних агентств, цей ріст прогнозується і в найближчому майбутньому очікується ще більше зростання обсягів цього сектора. Прогрес у сфері технологій, підвищена увага до здорового способу життя та розуміння важливості моніторингу серцевого здоров'я сприяють цьому зростанню.

Зокрема, статистичні дані показують зростання популярності носимих пристроїв, збільшення кількості користувачів мобільних додатків для моніторингу здоров'я, а також збільшення кількості телемедичних консультацій у сфері кардіології.

Прогнозується, що ринок інформаційних систем для моніторингу здоров’я серця буде продовжувати динамічний розвиток у наступні роки. Зростання обсягів цього ринку буде зумовлене не лише технологічними інноваціями, але й збільшеною увагою до проблем серцево-судинних захворювань у світлі демографічних та соціальних змін.

На сьогоднішній день на ринку інформаційних систем для моніторингу здоров’я серця можна виділити декілька основних гравців, які активно впливають на його розвиток:

Philips Healthcare: компанія відома своїми інноваційними рішеннями у галузі медичного обладнання та інформаційних систем для моніторингу здоров’я серця.

GE Healthcare: лідер у виробництві різноманітного медичного обладнання, включаючи системи для моніторингу та діагностики серцево-судинних захворювань.

Medtronic: відомий своїми імплантованими пристроями та системами для моніторингу серцевого стану пацієнтів.

Abbott: компанія спеціалізується на розробці інноваційних технологій для моніторингу та діагностики серцевих захворювань.

Siemens Healthineers: виробник широкого спектру медичних систем та обладнання, включаючи ті, що використовуються у кардіології та моніторингу здоров’я серця.

Технологічні тренди у сфері моніторингу здоров’я серця включають:

1. Телемедицина: розширення можливостей дистанційного моніторингу та консультування лікарів за допомогою віддалених технологій.

2. Мобільні додатки: зростання популярності мобільних додатків для моніторингу здоров’я серця та передачі даних між пацієнтами та лікарями.

3. Штучний інтелект: використання ШІ для аналізу великих обсягів медичних даних [5, 6] та покращення точності діагностики та передбачення стану серця.

4. Інтернет речей (IoT): застосування підключених до мережі пристроїв для постійного моніторингу та збору даних про стан серця.

Конкурентне середовище у сфері моніторингу здоров’я серця включає як великі медичні компанії, так і стартапи, що спеціалізуються на розробці інноваційних технологій. Ринок можна поділити на такі сегменти:

Споживчі пристрої: носимі пристрої та мобільні додатки для самомоніторингу та збору даних про стан серця.

Професійні медичні системи: спеціалізовані системи для лікарів та медичних установ для моніторингу та діагностики серцевих захворювань.

Зробимо оцінку викликів та можливостей для подальшого розвитку ринку:

Виклики:

Безпека та конфіденційність даних: завжди важливо забезпечувати надійний рівень захисту медичних даних, особливо у сфері серцево-судинної медицини, де дані можуть бути особливо чутливими.

Стандартизація технологій: різноманітність інформаційних систем та технологій може ускладнювати інтеграцію та взаємодію між ними, тому важливо працювати над стандартизацією для більшої сумісності.

Фінансування та доступність: деякі інноваційні технології можуть бути високо вартісними, що може стати перешкодою для їх широкого поширення та доступності для всіх шарів населення.

Можливості:

Розвиток штучного інтелекту та аналітики даних: ШІ може допомогти вдосконалити процеси аналізу та передбачення стану здоров’я серця, що веде до більш точних та індивідуалізованих рішень у медичній практиці.

Зростання популярності носимих пристроїв: послуги моніторингу здоров’я через носимі пристрої стають все більш доступними та зручними для користувачів, що сприяє збільшенню їхнього використання.

Розвиток телемедицини: віддалені консультації та моніторинг через телемедицину дозволяють здійснювати якісну медичну допомогу на відстані, що розширює можливості доступу до медичних послуг.

Інновації в медичній електроніці: нові технології та розробки в галузі медичних пристроїв дозволяють створювати більш ефективні та точні засоби для моніторингу та діагностики серцевих захворювань.

1.1.1 Аналіз проблеми використання інформаційних систем моніторингу здоров'я

Аналіз літературних джерел свідчить про багатовекторність підходів до розробки інформаційних систем у галузі охорони здоров’я, зокрема для моніторингу стану серця людини. У цій підсистемі дослідження особливу увагу зосереджено на питаннях управління ІТ-проєктами, використання штучного інтелекту та захисту персональних даних.

У роботі Г. Керцнера [1] представлено системний підхід до управління проєктами, що охоплює планування, контроль і реалізацію складних ініціатив. Автор наголошує на важливості чіткої структури WBS та інтегрованого контролю, що є релевантним у проєктах розробки ІС для медицини. Проблематика адаптації класичних моделей управління до високорегульованого середовища медицини залишається відкритою.

К. Швалбе у своїй праці [2] акцентує увагу на специфіці управління ІТ-проєктами, зокрема у контексті гнучких методологій. Вона пропонує поєднувати Waterfall і Agile у складних умовах, таких як розробка медичних систем. Важливою є проблема інтеграції ІТ у регульовану галузь із високими вимогами до безпеки.

К. Хелдман [3] зосереджується на сертифікаційних вимогах і формалізованих процесах управління, що можуть бути використані для підготовки проєктного середовища розробки ІС. У роботі представлено чіткі моделі ризик-менеджменту та комунікаційних процесів.

Офіційний посібник PMBOK (7-ме видання) [4] служить основою для стандартизації життєвого циклу продукту, орієнтуючись на цінності (value

delivery) та управління ризиками. Хоча підхід є узагальненим, він пропонує універсальні принципи, які можуть бути адаптовані для медичних ІС.

У статті Sayed Abdelgaber та співавт. [5] зроблено системний огляд ролі ШІ в медичній підтримці. Автори звертають увагу на проблеми недостатньої структурованості даних і відсутності уніфікованих протоколів. Як рішення пропонується використання ML-моделей, інтегрованих із системами IoT.

Робота A. Esteva [6] демонструє приклади використання глибокого навчання (DL) для медичних задач. Розглядаються питання підбору моделей, використання попередньо навчених архітектур та проблематика overfitting, що актуально для медичних ІС із обмеженими наборами даних.

У книзі Ч. Аґґарвала [7] детально описані принципи побудови та навчання нейронних мереж. Теоретична база, представлена у цій праці, є основою для майбутнього розширення функціоналу системи моніторингу із застосуванням нейромереж, наприклад, для аналізу ЕКГ.

B. Mesko у своїй роботі [8] наголошує на важливості цифрової грамотності медичних фахівців у контексті впровадження AI. Автор підтримує ідею застосування explainable AI (XAI) як інструменту формування довіри до автоматизованих діагностичних рішень.

Стаття M. Hassanalieragh та ін. [9] присвячена інтеграції IoT-пристроїв у медичні системи. Автори аналізують переваги edge computing, що дозволяє здійснювати обробку даних поблизу джерела їх виникнення, зменшуючи затримки та підвищуючи надійність.

Нормативний документ HIPAA [10], який встановлює правила конфіденційності та захисту медичної інформації. Дотримання вимог цього закону є критично важливим при проєктуванні інформаційних систем у сфері охорони здоров’я, особливо у контексті обробки персональних даних.

У статті Abdullah Aldwean і Dan Tenney [11] розглянуто основні виклики впровадження штучного інтелекту в охороні здоров’я. Автори акцентують увагу на проблемах етичного характеру, нестачі інфраструктури, а також опору змінам

серед медичного персоналу. Серед рішень – запровадження політик цифрової трансформації, покрокова інтеграція AI у робочі процеси.

Adebayo Augustine Adeniyi та Oluwademiladeayo Adeniyi [12] проводять огляд ролі ML і AI у модернізації охорони здоров'я. Підкреслено трансформаційний потенціал в частині діагностики, прогнозування та підтримки прийняття рішень. Автори вказують на виклики, пов'язані з якістю даних, та пропонують системи очищення та стандартизації як основу для ефективного застосування ML.

Ali Abdin у наративному огляді [13] аналізує застосування AI у діагностичній медицині. Основна увага приділяється ефективності алгоритмів у порівнянні з традиційними методами, зокрема при аналізі зображень. Підкреслено важливість інтерпретованості рішень, що генеруються ШІ.

Badr Alnasser [14] розглядає економічні аспекти впровадження AI в системи охорони здоров'я. Автор аналізує затрати на розгортання інфраструктури, навчання персоналу та супутні витрати. Робиться висновок, що при належному управлінні витрати компенсуються підвищенням ефективності медичних послуг і зниженням рівня помилок.

У статті Chayakrit Krittanawong та Hafeez Ul Hassan Virk [15] представлено метааналіз застосування машинного навчання у прогнозуванні серцево-судинних захворювань. Автори демонструють високу ефективність таких алгоритмів у виявленні ризиків. Обґрунтовується доцільність їх застосування у рамках систем первинного моніторингу.

Christopher Tsai і Justin Starren [16] аналізують роль медичних сестер у впровадженні телемедичних технологій у домашнє середовище. Окремо розглядається взаємодія з пацієнтом, підготовка обладнання та технічна підтримка. Праця є важливою з погляду забезпечення користувацької підтримки у децентралізованих ІС.

HIPAA Journal [17] детально пояснює вимоги до шифрування даних згідно з HIPAA. Розглянуто технічні аспекти впровадження шифрування в медичних

системах, включаючи вимоги до алгоритмів (AES, RSA), захисту ключів та контроль доступу до даних.

GDPR.eu [18] — офіційний портал з роз'ясненнями положень GDPR, який є базовим документом для забезпечення прав суб'єктів персональних даних. У контексті проєкту важливими є принципи прозорості, право на забуття, захист при передачі даних та вимоги до Data Protection by Design.

Michael Fitzgerald і Nina Kruschwitz [19] у дослідженні MIT Sloan аналізують стратегічну важливість цифрової трансформації для організацій. Для сфери охорони здоров'я актуальними є рекомендації щодо переходу до гнучкої інфраструктури, орієнтованої на дані та користувача.

У роботі Piyush Mathur [20] наведено підсумок впровадження AI у медичній практиці за рік. Розглянуто приклади успішних кейсів використання AI у прогнозуванні, клінічному рішенні та автоматизації. Робота підкріплює ідею поступового впровадження інтелектуальних підсистем у традиційні інформаційні рішення.

У роботі Ramo Sendelj [21] розглянуто ключові виклики кібербезпеки в системах охорони здоров'я. Автор виділяє високий ризик витоку даних та недосконаліть процесів автентифікації в медичних ІС. Як рішення пропонується впровадження систем багаторівневого захисту, постійного моніторингу загроз та навчання персоналу з питань цифрової гігієни.

Sabine Koch [22] аналізує роль інформаційних систем у забезпеченні медичної допомоги людям похилого віку. Особливу увагу приділено персоналізації догляду та інтеграції із соціальними службами. Автор підкреслює необхідність адаптації ІС до когнітивних і фізіологічних особливостей літніх користувачів.

Thomas H. Davenport [23] оцінює потенціал застосування AI у сфері охорони здоров'я. У праці визначено ключові напрями трансформації клінічної практики: підтримка прийняття рішень, автоматизація адміністративних процесів і персоналізована медицина. Автор акцентує увагу на необхідності регуляторної бази для ШІ в охороні здоров'я.

Ahmad Chaddad та Yihang Wu [24] аналізують концепцію federated learning для медичних додатків. Розглядається модель децентралізованого навчання, яка дозволяє залучати медичні дані без їх передачі до центрального сховища. Це забезпечує відповідність вимогам конфіденційності (GDPR, HIPAA) при використанні AI.

Casey Lynnette Overby та Peter Tarczy-Hornoch [25] у своїй статті зосереджуються на викликах трансляційної біоінформатики у персоналізованій медицині. Вказано на потребу у високоякісних даних, інтероперабельності між системами та стандартизації обміну медичною інформацією.

David W. Bates та співавт. [26] обґрунтовують зниження кількості медичних помилок за рахунок впровадження ІТ. Зокрема, пропонується автоматизація призначень, електронні медичні записи та інтегровані системи підтримки рішень. Вказується на роль інформаційної безпеки у запобіганні технічним збоєм.

Deniz Ozel та Uğur Bilge [27] описують повний цикл розробки веб-застосунку для підтримки рішень у відділенні інтенсивної терапії. Розглянуто питання архітектури, інтерфейсу користувача та тестування функціоналу. Праця демонструє практичну реалізацію ІС із критичними вимогами до надійності.

Dipak Kalra та David Ingram [28] аналізують концепцію електронних медичних записів у міжнародному контексті. Описано виклики щодо сумісності, юридичної валідності та захисту інформації. Наголошується на важливості єдиного інформаційного простору в медицині.

Fei Wang та Lawrence Peter Casalino [29] розглядають прогрес у впровадженні deep learning у медицині. Визначаються основні переваги: вища точність, можливість навчання на неструктурованих даних. Водночас зазначається про потребу у валідації моделей і запобіганні їх неконтрольованому використанню.

Indra Neil Sarkar [30] аналізує роль біомедичної інформатики в трансляційній медицині. Зазначається, що ефективна ІС повинна забезпечувати

швидкий перехід від наукових даних до клінічної практики. Це передбачає інтеграцію аналітики, дослідницьких платформ і медичних записів.

Min Chen та Yujun Ma [31] представляють концепцію "Wearable 2.0", що базується на інтеграції людини з хмарними інфраструктурами. Автори описують архітектуру, де носимі пристрої виконують функцію сенсорного шару, а обробка даних здійснюється на рівні хмари. Вказується на переваги щодо масштабованості та доступу до ресурсів в реальному часі.

Muzammil Khan та Muhammad Taqi Mehran [32] проводять огляд використання AI у боротьбі з COVID-19. У фокусі – системи моніторингу симптомів, прогнозування ризику захворювання, автоматизація тріажу. Автори зазначають, що подібні системи можуть бути адаптовані для хронічних хвороб, зокрема серцевих.

Nicola Rieke та Fausto Milletari [33] обґрунтовують застосування федеративного навчання в цифровій медицині. Автори вказують на проблеми конфіденційності при централізованому навчанні моделей і пропонують FL як технологію, що дозволяє зберігати дані локально, тренуючи спільну модель на розподілених пристроях.

Spyros Kitsiou [34] здійснює аналіз існуючих стандартів електронних медичних записів. Виділено ключові відмінності між HL7, CDA, OpenEHR та іншими. Автор наголошує на необхідності стандартизації структур і протоколів для досягнення сумісності в рамках мультиінституційних медичних ІС.

Yong Jiang та Yi Dong [35] у своєму огляді відстежують історію, поточний стан і майбутнє застосування AI в охороні здоров’я. Автори описують переходи від rule-based систем до сучасних моделей глибокого навчання, з акцентом на персоналізацію, точність та швидкість рішень.

OWASP Top Ten [36] описує основні ризики веб-додатків, включно з SQL-ін'єкціями, міжсайтовими скриптами (XSS), неправильним керуванням доступом. Для проєкту медичної ІС це джерело є критично важливим у контексті захисту API та клієнт-серверної взаємодії.

Security Pillar – AWS Well-Architected Framework [37] окреслює принципи побудови безпечної хмарної інфраструктури. Документ містить рекомендації щодо ізоляції обчислювальних середовищ, управління ключами, контролю доступу та журналювання дій користувачів.

NIST SP 800-63B [38] (Grassi, Fenton) надає специфікації щодо цифрової ідентифікації та управління автентифікацією. Особливо актуальні положення про багатофакторну автентифікацію (MFA), життєвий цикл паролів, збереження сесій та безпечне оновлення облікових даних.

Google Cloud Security [39] висвітлює підходи до шифрування даних «у спокої» (at rest) та «під час передачі» (in transit). Розглядаються методи ключового управління, використання апаратного безпечного модуля (HSM), автоматичне шифрування на рівні диска та сервісів.

Kubernetes Documentation [40] надає базову інформацію про архітектуру та взаємодію компонентів контейнеризованих застосунків. Описуються сервіси, поди, кластерна структура, що є корисним для розгортання масштабованої, ізольованої інфраструктури медичних ІС.

Wazuh Documentation [41] описує механізми забезпечення безперервного моніторингу безпеки в інформаційних системах. Зокрема, розглянуто можливості системи SIEM для виявлення аномальної поведінки користувачів, журналювання інцидентів безпеки та інтеграції з інструментами попередження вторгнень (IDS/IPS).

ISO/IEC 27035-1:2016 [42] є міжнародним стандартом, який регламентує управління інцидентами інформаційної безпеки. У документі описано етапи реагування на інциденти: ідентифікація, аналіз, мінімізація впливу, відновлення та навчання. Цей стандарт може бути інтегрований у політику безпеки медичної ІС.

Spring Security Docs [43] надає інструментарій для впровадження OAuth 2.0, OpenID Connect та RBAC у сучасних веб-застосунках. У контексті медичної системи це дозволяє гнучко реалізувати авторизацію різних ролей користувачів та захистити інтерфейси взаємодії.

Eric Ayintareba Akolgo [44] описує концепцію "злиття технологій" у сучасних медичних ІС. Автор підкреслює важливість інтеграції IoT, AI, Big Data та хмарних обчислень для досягнення безперервного моніторингу. Запропонована архітектура може бути основою для подальшого розвитку гібридних медичних систем.

Hassan Aziz [45] у статті про телемедицину акцентує увагу на розширенні доступу до медичних послуг через віддалені консультації. Автор підкреслює, що телемедицина, зокрема у поєднанні з ІС моніторингу, забезпечує оперативність реагування та зниження навантаження на лікарні.

Len Bass, Rick Kazman [46] у праці "Software Architecture in Practice" акцентують увагу на концепціях архітектурних стилів та шаблонів. Це дозволяє формалізувати структуру медичної ІС як багаторівневу систему з чітко розмежованими модулями відповідальності.

Martin Fowler [47] у роботі про архітектурні шаблони підприємств описує патерни побудови бізнес-логіки, взаємодії з базами даних та клієнтами. Це джерело є цінним для визначення структури прикладного рівня медичної системи.

IBM Watson Health [48] описує сучасні технологічні тренди у впровадженні AI в медичну практику. Представлені кейси стосуються обробки зображень, підтримки діагностичних рішень та інтеграції з клінічними даними.

S. M. Riazul Islam та Daehan Kwak [49] у своєму огляді детально аналізують застосування IoT у сфері охорони здоров’я. Вказано на переваги безперервного моніторингу, але також окреслено проблеми енергоспоживання, безпеки та масштабованості IoT-інфраструктури.

M. Jørgensen [50] у своєму метааналізі оцінки зусиль на розробку ПЗ досліджує похибки у прогнозуванні часу та вартості. Це має безпосереднє застосування у проєктному менеджменті медичних ІС, де планування ресурсів є критично важливим.

Таким чином, проаналізовані джерела свідчать про широку палітру підходів до створення та впровадження інформаційних систем у сфері охорони

здоров’я, з особливим акцентом на використання штучного інтелекту, управління ІТ-проєктами, захисту персональних даних і технічної реалізації. Попри численні успіхи в цій галузі, низка питань залишається нерозв’язаною або потребує подальшого вдосконалення. Зокрема, це стосується адаптації методологій управління проєктами до специфіки персоналізованих медичних рішень, інтеграції ІТ-інструментів у медичну практику з урахуванням етичних, правових і організаційних аспектів, а також створення ефективних команд розробки для таких проєктів.

Саме ці аспекти становлять основу даного дослідження. У межах кваліфікаційної роботи буде розглянуто процеси управління проєктом створення персоналізованої інформаційної системи для кардіомоніторингу, що дозволить не лише узагальнити наявний досвід, а й запропонувати власну концепцію реалізації такої системи з урахуванням сучасних технологій, вимог ринку та потреб охорони здоров’я

1.1.2 Загальна характеристика систем

Інформаційні системи для моніторингу здоров'я серця можуть бути різноманітними за своєю природою, функціональністю та технічними особливостями. Нижче наведено загальну характеристику таких систем:

- Стаціонарні системи: ці системи використовуються у медичних установах та лікарнях для постійного моніторингу стану серця пацієнтів. Вони зазвичай мають широкий функціонал та можуть обробляти великі обсяги даних.
- Портативні системи: такі системи можуть бути носимими пристроями або мобільними додатками для смартфонів та планшетів. Вони призначені для персонального моніторингу стану серця пацієнтів в режимі реального часу.
- Імплантовані системи: ці системи вбудовуються безпосередньо в організм пацієнта та здатні постійно відстежувати показники серцевої діяльності. Вони часто використовуються для пацієнтів зі складними серцевими захворюваннями.

Основні функціональні можливості систем:

- Моніторинг ЕКГ для виявлення ритмічних або структурних аномалій серця [15].
- Аналіз варіабельності серцевого ритму для оцінки автономної нервової системи та стресових реакцій.
- Виявлення аритмій, таких як фібриляція передсердь, та інших серцевих патологій.
- Моніторинг показників фізичної активності та їх вплив на серцеву діяльність.

Технологічні компоненти систем:

Датчики, які збирають дані про серцеву активність [9], включаючи ЕКГ, та передають їх до центральної системи.

Програмне забезпечення для аналізу та інтерпретації даних, що дозволяє виявляти аномалії та генерувати звіти для медичних фахівців та пацієнтів.

Комунікаційні модулі для передачі даних у реальному часі або в режимі віддаленого моніторингу до медичних центрів чи мобільних пристроїв.

Особливості роботи:

- Системи можуть працювати в режимі реального часу [28], надаючи миттєві дані про стан серця пацієнта.
- Деякі системи також забезпечують можливість віддаленого моніторингу, що дозволяє медичним фахівцям відстежувати стан пацієнтів на відстані та надавати необхідну допомогу.

Системи для моніторингу здоров'я серця повинні бути здатні інтегруватися з іншими медичними інформаційними системами, такими як Інформаційні системи електронної медичної картки (ЕМК), щоб забезпечити обмін даними між різними клінічними установами та забезпечити цілісну картину стану здоров'я пацієнта. Для цього системи повинні відповідати стандартам обміну медичною інформацією, таким як HL7 або FHIR, та мати можливість інтеграції з IT-інфраструктурою медичних закладів.

Крім того, системи моніторингу здоров'я серця повинні бути сумісними з різними типами медичного обладнання та мобільних пристроїв для забезпечення широкого спектру можливостей для пацієнтів та медичних фахівців. Наприклад, здатність працювати з різними моделями дефібриляторів, кардіостимуляторів або моніторів серцевої діяльності.

1.1.3 Порівняння функціональних можливостей існуючих систем

Зробимо порівняння існуючих систем розглянутих виробників (табл. 1.1):

Таблиця 1.1

Порівняння основних функцій систем

Виробники	Моніторинг ЕКГ	Аналіз варіабельності серцевого ритму	Оповіщення про аритмії	Віддалений моніторинг	Інтеграція з мобільними додатками
Philips IntelliVue	Так	Так	Так	Так	Так
GE Healthcare MUSE	Так	Так	Так	Так	Так
Medtronic CareLink	Так	Так	Так	Так	Так
Abbott Confirm Rx	Так	Так	Так	Так	Так
Boston Scientific LATITUDE	Так	Так	Так	Так	Так

Також наведемо оцінку точності та надійності (табл. 1.2), та порівняння зручності використання (табл. 1.3):

Таблиця 1.2

Оцінка точності та надійності

Виробники	Точність моніторингу	Надійність роботи	Кількість хибних тривог (%)
Philips IntelliVue	Висока	Висока	2%
GE Healthcare MUSE	Висока	Висока	2.5%
Medtronic CareLink	Висока	Висока	2%
Abbott Confirm Rx	Висока	Висока	1.5%
Boston Scientific LATITUDE	Висока	Висока	1.8%

Таблиця 1.3

Порівняння зручності використання

Виробники	Зручність для пацієнтів	Зручність для медичного персоналу
Philips IntelliVue	Висока	Висока
GE Healthcare MUSE	Висока	Висока
Medtronic CareLink	Висока	Висока
Abbott Confirm Rx	Висока	Висока
Boston Scientific LATITUDE	Висока	Висока

Зробимо аналіз додаткових функцій (табл. 1.4)

Таблиця 1.4

Аналіз додаткових функцій

Виробники	Підтримка телемедицини	Використання ШІ для аналізу даних
Philips IntelliVue	Так	Так
GE Healthcare MUSE	Так	Так
Medtronic CareLink	Так	Так
Abbott Confirm Rx	Так	Так
Boston Scientific LATITUDE	Так	Так

Розглянемо також відгуки користувачів та результати клінічних випробувань цих систем:

- Philips IntelliVue: користувачі відзначають високу точність та зручність використання системи. Клінічні випробування показали ефективність у ранньому виявленні серцевих патологій [20].
- GE Healthcare MUSE: система отримала позитивні відгуки за надійність та можливості аналітики. Клінічні випробування підтвердили високу точність моніторингу.
- Medtronic CareLink: відгуки користувачів свідчать про зручність інтеграції з іншими медичними системами та високу точність. Клінічні випробування продемонстрували ефективність системи у довгостроковому моніторингу.
- Abbott Confirm Rx: користувачі високо оцінюють компактність та зручність використання імплантованих пристроїв. Результати клінічних випробувань показали високу точність та надійність.
- Boston Scientific LATITUDE: система отримала позитивні відгуки за інтеграцію з мобільними додатками та підтримку телемедицини. Клінічні випробування підтвердили ефективність системи в умовах віддаленого моніторингу.

Проведений аналіз ринку інформаційних систем для моніторингу здоров'я серця показав, що сучасні системи, такі як Philips IntelliVue, GE Healthcare MUSE, Medtronic CareLink, Abbott Confirm Rx та Boston Scientific LATITUDE, забезпечують високий рівень функціональності, точності та надійності. Усі ці системи мають свої унікальні переваги, включаючи потужні можливості моніторингу та аналізу, інтеграцію з мобільними додатками, підтримку телемедицини та використання штучного інтелекту для обробки даних.

Зважаючи на те, що всі існуючі системи демонструють високу ефективність та надійність, при розробці нової персоналізованої інформаційної системи для моніторингу здоров'я серця слід взяти до уваги всі функціональні можливості та технологічні досягнення, які вже присутні в наявних рішеннях.

Нова система повинна об'єднати в собі всі ці переваги, додавши нові інноваційні функції, які ще більше підвищать її ефективність та зручність використання.

Основні напрями розвитку нової системи можуть включати:

- Розширені можливості аналізу даних з використанням штучного інтелекту та машинного навчання для більш точної діагностики та прогнозування серцевих захворювань.
- Інтеграція з новими мобільними платформами та носимими пристроями, що дозволить забезпечити ще більш зручний та доступний моніторинг здоров'я.
- Поліпшення інтерфейсів користувача для ще більшої зручності як для пацієнтів, так і для медичного персоналу.
- Додаткові функції телемедицини, що дозволять ще швидше реагувати на зміни в стані здоров'я пацієнтів та забезпечувати більш ефективне лікування на відстані.

1.2 Формулювання проблемної області

На сучасному ринку медичних інформаційних систем для моніторингу здоров'я серця існує значна кількість різноманітних рішень. Однак, більшість існуючих наукових робіт та досліджень зосереджені на технічних аспектах розробки цих систем, таких як програмування, дизайн інтерфейсу користувача, та забезпечення безпеки даних. Бракує досліджень, які детально описують процес впровадження персоналізованих інформаційних систем для моніторингу здоров'я серця з точки зору управління проєктами [25] та адаптації під індивідуальні потреби пацієнтів [30].

Впровадження персоналізованих систем моніторингу здоров'я серця є критично важливим аспектом для забезпечення ефективного та якісного медичного обслуговування. Це включає не лише технічну інтеграцію, але й врахування індивідуальних особливостей кожного пацієнта, налаштування порогів оповіщення, забезпечення безперервного моніторингу в режимі

реального часу та надання персоналізованих рекомендацій [3, 4]. Недостатня увага до цих аспектів може призвести до зниження ефективності системи, пропущених критичних показників здоров'я, та в результаті – до погіршення стану здоров'я пацієнтів.

Ця робота сприятиме розвитку наукових знань у галузі впровадження персоналізованих інформаційних систем для моніторингу здоров'я серця. Запропонований підхід може бути використаний як основа для майбутніх досліджень та проєктів [19], а також допоможе медичним установам ефективніше управляти своїми ресурсами та часом при розробці і впровадженні подібних систем. Це, у свою чергу, забезпечить вищий рівень медичного обслуговування та покращить якість життя пацієнтів.

1.2.1 SWOT аналіз

У цій частині ми розглянемо SWOT-аналіз системи відслідковування здоров’я серця. SWOT-аналіз-важливий інструмент стратегічного планування [1], який дозволяє нам оцінити сильні і слабкі сторони нашої платформи, визначити потенційні можливості для розвитку і загрози, з якими ми можемо зіткнутися. Аналіз дозволяє глибше зрозуміти зовнішнє та внутрішнє середовище, в якому працює платформа, і допомагає підготувати ефективні стратегії для досягнення довгострокових цілей.

Такий підхід є особливо актуальним для високотехнологічних IT-проєктів у сфері охорони здоров’я, де рішення повинні враховувати як технічні обмеження, так і регуляторні вимоги, поведінку користувачів і ринкову конкуренцію.

SWOT-аналіз (табл. 1.5), що, детально представляє кожен з цих аспектів, дозволяючи нам системно підходити до планування стратегій та прийняття обґрунтованих рішень.

Таблиця 1.5

SWOT-аналіз персоналізованої інформаційної системи відслідковування здоров'я серця

Сильні сторони	Слабкі сторони
• Використання новітніх технологій, таких як штучний інтелект, телемедицина, та Інтернет речей, робить систему більш ефективною та привабливою для користувачів.. • Система може адаптуватися до індивідуальних потреб кожного пацієнта, що підвищує точність діагностики та ефективність лікування. • Можливість постійного спостереження за станом здоров'я пацієнтів без необхідності їх постійної присутності в медичних установах. • Автоматичне виявлення та оповіщення про серцеві аномалії дозволяє швидко реагувати на критичні зміни у стані здоров'я пацієнта. • Наявність платформ для пацієнтів та медичних працівників, що забезпечує зручний обмін інформацією та зворотний зв'язок.	• Вартість обладнання та програмного забезпечення, а також необхідність навчання персоналу можуть бути значними. • Інтеграція нової системи з вже існуючими медичними інформаційними системами може викликати труднощі. • Підтримка високого рівня безпеки даних потребує додаткових ресурсів та може бути складною в реалізації. • Неполадки в роботі обладнання або програмного забезпечення можуть призвести до втрати даних або зниження якості моніторингу.
Можливості	**Загрози**
• Збільшення попиту на дистанційні медичні послуги та зростання числа користувачів персоналізованих систем моніторингу. • Можливість укладання контрактів з лікарнями та клініками для впровадження системи на їх базі [14]. • Інтеграція з іншими телемедичними сервісами для надання комплексних медичних послуг. • Система сприяє ранньому виявленню серцевих захворювань та своєчасному наданню медичної допомоги.	• Висока конкуренція на ринку медичних інформаційних систем може ускладнити вихід на ринок та утримання позицій. • Введення нових регулюючих норм щодо зберігання та обробки медичних даних може збільшити витрати на впровадження та експлуатацію системи. • Швидкий розвиток технологій може зробити існуючі рішення застарілими, вимагаючи постійних інвестицій в оновлення та модернізацію. • Економічна нестабільність та зниження платоспроможності населення можуть вплинути на попит на систему. • Недовіра пацієнтів до нових технологій та занепокоєння з приводу конфіденційності особистих даних можуть перешкоджати впровадженню системи [11, 21].

Проаналізувавши сильні та слабкі сторони, можливості та загрози можемо сказати, що для успішного впровадження системи потрібно забезпечити достатнє фінансування на всіх етапах проєкту, включаючи закупівлю обладнання, розробку та інтеграцію програмного забезпечення, навчання персоналу та забезпечення конфіденційності даних.

Співпраця з медичними установами та розвиток телемедицини можуть сприяти швидкому впровадженню системи та її популяризації серед пацієнтів та медичних працівників.

1.3 Аналіз стандартів і нормативних вимог (GDPR, HIPAA)

Для інформаційних систем, що працюють з медичними даними, одним із найбільш важливих аспектів є відповідність міжнародним нормативним вимогам, які регулюють конфіденційність і захист персональних та медичних даних. У рамках цієї магістерської роботи було обрано два ключових міжнародних стандарти: Загальний регламент щодо захисту даних (GDPR, General Data Protection Regulation, Регламент ЄС 2016/679 від 27 квітня 2016 року) та Закон про мобільність і підзвітність у сфері медичного страхування США (HIPAA, Health Insurance Portability and Accountability Act of 1996, Public Law 104-191, 110 Stat. 1936).

Загальний регламент щодо захисту даних (GDPR).

GDPR є обов'язковим нормативним актом, що набув чинності 25 травня 2018 року, і визначає правила збирання, обробки, зберігання та передачі персональних даних фізичних осіб на території Європейського Союзу. Він застосовується не лише до організацій, які перебувають у ЄС, але й до будь-яких інших, що працюють з даними громадян ЄС. GDPR був обраний для аналізу у рамках цієї роботи через високий рівень вимог до захисту персональних даних та чітко визначені технічні стандарти, що робить його ідеальним для забезпечення безпеки медичних даних.

Зокрема, в контексті розробки персоналізованої інформаційної системи моніторингу серцевої діяльності, найбільш важливими є наступні положення GDPR [18]:

- Стаття 5 («Принципи обробки персональних даних»): встановлює фундаментальні принципи обробки даних, які включають прозорість, мінімізацію даних, цільове обмеження, точність, конфіденційність та цілісність. Це означає, що наша система повинна обмежити збір інформації лише необхідними медичними та особистими даними пацієнтів, які безпосередньо стосуються мети проєкту – моніторингу та аналізу стану серця.
- Статті 15-22 («Права суб'єктів даних»): визначають права користувачів, серед яких право на доступ до своїх даних, право на їх виправлення, право на обмеження обробки та право на забуття (повне видалення). Для реалізації цих положень у системі буде створено окремий функціонал, що дозволить користувачам керувати своїми даними у зручний та прозорий спосіб.
- Стаття 32 («Безпека обробки даних»): передбачає впровадження сучасних методів захисту даних, таких як псевдонімізація, шифрування, регулярний аудит безпеки та оцінка ризиків. Система, що розробляється, передбачатиме багаторівневе шифрування даних, що забезпечить високий рівень безпеки медичної інформації.

Закон про мобільність і підзвітність у сфері медичного страхування США (HIPAA), був ухвалений у 1996 році й на сьогоднішній день є одним із найвідоміших і найважливіших нормативних документів у галузі захисту медичних даних у світі. HIPAA встановлює обов'язкові вимоги до організацій [10], що збирають, зберігають та обробляють медичну інформацію громадян США.

Найбільш релевантними для проєкту є наступні положення HIPAA:

- Правило конфіденційності (Privacy Rule, 45 CFR Part 160 та Subparts A і E Part 164): забезпечує захист конфіденційності персональної медичної інформації та встановлює умови, за яких можливий доступ до цих даних. Наша

система передбачає суворий контроль доступу з використанням сучасних аутентифікаційних механізмів, що відповідають вимогам цього правила.

- Правило безпеки (Security Rule, 45 CFR Part 160 та Subparts A і C Part 164): визначає стандарти для захисту електронних медичних даних, включаючи вимоги щодо шифрування [17], контролю доступу, аудиту безпеки та резервного копіювання. Ці вимоги реалізовуються через впровадження сучасних IT-рішень у нашу інформаційну систему.
- Правило повідомлення про порушення (Breach Notification Rule, 45 CFR §§ 164.400-414): зобов’язує організації інформувати користувачів у разі витоку або порушення безпеки медичних даних. Це правило потребує впровадження автоматизованих систем моніторингу безпеки та механізмів швидкого реагування на інциденти, що буде забезпечено в розробленій системі.

Обрання GDPR та HIPAA як основних нормативних стандартів для цієї роботи обумовлено необхідністю відповідності міжнародним стандартам безпеки та конфіденційності, а також перспективою виходу на міжнародний ринок, що підвищує потенціал комерціалізації системи. Дотримання цих стандартів дозволяє гарантувати безпечність, прозорість і надійність роботи інформаційної системи, а також уникнути серйозних юридичних і репутаційних ризиків, пов'язаних із недотриманням вимог до захисту медичних та персональних даних.

1.4 Постановка задачі дослідження, формулювання технічного завдання на розробку

Основною метою даного дослідження є швидке впровадження персоналізованої інформаційної системи відслідковування здоров’я серця з мінімальними затратами, створення застосунку для моніторингу здоров’я клієнтів та швидке реагування на інциденти.

Цілі та задачі дослідження наведено у табл. 1.6.

Таблиця 1.6

Цілі та задачі дослідження

Цілі проєкту	Задачі
• Впровадження системи для моніторингу здоров'я серця. • Підвищення якості та ефективності медичного обслуговування. • Зменшення витрат на медичне обслуговування.	• Аналіз існуючих рішень на ринку. • Деталізація функціональних та нефункціональних вимог до застосунку [2]. • Створення архітектури застосунку, визначення технологій та інструментів для розробки. • Зменшення необхідності фізичних візитів до лікаря шляхом впровадження дистанційного моніторингу та консультацій. Підвищення ефективності роботи медичного персоналу за рахунок автоматизації процесів моніторингу та аналізу даних.

Реалізація цього проєкту дозволить створити інструмент для моніторингу здоров’я клієнтів, можливість проводити онлайн консультації з лікарями та забезпечить швидке реагування на виклики з датчиків.

Функціональні вимоги:

- Система повинна забезпечувати безперервний моніторинг серцевого ритму.
- Виявлення та оповіщення про аномалії в режимі реального часу [13].
- Можливість створення індивідуальних профілів для кожного пацієнта.
- Налаштування порогів тривог для різних категорій пацієнтів.
- Підтримка віддаленого моніторингу з доступом до даних через мобільні додатки.
- Можливість надання консультацій пацієнтам в режимі онлайн.
- Інтеграція з електронними медичними записами (EMR) [28] та іншими медичними інформаційними системами.
- Підтримка стандартів обміну медичними даними, таких як HL7, FHIR [34].

- Можливість надання освітніх матеріалів для пацієнтів.
- Платформа для зворотного зв'язку між пацієнтами та медичними працівниками

Цей проєкт враховує як функціональні, так і нефункціональні вимоги, щоб забезпечити високу якість обслуговування та задоволення потреб користувачів.

Нефункціональні вимоги:

- Регулярне проведення аудитів безпеки.
- Система повинна мати високу доступність та надійність.
- Резервне копіювання даних та план відновлення після збоїв.
- Система повинна підтримувати збільшення кількості користувачів та оброблюваних даних без зниження продуктивності.
- Інтуїтивно зрозумілий та зручний для користувачів інтерфейс. Підтримка багатомовності та налаштувань для користувачів з обмеженими можливостями.
- Швидке реагування системи на запити користувачів.

1.5 Паспорт проєкту

Для реалізації успішного проєкту важливо чітко визначити його цілі та задачі, що дозволить структуровано підійти до кожного етапу роботи та забезпечити досягнення запланованих результатів. Розглянемо конкретні вимірювані цілі та задачі, сформульовані відповідно до методології SMART, що допоможе зробити проєкт максимально ефективним та результативним.

Для виконання проєкту було сформовано конкретні вимірюванні цілі та задачі за SMART. Цілі проєкту та задачі, які необхідно вирішити в межах проєкту для досягнення поставлених цілей наведені у табл. 1.7

Таблиця 1.7

Цілі та задачі проєкту

Ціль	Конкретна	Вимірювана	Досяжна	Актуальна	Часові обмеження
Досягти 10,000 користувачів за рік	Розробити та впровадити систему для моніторингу здоров'я серця	Кількість користувачів (10,000)	Розраховано на масове використання [20]	Важливо для збільшення впізнаваності	Рік
Забезпечити прибуток в 1,000,000 грн	Отримати прибуток від підписок та продажу пристроїв	Загальний дохід (1,000,000 грн)	Можливо через оптимізацію витрат [14] і ефективний маркетинг	Відповідає бізнес-цілям	Рік
Забезпечити конфіденційність та безпеку даних	Впровадити сучасні методи шифрування та захисту даних	Відсутність витоків даних	Можливо через використання сучасних технологій	Важливо для довіри користувачів	Постійно

Головними завданнями проєкту є:

- Розробка технічного завдання та вимог до системи.
- Закупівля необхідного обладнання.
- Розробка програмного забезпечення.
- Забезпечення безпеки та конфіденційності даних.
- Маркетингова кампанія для залучення користувачів.

Цільова група: пацієнти з серцево-судинними захворюваннями [15], люди, що потребують регулярного моніторингу здоров'я серця, медичні працівники, які надають послуги з моніторингу та діагностики серцевих захворювань.

Потреби: оперативне виявлення та сповіщення про аномалії, регулярний та точний моніторинг серцевого стану.

Продукт:

- Мобільний застосунок.
- Носимий пристрій для моніторингу серцевої діяльності [31].
- Портал для взаємодії з медичним персоналом

- База даних для зберігання та аналізу медичної інформації.

Опис цінності:

- Поліпшення якості життя пацієнтів завдяки оперативному моніторингу здоров'я серця [22].
- Зменшення ризику серцевих ускладнень через своєчасне виявлення проблем.
- Підвищення ефективності роботи медичного персоналу завдяки інтегрованій системі моніторингу та аналізу даних.

Зацікавлені сторони:

- Медичний персонал (лікарі, медсестри, адміністратори медичних закладів).
- Пацієнти з серцево-судинними захворюваннями.
- Родичі та опікуни пацієнтів.
- Постачальники медичного обладнання.
- Розробники програмного забезпечення.
- Інвестори та партнери проєкту.

Була розрахована детальна оцінка проєкту. У таблиці 1.8 наведені витрати впровадження та підтримку персоналізованої інформаційної системи відслідковування здоров’я серця.

1. Визначення вихідних даних для розрахунку:

- Загальні інвестиції: $IC = 2\,008\,000$ грн
- Очікуваний річний дохід (вигода): $CF = 1\,500\,000$ грн
- Період розрахунку ефективності: 5 років
- Ставка дисконтування: $r = 12\%$ або $r = 0{,}12$

2. Розрахунок чистої приведеної вартості (NPV):

Чиста приведена вартість (NPV) обчислюється за формулою (1.1):

$$NPV = \sum_{t=1}^{n} \frac{CF_t}{(1+r)^t} - IC \quad (1.1)$$

де:

CF_t – грошовий потік за період t;

r – ставка дисконтування;

IC – початкові інвестиції;

n – кількість періодів (років).

3. Внутрішня ставка дохідності (IRR):

Внутрішня ставка дохідності (IRR) визначається ставкою дисконтування (1.2), за якої NPV = 0:

$$0 = \sum_{t=1}^{n} \frac{CF_t}{(1+IRR)^t} - IC \quad (1.2)$$

Підбір IRR є ітеративним. Раніше виконані розрахунки показали, що IRR ≈ 69%.

Це значно більше за ставку дисконтування (12%), отже проєкт дуже привабливий з фінансової точки зору.

4. Рентабельність інвестицій (ROI):

Рентабельність інвестицій обчислюється за формулою (1.3):

$$ROI = \frac{\text{Загальний прибуток за період} - IC}{IC} \times 100\% \quad (1.3)$$

5. Термін окупності проєкту (Payback period):

Термін окупності визначається за формулою (1.4):

$$\text{Payback period} = \frac{IC}{CF} \quad (1.4)$$

Підставляємо значення:

$$\text{Payback period} = \frac{2\,008\,000}{1\,500\,000} \approx 1{,}34 \text{ року (близько 16 місяців)}$$

Зробимо розрахунки IRR, ROI, NPV у Python програмі (рис. 1.1):

```python
import numpy as np

# Вхідні дані
investment = 2_008_000
annual_benefit = 1_500_000
years = 5
discount_rate = 0.12

# Грошові потоки: -investment у t=0, +annual_benefit у t=1..5
cashflows = [-investment] + [annual_benefit] * years

# 1) NPV
npv = sum(cf / ((1 + discount_rate) ** i) for i, cf in enumerate(cashflows))

# 2) IRR через корені полінома
coeffs = cashflows
roots = np.roots(coeffs)

# обираємо єдиний дійсний корінь > 0
irr_root = next(r.real for r in roots if abs(r.imag) < 1e-6 and r.real > 0)
irr = irr_root - 1

# 3) ROI
total_benefit = sum(cashflows[1:])
roi = (total_benefit - investment) / investment * 100

# 4) Термін окупності (лінеарно між роками)
cumulative = np.cumsum(cashflows)
payback_year = next(i for i, val in enumerate(cumulative) if val >= 0)
prev_cum = cumulative[payback_year - 1]
payback_frac = (investment - cumulative[payback_year - 1]) / annual_benefit
payback_period = payback_year - 1 + payback_frac

# Вивід результатів
print(f"NPV: {npv:,.0f} грн")
print(f"IRR: {irr * 100:.2f}%")
print(f"ROI: {roi:.2f}%")
print(f"Payback Period: {payback_period:.2f} years")
```

Рис. 1.1. Розрахунок NPV, IRR, ROI

Та оцінимо результати обчислень (рис.1.2)

```
X:\PyProjects\Education\venv\Scripts\python.exe X:\PyProjects\Education\Upr_Rishennia\IRR_ROI_NPV.py
NPV: 3,399,164 грн
IRR: 69.34%
ROI: 273.51%
Payback Period: 2.68 years

Process finished with exit code 0
```

Рис. 1.2. Результати обчислень

Оскільки $NPV > 0$, проєкт економічно ефективний.

У попередньому розрахунку терміну окупності проєкту використовувалась спрощена модель, що ґрунтується на діленні обсягу інвестицій на річний грошовий потік. Такий підхід дає швидку оцінку ефективності, проте не враховує реальну динаміку повернення інвестицій у часі. У даному розділі для підвищення

точності оцінки застосовано уточнений метод, заснований на поетапному накопиченні грошових потоків (кумулятивна сума).

Суть підходу полягає у тому, що спочатку обчислюється сумарний грошовий потік на кінець кожного року, і визначається момент, коли цей показник вперше перевищує суму інвестицій. Після цього з використанням лінійної інтерполяції обраховується частка року, необхідна для повного повернення інвестованих коштів. Такий підхід дозволяє отримати реальний період окупності у роках і місяцях, а не лише умовну середню величину.

Використання точного методу доцільне у випадках, коли необхідно планувати фінансові потоки, обґрунтовувати строки досягнення беззбитковості проєкту або проводити аналіз інвестиційної привабливості з урахуванням часових параметрів. З практичної точки зору, такий підхід забезпечує більшу достовірність і є більш релевантним для управлінських та інвестиційних рішень.

Проведене порівняння існуючих інформаційних систем моніторингу здоров’я серця (табл. 1.8) дозволяє виявити ключові функціональні та технічні характеристики, які є визначальними для їх ефективного використання в медичній практиці. На основі аналізу визначено сильні сторони, такі як наявність модулів для віддаленого спостереження, інтеграція з носимими пристроями та відповідність стандартам безпеки, а також виявлено обмеження, серед яких — недостатня гнучкість налаштувань, відсутність елементів персоналізації або обмежена аналітика даних.

Таблиця 1.8

Оцінка вартості проєкту

Показник економічної ефективності	Значення	Рекомендований критерій	Висновок
Загальні інвестиції	2 008 000 грн	Не перевищує планового бюджету	Відповідає бюджету

Продовження табл.1.8

Очікуваний річний дохід	1 500 000 грн	Достатній для прибутковості	Високий потенціал доходу
NPV (чиста приведена вартість)	3 399 164 грн	NPV > 0	Висока економічна ефективність
IRR (внутрішня ставка дохідності)	≈69%	IRR > 12%	Дуже привабливий показник
ROI (рентабельність інвестицій)	273,47%	Позитивне значення	Надзвичайно ефективний проєкт
Період окупності	1,34 року (≈16 міс.)	Якнайменший	Швидке повернення коштів

На основі розрахунків, загальні витрати на реалізацію проєкту становлять 2,008,000 гривень. Основні витрати підуть на закупівлю обладнання (датчики, портативні носії), розробку програмного забезпечення та зарплату персоналу.

Запланований бюджет охоплює всі основні аспекти проєкту, включаючи розробку, тестування, запуск та підтримку продукту.

Дерево цілей проєкту порталу на рис. 1.3. описує основні цілі та підцілі, які повинні бути досягнуті для успішної реалізації цього проєкту. Основна мета полягає в впровадженні персоналізованої інформаційної системи для моніторингу здоров'я серця, що забезпечить покращення якості медичного обслуговування, зниження смертності від серцево-судинних захворювань та підвищення економічної ефективності системи охорони здоров’я.

Основні цілі:

- Розробка та впровадження системи
- Забезпечення якості обслуговування та підтримки.
- Маркетинг та залучення користувачів.

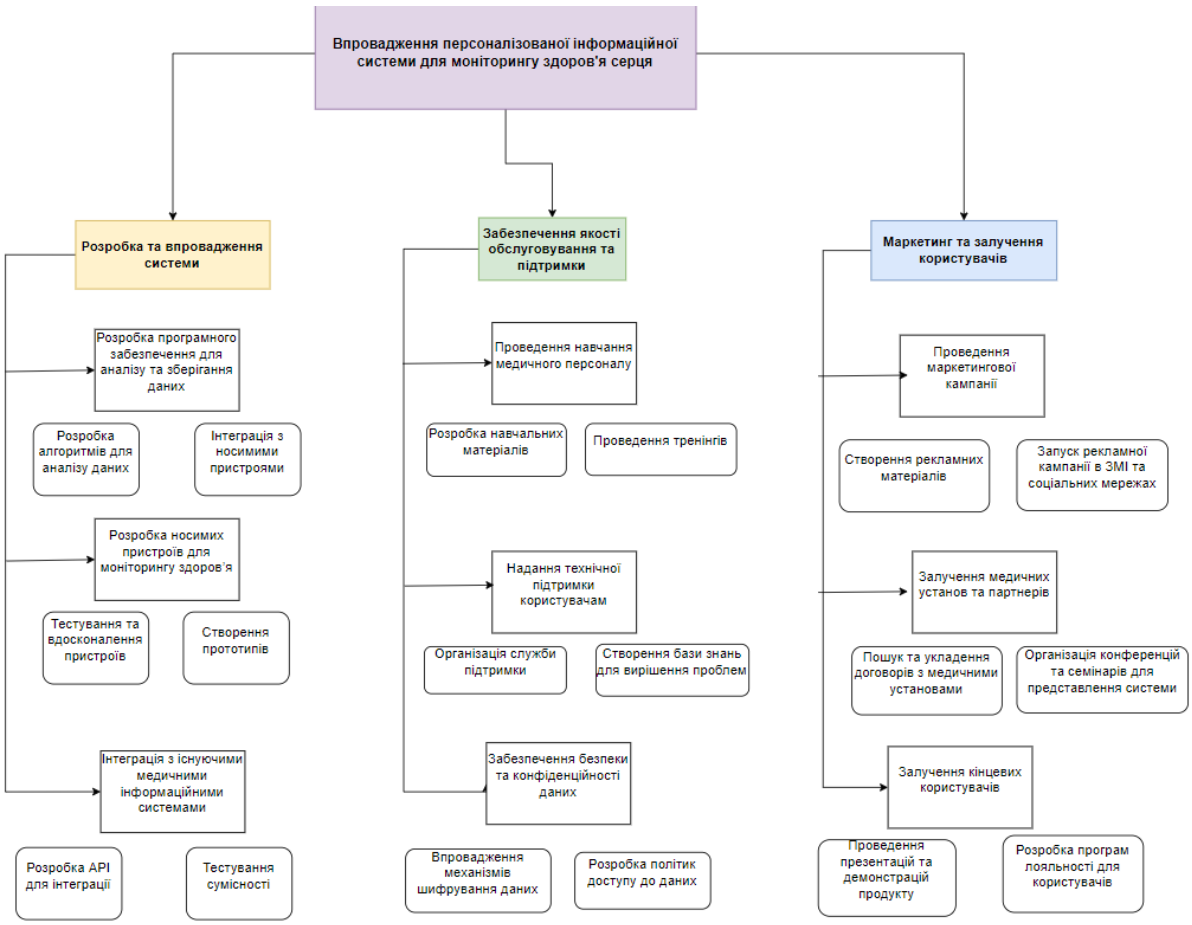


Рис. 1.3 Дерево цілей проєкту

На основі проведеного аналізу ринку інформаційних систем для моніторингу здоров'я серця, визначення проблемної області, SWOT-аналізу, а також вивчення нормативних вимог (GDPR, HIPAA) і сформованого дерева цілей продукту, були визначені наступні завдання дослідження:

1. Провести розробку концептуальної моделі інформаційної системи, яка інтегруватиме мобільні пристрої та сучасні технології моніторингу серцево-судинних показників.

2. Обґрунтувати вибір архітектури інформаційної системи, що відповідатиме вимогам масштабованості, стабільності та безпеки.

3. Розробити математичну модель для аналізу медичних даних з метою раннього прогнозування серцевих захворювань [6].

4. Визначити алгоритми роботи програмного забезпечення, орієнтовані на використання штучного інтелекту для автоматизованої обробки даних,

діагностики і формування рекомендацій щодо профілактики серцевих захворювань.

5. Сформувати інформаційне забезпечення проєкту шляхом розробки та побудови ефективної бази даних з урахуванням нормативних вимог GDPR та HIPAA щодо захисту персональних і медичних даних.

6. Реалізувати програмне забезпечення для взаємодії з мобільними пристроями та носимими смарт-гаджетами, з метою забезпечення зручності і простоти використання кінцевими користувачами.

7. Провести комплексне тестування розробленої інформаційної системи, оцінити точність та достовірність отриманих результатів і порівняти її ефективність з аналогічними системами, що існують на ринку.

8. Визначити можливі напрямки подальшого розвитку та перспективи масштабування створеної інформаційної системи для забезпечення її сталого функціонування і комерційного потенціалу в майбутньому.

Виконання вищезазначених завдань дозволить розробити ефективну інформаційну систему для моніторингу здоров’я серця, яка буде конкурентоспроможною на ринку та відповідатиме сучасним вимогам до функціональності й безпеки.

РОЗДІЛ 2. РОЗРОБКА КОНЦЕПТУАЛЬНОЇ МОДЕЛІ СИСТЕМИ.

2.1 Опис архітектури системи.

Інформаційна система для моніторингу здоров’я серця розробляється на основі клієнт-серверної архітектури, що дозволяє ефективно розподіляти функціональність між пристроєм користувача (клієнтською частиною) та серверною частиною, яка виконує обробку даних, зберігання, аналітику та керування.

Загалом, архітектура системи має багаторівневу структуру, яка включає наступні шари:

1. Шар збору даних (Data Collection Layer): Реалізується за допомогою носимих пристроїв (смарт-годинники, фітнес-браслети, ЕКГ-сенсори), які передають фізіологічні показники користувача через мобільний застосунок.

2. Клієнтський шар (Client Layer): Представлений мобільним застосунком, який виконує роль інтерфейсу користувача, дозволяючи переглядати медичні показники, отримувати повідомлення, та синхронізувати дані з сервером.

3. Серверний логічний шар (Application Layer): Включає бізнес-логіку системи, обробку запитів, валідацію, збереження даних та підготовку їх до аналітики. Основні компоненти реалізовані як RESTful API-сервіси, що взаємодіють із клієнтом через HTTPS.

4. База даних та сховище (Data Layer): Постійне зберігання даних реалізовано через реляційну базу даних (наприклад, PostgreSQL). Для великих масивів телеметричних даних використовується Time-Series DB (InfluxDB або аналогічна). Дані зберігаються у захищеному вигляді відповідно до вимог GDPR та HIPAA.

5. Аналітичний та ML-шар (AI/ML Layer — у перспективі): Передбачає інтеграцію модулів для інтелектуального аналізу медичних показників, що працюватимуть окремо як мікросервіси. Застосування цього шару можливе на наступному етапі розвитку системи.

Зобразимо загальний вигляд архітектури системи (рис. 2.1)

Рис. 2.1. Загальний вигляд архітектури системи

Опис взаємодії між компонентами системи

- Користувач через мобільний застосунок взаємодіє з пристроями збору даних (датчики) через Bluetooth або Wi-Fi.
- Зібрані дані передаються на сервер, де проходять перевірку, очищення та зберігання у базі даних.
- У разі перевищення порогових значень або виявлення аномалій, сервер генерує повідомлення, яке надсилається назад у мобільний застосунок користувача.
- За потреби, дані та рекомендації можуть надсилатися медичному фахівцю через окремий інтерфейс доступу.
- У перспективі, частина даних передаватиметься до AI/ML-модуля, який здійснюватиме автоматизоване прогнозування та формування рекомендацій. Взаємодія відбуватиметься через API Gateway, що сприятиме безпечному масштабуванню.

Функціональність основних модулів системи

1. **Мобільний застосунок (клієнт):**
   - Отримання даних із сенсорів.
   - Відображення показників у вигляді графіків, таблиць, статусів.
   - Надсилання даних на сервер.
   - Отримання повідомлень і рекомендацій.
   - Ідентифікація користувача, захист доступу.
2. **API-сервер (бізнес-логіка):**
   - Прийом, обробка та маршрутизація запитів від мобільного клієнта.
   - Валідація даних, перевірка прав доступу.
   - Робота із зовнішніми модулями (аналітика, AI/ML).
   - Обробка станів пацієнта та логіка інформування.
3. **База даних:**
   - Структуроване збереження особистої, медичної та статистичної інформації.
   - Історичне логування даних телеметрії.
   - Кешування критичних показників для швидкого доступу.
4. **Модуль аналітики / AI/ML (перспективний):**
   - Навчання моделей на основі накопичених даних.
   - Виявлення шаблонів та відхилень у показниках.
   - Побудова прогнозів стану здоров’я користувача.
   - Генерація персоналізованих рекомендацій.
5. **Інтерфейс медичного працівника (Web-клієнт):**
   - Доступ до даних пацієнтів.
   - Аналіз динаміки змін.
   - Відправлення індивідуальних рекомендацій.

Інформаційна система моніторингу здоров’я серця передбачає тісну взаємодію з внутрішніми та зовнішніми ресурсами, включно з базами даних та хмарними сервісами.

Для зберігання та обробки структурованих медичних та персональних даних використовується реляційна база даних (RDBMS), наприклад, PostgreSQL.

Усі запити до бази даних реалізовано через проміжний шар — сервер застосунків (API-сервер), який виконує такі завдання:

- Формування та виконання SQL-запитів (CRUD-операції) через безпечні параметризовані запити.
- Управління транзакціями для збереження цілісності та узгодженості даних.
- Забезпечення шифрування даних (наприклад, за стандартом AES-256) відповідно до вимог GDPR та HIPAA.
- Використання механізму ORM (Object-Relational Mapping) для спрощення взаємодії між об'єктною моделлю системи та таблицями бази даних.

Крім того, система використовує спеціалізовану базу даних для зберігання великих обсягів часових рядів (наприклад, InfluxDB). Вона забезпечує ефективний доступ до хронологічних даних, які отримуються з носимих пристроїв.

Взаємодія з хмарними та зовнішніми сервісами

Для підвищення надійності та ефективності роботи системи передбачено інтеграцію з хмарними сервісами, такими як Amazon Web Services (AWS) або Google Cloud Platform (GCP). Основні функції зовнішніх сервісів:

- Хмарне сховище (Cloud Storage): використовується для зберігання резервних копій даних і медіа-даних (наприклад, результати вимірювань у вигляді графіків та звітів).
- Хмарні обчислення (Cloud Compute Services): за потреби забезпечують масштабування обчислювальних потужностей, що особливо важливо для обробки та аналітики великих наборів даних.
- Сервіси автентифікації та авторизації: для безпечного доступу користувачів до системи (наприклад, OAuth 2.0).
- Сервіси сповіщень (Push Notifications): для своєчасного інформування користувачів про критичні зміни їх стану здоров'я.

Взаємодія із зовнішніми сервісами виконується через REST API, забезпечуючи високу гнучкість та незалежність компонентів системи.

Архітектура створеної інформаційної системи дозволяє ефективно масштабувати її можливості як горизонтально (збільшення кількості серверних вузлів), так і вертикально (збільшення ресурсів серверів). Система побудована на основі мікросервісної архітектури, що дозволяє інтегрувати додаткові компоненти без суттєвої зміни загальної архітектури.

Горизонтальне масштабування досягається завдяки:

- Використанню контейнеризації сервісів (Docker, Kubernetes).
- Використанню балансувальника навантаження (Load Balancer), що рівномірно розподіляє навантаження між вузлами.

Вертикальне масштабування можливе завдяки:

- Можливості підвищення потужності окремих серверних вузлів (CPU, RAM, дисковий простір).
- Гнучкому збільшенню ресурсів хмарних обчислювальних середовищ.

Підключення AI/ML-модулів передбачено на наступних етапах розвитку системи (детально описано в підрозділі 2.5). Для цього в архітектурі вже передбачені такі можливості:

- Можливість накопичення та обробки великих обсягів даних, які необхідні для навчання алгоритмів машинного навчання.
- Інтеграція AI/ML як окремих незалежних мікросервісів через API, що не вимагає змін основного програмного забезпечення.
- Використання спеціалізованих бібліотек для роботи з AI/ML (TensorFlow, Keras, PyTorch), які легко інтегруються з поточними технологіями проєкту.
- Передбачений механізм CI/CD для швидкої розробки, тестування та впровадження нових AI/ML-модулів.

Таким чином, створена архітектура є гнучкою, масштабованою та готовою до інтеграції AI/ML-компонентів у майбутньому, що гарантує подальший розвиток та підтримку високого рівня конкурентоспроможності розробленої інформаційної системи.

2.2 Розробка концептуальної моделі інформаційної системи.

У цій частині нашого курсового проєкту можете побачити концептуальну модель, зображену на рис. 2.2, яка є ключовим елементом дослідження. Концептуальна модель демонструє основні компоненти системи, їх взаємозв'язки та процеси, що відбуваються в рамках системи. Модель допомагає визначити структурні елементи платформи та зрозуміти, як вони взаємодіють для досягнення бізнес-цілей.

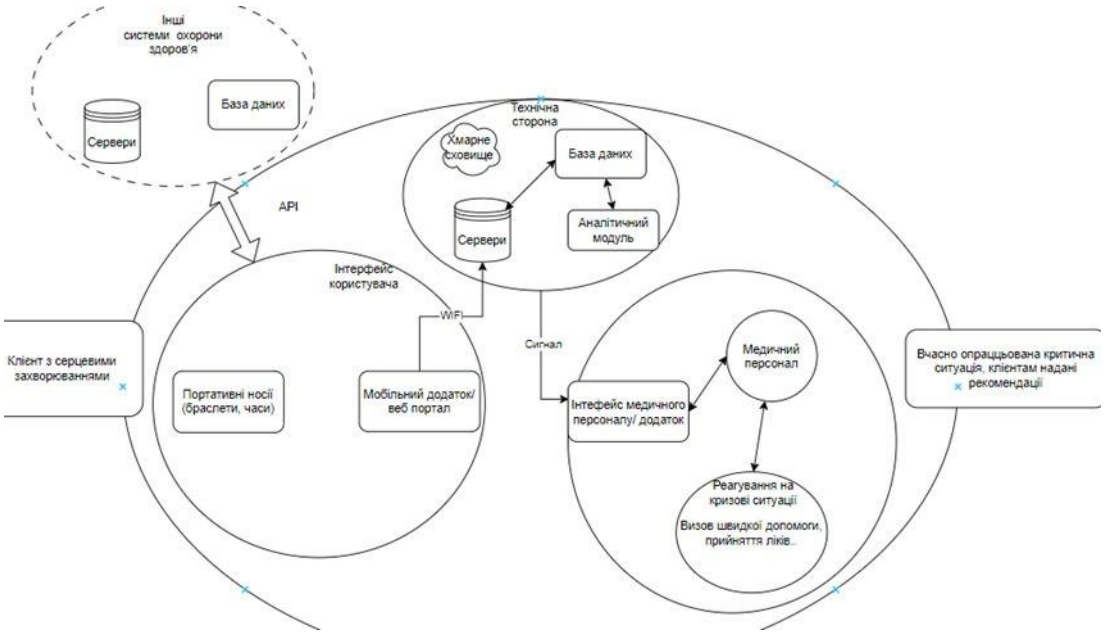


Рис. 2.2. Концептуальна модель інформаційної системи

Розглядаючи цю модель, можна зрозуміти не тільки те, як платформа обробляє поточні операції, але й як можливо оптимізувати та вдосконалити різні аспекти діяльності. Рисунок 2.2 є фундаментом для аналізу ефективності різних компонентів системи та розробки стратегічних рішень, спрямованих на підвищення продуктивності та задоволення потреб наших клієнтів.

Приклад взаємодії компонентів системи моніторингу роботи серця (рис. 2.3):

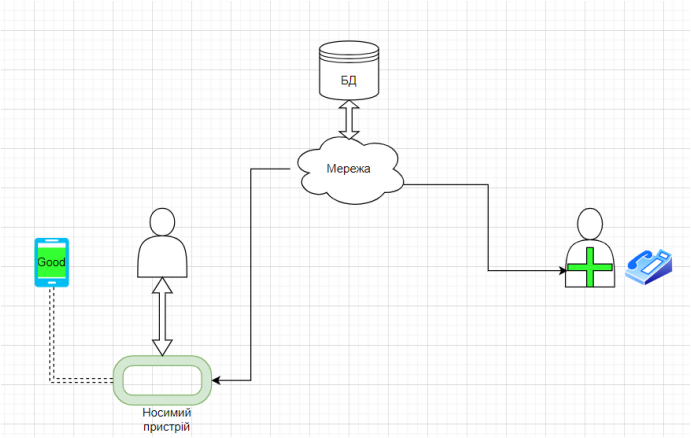


Рис. 2.3. Моніторинг роботи серця користувача за допомогою носимих пристроїв.

При погіршенні показників, оповіщення прийде на телефон клієнта, та сповістить медичного працівника про ситуацію (рис. 2.4).

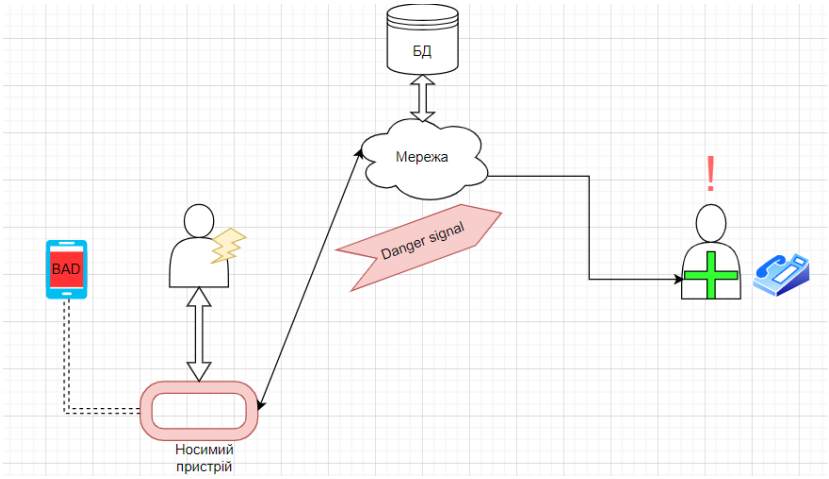


Рис. 2.4. Сповіщення медичного персоналу

Медичний працівник реагує на ситуацію, запис в базу даних. (рис. 2.5)

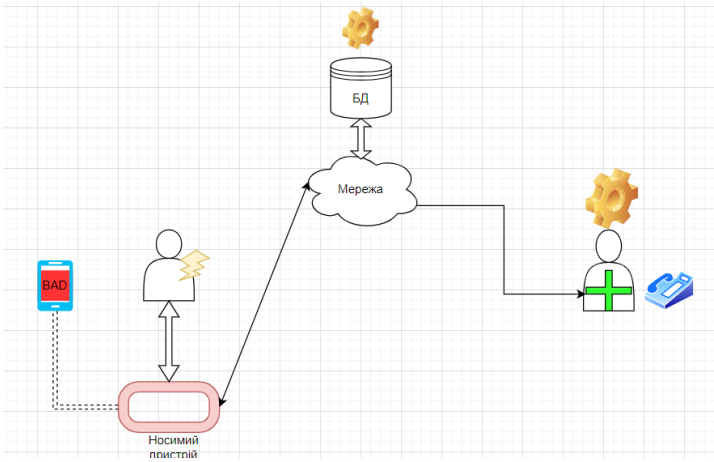


Рис. 2.5. Реакція медичного персоналу на критичний стан

Представлена архітектура системи ілюструє взаємодію ключових компонентів, включаючи базу даних, інтерфейс користувача, модулі обробки даних та взаємодії з мобільними пристроями. Такий структурний підхід дозволяє забезпечити масштабованість, ефективну обробку інформації про стан серця користувача, а також гнучке підключення до зовнішніх пристроїв. У результаті створено платформу, що є технічно придатною для впровадження в медичну практику та подальшої адаптації під специфічні вимоги користувачів і клінік.

2.3 Формалізація математичної моделі та постановка задачі в математичному вигляді

Для швидкого впровадження та максимізації доходу від персоналізованої інформаційної системи для моніторингу здоров'я серця, ми розглянемо оптимальний підхід до закупівлі обладнання, реклами та розробки підпискових планів.

Для підрахування початкових витрат використаємо формулу (2.1):

$$C_{initial} = C_{\text{equipment}} + C_{\text{software}} + C_{\text{infrastructure}} + C_{\text{training}} + C_{\text{marketing}} \quad (2.1)$$

де $C_{initial}$ – початкова вартість;

$C_{\text{equipment}}$ - вартість датчиків, серверів, робочих станцій тощо.

C_{software} - включає витрати на розробку та тестування ПЗ.

$C_{\text{infrastructure}}$ - вартість мережевого обладнання, хмарних сервісів, баз даних.

C_{training} - витрати на навчання медичних працівників та адміністраторів.

$C_{\text{marketing}}$ - витрати на рекламну кампанію для залучення нових користувачів.

Для підрахування операційних щомісячниї витрат використаємо формулу (2.2):

$$C_{operational} = C_{\text{maintenance}} + C_{\text{support}} + C_{\text{salaries}} \quad (2.2)$$

де $C_{operational}$ – щомісячні витрати;

$C_{\text{maintenance}}$ - регулярні технічні обслуговування..

C_{support} - підтримка користувачів та технічна підтримка.

C_{salaries} - зарплати персоналу, що займається обслуговуванням системи.

Для прибутку з підписок використаємо формулу (2.3):

$$\text{Revenue} = N_{base} * P_{base} + N_{premium} * P_{premium} \quad (2.3)$$

де Revenue – прибуток з підписок щомісяці;

N_{base} - кількість базових підписок.

P_{base} - вартість базової підписки.

$\text{N}_{\text{premium}}$ - кількість преміум підписок.

$\text{P}_{\text{premium}}$ - вартість преміум підписки.

Для підрахунку абсолютного прибутку використаємо формулу (2.4):

$$\text{Profit} = \text{Revenue} * \text{n} - \left(\text{C}_{\text{operational}} * \text{n} + \text{C}_{\text{initial}}\right) \quad (2.4)$$

де Profit– загальний прибуток/витрати за період;

n – кількість місяців.

2.4 Розрахунок розробленої моделі

Для розрахунку за розробленою моделлю порахуємо приблизну вартість:

Підставимо вартість початкових елементів:

$C_{\text{equipment}}$ = 1,000,000.

C_{software} = 1,000,000.

$C_{\text{infrastructure}}$ = 200,000.

C_{training} = 200,000.

$C_{\text{marketing}}$ = 5000.

Використаємо формулу (2.1) для розрахунку початкової вартості:

$$\text{C}_{\text{initial}} = C_{\text{equipment}} + C_{\text{software}} + C_{\text{infrastructure}} + C_{\text{training}} + C_{\text{marketing}}$$

C_{initial} = 2,405,000.

Підставимо вартість місячних витрат:

$C_{\text{maintenance}}$ = 50000.

C_{support} = 50000.

C_{salaries} = 100000.

Використаємо формулу (2.2) для розрахунку операційних витрат:

$$C_{operational} = C_{\text{maintenance}} + C_{\text{support}} + C_{\text{salaries}}$$

$C_{operational}$ = 200,000

Нехай вартість базової підписки буде 400грн, преміум 1000 грн.

N_{base} = 600.

N_{premium} = 350.

Використаємо формулу (2.3) для розрахування місячного прибутку за рахунок підписок:

$$\text{Revenue} = N_{\text{base}} * P_{\text{base}} + N_{\text{premium}} * P_{\text{premium}}$$

$$\text{Revenue} = 600 * 400 + 350 * 1000 = 590{,}000$$

Порахуємо дохід за 12 місяців, дял цього використаємо формулу (2.4):

$$\text{Profit} = \text{Revenue} * n - (C_{\text{operational}} * n + C_{\text{initial}})$$

$$\text{Profit} = 590{,}000 * 12 - (200{,}000 * 12 + 2{,}405{,}000) = 2{,}275{,}000.$$

Для більш точного розрахування витрат та прибутку був розроблена програма мовою програмування python (див. Додаток А).

Результатом виконання програми є графіки місячного прибутку в залежності від кількості підписників (рис. 2.6) та загальний прибуток помісячно (рис. 2.7).

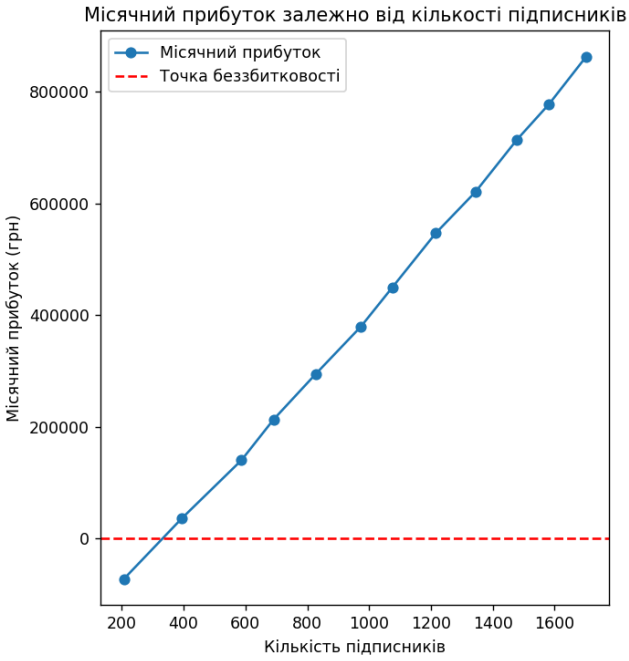


Рис. 2.6 Графік місячного прибутку залежно від кількості підписників

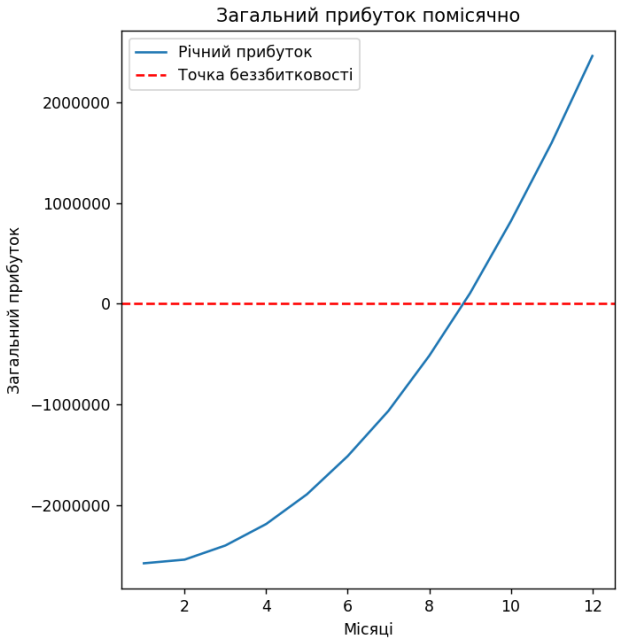


Рис. 2.7 Графік загального прибутку помісячно

З графіку можна зробити висновок, що перші місяці ми будемо відбивати початкові внески на дорогі технології, але з набуттям більшої кількості підписок щомісячний дохід буде рости. Набуття менш ніж двох тисяч користувачів за рік дозволить за 9 місяців почати виходи покрити всі затратні кошти.

При збільшенні витрат на рекламу в 10 разів на початковому етапі та фінансуванні реклами щомісячно кількість підписників очікувано збільшиться, що дозволить нам виконати ціль в 10,000 підписників за рік (рис. 2.8) :

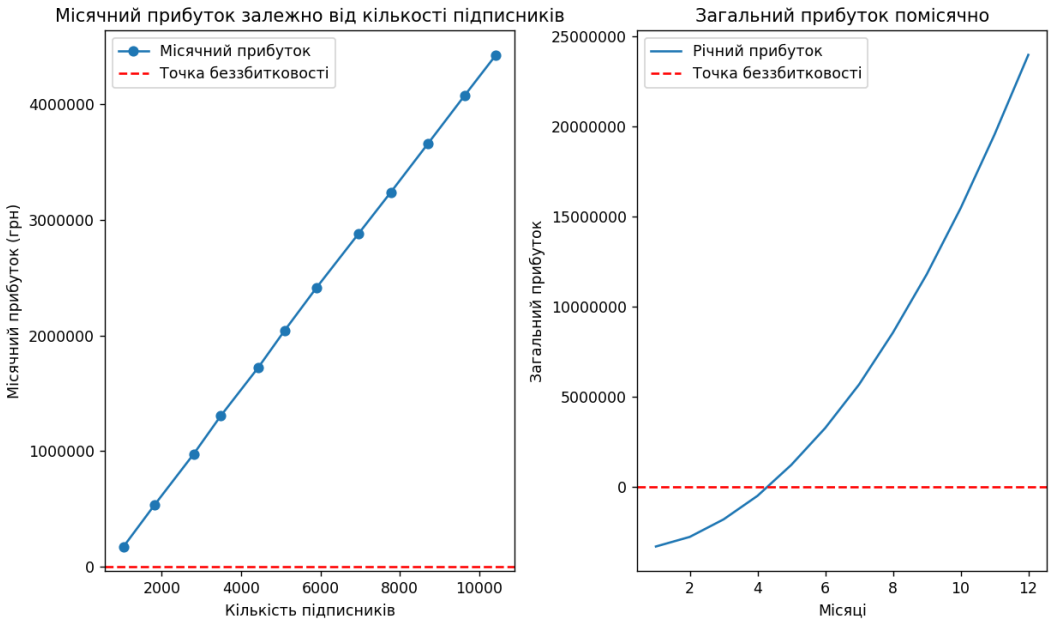


Рис. 2.8 Графіки при збільшенні фінансуванні реклами системи

Отже, збільшення витрат на рекламну кампанію на початкових етапах розвитку проєкту має суттєвий позитивний вплив на темпи залучення користувачів. Завдяки інтенсивній рекламній активності можна досягти запланованого рівня залучення аудиторії у стислі терміни, що сприятиме підвищенню впізнаваності продукту на ринку.
Такий підхід підтверджує важливість стратегічного планування маркетингової діяльності для успішної реалізації інформаційної системи моніторингу здоров’я серця.

2.5 Використання AI/ML у діагностиці серцевих захворювань

Сучасні інформаційні технології постійно розвиваються та відкривають нові можливості для підвищення якості медичних послуг, зокрема у сфері діагностики серцево-судинних захворювань. Значний потенціал мають такі передові технології, як штучний інтелект (Artificial Intelligence, AI) та машинне навчання (Machine Learning, ML), що дозволяють автоматизувати та вдосконалити процеси аналізу медичних даних, прогнозування станів здоров'я та ухвалення медичних рішень.

За даними Всесвітньої організації охорони здоров’я (ВООЗ), серцево-судинні захворювання є однією з головних причин смертності у світі. Це зумовлює нагальну необхідність пошуку нових методів ранньої діагностики, профілактики та ефективного лікування цих хвороб. Впровадження AI та ML в інформаційні системи моніторингу здоров’я має потенціал суттєво вплинути на ефективність діагностичних процедур і прогнозування ризиків для здоров'я пацієнтів.

Переваги впровадження AI/ML у діагностиці серцевих захворювань:

- Автоматизація процесу діагностики

Технології AI дозволяють проводити аналіз значних обсягів медичних даних, таких як показники електрокардіограми (ЕКГ), рівень артеріального тиску, варіабельність серцевого ритму, фізична активність тощо. Алгоритми ML здатні

виявляти приховані закономірності у даних, що недоступні для людського сприйняття, забезпечуючи високий рівень точності діагностики.

- Швидке та точне прогнозування ризиків

Застосування алгоритмів машинного навчання дозволяє проводити швидке і точне прогнозування можливих захворювань на основі аналізу накопичених медичних даних. Це забезпечує можливість вчасно розпочати профілактичні заходи, що значно знижує ризики розвитку серцевих захворювань та підвищує ефективність лікування.

- Персоналізована медицина

AI та ML дозволяють персоналізувати підхід до кожного пацієнта, враховуючи його індивідуальні особливості організму, історію хвороби, спосіб життя. Це надає можливість формувати індивідуальні рекомендації щодо харчування, фізичних навантажень, медикаментозної терапії, які максимально відповідають потребам конкретної людини.

- Зменшення навантаження на медичний персонал

Автоматизація первинної діагностики та аналізу даних за допомогою AI значно зменшує навантаження на лікарів та інших медичних працівників, дозволяючи їм зосередитись на складніших та критичних випадках.

- Підвищення економічної ефективності

Впровадження алгоритмів AI/ML у медичні інформаційні системи може суттєво зменшити витрати, пов’язані з лікуванням пацієнтів, завдяки ранньому виявленню потенційних ризиків та зменшенню кількості дороговартісних медичних процедур і госпіталізацій.

Попри те, що на поточному етапі розвитку проєкту ще немає технічних та фінансових ресурсів для повноцінної реалізації AI/ML компонентів, їхнє майбутнє впровадження є стратегічно важливим напрямком розвитку системи. Це забезпечить додаткову конкурентоспроможність продукту та його відповідність сучасним стандартам якості та інноваційності у сфері медичних технологій.

Застосування штучного інтелекту (AI) та машинного навчання (ML) у майбутній версії інформаційної системи для моніторингу здоров'я серця може бути реалізоване за кількома перспективними сценаріями:

1. Прогнозування ризику серцево-судинних захворювань

У цьому сценарії система регулярно збирає медичні дані користувача (ЕКГ, пульс, артеріальний тиск, фізична активність). Алгоритми машинного навчання, такі як Random Forest, Gradient Boosting, або нейронні мережі, аналізуватимуть ці дані, визначаючи ймовірність виникнення серцево-судинних захворювань.

2. Виявлення аномальних станів у реальному часі

Цей сценарій передбачає використання алгоритмів класифікації (наприклад, нейронних мереж або LSTM-моделей), які здатні швидко визначати відхилення показників здоров'я від норми в реальному часі. Якщо алгоритм виявить критичні зміни, система негайно повідомить пацієнта та медичний персонал.

3. Персоналізовані рекомендації та плани профілактики

На основі аналізу медичних показників і способу життя користувача, алгоритми AI/ML будуть формувати індивідуальні плани профілактики захворювань, включаючи рекомендації щодо фізичної активності, харчування та режиму дня.

Технічно взаємодія інформаційної системи з AI/ML буде реалізовуватись таким чином:

- Збір та попередня обробка даних: медичні показники надходять від носимих пристроїв і мобільних застосунків, після чого передаються у хмарне сховище для подальшої обробки.
- Підготовка та очищення даних: отримані дані проходять процес очищення, стандартизації, нормалізації та підготовки до аналізу.
- Інтеграція з AI/ML-модулями: підготовлені дані передаються в модулі AI/ML, де відбувається навчання моделей та прогнозування ризиків і аномалій.

- Формування вихідних даних та рекомендацій: результати роботи алгоритмів повертаються в систему, де на основі цих даних формуються медичні рекомендації та повідомлення для користувачів.

Наведемо приблизну структуру архітектури системи з використанням AI/ML:

1. Збір даних (Edge Level)

- Носимі пристрої: фітнес-браслети, смарт-годинники, ЕКГ-датчики тощо.
- Мобільні пристрої: смартфони збирають дані з сенсорів через Bluetooth або Wi-Fi.

2. Передача та зберігання даних

- Дані надходять до хмарного сховища, де проходять попередню обробку (очищення, стандартизація, перевірка на аномалії).
- Здійснюється логування та шифрування відповідно до стандартів GDPR/HIPAA.

3. AI/ML-модуль

- Тут реалізовано алгоритми класифікації та прогнозування:
    - Визначення ризику серцевих захворювань.
    - Виявлення аномальних змін у показниках.
    - Формування персоналізованих профілактичних рекомендацій.
- Алгоритми можуть включати нейронні мережі, Random Forest, LSTM тощо.

4. Модуль прийняття рішень

- Результати AI/ML-модулів інтерпретуються та передаються в модуль логіки системи.
- Визначається, чи потрібне негайне сповіщення користувача або медичного фахівця.

5. Інтерфейси взаємодії

- Мобільний застосунок користувача: відображає рекомендації, попередження, статус здоров’я.

- Панель медичного фахівця: дає змогу переглядати історію пацієнтів, діагностичні висновки, графіки.

Для забезпечення надійної, масштабованої та безпечної роботи інформаційної системи була наведена архітектура програмного забезпечення (рис. 2.8), яка поєднує клієнтські, серверні та аналітичні компоненти. Такий підхід дозволяє ефективно організувати обробку даних, взаємодію між пристроями та модулями, а також забезпечити гнучкість у розгортанні системи у різних середовищах.

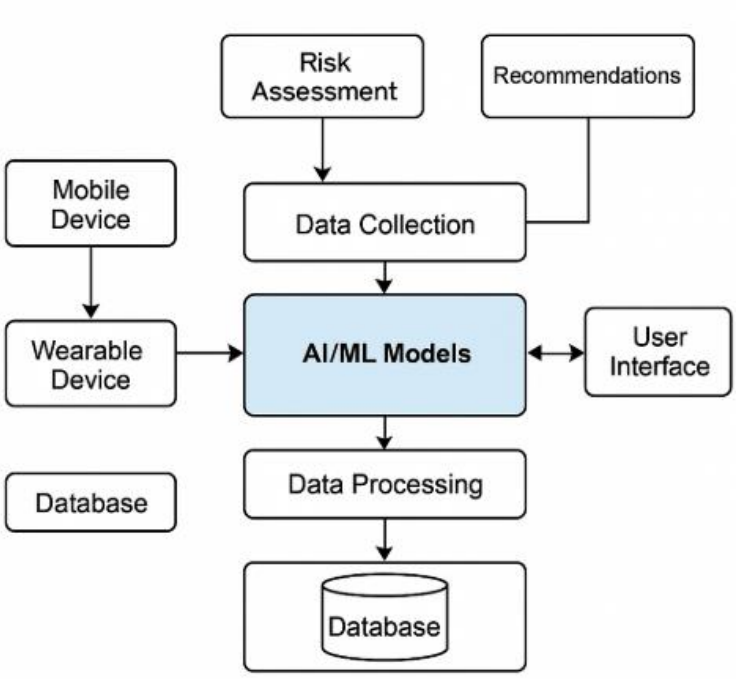


Рис. 2.8. Поверхневу архітектуру інформаційної системи моніторингу здоров’я серця з інтеграцією AI/ML

На зображенні (рис. 2.8) представлено поверхневу архітектуру інформаційної системи моніторингу здоров’я серця з інтеграцією AI/ML. Схема демонструє основні етапи обробки даних — від збору до формування результатів і взаємодії з користувачем.

РОЗДІЛ 3. РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ

3.1 Розробка концептуальної моделі бази даних проєкту

Для розробки концептуальної моделі бази даних проєкту, важливо визначити основні сутності, їх атрибути та взаємозв’язки між ними. Ось ключові сутності, які мають бути в базі даних:

1. Пацієнти

Атрибути: Ідентифікатор користувача, ім'я, прізвище, дата народження, електронна пошта, пароль, номер телефону, адреса, гендер, медична історія, підписка.

Зв'язки: зберігає тип підписки.

2. Лікарі

Атрибути: Ідентифікатор користувача, ім'я, прізвище, спеціальність, пошта, пароль, номер телефону, ідентифікатор лікарні.

Зв'язки: зберігає ідентифікатор лікарні де працює.

3. Пристрої для моніторингу

Атрибути: Ідентифікатор, тип девайсу, модель, ідентифікатор пацієнту, дата активації.

Взаємовідносини: зберігає ідентифікатор пацієнта.

4. Дані моніторингу

Атрибути: Ідентифікатор, ідентифікатор девайсу, час, дані, рекомендації.

Взаємозв'язок: зберігає ідентифікатор девайсу.

5. Попередження

Атрибути: Ідентифікатор, ID пацієнту, ID даних, тип, сповіщення, час, оброблене.

Зв'язки: зберігає ідентифікатор пацієнта та даних моніторингу.

6. Підписки

Атрибути: Ідентифікатор, тип, ціна, бенефіти.

7. Лікарні

Атрибути: Ідентифікатор, назва, адреса, телефон.

Зв'язки: Відгуки пов'язані з товарами та користувачами.

8. Пацієнт-історія попереджень

Історія попереджень для кожного пацієнта.

Концептуальна модель бази даних виконана за допомогою Діаграм Чена. Це модель даних, яка дозволяє описувати концептуальні схеми за допомогою узагальнених конструкцій блоків (див. ДОДАТОК Б).

Мета інфологічного (концептуального) моделювання – забезпечення найбільш природних для людини способів збору та представлення тієї інформації, яку передбачається зберігати в базі даних. Ця модель підтримує функціональність додатку, ефективно організовуючи дані.

3.2 Побудова логічної моделі бази даних проєкту

На етапі побудови логічної моделі було додано додаткову сутність

Пацієнт-лікар

Атрибути: Ідентифікатор пацієнту, ідентифікатор лікаря.

На рис. 3.1 представлена даталогічна модель бази даних, яка показує структуру таблиць, їх атрибути та взаємозв’язки між сутностями. Дані в цій моделі нормалізовані: важливим кроком у розробці бази даних є нормалізація для забезпечення оптимальної структури, зменшення дублювання інформації та підвищення ефективності. У даному проєкті нормалізація проведена до третьої нормальної форми (3НФ).

На основі цієї моделі бази даних можна зробити висновок, що структура проєкту забезпечує повну та ефективну обробку інформації, необхідної для функціонування системи персоналізованого моніторингу здоров’я серця. Логічна модель включає всі основні сутності, такі як пацієнти, лікарі, пристрої, дані моніторингу, попередження, лікарні та підписки, і відображає взаємозв'язки між ними.

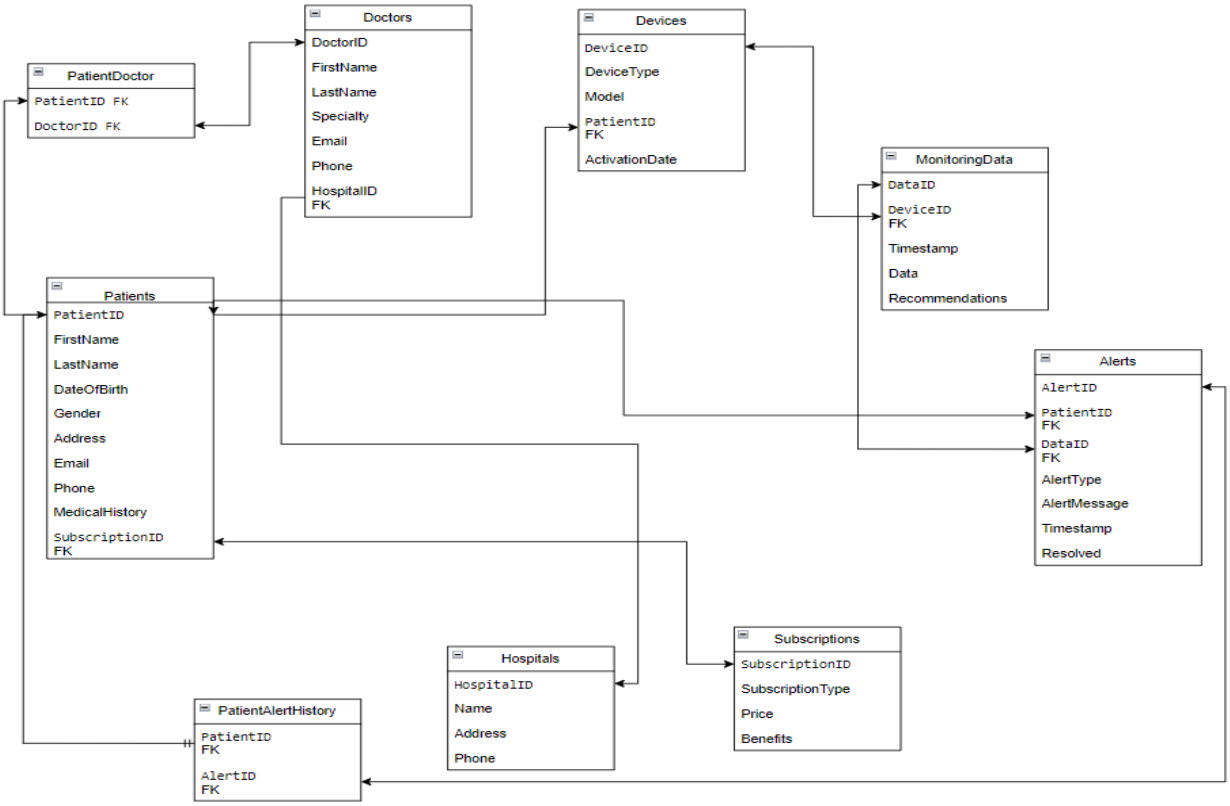


Рис. 3.1 Логічна (даталогічна) модель бази даних

Ця модель бази даних сприяє забезпеченню високого рівня обслуговування користувачів та дозволяє здійснювати надійне управління даними. Вона також забезпечує можливість розширення та масштабування системи в майбутньому, що є важливим для підтримання зростання бізнесу та адаптації до нових вимог ринку.

Логічна модель бази даних є фундаментальною частиною проєкту, що сприяє його успішній реалізації та подальшому розвитку.

3.3 Архітектура системи

Основні компоненти архітектури включають діаграму прецедентів, блок-схему процесу авторизації користувачів, а також діаграму бази даних, яка показує структуру даних і їхні зв’язки.

Блок-схема діяльності (рис. 3.2) описує процес авторизації користувачів у системі. Процес починається з входу до системи та перевірки рівня доступу користувача. Залежно від результату цієї перевірки, користувач може отримати

права адміністратора, пацієнта або лікаря. Після перевірки доступу користувач може або продовжити роботу в системі, або вийти з неї.

Діаграма прецедентів (рис. 3.3) ілюструє основні функціональні можливості системи та взаємодії між користувачами та системою після входу в систему. На діаграмі представлені такі ролі користувачів, як пацієнт, лікар та адміністратор. Основні дії включають перегляд рекомендацій, купівля підписки, перегляд медичні дані, додавання медичних записів, відправлення рекомендацій, перегляд статистики, керування дозволів. Всі ці дії взаємодіють з базою даних (БД), яка є центральним елементом системи.

Ці візуальні моделі дозволяють чітко сформулювати вимоги до системи, забезпечити прозоре розуміння її логіки та функціональності для всіх учасників проєкту. Вони також слугують основою для подальшої реалізації програмного забезпечення та тестування ключових сценаріїв взаємодії.

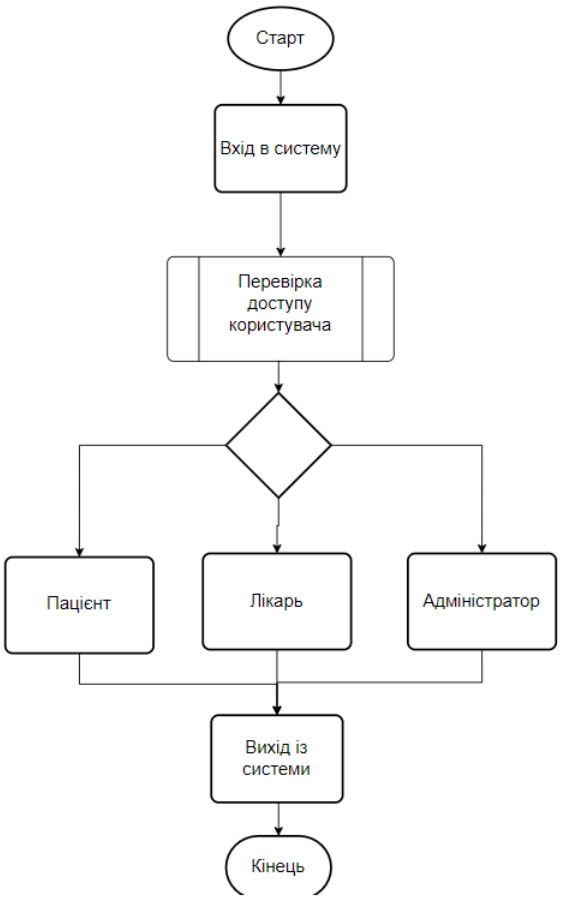


Рис. 3.2. Блок-схема діяльності – авторизація в порталі

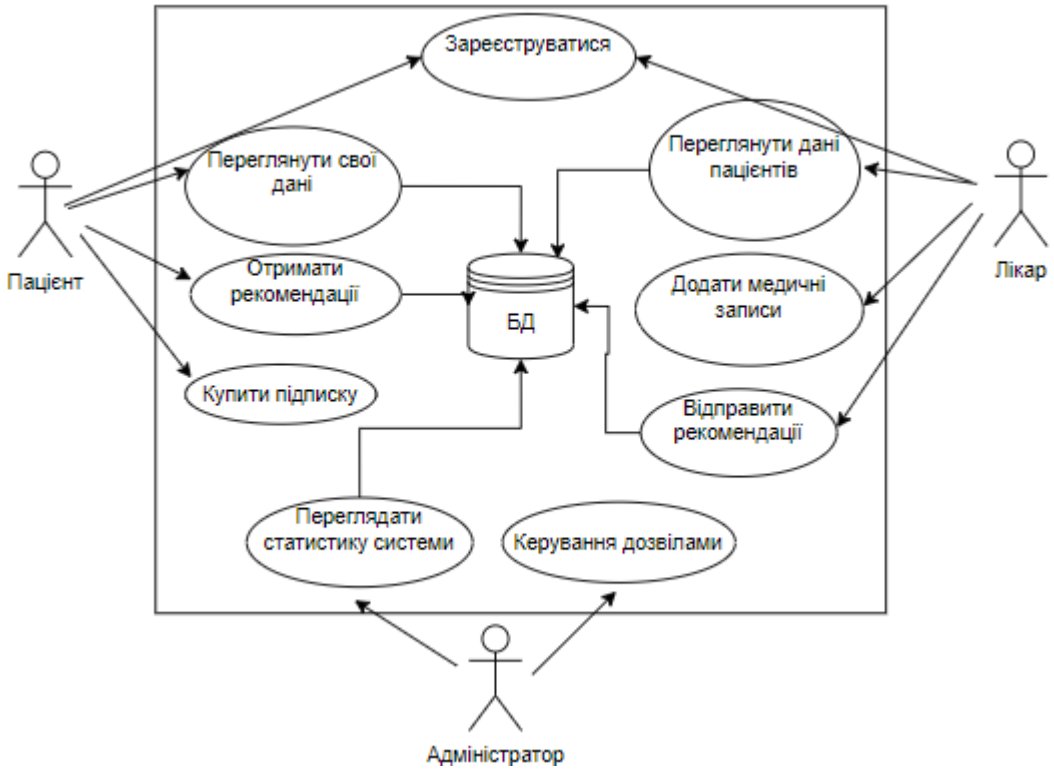


Рис. 3.3. Діаграма прецедентів

Архітектура системи програми забезпечує комплексний підхід до управління товарами, користувачами та замовленнями. Використання діаграми прецедентів та блок-схеми діяльності дозволяє чітко визначити ролі користувачів і їхні взаємодії з системою. Діаграма бази даних відображає логічну структуру даних та зв'язки між таблицями, що гарантує цілісність та ефективне управління інформацією. Такий підхід дозволяє створити надійну та функціональну систему для електронної комерції.

Поняття планування робіт у проєкті

Планування робіт є однією з ключових складових процесу управління проєктами та має вирішальне значення для досягнення цілей проєкту в межах встановлених обмежень часу, бюджету та якості. Згідно з PMBOK® Guide (7-е видання), планування визначається як процес визначення цілей, розробки стратегії їх досягнення, оцінки необхідних ресурсів та встановлення послідовності виконання робіт.

Процес планування робіт дозволяє:

- Визначити обсяг необхідних завдань та їх розбиття на дрібніші частини для полегшення управління;
- Оцінити строки та ресурси, необхідні для кожної діяльності;
- Встановити логічні залежності між роботами;
- Ідентифікувати критичний шлях проєкту;
- Сформувати основу для подальшого моніторингу та контролю за виконанням робіт.

У контексті розробки інформаційної системи для моніторингу здоров'я серця планування робіт дозволяє структурувати весь обсяг завдань від етапу ініціалізації проєкту до фінального впровадження системи у клінічне середовище. Це забезпечує можливість своєчасної оцінки прогресу, виявлення ризиків затримок та своєчасного корегування плану.

Планування робіт формує основу для подальшого створення таких інструментів управління, як ієрархічна структура робіт (WBS), діаграма Ганта, матриця відповідальності та графік ресурсів, що забезпечує цілісне бачення виконання проєкту.

Ієрархічна структура робіт (WBS, Work Breakdown Structure) — це методологія декомпозиції проєкту на більш дрібні, керовані компоненти, які називаються пакетами робіт. WBS створює ієрархічне дерево завдань, що дозволяє чітко визначити весь обсяг робіт, який має бути виконаний для досягнення цілей проєкту.

WBS є основою для подальшого планування ресурсів, строків, витрат та контролю виконання проєкту. Кожен рівень ієрархії деталізує завдання до рівня, на якому вони можуть бути ефективно заплановані, виконані та проконтрольовані.

Основні переваги побудови WBS:

- Забезпечення повноти охоплення всіх робіт проєкту;
- Полегшення управління великими та складними проєктами за рахунок поділу на зрозумілі компоненти;
- Визначення чітких зон відповідальності для членів команди;

- Підвищення прозорості й контрольованості виконання завдань.

У межах розробки інформаційної системи для моніторингу здоров'я серця була побудована детальна WBS, яка включає чотири основні фази: Ініціалізація, Аналіз і проєктування, Реалізація та тестування, Впровадження і завершення. Кожна фаза розділена на конкретні пакети робіт, які, у свою чергу, деталізуються на індивідуальні завдання. Такий підхід забезпечує послідовність, логічність та контрольованість усіх етапів розробки системи.

Для наочного представлення загальної структури проєкту було сформовано ієрархію робіт у середовищі MS Project (рис. 3.4). Така структура дозволяє систематизувати всі етапи життєвого циклу проєкту, визначити тривалість кожної фази, послідовність виконання завдань, а також логічні залежності між ними. Це є основою для ефективного планування, контролю виконання та подальшого управління ресурсами в межах реалізації інформаційної системи для моніторингу здоров'я серця. Узагальнена ієрархічна структура представлена нижче.

	Task Mode	Task Name	Duration	Start	Finish	Predecessors
0		Система для моніторинг	350 days	Mon 21.04.25	Fri 21.08.26	
1		Ініціалізація проєкту	35 days	Mon 21.04.25	Fri 06.06.25	
14		Аналіз і проєктування	75 days	Mon 09.06.25	Fri 19.09.25	10
27		Реалізація та тестування	180 days	Mon 22.09.25	Fri 29.05.26	14
40		Впровадження та завер	60 days	Mon 01.06.26	Fri 21.08.26	27

Рис. 3.4. Згорнута ієрархія робіт MS Project

Для детального планування етапів розробки інформаційної системи була сформована розгорнута ієрархія робіт, яка охоплює всі фази життєвого циклу проєкту — від ініціалізації та аналізу вимог до розробки, тестування та впровадження (рис. 3.5, рис. 3.6). Така деталізація дозволяє чітко структурувати завдання, визначити їхню послідовність, встановити залежності між етапами, а також контролювати використання часу та ресурсів. Це є необхідною умовою

для ефективного управління проєктом, забезпечення його своєчасного виконання та досягнення очікуваних результатів.

Діаграма Ганта є одним із найпоширеніших інструментів візуалізації плану виконання проєкту. Вона дозволяє графічно відобразити послідовність, тривалість і взаємозв'язки між окремими роботами, що входять до складу проєкту. Створена Генрі Гантом у 1910-х роках, ця діаграма залишається ключовим елементом планування і контролю проєктів у сучасному управлінні.

Діаграма Ганта складається з горизонтальної шкали часу та списку завдань, кожне з яких зображується у вигляді прямокутного блоку. Довжина блоку відповідає тривалості завдання, а взаємозв’язки між завданнями показують залежності та можливі обмеження (наприклад, завдання може початися тільки після завершення попереднього).

Task Mode	Task Name	Duration	Start	Finish	Predecessors
	Система для моніторингу здоров'я серця людини	**350 days?**	**Mon 21.04.25**	**Fri 21.08.26**	
	1 Ініціалізація проєкту	**35 days?**	**Mon 21.04.25**	**Fri 06.06.25**	
	1.1 Розробка концепції	**10 days?**	**Mon 21.04.25**	**Fri 02.05.25**	
	1.1.1	1 day?	Mon 21.04.25	Mon 21.04.25	
	1.1.2 Аналіз ринку медичних ІС	3 days	Mon 21.04.25	Wed 23.04.25	
	1.1.3 Визначення цілей і задач системи	4 days	Thu 24.04.25	Tue 29.04.25	4
	1.1.4 Визначення функціональних обмежень	3 days	Wed 30.04.25	Fri 02.05.25	5
	1.2 Обґрунтування та ТЕО	**15 days**	**Mon 05.05.25**	**Fri 23.05.25**	**6**
	1.2.1 Розрахунок бюджету	5 days	Mon 05.05.25	Fri 09.05.25	
	1.2.2 Аналіз зацікавлених сторін	5 days	Mon 12.05.25	Fri 16.05.25	8
	1.2.3 Складання попереднього плану	5 days	Mon 19.05.25	Fri 23.05.25	9
	1.3 Затвердження ініціативи	**10 days**	**Mon 26.05.25**	**Fri 06.06.25**	**10**
	1.3.1 Узгодження з науковим керівником	3 days	Mon 26.05.25	Wed 28.05.25	
	1.3.2 Підготовка обґрунтування у вигляді презентації	4 days	Thu 29.05.25	Tue 03.06.25	12
	1.3.3 Формування команди (аналітик, розробник, тестувальник)	3 days	Wed 04.06.25	Fri 06.06.25	13
	2 Аналіз і проєктування	**75 days**	**Mon 09.06.25**	**Fri 19.09.25**	**11**
	2.1 Збір та аналіз вимог	**25 days**	**Mon 09.06.25**	**Fri 11.07.25**	
	2.1.1 Опитування цільової аудиторії (пацієнти, лікарі)	8 days	Mon 09.06.25	Wed 18.06.25	
	2.1.2 Визначення функціональних та нефункціональних вимог	10 days	Thu 19.06.25	Wed 02.07.25	17
	2.1.3 Формалізація специфікації вимог (SRS)	7 days	Thu 03.07.25	Fri 11.07.25	18
	2.2 Архітектура системи	**20 days**	**Mon 14.07.25**	**Fri 08.08.25**	**16**
	2.2.1 Вибір клієнт-серверної моделі	6 days	Mon 14.07.25	Mon 21.07.25	
	2.2.2 Визначення основних компонентів (БД, API, Front-End)	7 days	Tue 22.07.25	Wed 30.07.25	21
	2.2.3 Розробка загальної схеми взаємодії модулів	7 days	Thu 31.07.25	Fri 08.08.25	22
	2.3 Технічне проєктування	**30 days**	**Mon 11.08.25**	**Fri 19.09.25**	**20**
	2.3.1 Розробка ER-моделі БД	10 days	Mon 11.08.25	Fri 22.08.25	
	2.3.2 Прототипування інтерфейсу	10 days	Mon 25.08.25	Fri 05.09.25	25
	2.3.3 Визначення протоколів взаємодії (REST, HTTPS)	10 days	Mon 08.09.25	Fri 19.09.25	26
	3 Реалізація та тестування	**180 days**	**Mon 22.09.25**	**Fri 29.05.26**	**15**
	3.1 Розробка системи	**100 days**	**Mon 22.09.25**	**Fri 06.02.26**	
	3.1.1 Програмування бекенду (Python + Flask/FastAPI)	40 days	Mon 22.09.25	Fri 14.11.25	
	3.1.2 Розробка фронтенду (HTML/CSS/JS або React)	30 days	Mon 17.11.25	Fri 26.12.25	30
	3.1.3 Створення системи автентифікації	30 days	Mon 29.12.25	Fri 06.02.26	31
	3.2 Тестування	**40 days**	**Mon 09.02.26**	**Fri 03.04.26**	**29**
	3.2.1 Функціональне тестування (юніт-тести)	15 days	Mon 09.02.26	Fri 27.02.26	

Рис. 3.5. Ієрархія робіт проєкту в MS Project

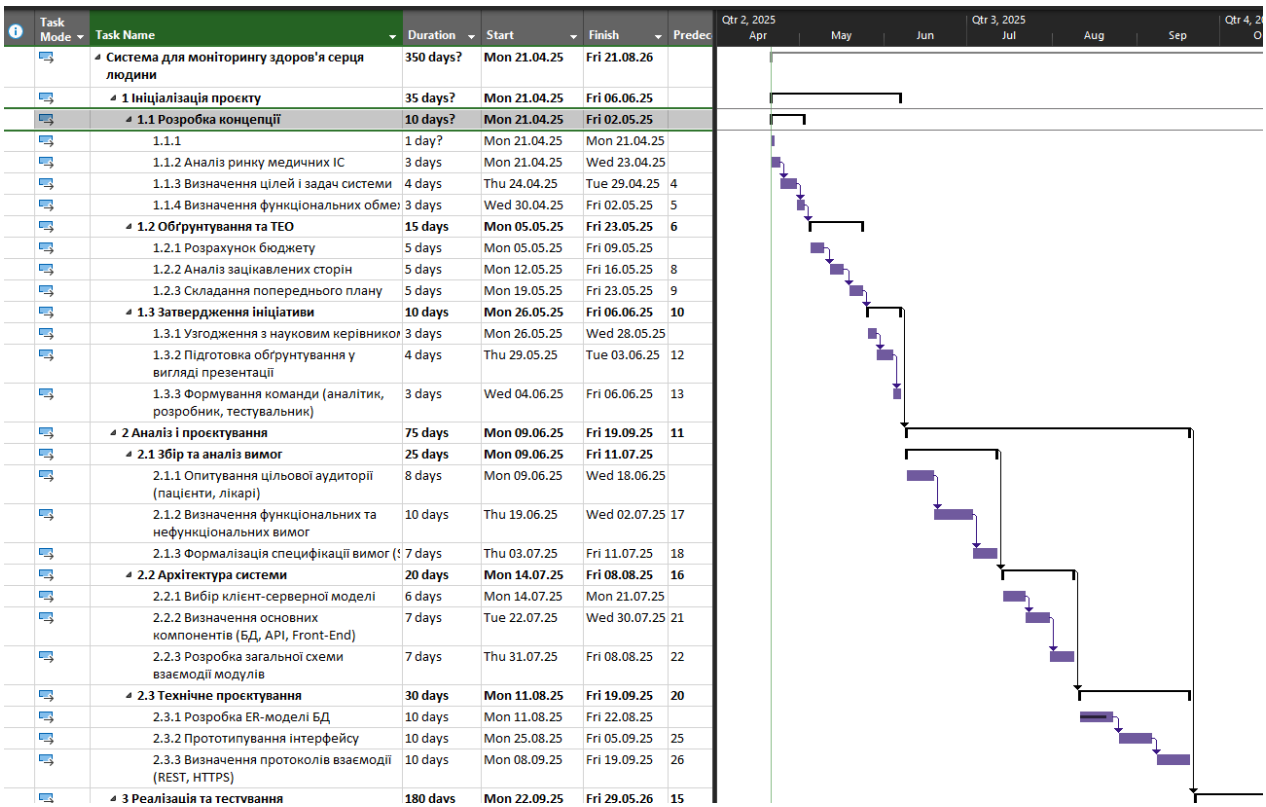
		3.2 Тестування	**40 days**	**Mon 09.02.26**	**Fri 03.04.26**	**29**
		3.2.1 Функціональне тестування (юніт-тести)	15 days	Mon 09.02.26	Fri 27.02.26	
		3.2.2 Тестування безпеки та валідації даних	10 days	Mon 02.03.26	Fri 13.03.26	34
		3.2.3 Навантажувальне тестування API	15 days	Mon 16.03.26	Fri 03.04.26	35
		3.3 Інтеграція з сенсорами	**40 days**	**Mon 06.04.26**	**Fri 29.05.26**	**33**
		3.3.1 Обробка даних з BLE/IoT пристроїв	15 days	Mon 06.04.26	Fri 24.04.26	
		3.3.2 Синхронізація з базою даних	10 days	Mon 27.04.26	Fri 08.05.26	38
		3.3.3 Налаштування сповіщень при виявленні аномалій	15 days	Mon 11.05.26	Fri 29.05.26	39
	4 Впровадження та завершення		**60 days**	**Mon 01.06.26**	**Fri 21.08.26**	**28**
		4.1 Впровадження MVP	**20 days**	**Mon 01.06.26**	**Fri 26.06.26**	
		4.1.1 Розгортання у хмарному середовищі (наприклад, Heroku/AWS)	10 days	Mon 01.06.26	Fri 12.06.26	
		4.1.2 Проведення демонстрації системи	5 days	Mon 15.06.26	Fri 19.06.26	43
		4.1.3 Отримання зворотного зв'язку	5 days	Mon 22.06.26	Fri 26.06.26	44
		4.2 Документація та навчання	**20 days**	**Mon 29.06.26**	**Fri 24.07.26**	**42**
		4.2.1 Підготовка технічної документації	7 days	Mon 29.06.26	Tue 07.07.26	
		4.2.2 Написання інструкції для користувача	7 days	Wed 08.07.26	Thu 16.07.26	47
		4.2.3 Проведення тренінгу (як приклад, запис відео)	6 days	Fri 17.07.26	Fri 24.07.26	48
		4.3 Аналіз ефективності та завершення	**20 days**	**Mon 27.07.26**	**Fri 21.08.26**	**46**
		4.3.1 Підготовка підсумкового звіту	7 days	Mon 27.07.26	Tue 04.08.26	
		4.3.2 Оцінка результатів порівняно з цілями	7 days	Wed 05.08.26	Thu 13.08.26	51
		4.3.3 Закриття проєкту	6 days	Fri 14.08.26	Fri 21.08.26	52

Рис. 3.6. Ієрархія робіт проєкту в MS Project Продовження

Основні переваги використання діаграми Ганта:

- Наглядне відображення загальної картини виконання проєкту;
- Можливість відстеження фактичного прогресу відносно запланованого графіка;
- Виявлення критичних точок і потенційних затримок;
- Підвищення ефективності комунікації між учасниками проєкту.

У проєкті розробки інформаційної системи для моніторингу здоров'я серця діаграма Ганта (рис. 3.7) була побудована на основі ієрархічної структури робіт. Вона відображає тривалість і послідовність усіх основних фаз — від аналізу ринку медичних інформаційних систем до завершального етапу впровадження та оцінки результатів. Це дозволяє ефективно управляти строками виконання, контролювати залежності між роботами та оперативно реагувати на можливі ризики порушення графіка.



Task Mode	Task Name	Duration	Start	Finish	Predec
	Система для моніторингу здоров'я серця людини	**350 days?**	**Mon 21.04.25**	**Fri 21.08.26**	
	1 Ініціалізація проєкту	**35 days?**	**Mon 21.04.25**	**Fri 06.06.25**	
	1.1 Розробка концепції	**10 days?**	**Mon 21.04.25**	**Fri 02.05.25**	
	1.1.1	1 day?	Mon 21.04.25	Mon 21.04.25	
	1.1.2 Аналіз ринку медичних ІС	3 days	Mon 21.04.25	Wed 23.04.25	
	1.1.3 Визначення цілей і задач системи	4 days	Thu 24.04.25	Tue 29.04.25	4
	1.1.4 Визначення функціональних обме	3 days	Wed 30.04.25	Fri 02.05.25	5
	1.2 Обґрунтування та ТЕО	**15 days**	**Mon 05.05.25**	**Fri 23.05.25**	**6**
	1.2.1 Розрахунок бюджету	5 days	Mon 05.05.25	Fri 09.05.25	
	1.2.2 Аналіз зацікавлених сторін	5 days	Mon 12.05.25	Fri 16.05.25	8
	1.2.3 Складання попереднього плану	5 days	Mon 19.05.25	Fri 23.05.25	9
	1.3 Затвердження ініціативи	**10 days**	**Mon 26.05.25**	**Fri 06.06.25**	**10**
	1.3.1 Узгодження з науковим керівнико	3 days	Mon 26.05.25	Wed 28.05.25	
	1.3.2 Підготовка обґрунтування у вигляді презентації	4 days	Thu 29.05.25	Tue 03.06.25	12
	1.3.3 Формування команди (аналітик, розробник, тестувальник)	3 days	Wed 04.06.25	Fri 06.06.25	13
	2 Аналіз і проєктування	**75 days**	**Mon 09.06.25**	**Fri 19.09.25**	**11**
	2.1 Збір та аналіз вимог	**25 days**	**Mon 09.06.25**	**Fri 11.07.25**	
	2.1.1 Опитування цільової аудиторії (пацієнти, лікарі)	8 days	Mon 09.06.25	Wed 18.06.25	
	2.1.2 Визначення функціональних та нефункціональних вимог	10 days	Thu 19.06.25	Wed 02.07.25	17
	2.1.3 Формалізація специфікації вимог (S	7 days	Thu 03.07.25	Fri 11.07.25	18
	2.2 Архітектура системи	**20 days**	**Mon 14.07.25**	**Fri 08.08.25**	**16**
	2.2.1 Вибір клієнт-серверної моделі	6 days	Mon 14.07.25	Mon 21.07.25	
	2.2.2 Визначення основних компонентів (БД, API, Front-End)	7 days	Tue 22.07.25	Wed 30.07.25	21
	2.2.3 Розробка загальної схеми взаємодії модулів	7 days	Thu 31.07.25	Fri 08.08.25	22
	2.3 Технічне проєктування	**30 days**	**Mon 11.08.25**	**Fri 19.09.25**	**20**
	2.3.1 Розробка ER-моделі БД	10 days	Mon 11.08.25	Fri 22.08.25	
	2.3.2 Прототипування інтерфейсу	10 days	Mon 25.08.25	Fri 05.09.25	25
	2.3.3 Визначення протоколів взаємодії (REST, HTTPS)	10 days	Mon 08.09.25	Fri 19.09.25	26
	3 Реалізація та тестування	**180 days**	**Mon 22.09.25**	**Fri 29.05.26**	**15**

Рис. 3.7. Діаграма Ганта

Наведена діаграма дає можливість візуалізувати критичний шлях проєкту та своєчасно виявляти потенційні затримки в реалізації окремих етапів. Завдяки цьому управлінська команда може приймати обґрунтовані рішення щодо перерозподілу ресурсів або зміни пріоритетів для забезпечення своєчасного завершення проєкту.

3.4 Захист та безпека персональних даних

Одним з ключових методів забезпечення конфіденційності персональних і медичних даних користувачів інформаційної системи є застосування шифрування. У розроблюваній системі передбачено дві основні форми шифрування:

- Шифрування даних при передачі: Для забезпечення конфіденційності та цілісності даних під час їх передачі між мобільним застосунком, сервером і зовнішніми сервісами використовується протокол безпечного зв'язку TLS (Transport Layer Security) версії 1.3. Це дозволяє захистити дані від перехоплення та зміни сторонніми особами. Використовуються криптографічні алгоритми сімейства AES-GCM з розміром ключа 256 біт.

- Шифрування даних у стані спокою (at rest): Для захисту даних, що зберігаються у базі даних та резервних сховищах, застосовується алгоритм симетричного шифрування AES-256. Ключі шифрування регулярно оновлюються (процедура ротації ключів) кожні 30 днів і зберігаються у спеціалізованому захищеному сховищі ключів (Key Management Service, KMS). Це забезпечує високий рівень безпеки і мінімізує ризики у випадку компрометації одного з ключів.

Застосування таких методів шифрування повністю відповідає міжнародним стандартам захисту даних GDPR та HIPAA, які регламентують безпечну обробку та зберігання персональних медичних даних.

Аутентифікація та авторизація

Система моніторингу здоров'я серця передбачає комплексний підхід до автентифікації та авторизації користувачів з метою забезпечення доступу лише авторизованих осіб та запобігання несанкціонованому доступу:

- Аутентифікація користувачів: Реалізована на базі протоколу OAuth 2.0 з підтримкою OpenID Connect. Додатковим рівнем захисту є застосування двофакторної автентифікації (Two-Factor Authentication, 2FA), яка вимагає від користувача підтвердження доступу додатковим фактором — кодом, отриманим через SMS-повідомлення або за допомогою мобільного застосунку генерації тимчасових паролів (Time-based One-Time Password, TOTP).
- Авторизація доступу: Система використовує модель контролю доступу на основі ролей (Role-Based Access Control, RBAC). Це дозволяє чітко розмежовувати рівні доступу залежно від ролі користувача в системі:
  - Користувач (пацієнт): має доступ лише до власних медичних даних і рекомендацій.
  - Медичний спеціаліст (лікар): має право доступу до даних пацієнтів, які закріплені за ним, з метою моніторингу, діагностики і формування медичних рекомендацій.

- Адміністратор системи: має обмежений доступ до технічних налаштувань та управління системою без права доступу до персональних медичних даних пацієнтів.

Взаємодія між компонентами системи, зокрема сервером та клієнтськими застосунками, здійснюється за допомогою API-інтерфейсів, які мають високий рівень безпеки:

- API Gateway: Центральний компонент, через який проходять усі запити до серверної частини. API Gateway забезпечує:
  - Перевірку автентичності JWT-токенів (JSON Web Token).
  - Моніторинг та обмеження частоти запитів (rate limiting).
  - Захист від DDoS-атак завдяки вбудованим механізмам фільтрації запитів.
- Політика CORS (Cross-Origin Resource Sharing): Чітко визначено список дозволених доменів, з яких можлива взаємодія з API-сервісами, що зменшує ризики несанкціонованих запитів.
- Захист від атак типу Brute-force: Система автентифікації передбачає механізми блокування тимчасового доступу у разі численних невдалих спроб входу, що ефективно перешкоджає автоматичним підбором паролів.

Важливим елементом забезпечення інформаційної безпеки є детальний контроль змін та ретельне ведення логів (журналів):

- Журналювання операцій (Audit logging): Усі дії користувачів, включаючи доступ до даних, зміни та спроби входу, реєструються з обов'язковим зазначенням дати, часу, ідентифікатора користувача та характеру дії.
- Immutable logs: Логи зберігаються у спеціальному захищеному сховищі з використанням технологій блокчейн або подібних методів, що унеможливлює зміну журналів після їх створення.
- Моніторинг та реагування на події: Інтеграція логів з системами моніторингу та управління подіями безпеки (Security Information and Event Management, SIEM). Це дозволяє оперативно реагувати на потенційні загрози та аномальні події у роботі системи.

Резервне копіювання даних є невід'ємною частиною стратегії забезпечення безпеки інформаційної системи моніторингу здоров’я серця. У рамках проєкту реалізується автоматизоване резервне копіювання бази даних та критично важливих конфігураційних файлів, яке виконується щоденно та щотижнево залежно від категорії даних.

Для цього використовуються технології повних (full backup) та інкрементних резервних копій (incremental backup). Повні резервні копії виконуються щотижня та містять повні набори даних системи, а інкрементні – щоденно, вони фіксують тільки зміни, які відбулися з моменту останнього повного резервного копіювання. Це дозволяє значно зменшити обсяги інформації, що зберігається, та скоротити час, необхідний на відновлення.

Відновлення даних відбувається за допомогою спеціальних процедур, розроблених і протестованих у рамках проєкту. Кожна резервна копія перевіряється на цілісність за допомогою алгоритмів контрольних сум (наприклад, MD5 або SHA-256), що гарантує можливість повного та коректного відновлення інформації.

Для зберігання резервних копій використовується хмарне рішення (наприклад, Amazon S3 або Google Cloud Storage), що забезпечує високу доступність, надійність і масштабованість. Всі дані, що передаються у сховище, шифруються за допомогою протоколу AES-256.

Виявлення вторгнень та інцидент-менеджмент

Для своєчасного виявлення та реагування на потенційні загрози безпеці в системі реалізовано автоматизовану систему моніторингу та виявлення вторгнень (Intrusion Detection System, IDS). В якості IDS використовується програмне рішення на базі Snort або Suricata, що здійснює постійний аналіз мережевого трафіку для виявлення аномальної активності.

Система налаштована на ідентифікацію різних типів загроз, таких як SQL-ін'єкції, DoS-атаки, несанкціоновані спроби доступу до ресурсів, сканування портів тощо. При виявленні загрози IDS автоматично генерує сповіщення, які

надсилаються адміністратору безпеки через електронну пошту та системи оперативних повідомлень (наприклад, Slack, Telegram).

Інцидент-менеджмент реалізується відповідно до міжнародних стандартів ISO/IEC 27035. У рамках проєкту визначено процедури, що включають:

- Початковий аналіз інциденту та категоризацію за рівнем критичності;
- Встановлення карантину для ізоляції уражених систем;
- Детальний аналіз та документування інциденту;
- Усунення наслідків інциденту та відновлення нормальної роботи системи;
- Аналіз причин виникнення інциденту та впровадження заходів для запобігання подібних ситуацій у майбутньому.

Для візуалізації принципів забезпечення безпеки персональних даних у розробленій інформаційній системі було сформовано архітектуру захисту даних, яка базується на використанні сучасних стандартів шифрування та автентифікації. На рисунку 3.8 представлено основні механізми захисту даних під час їх передачі та зберігання, а також механізми контролю доступу відповідно до вимог GDPR та HIPAA.

Як показано на рисунку, передача даних між клієнтом і сервером здійснюється з використанням протоколу TLS 1.3, що забезпечує високий рівень захисту даних у процесі комунікації (encryption in transit).

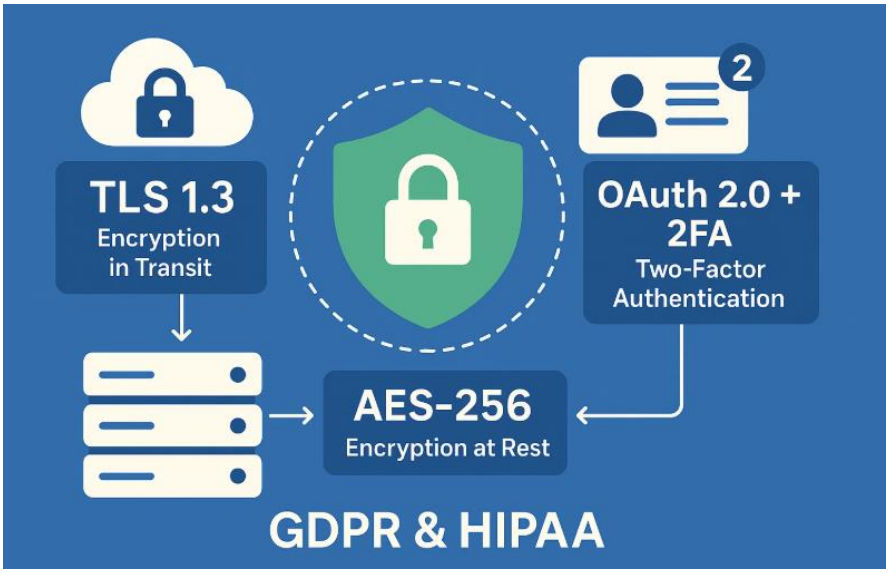


Рис. 3.8. Основні механізми захисту під час передачі та зберігання

Дані, що зберігаються на сервері, захищені за допомогою алгоритму симетричного шифрування AES-256 (encryption at rest). Для автентифікації користувачів впроваджено протокол OAuth 2.0 з підтримкою двофакторної аутентифікації (2FA), що суттєво підвищує стійкість системи до несанкціонованого доступу. Дотримання стандартів GDPR і HIPAA підтверджує відповідність розробленої системи міжнародним вимогам у сфері захисту персональних медичних даних.

Захист інформації реалізовано на кількох рівнях. На рівні передачі даних між клієнтом і сервером застосовується протокол TLS 1.3, який забезпечує шифрування даних під час транспортування (encryption in transit), зводячи до мінімуму ризики перехоплення або модифікації трафіку зловмисниками. Даний протокол підтримує сучасні механізми узгодження ключів, автентифікації та захищеної сесії, що забезпечує конфіденційність, цілісність та автентичність даних.

На рівні збереження даних реалізовано застосування алгоритму AES-256, який вважається одним із найбільш захищених у галузі симетричного шифрування. Збережені медичні записи, журнали подій, облікові дані користувачів та інша критично важлива інформація захищені як фізично (через серверну інфраструктуру), так і логічно (через політику прав доступу, рольову модель користувачів та ізоляцію даних).

Для реалізації вимог HIPAA також реалізовані механізми журналювання доступу до медичних даних, що дозволяє здійснювати аудит активності користувачів і виявляти потенційні спроби порушення політики безпеки. Кожен запит до системи автентифікується, реєструється з міткою часу, ID користувача та IP-адресою, що забезпечує простежуваність усіх дій у системі.

РОЗДІЛ 4. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ РЕАЛІЗАЦІЇ ІТ ПРОЄКТУ

4.1 Опис структури програмного забезпечення

Система персоналізованого моніторингу здоров'я серця складається з трьох основних компонентів: клієнтської частини (Front-end), серверної частини (Back-end) та зовнішніх сервісів. Кожен з цих компонентів має свої особливості та виконує визначені функції для забезпечення ефективної роботи всієї системи.

1. Клієнтська частина (Front-end)

Клієнтська частина — це інтерфейс, з яким взаємодіють кінцеві користувачі системи: пацієнти, лікарі та адміністратори. Основні компоненти клієнтської частини:

Веб-інтерфейс: інтерактивний веб-сайт, через який користувачі можуть зареєструватися, увійти в систему, переглядати свої дані, отримувати рекомендації та спілкуватися з лікарями.

Мобільний додаток: додаток для смартфонів, який дозволяє користувачам здійснювати всі ті ж дії, що й на веб-сайті, але зручніше для використання на мобільних пристроях.

Інтерфейс адміністратора: спеціалізований інтерфейс для адміністраторів, де вони можуть керувати користувачами, налаштуваннями системи, переглядати статистику та звіти.

Технології: HTML, CSS, JavaScript, React, Angular, Vue.js, Flutter (для мобільних додатків).

2. Серверна частина (Back-end)

Серверна частина відповідає за обробку даних, бізнес-логіку та зберігання інформації. Основні компоненти серверної частини:

Сервер обробки даних: відповідає за отримання, обробку та зберігання даних від клієнтів, таких як результати моніторингу, медичні записи, рекомендації лікарів тощо.

База даних: зберігає всю інформацію про користувачів, їхні медичні дані, налаштування системи, історію тривог та інше.

API: надає програмні інтерфейси для взаємодії клієнтської частини та зовнішніх сервісів із сервером.

Технології: Node.js, Python (Django, Flask), Ruby on Rails, Java (Spring), бази даних MySQL, PostgreSQL, MongoDB.

3. Зовнішні сервіси

Зовнішні сервіси забезпечують додаткові функціональні можливості системи шляхом інтеграції з іншими платформами та службами. Основні компоненти зовнішніх сервісів:

Служби телемедицини: забезпечують можливість віддалених консультацій між пацієнтами та лікарями через відеоконференції або чати.

Служби аналітики: інтеграція з сервісами аналітики для обробки великих обсягів даних та надання індивідуальних рекомендацій на основі машинного навчання.

Служби оповіщення: використовуються для надсилання повідомлень пацієнтам і лікарям про критичні події, нагадування про прийом ліків тощо.

Платіжні системи: інтеграція з платіжними сервісами для обробки оплат (наприклад, PayPal, Binance).

Служби доставки: інтеграція з сервісами доставки для автоматизації відправлення ліків.

Технології: RESTful API, SOAP, JSON, XML, WebRTC (для відеоконференцій), сторонні API для інтеграції (наприклад, Twilio для SMS і телефонних викликів, Firebase для пуш-повідомлень).

Структура програмного забезпечення для персоналізованої інформаційної системи моніторингу здоров'я серця включає клієнтську частину для взаємодії з користувачами, серверну частину для обробки та зберігання даних, а також зовнішні сервіси для розширення функціональності. Такий підхід дозволяє створити гнучку, масштабовану та ефективну систему, яка задовольняє потреби всіх зацікавлених сторін.

Вибір та реалізація структури програмного забезпечення для персоналізованої інформаційної системи моніторингу здоров’я серця базується на необхідності забезпечення стабільності, масштабованості та простоти використання для кінцевого користувача. Клієнтська частина (frontend) дозволяє створити інтуїтивно зрозумілий і привабливий користувальницький інтерфейс, який сприяє позитивному досвіду користувача. Серверна частина (backend) забезпечує надійну обробку запитів і взаємодію з базою даних, підтримуючи високу продуктивність системи.

Інтеграція із зовнішніми службами, такими як платіжна системиа PayPal, служби доставки та служби електронної пошти, надають додаткові функції та підвищують рівень автоматизації ваших бізнес-процесів. Хмарні сервіси AWS дають вам можливість масштабувати свої програми та забезпечити високу доступність.

Таким чином, структура програмного забезпечення дає основу для подальшого розвитку та розширення функціональності застосунку, забезпечуючи його конкурентоспроможність на ринку та задоволення потреб користувачів. Це дозволяє ефективно обслуговувати та допомагати пацієнтам, забезпечуючи їм рекомендації від лікарів, моніторинг за їх станом здоров'я та швидке реагування на інциденти.

4.2 Розробка алгоритмів та інтерфейсів програмного забезпечення

В цьому розділі розглядаються ключові аспекти розробки алгоритмів та інтерфейсів програмного забезпечення для застосунку персоналізованої інформаційної системи відслідковування здоров’я серця.

Основною метою є забезпечення ефективної, зручної та інтуїтивної взаємодії користувачів з системою. Це досягається шляхом розробки логічних і структурованих алгоритмів, які реалізують основні бізнес-процеси, а також створенням інтерфейсів, що забезпечують легкість використання. Загальний вигляд застосунку наведено на рис. 4.1.

Було розроблено інтерфейси для наступних сторінок: реєстрація, авторизація, головна сторінка, профіль користувача, налаштування, репорти та сторінка в разі виникнення помилки (див. Додаток В).

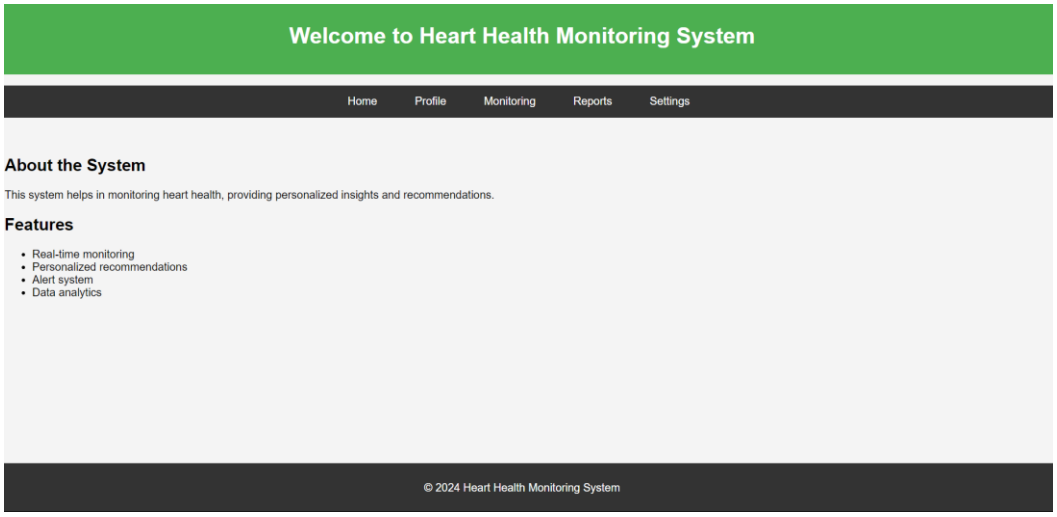


Рис. 4.1 Головна сторінка сайту

Розробка алгоритмів включає в себе проєктування і реалізацію основних процесів, таких як реєстрація та авторизація користувачів, перевірка рекомендацій, моніторинг підключеного девайсу, надбання підписки, консультація із лікарем і інтеграція з зовнішніми службами доставки дивись табл. 4.1.

Таблиця 4.1

Алгоритми програмного забезпечення

Алгоритм реєстрації користувача	1. Користувач заповнює форму реєстрації. 2. Сервер перевіряє коректність даних (наприклад, унікальність email). 3. Якщо дані коректні, користувач заноситься до бази даних. 4. Надсилається підтвердження реєстрації на email користувача.
Алгоритм надбання підписки	1. Користувач обирає підписку та робить оплату через платіжну систему. 2. Сервер оновлює дані про підписку користувача у базі даних. 3. Користувачу набуваю певних привілеїв наданих підпискою.

Продовження табл.4.1

Алгоритм запису на консультацію до лікаря	1. Користувач переходить до пошуку спеціалістів 2. Дивиться спеціальність лікаря та запрошує в нього консультацію.
Інтерфейс реєстрації та авторизації (рис.4.2)	1. Форма для введення особистих даних (ім'я, email, пароль). 2. Кнопка для підтвердження реєстрації. 3. Сторінка авторизації з полями для введення email та пароля.
Інтерфейс своїх медичних даних(рис.4.3)	1. Завантажені користувачем медичні дані. 2. Медична історія користувача 3. Пошук спеціалістів
Інтерфейс налаштування свого профілю(рис.4.4)	1. Інструменти управління обліковим записом. 2. Завантаження фото. 3. Кнопка для перегляду медичних даних.

Ці компоненти та алгоритми забезпечують повний цикл взаємодії користувача з застосунком, від реєстрації (рис. 4.2) до надбання підписок та консультацій з лікарями.

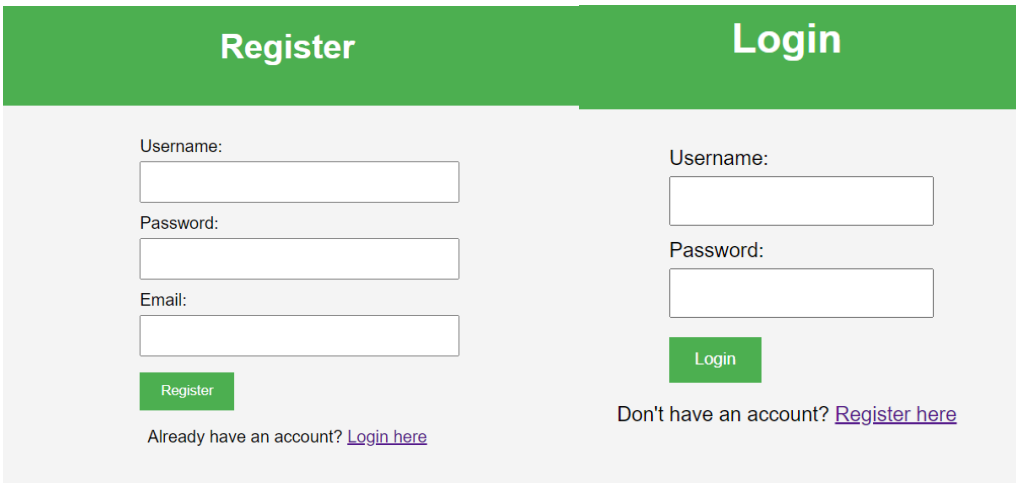


Рис. 4.2 Інтерфейс вікна реєстрації та авторизації

Щоб зареєструватись на платформі достатньо ввести ім’я користувача, пошту та пароль і буде створено профіль користувача. Вхід відповідно можна здійснити за допомогою ім’я користувача чи пошти та пароля, зважаючи на дані які було введено під час реєстрації.

Одна з головних сторінок – перегляд історії захворювань, завантаження медичних документів (рис. 4.3). Таким чином у лікарів та користувачів буде швидкий та зручний доступ до потрібних даних.

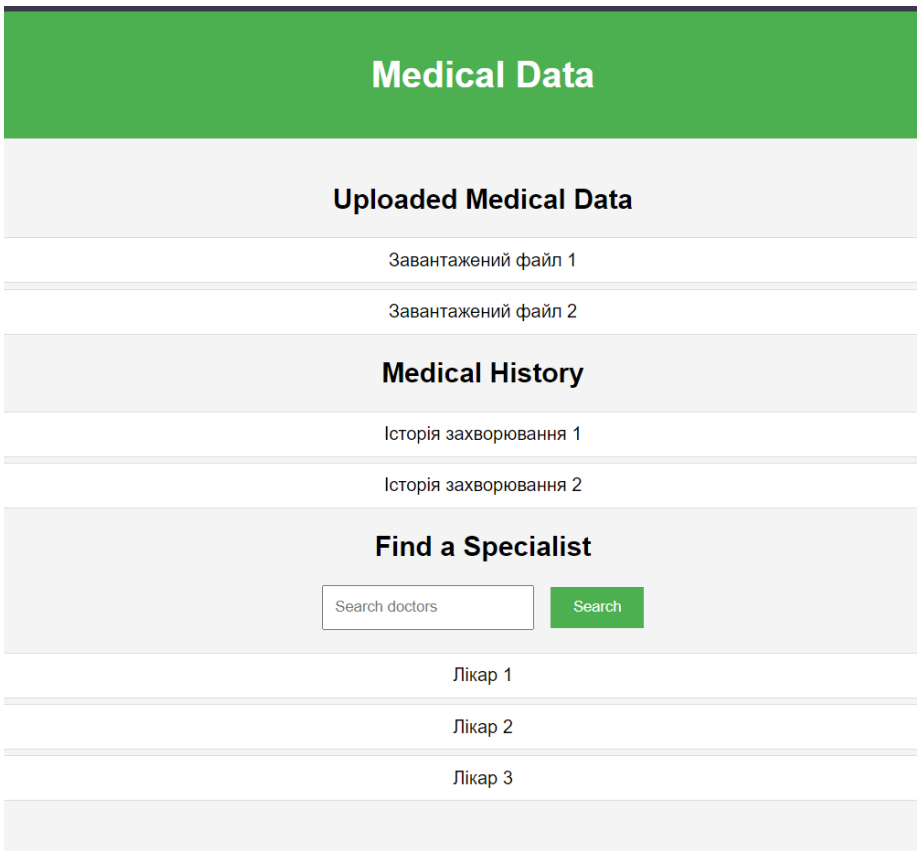


Рис. 4.3 Інтерфейс перегляду медичних даних

Клієнт може вводити та змінювати особисті дані на сторінці зміни профілю користувача (рис. 4.4):

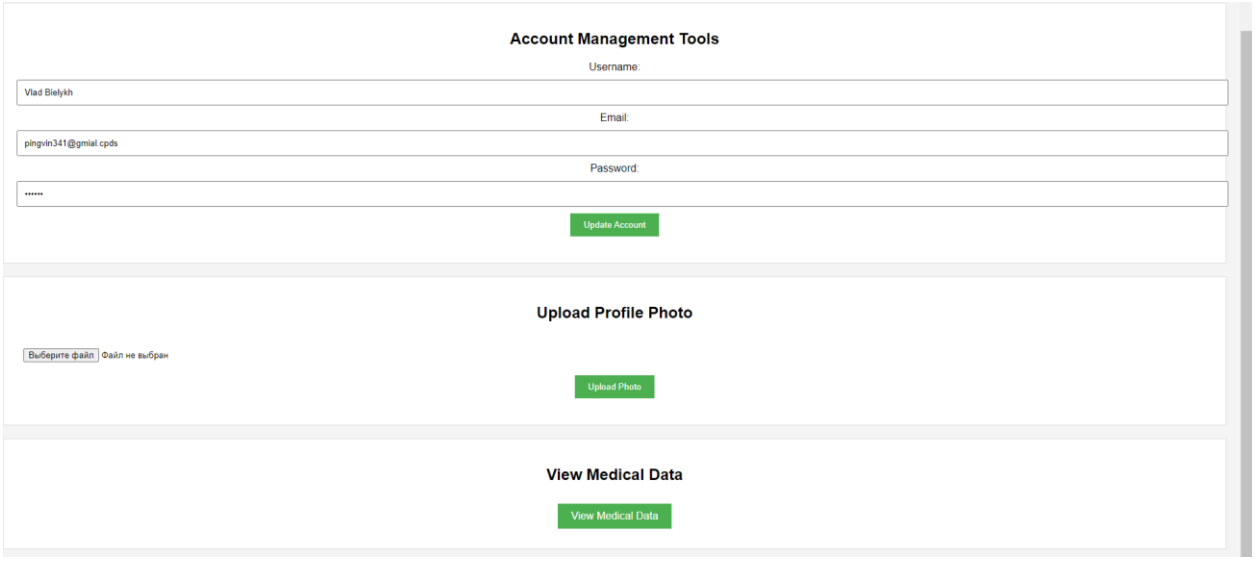


Рис. 4.4 Інтерфейс налаштування профілю

На рис. 4.5 вказана частина кода програми де завантажуються всі сторінки:

```python

@app.route( rule: '/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        return redirect(url_for('home'))
    return render_template('login.html')

@app.route('/medical_data')
def medical_data():
    user_data = {
        "uploaded_data": ["Завантажений файл 1", "Завантажений файл 2"],
        "medical_history": ["Історія захворювання 1", "Історія захворювання 2"],
        "doctors": ["Лікар 1", "Лікар 2", "Лікар 3"]
    }
    return render_template( template_name_or_list: 'medical_data.html', user_data=user_data)

@app.route( rule: '/profile', methods=['GET', 'POST'])
def profile():
    if request.method == 'POST':
        profile_photo = request.files['profile_photo']
        return redirect(url_for('profile'))
    return render_template('profile.html')

if __name__ == '__main__':
    app.run(debug=True)
```

Рис. 4.5 Частина python коду

Сайт був написаний мовою програмування python бібліотекою flask, в той час як дизайн був зроблений на html та спеціальною мовою стилю сторінок css.

Html файл головної сторінки сайту (рис. 4.6):

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Heart Health Monitoring System</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
    <header>
        <h1>Welcome to Heart Health Monitoring System</h1>
    </header>
    <nav>
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">Profile</a></li>
            <li><a href="#">Monitoring</a></li>
            <li><a href="#">Reports</a></li>
            <li><a href="#">Settings</a></li>
        </ul>
    </nav>
    <main>
        <section>
            <h2>About the System</h2>
            <p>This system helps in monitoring heart health, providing personalized insights and recommendations.</p>
        </section>
        <section>
            <h2>Features</h2>
            <ul>
                <li>Real-time monitoring</li>
                <li>Personalized recommendations</li>
                <li>Alert system</li>
                <li>Data analytics</li>
```

Рис. 4.6 html файл головної сторінки сайту

4.3 Інтеграція з мобільними пристроями та смарт-гаджетами

Одним із важливих елементів розробки інформаційної системи моніторингу здоров’я серця людини є забезпечення інтеграції з широким спектром мобільних пристроїв та носимих гаджетів. Актуальність цієї задачі обумовлена тим, що дані, отримані від носимих пристроїв (наприклад, фітнес-браслетів, смарт-годинників, медичних сенсорів), є ключовими для моніторингу фізіологічних показників користувачів у реальному часі. Водночас інтеграція з такими пристроями на практиці виявляється надзвичайно складною з управлінської точки зору, оскільки включає низку технічних, організаційних та ризикових факторів.

Проблема полягає у різноманітті моделей пристроїв, їх протоколів взаємодії (BLE, Wi-Fi, пропріетарні API виробників), різній структурі переданих даних та відсутності єдиного стандарту сумісності. Крім того, постійне оновлення операційних систем та API гаджетів (наприклад, Apple HealthKit, Google Fit) ускладнює підтримку стабільної взаємодії протягом життєвого циклу проєкту. Іншим критичним фактором є потреба забезпечення безпеки медичних даних відповідно до вимог стандартів GDPR і HIPAA, що суттєво обмежує варіанти технічних рішень.

З управлінської перспективи інтеграція з мобільними пристроями у рамках проєкту виступає як окрема підсистема з власними вимогами до планування ресурсів, контролю ризиків, тестування та гнучкості впровадження. Ігнорування цих особливостей призводить до збоїв в інтеграції, затримок у розгортанні системи, збільшення витрат та зниження якості кінцевого продукту.

Таким чином, інтеграція мобільних пристроїв є складною і багатофакторною задачею, яка вимагає окремого підходу до управління в рамках загального проєкту розробки інформаційної системи.

З огляду на складність та динамічність процесу інтеграції мобільних пристроїв у проєкт інформаційної системи моніторингу здоров’я серця, було прийнято рішення застосувати ітеративну модель життєвого циклу розробки

програмного забезпечення (SDLC). Такий підхід дозволяє здійснювати поетапне планування, розробку, інтеграцію та тестування кожного типу пристроїв окремо, що забезпечує гнучкість, контроль ризиків і швидке виявлення помилок на ранніх стадіях.

Ітеративний підхід базується на циклічній структурі процесу: після кожної ітерації (яка охоплює інтеграцію певного класу пристроїв) відбувається оцінка досягнутих результатів, виявлення недоліків і внесення змін до плану наступної ітерації. Це дозволяє враховувати змінні умови — наприклад, оновлення протоколів гаджетів або зміни вимог безпеки — без необхідності масштабної переробки всієї системи (рис. 4.7).

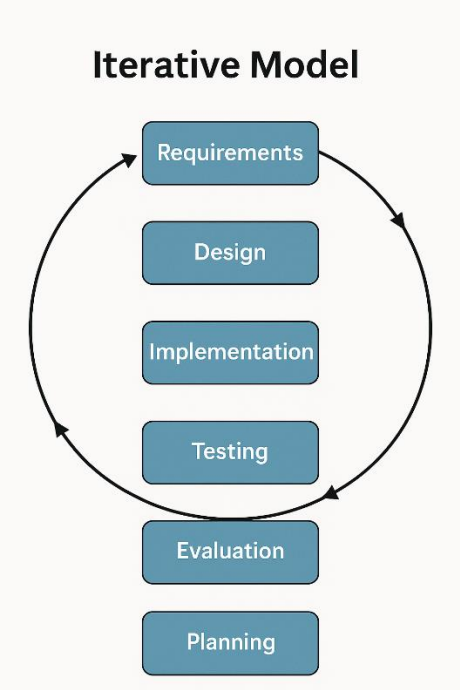


Рис. 4.7 Ітеративна модель SDLC

На етапі планування інтеграції система пристроїв була класифікована за категоріями (фітнес-браслети, смарт-годинники, медичні сенсори), а кожна категорія відповідала окремій ітерації процесу розробки. Кожна ітерація включала детальне вивчення API пристроїв, проектування механізму обробки даних, реалізацію підключення, тестування та впровадження у загальну систему.

Такий підхід дозволяє мінімізувати ризики накопичення критичних помилок, забезпечує гнучкість графіка розробки, дозволяє швидко інтегрувати нові пристрої у майбутньому та підвищує загальну якість інтеграції. Крім того, поділ інтеграції на автономні ітерації сприяє ефективнішому розподілу ресурсів і спрощує контроль за виконанням завдань у рамках загального плану проєкту.

Таким чином, застосування ітеративної моделі SDLC стало обґрунтованим і ефективним рішенням для організації процесу інтеграції мобільних пристроїв та смарт-гаджетів у інформаційну систему, дозволяючи гнучко адаптувати проєкт до технічних викликів і мінімізувати проєктні ризики.

Етап 1. Планування інтеграції пристроїв

На етапі планування було проведено класифікацію пристроїв за категоріями:

- фітнес-браслети (MiBand, Fitbit),
- смарт-годинники (Apple Watch, Samsung Watch),
- медичні сенсори (пульсоксиметри, портативні ЕКГ-пристрої).

Було сформульовано вимоги до протоколів обміну даними (Bluetooth Low Energy, Wi-Fi Direct, API HealthKit/Google Fit), встановлено пріоритети за критеріями поширеності пристроїв та простоти інтеграції. Паралельно оцінено ризики нестабільності з’єднань та проблеми відповідності стандартам безпеки.

Етап 2. Збір та аналіз технічних вимог

На другому етапі для кожної категорії пристроїв було зібрано технічну документацію:

- специфікації API,
- вимоги до BLE-підключення,
- формати обміну даними,
- вимоги до безпеки аутентифікації.

Особливу увагу приділяли підтримці шифрування даних, способам валідації отриманої інформації та можливостям пристроїв щодо надання історичних даних.

Етап 3. Проєктування архітектури інтеграції

На основі зібраних вимог було спроєктовано загальну архітектуру інтеграції:

- Локальна взаємодія через BLE для прямих сенсорів.
- Синхронізація агрегованих даних через API смартфонів для платформ HealthKit/Google Fit.
- Валідація даних за схемами JSON та контроль автентичності пристроїв.
- Організація проміжної буферизації даних у випадку втрати з'єднання з мережею.

Проєктування передбачало мінімізацію залежності від конкретних виробників через використання адаптерів взаємодії.

Етап 4. Реалізація інтеграції пристроїв

Реалізація відбувалася ітераційно: підключення одного класу пристроїв за цикл. Для кожного типу гаджетів здійснювалось:

- розроблення драйверів взаємодії,
- налаштування обробки форматів даних,
- впровадження механізмів перевірки достовірності отриманої інформації,
- шифрування переданих даних за протоколами TLS або BLE Secure Connections.

На завершення кожної ітерації відбувалося локальне тестування інтегрованого пристрою в ізольованому середовищі.

Етап 5. Тестування інтеграції пристроїв

Після розробки кожного адаптера для конкретного типу пристрою проводилося комплексне тестування інтеграції. Тестування включало такі напрямки:

- **Функціональне тестування:** перевірка правильності отримання, обробки та передачі даних системою для кожного типу пристроїв. Перевірялася відповідність даних встановленим форматам JSON або структурі агрегованої інформації HealthKit/Google Fit.

- **Тестування стійкості з’єднання:** моделювались сценарії нестабільної мережі та розриву Bluetooth-з’єднання для оцінки здатності системи до відновлення передачі даних та коректного кешування.
- **Безпекове тестування:** здійснювалася перевірка правильності шифрування каналів передачі даних, перевірка автентифікації пристроїв, тестування стійкості до атак типу "man-in-the-middle".
- **Навантажувальне тестування:** перевірка роботи системи при одночасному підключенні декількох пристроїв та моделювання ситуацій з великою кількістю одночасних обмінів даними.

Результати тестування документувалися у вигляді звітів про ітераційні дефекти та усувалися до переходу до наступного етапу.

Етап 6. Впровадження інтеграційних рішень і подальша підтримка

Після успішного тестування інтегровані пристрої впроваджувалися у середовище тестової експлуатації всієї системи моніторингу здоров’я серця. На цьому етапі здійснювалася інтеграція модулів збору даних у центральну інформаційну архітектуру системи.

Кожна ітерація завершувалася аналізом продуктивності взаємодії, збіром зворотного зв’язку та за потреби — корекцією механізмів обробки даних або протоколів підключення. Для забезпечення стійкої підтримки інтеграції розроблялися політики регулярної перевірки сумісності з оновленнями пристроїв та API.

У межах плану підтримки також було передбачено періодичні оновлення драйверів взаємодії та впровадження механізмів автоматичної адаптації до змін у специфікаціях протоколів пристроїв.

РОЗДІЛ 5. ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ

5.1 Тестування працездатності та продуктивності

Тестування інформаційної системи моніторингу серцевої діяльності є критично важливим етапом, що дозволяє оцінити її функціональність, надійність, продуктивність та відповідність встановленим вимогам. Оскільки система призначена для роботи з персональними фізіологічними даними, її стабільність і точність є ключовими характеристиками якості.

У рамках тестування було використано принципи, визначені у стандарті ISTQB (International Software Testing Qualifications Board). Випробування системи проводилося відповідно до розроблених сценаріїв тестування, які охоплювали як базову функціональність, так і специфічні ситуації, що можуть виникати при реальній експлуатації.

Відповідно до стандартів ISTQB, тестування поділяється на функціональне та нефункціональне, кожне з яких включає окремі типи тестів, обґрунтовані специфікою проєкту.

Функціональне тестування

- Тестування функціональності інтерфейсу користувача:

У нашій системі кінцевим користувачем є не тільки технічно підготовлений персонал, але й звичайні пацієнти, серед яких можливі літні люди або особи з обмеженими технічними навичками. Надійність роботи інтерфейсу є критичною: некоректне відображення інформації або складність навігації можуть призвести до того, що користувач не помітить попередження про критичний стан здоров'я. Тому ми обов’язково тестуємо усі сценарії роботи з інтерфейсом, включно з реєстрацією, переглядом результатів, отриманням сповіщень.

- Тестування обробки даних та сповіщень:

Зважаючи на те, що система збирає медичні показники в режимі реального часу, від правильності обробки цих даних залежить здоров'я і навіть життя користувачів. Тестування правильності обробки даних дозволяє виявити та

усунути помилки в логіці аналізу та формування сповіщень, гарантувати оперативність і точність реакції системи на критичні відхилення.

- Тестування авторизації та доступу:

Оскільки ми працюємо із надчутливою інформацією (медичними даними), відповідність вимогам GDPR і HIPAA є обов'язковою. Тестування автентифікації та авторизації необхідне для запобігання несанкціонованому доступу до персональних медичних даних пацієнтів і лікарів. Недостатній рівень контролю доступу може призвести не тільки до юридичних санкцій, але й до серйозних репутаційних втрат.

- Тестування інтеграції з пристроями:

У нашій системі важливо, щоб мобільні застосунки безпомилково приймали дані від носимих пристроїв (фітнес-браслетів, годинників, ЕКГ-сенсорів). Помилки інтеграції можуть призвести до втрати даних або до їх некоректного зчитування, що поставить під сумнів увесь процес моніторингу. Саме тому обов'язковим є перевірка стабільності обміну даними через Bluetooth/Wi-Fi на різних пристроях.

Нефункціональне тестування

- Тестування продуктивності (Performance Testing)

У разі виявлення відхилень у серцевій діяльності, система повинна оперативно повідомити про це як користувача, так і медичного працівника. Якщо час реакції системи перевищить допустимі значення, це може спричинити затримку у наданні медичної допомоги, що прямо впливає на безпеку пацієнтів. Тому обов'язково перевіряємо час відгуку системи на запити, час обробки даних, час доставки сповіщень.

- Тестування навантаження (Load Testing)

Проєкт передбачає роботу одночасно з великою кількістю користувачів, особливо якщо система буде інтегрована в клініки чи страхові компанії. Система повинна справлятися з тисячами запитів і оновлень даних у реальному часі. Тестування навантаження дозволяє оцінити, чи не призведе збільшення кількості користувачів до падіння продуктивності або до збоїв у роботі системи.

- Тестування стабільності та надійності (Stability and Reliability Testing)

Моніторинг здоров’я потребує постійної роботи системи без перерв, оскільки критичні стани серця можуть виникнути будь-якої миті. Якщо система нестабільна і періодично виходить з ладу, це може стати причиною втрати важливих медичних даних або пропуску тривожних сигналів. Стабільність роботи протягом тривалого часу під навантаженням є обов'язковою умовою.

- Тестування безпеки (Security Testing)

У сфері охорони здоров’я витік персональних даних або їх спотворення може мати катастрофічні наслідки — від шкоди репутації до юридичної відповідальності за порушення GDPR і HIPAA. Безпечність системи моніториться на рівні шифрування даних, автентифікації користувачів, захисту від типових атак (SQL-ін’єкцій, XSS, CSRF). Безпека є базовою вимогою для будь-якої системи, що працює з медичними даними.

- Тестування зручності використання (Usability Testing)

Ефективність медичної системи безпосередньо залежить від того, наскільки швидко і безпомилково користувач може взаємодіяти з нею. Особливо це важливо для літніх людей або пацієнтів у стресових станах. Тестування юзабіліті дозволяє зробити інтерфейс простим, інтуїтивно зрозумілим і зменшити ймовірність помилок через неправильне використання.

- Тестування сумісності (Compatibility Testing)

Оскільки користувачі можуть використовувати різні пристрої (Android, iOS, різні бренди та моделі телефонів), важливо забезпечити коректну роботу застосунку на різних платформах і пристроях. Несумісність може призвести до того, що частина аудиторії не зможе використовувати продукт, що суперечить цілям масового впровадження системи.

У рамках підготовки тестування було розроблено набір базових тест-кейсів, що охоплюють ключові функціональні та нефункціональні аспекти роботи інформаційної системи для моніторингу здоров'я серця людини. Тест-кейси структуровані відповідно до стандартів тестування та містять докладний

опис дій, очікуваних результатів для кожного кроку, що дозволяє забезпечити повноцінну перевірку працездатності, надійності та стабільності системи.

Тест-кейс 1: Перевірка реєстрації нового користувача.

Перевірка можливості реєстрації нового користувача в системі через мобільний застосунок (табл. 5.1).

Таблиця 5.1

Тест-кейс 1. Перевірка реєстрації нового користувача

1. Відкрити мобільний застосунок.	1. Застосунок завантажується, відображається екран входу.
2. Натиснути кнопку "Зареєструватися".	2. Відкривається форма реєстрації.
3. Заповнити обов'язкові поля (ім'я, e-mail, пароль).	3. Дані вводяться без помилок, поля валідуються (наприклад, формат e-mail правильний).
4. Натиснути "Підтвердити реєстрацію".	4. Після успішної реєстрації система відображає повідомлення "Реєстрація успішна" і переадресовує користувача на головний екран системи.

Тест-кейс 2: Перевірка авторизації користувача з багатофакторною автентифікацією.

Перевірити процес входу до системи із застосуванням двофакторної автентифікації (табл. 5.2).

Таблиця 5.2

Тест-кейс 2. Перевірка авторизації користувача з багатофакторною автентифікацією

1. Відкрити мобільний застосунок.	1. Відображається екран входу.
2. Ввести e-mail і пароль користувача.	2. Система приймає дані, перевіряє їх правильність і переходить до наступного кроку.
3. Ввести код, надісланий у SMS або на e-mail.	3. Якщо код правильний, користувач отримує доступ до системи та переходить на головний екран. Якщо код неправильний — виводиться повідомлення про помилку.

Тест-кейс 3: Перевірка надходження даних від носимого пристрою.

Перевірка надходження та обробки даних серцевого ритму від підключеного пристрою до мобільного застосунку (табл. 5.3).

Таблиця 5.3

Тест-кейс 3. Перевірка надходження даних від носимого пристрою

1. Увімкнути носимий пристрій (наприклад, фітнес-браслет або ЕКГ-сенсор).	1. Пристрій запускається, індикатор роботи активний.
2. Відкрити мобільний застосунок і перейти до розділу "Моніторинг".	2. Відображається поточний статус підключення пристрою (Connected).
3. Дочекатися перших даних серцевого ритму.	3. На екрані з’являються числові значення пульсу або ЕКГ-графік, що оновлюються кожні кілька секунд.
4. Перевірити безперервність надходження даних протягом 5 хвилин.	4. Жодних розривів з’єднання або втрати даних не спостерігається.

Тест-кейс 4: Перевірка формування сповіщення про критичний стан.

Перевірка формування сповіщення при фіксації критичного показника серцевого ритму (табл. 5.4).

Таблиця 5.4

Тест-кейс 4. Перевірка формування сповіщення про критичний стан

1. Підключити носимий пристрій до застосунку.	1. Відображається статус "Пристрій підключено".
2. Симулювати критичне значення серцевого ритму (вище 180 ударів за хвилину або нижче 40).	2. Система фіксує вихід показників за межі нормальних значень.
3. Очікувати сповіщення на телефоні.	3. Користувач отримує push-повідомлення з текстом "Увага: виявлені критичні показники серцевого ритму".
4. Перевірити надсилання даних медичному працівнику.	4. У базі даних формується запис про критичну подію, медичному працівнику надсилається повідомлення через API.

Тест-кейс 5: Load testing.

Перевірка роботи при 1000 одночасно підключених користувачів (табл. 5.5).

Таблиця 5.5

Тест-кейс 5. Load testing

1. Запустити емуляцію підключення 1000 віртуальних користувачів до серверу.	1. Сервер приймає усі з'єднання без помилок або зависань.
2. Налаштувати надсилання даних від кожного користувача кожні 5 секунд.	2. Сервер приймає дані, час відповіді не перевищує 1-2 секунди при піковому навантаженні.
3. Вести моніторинг роботи серверної бази даних і API.	3. Відсутність помилок 5xx, кількість оброблених запитів відповідає кількості надісланих запитів.
4. Завершити навантаження та перевірити журнали помилок.	4. У логах відсутні критичні помилки або падіння служб. Система продовжує працювати стабільно.

Проведена розробка тест-кейсів дозволяє здійснити комплексне тестування ключових функцій інформаційної системи моніторингу здоров’я серця людини. Запропоновані сценарії охоплюють основні бізнес-процеси, критичні для безпеки, стабільності та ефективності роботи системи, включаючи перевірку реєстрації користувачів, обробку даних від носимих пристроїв, формування сповіщень про критичні стани, а також оцінку поведінки системи під високим навантаженням.

На основі виконаних тестів буде проведено попередню оцінку якості програмного забезпечення, що дозволить виявити можливі недоліки на ранніх етапах розробки та своєчасно впровадити необхідні поліпшення перед введенням системи в експлуатацію.

У процесі організації тестування всі тести зазвичай оформлюються у вигляді STS-документів (Steps to reproduce). Такий формат відповідає вимогам

до формалізованого опису сценаріїв тестування, встановленим міжнародними стандартами якості програмного забезпечення (наприклад, IEEE 829).

Кожен STS-документ містить структурований опис тесту:

- Унікальний ідентифікатор тест-кейсу
- Назву тест кейсу
- Короткий опис мети тесту
- Передумови виконання
- Докладні кроки тестування
- Очікувані результати для кожного кроку

Для забезпечення відповідності встановленим стандартам, усі розроблені тест-кейси в межах цієї роботи були оброблені та представлені у форматі, що відповідає принципам STS-документування (рис. 5.1). Такий підхід дозволяє не лише підвищити якість тестування, але й значно спрощує подальший аналіз результатів, управління тестовим процесом та забезпечення простоти відтворення тестів у майбутньому.

ID	TC_NAME	DESCRIPTION	PRECONDITIONS
TC-01	Реєстрація нового користувача	Перевірка можливості реєстрації нового користувача в системі через мобільний застосунок.	Встановлений мобільний застосунок, доступ до інтернету.
TC-02	Авторизація користувача з 2FA	Перевірити процес входу до системи із застосуванням двофакторної автентифікації.	Створений обліковий запис, підключено двофакторну автентифікацію через SMS або email.
TC-03	Надходження даних з носимого пристрою	Перевірка надходження та обробки даних серцевого ритму від підключеного пристрою до мобільного застосунку.	Пристрій синхронізований із застосунком, Bluetooth увімкнено.
TC-04	Сповіщення про критичний стан	Перевірка формування сповіщення при фіксації критичного показника серцевого ритму.	Система налаштована на оповіщення при перевищенні встановлених порогових значень.
TC-05	Тест навантаження: 1000 одночасних користувачів	Оцінка стабільності серверної частини при одночасному підключенні великої кількості користувачів.	Налаштоване тестове середовище навантаження (емуляція пристроїв, скрипти генерації запитів).

Рис. 5.1. STS-документ

У кожному STS-документі мають бути очікувані результати для кожного виконаного кроку (рис. 5.2). При тестування ми маємо бути впевнені, що поведінка програми відповідає очікуванню.

Steps to reproduce	Expected results
1. Відкрити мобільний застосунок. 2. Натиснути кнопку "Зареєструватися". 3. Заповнити обов'язкові поля (ім'я, e-mail, пароль). 4. Натиснути "Підтвердити реєстрацію".	1. Застосунок завантажується, відображається екран входу. 2. Відкривається форма реєстрації. 3. Дані вводяться без помилок, поля валідуються (наприклад, формат e-mail правильний). 4. Після успішної реєстрації система відображає повідомлення "Реєстрація успішна" і переадресовує користувача на головний екран системи.
1. Відкрити мобільний застосунок. 2. Ввести e-mail і пароль користувача. 3. Ввести код, надісланий у SMS або на e-mail.	1. Відображається екран входу. 2. Система приймає дані, перевіряє їх правильність і переходить до наступного кроку. 3. Якщо код правильний, користувач отримує доступ до системи та переходить на головний екран. Якщо код неправильний — виводиться повідомлення про помилку.
1. Увімкнути носимий пристрій (наприклад, фітнес-браслет або ЕКГ-сенсор). 2. Відкрити мобільний застосунок і перейти до розділу "Моніторинг". 3. Дочекатися перших даних серцевого ритму. 4. Перевірити безперервність надходження даних протягом 5 хвилин.	1. Пристрій запускається, індикатор роботи активний. 2. Відображається поточний статус підключення пристрою (Connected). 3. На екрані з'являються числові значення пульсу або ЕКГ-графік, що оновлюються кожні кілька секунд. 4. Жодних розривів з'єднання або втрати даних не спостерігається.
1. Підключити носимий пристрій до застосунку. 2. Симулювати критичне значення серцевого ритму (вище 180 ударів за хвилину або нижче 40). 3. Очікувати сповіщення на телефоні. 4. Перевірити надсилання даних медичному працівнику.	1. Відображається статус "Пристрій підключено". 2. Система фіксує вихід показників за межі нормальних значень. 3. Користувач отримує push-повідомлення з текстом "Увага: виявлені критичні показники серцевого ритму". 4. У базі даних формується запис про критичну подію, медичному працівнику надсилається повідомлення через API.
1. Запустити емуляцію підключення 1000 віртуальних користувачів до серверу. 2. Налаштувати надсилання даних від кожного користувача кожні 5 секунд. 3. Вести моніторинг роботи серверної бази даних і API. 4. Завершити навантаження та перевірити журнали помилок.	1. Сервер приймає усі з'єднання без помилок або зависань. 2. Сервер приймає дані, час відповіді не перевищує 1-2 секунди при піковому навантаженні. 3. Відсутність помилок 5xx, кількість оброблених запитів відповідає кількості надісланих запитів. 4. У логах відсутні критичні помилки або падіння служб. Система продовжує працювати стабільно.

Рис. 5.2. Очікувані результати кроків тестування

Наведена таблиця дозволяє систематизувати процес тестування і забезпечити узгодженість між функціональними вимогами та фактичною поведінкою програмного забезпечення. Завдяки чіткому формулюванню очікуваних результатів, тестувальники можуть оперативно виявляти відхилення, оцінювати якість реалізації функціоналу та підтверджувати готовність системи до впровадження у виробниче середовище. Такий підхід значно підвищує надійність та передбачуваність роботи всієї інформаційної системи.

5.2 Створення Тест Плану

Розробка тест-плану є важливим етапом забезпечення якості програмного продукту. Тест-план визначає загальну стратегію тестування, обсяги тестування, ресурси, строки виконання тестів, а також критерії прийняття або відхилення результатів. Для проєкту створення інформаційної системи моніторингу здоров'я серця тест-план містить такі основні розділи:

1. Ідентифікатор тест-плану (Test Plan ID)

Кожен тест-план повинен мати унікальний ідентифікатор для відстеження змін та контролю версій документації. В нашому випадку ідентифікатор формується за схемою:

TP-HMS-01 — де TP означає "Test Plan", HMS — "Heart Monitoring System", 01 — перша версія документа.

Наявність унікального ідентифікатора дозволяє швидко знайти потрібний тест-план серед кількох документів проєкту, особливо якщо система з часом розширюватиметься.

2. Розділ "Вступ" пояснює контекст тестування та його мету у проєкті.

У нашому випадку, тестування інформаційної системи для моніторингу здоров’я серця є критично важливим, оскільки система обробляє медичні дані, життєво важливі для пацієнтів. Мета тестування — підтвердити, що система працює відповідно до визначених функціональних і нефункціональних вимог, забезпечує безпеку, стабільність і є зручною для використання як пацієнтами, так і медичними працівниками.

Також розділ встановлює зв'язок тестування із загальним процесом розробки та випуску системи.

3. Об'єкти тестування

Вказуються частини продукту, які підлягають перевірці:

- Мобільний застосунок користувача — інтерфейс реєстрації, перегляду даних, отримання сповіщень.

- Серверна частина системи — обробка даних сенсорів, логіка прийняття рішень, API взаємодії.
- База даних — зберігання особистої і медичної інформації відповідно до GDPR та HIPAA.
- Модуль інтеграції з носимими пристроями — обробка даних з ЕКГ-браслетів, смарт-годинників.
- Система генерації критичних сповіщень — своєчасне повідомлення користувача та медичного персоналу.

4. Функції, які будуть протестовані

Включають перелік основних можливостей системи, які будуть піддані тестуванню:

- Реєстрація нового користувача з перевіркою валідації даних.
- Вхід у систему з використанням багатофакторної автентифікації.
- Додавання, синхронізація та обробка даних від сенсорних пристроїв.
- Виявлення критичних змін показників серцевої діяльності.
- Формування історії спостережень і графіків показників.
- Передача даних до електронної медичної картки.
- Генерація повідомлень та тривог в реальному часі.

5. Функції, які не будуть протестовані

Перелік функцій, тестування яких виходить за рамки даного тест-плану:

- Оновлення мобільного застосунку через офіційні магазини додатків (Google Play, App Store).
- Інтеграція зі сторонніми страхувальними компаніями або CRM-системами.
- Робота застосунку в екзотичних мовах, окрім української та англійської.
- Ці функції будуть протестовані окремо в рамках майбутніх етапів розвитку продукту.

6. Типи тестування

Для забезпечення повного охоплення тестами ми застосуємо такі види:

- Функціональне тестування — перевірка, чи реалізовані функції працюють відповідно до вимог.
- Тестування продуктивності — вимір часу обробки запитів та реакції системи.
- Тестування навантаження — оцінка поведінки системи при високому одночасному навантаженні.
- Тестування безпеки — перевірка захисту даних і стійкості до атак.
- Тестування сумісності — перевірка роботи на різних платформах і пристроях.
- Тестування стабільності — аналіз безперервної роботи системи протягом тривалого часу.
- Тестування юзабіліті — оцінка зручності інтерфейсу та досвіду користувача.

7. Критерії прийняття та відхилення результатів

Критерії, за якими визначається успішність тестування:

- 95% всіх тест-кейсів повинні бути пройдені успішно.
- Усі критичні та високі дефекти повинні бути усунені і перевірені.
- Система не повинна мати витоків пам'яті після 72 годин безперервної роботи.
- Час відповіді критичних запитів повинен бути меншим за 1 секунду.

8. Критерії зупинки та відновлення тестування

Критерії зупинки:

- Виявлення критичного дефекту, що блокує основну функціональність.
- Виникнення технічних проблем у тестовому середовищі.
- Порушення строків розробки, що унеможливлює продовження тестування.

Критерії відновлення:

- Виправлення всіх критичних дефектів.
- Відновлення нормальної роботи тестового середовища.

- Підтвердження готовності розробників до повторного тестування.

9. Методологія тестування

Методологія тестування буде комбінованою:

- Ручне тестування основних сценаріїв використання користувачами.
- Автоматизоване тестування стабільних функцій для підвищення ефективності.
- Інтеграційне тестування API для забезпечення коректної взаємодії модулів.
- Регресійне тестування після кожного оновлення для перевірки вже протестованого функціоналу.

Тестування буде інтегрованим у процес розробки за принципами Agile (тестування після кожного спринту).

10. Інструменти тестування

Планується використання таких інструментів:

- Postman — для тестування REST API.
- JMeter — для тестування продуктивності та навантаження.
- OWASP ZAP — для перевірки безпеки веб-сервісів.
- Appium — для автоматизації тестування мобільного застосунку.
- TestRail / Excel — для управління тест-кейсами і формування тест-звітів.

11. Потреби тестового середовища (Environmental Needs)

Необхідне середовище для проведення тестування включає:

- Мобільні пристрої: смартфони на Android (10+) і iOS (14+).
- Носимі пристрої: смарт-годинники та ЕКГ-браслети.
- Сервер із встановленою серверною частиною (Docker-контейнери).
- Наявність тестового середовища бази даних PostgreSQL.
- Безпечне Wi-Fi середовище для імітації передачі медичних даних.
- Інструменти моніторингу продуктивності та логування системних подій.

12. План тестових активностей і розподіл ресурсів

План активностей:

- Підготовка тест-кейсів та сценаріїв тестування — 2 тижні.
- Виконання ручного і автоматизованого тестування — 4 тижні.
- Регресійне тестування після виправлень — 2 тижні.
- Формування фінального тестового звіту — 1 тиждень.

Ресурсний план:

- 1 QA Lead.
- 2 Manual QA Engineers.
- 1 Automation QA Engineer.

13. Графік тестування

Тестування відбуватиметься за наступним графіком:

- Спринт 1-2: Функціональне тестування базових сценаріїв.
- Спринт 3: Тестування інтеграцій з пристроями.
- Спринт 4: Нефункціональне тестування (навантаження, продуктивність, безпека).
- Фінальні 2 тижні: Повна регресія та підготовка тест-звіту.

14. Ризики та стратегії мінімізації

Основні ризики:

- Відсутність деяких моделей пристроїв для тестування → Використання емуляторів і партнерських лабораторій.
- Висока динаміка змін у системі → Постійне оновлення тест-кейсів та автоматизація основних перевірок.
- Нестабільність тестового середовища → Використання ізольованих тестових стендів.

15. Результати тестування

Очікувані результати:

- Підсумковий звіт про проходження тест-кейсів.
- Перелік знайдених дефектів із їх пріоритезацією.
- Оцінка рівня готовності системи до продуктивного середовища.
- Рекомендації щодо подальшого удосконалення якості системи.

Для забезпечення високої якості розробленої інформаційної системи моніторингу здоров’я серця був розроблений приблизний тест-план, що визначає основні напрями та підходи до тестування. У тест-плані стисло розглянуті всі ключові аспекти тестового процесу, включаючи об'єкти тестування, функції, що підлягають перевірці, типи тестування, критерії успішності та відмови, вимоги до тестового середовища, а також можливі ризики і стратегії їх мінімізації. Розроблений тест-план є базовим документом для організації, контролю і аналізу якості тестування системи на всіх етапах її розробки та впровадження.

5.3 Організація контролю якості продукту на етапі тестування

Тестування інформаційної системи виступає одним із ключових етапів процесу управління проєктом, спрямованого на забезпечення відповідності розробленого продукту встановленим вимогам якості, функціональності та надійності. У контексті управління проєктами тестування розглядається не лише як технічна процедура перевірки працездатності, а насамперед як інструмент забезпечення контролю за виконанням проєкту відповідно до цілей, визначених на початкових етапах планування.

Одним із основних принципів ефективного управління проєктами є регулярна верифікація та валідація продукту на кожному з його життєвих циклів. Тестування реалізує ці процеси, забезпечуючи об'єктивну оцінку поточного стану продукту відносно початкових специфікацій. Завдяки своєчасному тестуванню виявляються відхилення від запланованих характеристик, що дає можливість оперативно вносити корективи у процес розробки, мінімізуючи ризики невідповідності кінцевого результату очікуванням замовника.

У рамках розробки інформаційної системи моніторингу здоров’я серця тестування має особливо високу вагу через специфіку продукту. Оскільки система обробляє медичні дані пацієнтів та забезпечує критичні функції сповіщення про стан здоров’я, недоліки в її роботі можуть мати серйозні наслідки. Відповідно, тестування в даному проєкті набуває характеру не тільки

перевірки функціональності, а й гарантування безпеки, точності та своєчасності реакції системи на зміни життєво важливих показників.

У проєктному менеджменті тестування інтегрується в більш широкі процеси управління якістю, такі як:

- Планування якості — встановлення стандартів тестування та критеріїв прийняття продукту;
- Забезпечення якості — моніторинг відповідності процесів розробки встановленим стандартам;
- Контроль якості — безпосереднє тестування продукту та оцінка отриманих результатів.

Таким чином, тестування виконує у проєкті роль основного інструмента підтвердження того, що:

- Розробка системи йде відповідно до технічного завдання;
- Функціональні та нефункціональні вимоги реалізовані правильно;
- Система здатна безпечно і стабільно працювати в реальних умовах експлуатації.

Тестування також є важливою частиною механізму прийняття рішення щодо переходу проєкту на наступні етапи — наприклад, від розробки до впровадження, або від внутрішнього тестування до бета-тестування з реальними користувачами. У випадку виявлення критичних дефектів саме результати тестування стають підставою для коригування плану проєкту або перенесення строків релізу.

Інтеграція тест-плану у загальний план управління проєктом

Ефективна організація тестування в проєкті неможлива без його чіткої інтеграції в загальну систему планування робіт. Успішне управління проєктом передбачає включення тест-плану до основного плану проєкту як окремої підфази забезпечення якості, з чітко визначеними цілями, термінами, ресурсами та відповідальністю.

Тест-план визначає стратегію проведення тестування: обсяг тестових робіт, підходи до тестування, критерії успішності, критерії виходу і входу на етапи тестування. Він тісно пов'язаний з іншими планами проєкту:

- Планом розробки — для визначення, на якій стадії якого функціонального блоку проводиться тестування.
- Планом управління ризиками — для оцінки впливу можливих дефектів на строки і вартість проєкту.
- Планом забезпечення якості — для визначення стандартів та методів оцінки якості продукту.

У процесі планування тест-план має бути інтегрований з графіком проєкту за допомогою встановлення контрольних точок (milestones), наприклад:

- Завершення створення тест-кейсів;
- Завершення внутрішнього функціонального тестування;
- Завершення навантажувального тестування;
- Готовність до тестування приймання (acceptance testing).

Відсутність інтеграції тест-плану в загальний план проєкту може призвести до неправильного розподілу ресурсів, затримок у виявленні критичних дефектів і, як наслідок, до порушення строків завершення проєкту. Тому у межах управління проєктом тестування розглядається як обов'язковий етап, що вимагає свого планування, координації та контролю на рівні всього проєктного процесу.

Процедури організації тестування

Організація тестування в рамках проєкту передбачає не лише розробку тест-плану, але й чітке визначення процедур підготовки, виконання і аналізу результатів тестування. Процедури тестування повинні бути стандартизованими, щоб забезпечити повторюваність процесів та об’єктивність оцінки результатів.

Основні етапи організації тестування включають:

- Формування вимог до тестування: Визначення того, які функціональні та нефункціональні вимоги проєкту мають бути перевірені.

- Розробка тест-кейсів: Створення набору тестових сценаріїв, що покривають як основні, так і граничні випадки використання системи.
- Підготовка тестового середовища: Налаштування апаратного та програмного забезпечення для проведення тестування в умовах, наближених до реальних.
- Розподіл ролей і відповідальностей: Призначення членів команди, відповідальних за розробку тестів, проведення тестування, аналіз дефектів та формування звітів.
- Виконання тестування: Проведення тестів відповідно до розроблених тест-кейсів із фіксацією результатів.
- Аналіз результатів тестування: Оцінка виявлених дефектів, їх пріоритезація за критичністю, визначення необхідних дій для їх усунення.
- Регресійне тестування: Повторна перевірка функціональності після внесення змін до коду або системи.

Процедури тестування повинні бути формалізованими у вигляді документації (Steps to reproduce) і узгоджені з усіма зацікавленими сторонами проєкту, включаючи розробників, тестувальників та менеджерів проєкту.

Особлива увага приділяється контролю якості проведеного тестування: регулярно проводяться тест-рев'ю (перевірка повноти тест-кейсів), використовується подвійна верифікація критичних дефектів, формуються звіти про покриття тестуванням вимог системи.

Відповідна організація процедур тестування забезпечує високу прозорість процесу контролю якості в проєкті, дозволяє оперативно виявляти проблеми та коригувати плани проєкту відповідно до реального стану продукту.

Метрики контролю якості на етапі тестування

Для об’єктивної оцінки ефективності тестування та загальної якості інформаційної системи необхідне використання чітко визначених метрик контролю якості. Метрики дозволяють кількісно виміряти результати тестових активностей, порівняти їх із запланованими показниками і виявити потенційні проблеми у процесі розробки та впровадження проєкту.

У рамках даного проєкту запропоновано використовувати наступні основні метрики тестування:

- Рівень проходження тест-кейсів (Test Case Pass Rate)

Для визначення відсотку тест-кейсів, що завершились успішно від загальної кількості виконаних тестів використаємо формулу 5.1:

$$Pass\ Rate = \left(\frac{\text{Кількість успішних тестів}}{\text{Загальна кількість тестів}}\right) * 100\% \quad (5.1)$$

- Щільність дефектів (Defect Density)

Відношення кількості знайдених дефектів до розміру функціональності, що тестувалася (наприклад, на 1000 рядків коду або на одну функціональність). Допомагає виявити найбільш "проблемні" частини системи.

- Середній час виявлення дефекту (Average Defect Detection Time) Час, витрачений на виявлення дефекту від початку тестування функціональності. Високі значення можуть вказувати на складність тестування або погано складені тест-кейси.

- Час відповіді системи (System Response Time) Оцінює продуктивність: середній час, який потрібен системі для обробки користувацьких запитів під стандартним навантаженням. Для критичних запитів він має бути не більше 1 секунди.

- Рівень дефектів високої критичності (Critical Defects Rate) Відсоток дефектів, що класифікуються як критичні (critical severity), відносно загальної кількості знайдених дефектів. Ця метрика дозволяє оцінити рівень ризику використання продукту.

Використання даних метрик дозволяє:

- Здійснювати моніторинг процесу тестування в режимі реального часу.
- Прогнозувати строки завершення тестування.
- Приймати обґрунтовані рішення про готовність системи до впровадження.
- Вчасно виявляти зони підвищеного ризику в системі.

Роль тестувальників та команди управління проєктом

Тестувальники в проєкті виконують не лише технічну функцію перевірки розробленого продукту, але й є важливими учасниками загального процесу управління якістю. Їхня діяльність тісно інтегрована у команду проєкту та підтримує ухвалення стратегічних рішень щодо подальших етапів розробки.

Основні ролі тестувальників у проєкті:

- Розробка тестових сценаріїв: На основі вимог проєкту тестувальники створюють детальні тест-кейси, що забезпечують повне покриття функціональності системи.
- Проведення тестування: Виконання тестів, документування результатів, виявлення і реєстрація дефектів у системах відстеження помилок.
- Аналіз якості продукту: Підготовка регулярних звітів для менеджерів проєкту щодо виявлених проблем, ступеня готовності продукту, рівня ризиків.
- Регресійне тестування: Повторна перевірка системи після виправлення дефектів для гарантування, що нові зміни не призвели до погіршення якості.
- Підтримка приймального тестування (User Acceptance Testing, UAT): Підготовка середовища і допомога кінцевим користувачам у тестуванні системи перед введенням її в експлуатацію.

Роль команди управління проєктом у контексті тестування:

- Планування ресурсів тестування: Менеджери проєкту забезпечують наявність достатньої кількості часу, людей та обладнання для проведення повного циклу тестування.
- Моніторинг прогресу тестування: Регулярне відстеження стану тестування, аналіз метрик якості та своєчасне вжиття коригувальних заходів у разі виявлення проблем.
- Пріоритезація дефектів: Визначення пріоритетів виправлення дефектів на основі їх впливу на функціональність і ризики для проєкту.

- Прийняття рішень на основі результатів тестування: Вирішення питань про готовність системи до випуску, про перенесення строків або необхідність додаткової роботи.

Управління ризиками тестування

Тестування інформаційної системи, як і будь-яка інша стадія проєкту, супроводжується певними ризиками, які можуть впливати на якість кінцевого продукту, строки реалізації та загальні результати проєкту. В межах управління проєктом необхідно своєчасно ідентифікувати потенційні ризики, оцінювати їх можливий вплив і розробляти стратегії для їх мінімізації.

Основні ризики на етапі тестування інформаційної системи:

- Недостатнє покриття тест-кейсами. Ризик того, що частина функціональності не буде перевірена, що може призвести до появи прихованих дефектів у продукті.
- Брак часу на тестування. У випадку затримок на попередніх етапах проєкту тестування може бути скорочене, що впливає на глибину перевірки.
- Неправильне пріоритезування дефектів. Якщо команда невірно оцінює критичність виявлених помилок, можливе впровадження системи із серйозними дефектами.
- Невідповідність тестового середовища реальному. Відмінності між середовищем тестування і реальними умовами експлуатації можуть призвести до того, що в реальній роботі система поводитиметься інакше.
- Обмеженість ресурсів тестування. Недостатня кількість тестувальників, відсутність необхідного обладнання чи обмеження у використанні реальних пристроїв для перевірки.
- Людський фактор. Помилки в розробці тест-кейсів, некоректне виконання тестів або неправильна інтерпретація результатів.

Стратегії мінімізації ризиків тестування:

- Впровадження чітких критеріїв покриття тестами (наприклад, 100% покриття критичних функцій).

- Резервування буферного часу для етапу тестування в загальному плані проєкту.
- Використання інструментів для пріоритезації дефектів за шкалою критичності.
- Організація тестування в середовищах, максимально наближених до реальних умов експлуатації.
- Залучення додаткових ресурсів або використання емуляторів для імітації реальних пристроїв.
- Проведення регулярних аудитів тестових сценаріїв і ретельна перевірка результатів тестування.

У межах розділу було здійснено комплексне опрацювання питань тестування та оцінки ефективності розробленої інформаційної системи для моніторингу здоров’я серця людини. Проведено аналіз необхідних видів тестування, визначено їх доцільність у контексті специфіки продукту, зокрема з урахуванням критичності обробки медичних даних і вимог до стабільності та продуктивності системи.

Розроблений тест-план дозволив системно організувати процес тестування, визначити об’єкти і функції, що підлягають перевірці, критерії успішності проведення тестів, правила управління тестовими активностями та ризиками. Визначено основні метрики контролю якості, що дозволяють об’єктивно оцінити результати тестування та стан готовності продукту до подальших етапів розгортання.

Окрема увага була приділена організації контролю якості продукту в межах загального процесу управління проєктом. Розглянуто, як тестування інтегрується у проектний цикл, сприяє ранньому виявленню відхилень, мінімізації ризиків і забезпеченню відповідності результатів проєкту встановленим вимогам. Описано ключові ролі тестувальників, механізми взаємодії команди тестування з менеджментом проєкту та заходи з управління потенційними ризиками тестового процесу.

ВИСНОВКИ

У роботі здійснено комплексне дослідження процесів управління проєктом розробки персоналізованої інформаційної системи для моніторингу здоров’я серця людини. Було детально проаналізовано ринок сучасних інформаційних систем для кардіомоніторингу, ідентифіковано основні тенденції розвитку та визначено актуальність впровадження таких систем у медичну практику. Проведений SWOT-аналіз дозволив визначити сильні сторони, слабкості, можливості та загрози проєкту, сформувавши чітку базу для стратегічних рішень.

Сформовано чітке технічне завдання, на основі якого розроблено концептуальну модель системи, що забезпечує інтеграцію різних компонентів, включаючи мобільні додатки, носимі пристрої та серверну частину з використанням штучного інтелекту для аналізу медичних даних. Побудовано математичну модель для ефективного аналізу та прогнозування стану серця пацієнта. Окрему увагу приділено проєктуванню логічної та фізичної структур бази даних, яка відповідає вимогам безпеки даних відповідно до міжнародних стандартів GDPR і HIPAA.

Розроблена архітектура програмного забезпечення дозволяє ефективно керувати потоками даних, інтегрувати сучасні методи машинного навчання та забезпечує стабільну роботу на різних програмних і апаратних платформах. У рамках управління проєктом було використано сучасні методики: сформовано ієрархічну структуру робіт (WBS), створено детальні діаграми Ганта, здійснено аналіз ризиків та проведено бюджетування, що забезпечило ефективність управління ресурсами і зменшення ризиків.

Проведено ретельне тестування системи за стандартами ISTQB, що включало функціональне, продуктивне, інтеграційне тестування, а також тести безпеки та зручності використання. Тестові сценарії підтвердили, що система

повністю відповідає заявленим вимогам, має високу продуктивність і стабільність, що забезпечує її готовність до реального впровадження.

У рамках економічного аналізу проєкту було здійснено розрахунок фінансових показників, таких як NPV, IRR, ROI та період окупності. Отримані результати продемонстрували високу економічну ефективність і привабливість проєкту, що підтверджує його інвестиційну доцільність та довгострокову перспективність.

У процесі виконання магістерської роботи були реалізовані всі поставлені завдання, що охоплюють як теоретичну, так і прикладну складову проєкту. У межах аналітичного розділу було проведено дослідження сучасного стану ринку інформаційних систем моніторингу серцево-судинного здоров’я, проаналізовано їхні функціональні можливості та обґрунтовано доцільність створення нової персоналізованої системи. На цій основі було сформовано технічне завдання та розроблено концептуальну модель інформаційної системи, яка інтегрує мобільні пристрої та смарт-гаджети з технологіями безперервного моніторингу стану серця.

У розділі, присвяченому архітектурному і математичному моделюванню, обґрунтовано вибір багаторівневої архітектури, що забезпечує масштабованість, надійність і відповідність вимогам інформаційної безпеки. Побудовано математичну модель, орієнтовану на аналіз медичних даних із метою виявлення ризиків серцевих захворювань, та визначено алгоритми, які базуються на застосуванні елементів штучного інтелекту для автоматизованої обробки інформації, формування діагностичних висновків і рекомендацій щодо профілактики. У рамках того ж розділу було розроблено структуру бази даних та визначено принципи збереження персональної інформації відповідно до вимог GDPR та HIPAA.

У розділі, присвяченому реалізації проєкту, описано побудову програмного забезпечення з урахуванням кросплатформеної взаємодії з мобільними пристроями, носимими сенсорами та хмарною інфраструктурою. Здійснено моделювання життєвого циклу проєкту засобами MS Project із застосуванням структурної декомпозиції робіт, діаграм Ганта та інструментів управління ризиками і ресурсами. Також було сформовано фінансове обґрунтування проєкту, проведено розрахунки NPV, IRR, ROI, періоду окупності, що підтвердили його інвестиційну привабливість і економічну ефективність.

Розділ тестування включає створення тест-кейсів і сценаріїв перевірки працездатності, продуктивності, точності та стабільності розробленої системи відповідно до стандартів ISTQB. Проведене функціональне, інтеграційне й навантажувальне тестування продемонструвало готовність продукту до використання у реальних умовах. У завершальному розділі роботи були визначені перспективи подальшого розвитку системи, зокрема впровадження додаткових алгоритмів штучного інтелекту, розширення функціоналу для підтримки інших медичних показників, а також можливості інтеграції з електронними медичними картками та медичними закладами.

Таким чином, усі поставлені завдання були виконані в повному обсязі, а результати дослідження підтверджують актуальність, доцільність і практичну цінність створеної інформаційної системи для моніторингу здоров’я серця людини.

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Kerzner, Harold. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. 12-те вид. URL: https://library.ucyp.edu.my/wp-content/uploads/2024/01/Kerzner-Harold-Project-management-a-systems-approach-to-planning-scheduling-and-controlling-2017-Wiley-libgen.li_.pdf.
2. Schwalbe, K. Information Technology Project Management. 2019. URL: https://unidel.edu.ng/focelibrary/books/Information%20Technology%20Project%20Management%20by%20Kathy%20Schwalbe%20(z-lib.org).pdf.
3. Heldman, K. PMP Project Management Professional Exam Study Guide. 3-тє вид. URL: https://students.aiu.edu/submissions/profiles/resources/onlineBook/N6k3x5_Project%20Mgt.pdf.
4. A Guide to the Project Management Body of Knowledge (PMBOK Guide). 7-ме вид. 2021. URL: https://tegnum.edu.pe/wp-content/uploads/2023/09/Project-Management-Institute-A-Guide-to-the-Project-Management-Body-of-Knowledge-PMBOK-R-Guide-PMBOK%C2%AE%EF%B8%8F-Guide-Project-Management-Institute-2021.pdf.
5. Sayed Abdelgaber, Abdelrahman M. Helmi. A Comprehensive Survey on the Role of the Medical AI for the Healthcare Support. URL: https://doi.org/10.5120/ijca2023923073.
6. Andre Esteva. A guide to deep learning in healthcare. 2019. 24–29 с. URL: https://doi.org/10.1038/s41591-018-0316-z.
7. Charu C. Aggarwal. Neural Networks and Deep Learning. URL: https://www.academia.edu/42981452/Neural_Networks_and_Deep_Learning_Charu_C_Aggarwal.
8. Bertalan Mesko. A short guide for medical professionals in the era of artificial intelligence. 1–8. с. URL: https://doi.org/10.1038/s41746-020-00333-z.
9. Moeen Hassanalieragh. Health Monitoring and Management UsingInternet-of-Things (IoT) Sensing with Cloud-basedProcessing: Opportunities and Challenges. IEEE, 2015. 285–292 с. URL: https://doi.org/10.1109/SCC.2015.47.
10. Health Insurance Portability and Accountability Act of 1996 (HIPAA). URL: https://www.cdc.gov/phlp/php/resources/health-insurance-portability-and-accountability-act-of-1996-hipaa.html.
11. Abdullah Aldwean, Dan Tenney. Artificial Intelligence in Healthcare Sector: A Literature Review of the Adoption Challenges. Technology Management Department, University of Bridgeport, Bridgeport, USA. URL: https://doi.org/10.4236/ojbm.2024.121009.
12. Adebayo Augustine Adeniyi, Oluwademiladeayo Samuel Adeniyi. Revolutionizing Healthcare: The Impact of Machine Learning and Artificial Intelligence. Department

of Computer Science, Bowen University, Iwo, Nigeria. URL: https://doi.org/10.4236/etsn.2024.134006.
13. Ali Abdin. The Role of Artificial Intelligence in Diagnostic Medicine: A Narrative Review. Limassol General Hospital, University of Nicosia, Limassol, Cyprus. URL: https://doi.org/10.4236/oalib.1112432.
14. Badr Alnasser. A Review of Literature on the Economic Implications of Implementing Artificial Intelligence in Healthcare. Department of Health Management, University of Hail, Hail, Saudi Arabia. URL: https://doi.org/10.4236/etsn.2023.123003.
15. Chayakrit Krittanawong, Hafeez Ul Hassan Virk. Machine learning prediction in cardiovascular diseases: a meta-analysis. URL: https://doi.org/10.1038/s41598-020-72685-1.
16. Christopher Tsai, Justin Starren. The Role of Nurses in Installing Telehealth Technology in the Home. URL: https://doi.org/10.1097/00024665-200507000-00004.
17. HIPAA Journal. HIPAA Encryption Requirements Explained.. URL: https://www.hipaajournal.com/hipaa-encryption-requirements/.
18. GDPR.eu. General Data Protection Regulation – Full Text & Guidelines.. URL: https://gdpr.eu/.
19. Michael Fitzgerald, Nina Kruschwitz. Embracing Digital Technology: A New Strategic Imperative. MIT Sloan Management Review. 2013. URL: https://sloanreview.mit.edu/projects/embracing-digital-technology/.
20. Piyush Mathur. Artificial Intelligence in Healthcare: 2021 Year in Review. URL: https://doi.org/10.13140/RG.2.2.25350.24645/1.
21. Ramo Sendelj. Cybersecurity Challenges in Healthcare. 2022. URL: https://doi.org/10.3233/SHTI220951.
22. Sabine Koch. Health informatics and the delivery of care to older people. 195–199 c. URL: https://doi.org/10.1016/j.maturitas.2009.03.023.
23. Thomas H. Davenport. The potential for artificial intelligence in healthcare. Future Hospital Journa. 94–98. c. URL: https://doi.org/10.7861/futurehosp.6-2-94.
24. Ahmad Chaddad, Yihang Wu. Federated Learning for Healthcare Applications. URL: https://doi.org/10.1109/JIOT.2023.3325822.
25. Casey Lynnette Overby, Peter Tarczy-Hornoch. Personalized medicine: Challenges and opportunities for translational bioinformatics. URL: https://doi.org/10.2217/pme.13.30.
26. David W Bates, Lucian L. Leape, J. Marc Overhage Anthem Inc. Reducing the Frequency of Errors in Medicine Using Information Technology. 2001. 299–308 c. URL: https://doi.org/10.1136/jamia.2001.0080299.
27. Deniz Ozel, Uğur Bilge. A web-based intensive care clinical decision support system: From design to evaluation. 2012. URL: https://doi.org/10.3109/17538157.2012.710687.

28. Dipak Kalra, David Ingram. Electronic Health Records. 2007. URL: https://doi.org/10.1007/1-84628-141-5_7.
29. Fei Wang, Lawrence Peter Casalino. Deep Learning in Medicine—Promise, Progress, and Challenges. 2018. 293–294 c. URL: https://doi.org/10.1001/jamainternmed.2018.7117.
30. Indra Neil Sarkar. Biomedical informatics and translational medicine. 2010. URL: https://translational-medicine.biomedcentral.com/articles/10.1186/1479-5876-8-22.
31. Min Chen, Yujun Ma. Wearable 2.0: Enabling Human-Cloud Integration in Next Generation Healthcare Systems. 2017. 54–61 c. URL: https://10.1109/MCOM.2017.1600410CM.
32. Muzammil Khan, Muhammad Taqi Mehran. Applications of artificial intelligence in COVID-19 pandemic: A comprehensive review. URL: https://doi.org/10.1016/j.eswa.2021.115695.
33. Nicola Rieke, Fausto Milletari. The future of digital health with federated learning. npj Digital Medicine, 2020. 119 c. URL: https://doi.org/10.1038/s41746-020-00323-1.
34. Spyros Kitsiou. Overview and Analysis of Electronic Health Record Standards. 2009. URL: https://www.academia.edu/981667/Overview_and_Analysis_of_Electronic_Health_Record_Standards.
35. Yong Jiang, Yi Dong. Artificial intelligence in healthcare: past, present and future. 2017. 230–243 c. URL: https://doi.org/10.1136/svn-2017-000101.
36. It represents a broad consensus about the most critical security risks to web applications. OWASP Top Ten. URL: https://owasp.org/www-project-top-ten/.
37. Security Pillar - AWS Well-Architected Framework. 2024. URL: https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/welcome.html.
38. Paul A. Grassi, James L. Fenton . *NIST Special Publication 800-63B*. 2023. Digital Identity Guidelines. URL: https://pages.nist.gov/800-63-3/sp800-63b.html.
39. Data encryption. security/encryption. URL: https://cloud.google.com/security/encryption.
40. Kubernetes Documentation. 2023. URL: https://kubernetes.io/docs/concepts/overview/.
41. Wazuh Documentation.. Security information and event management (SIEM).. URL: https://documentation.wazuh.com/current/index.html.
42. ISO/IEC 27035-1:2016. Information security incident management – Principles of incident management. URL: https://www.iso.org/standard/60803.html.
43. Spring Security Docs. OAuth 2.0 Login and RBAC in Modern Applications.. URL: https://docs.spring.io/spring-security/reference/index.html.
44. Eric Ayintareba Akolgo. Artificial Intelligence in Healthcare: A Fusion of Technologies. URL: https://doi.org/10.4236/jcc.2024.1212008.
45. Hassan Aziz. Telemedicine. American Society for Clinical Laboratory Science. URL: https://doi.org/10.29074/ascls.28.4.256.

46. Len Bass, Rick Kazman. Software Architecture In Practice. Addison-Wesley Longman, 2003. URL: https://www.researchgate.net/publication/224001127_Software_Architecture_In_Practice.
47. Martin Fowler. Patterns of Enterprise Application Architecture. URL: https://github.com/jjcolumb/awesome-xaf/blob/master/Patterns%20of%20Enterprise%20Application%20Architecture%20-%20Martin%20Fowler.pdf.
48. IBM Watson Health. Artificial Intelligence Technology Trends and IBM Watson References in the Medical Field. URL: https://doi.org/10.17496/kmer.2016.18.2.51.
49. S. M. Riazul Islam, Daehan Kwak. The Internet of Things for Health Care: A Comprehensive Survey. IEEE, 2015. URL: https://ieeexplore.ieee.org/document/7113786.
50. M. Jørgensen. Jorgensen, M.: A Review of Studies on Expert Estimation of Software Development Effort. Journal of Systems and Software. 2004. URL: https://doi.org/10.1016/S0164-1212(02)00156-5.

ДОДАТКИ

Додаток А

Частина Python-коду для розрахунку місячного та річного прибутку

```
import matplotlib.pyplot as plt
import random

# Початкові дані
equip_cost = 1000000
software_cost = 1000000
infrastructure_cost = 200000
marketing_cost = 5000
salary_cost = 300000
initial_cost = equip_cost + software_cost + marketing_cost + infrastructure_cost
+ salary_cost

maintenance_cost = 50000
support_cost = 50000
salary_cost = 100000
monthly_operational_cost = []

monthly_revenue_base = 400
monthly_revenue_premium = 1000
num_base_users = []
num_premium_users = []

base_users = 0
premium_users = 0
for i in range(1, 13):
    new_base_users = random.randint(50, 150)
    new_premium_users = random.randint(25, 75)
    base_users += new_base_users
    premium_users += new_premium_users
    num_base_users.append(base_users)
    num_premium_users.append(premium_users)
    monthly_cost = maintenance_cost + support_cost + salary_cost
    if i % 3 == 0:
        monthly_cost += marketing_cost
    monthly_operational_cost.append(monthly_cost)


# Розрахунки
monthly_profit = [(a * monthly_revenue_base + b * monthly_revenue_premium) - c
                  for a, b, c in zip(num_base_users, num_premium_users,
monthly_operational_cost)]

usr = [usr_base + usr_prem for usr_base, usr_prem in zip(num_base_users,
num_premium_users)]
# Загальний прибуток за рік
total_profit_year = []
temp_initial_cost = initial_cost
for i in range(1, 13):
    profit = monthly_profit[i - 1] - temp_initial_cost
    total_profit_year.append(profit)
    temp_initial_cost -= monthly_profit[i - 1]

# Побудова графіків
plt.figure(figsize=(10, 6))

plt.subplot(1, 2, 1)
```

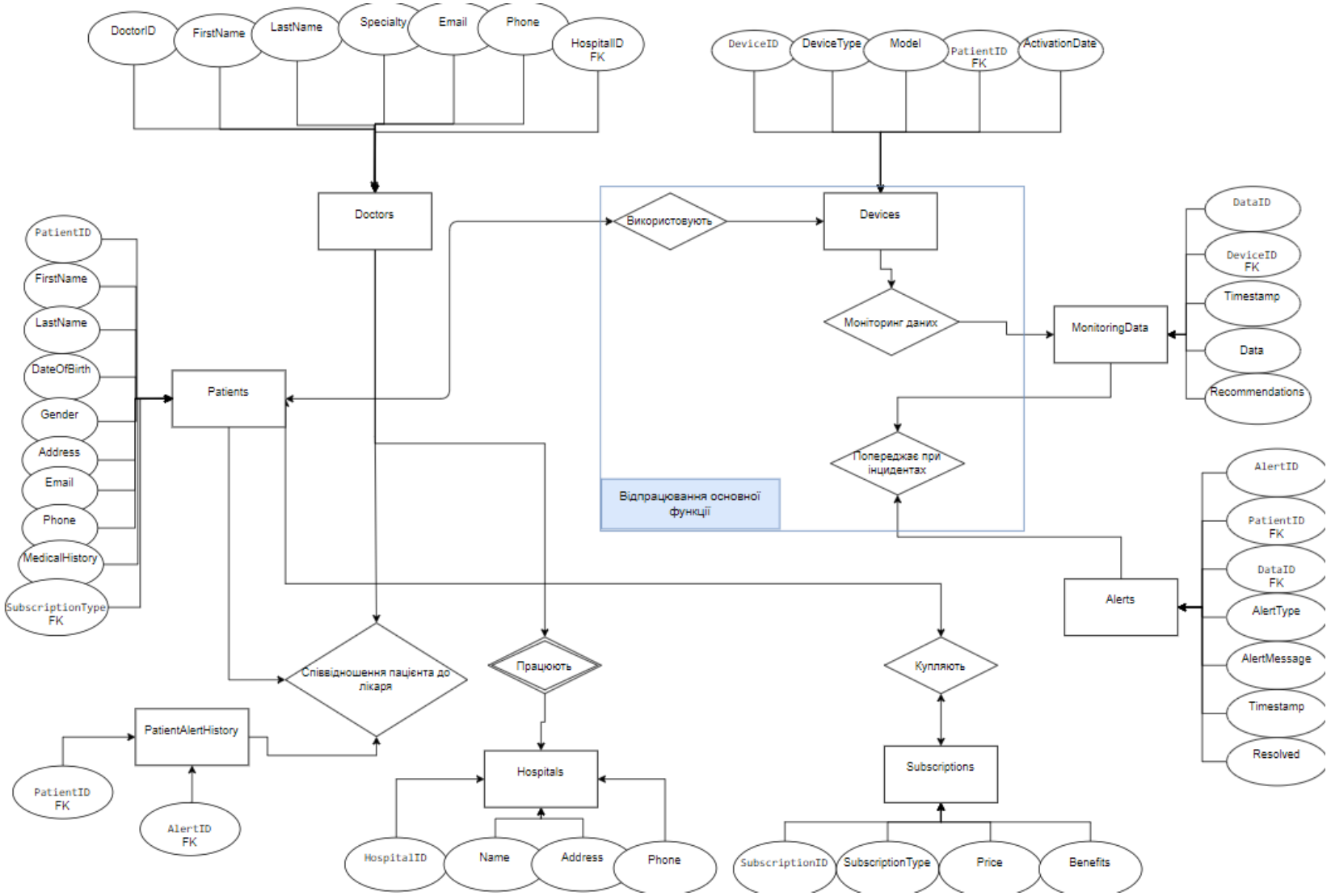
```
plt.plot(usr, monthly_profit, marker='o', label='Місячний прибуток')
plt.axhline(0, color='r', linestyle='--', label='Точка беззбитковості')
plt.xlabel('Кількість підписників')
plt.ylabel('Місячний прибуток (грн)')
plt.title('Місячний прибуток залежно від кількості підписників')
plt.legend()

plt.subplot(1, 2, 2)
plt.ticklabel_format(style='plain')
plt.plot(range(1, 13), total_profit_year, label='Річний прибуток')
plt.axhline(0, color='r', linestyle='--', label='Точка беззбитковості')
plt.xlabel('Місяці')
plt.ylabel('Загальний прибуток')
plt.title('Загальний прибуток помісячно')
plt.legend()

plt.tight_layout()
plt.show()
```

Додаток Б

Концептуальна модель бази даних



Додаток В

Інтерфейс сторінки що повертається при спробі перейти на неіснуючий ресурс