

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 122 «Комп'ютерні науки»
(освітня програма «Комп'ютерна графіка та розробка ігор»)
на тему:

«Система для моніторингу стану здоров'я»

Виконав студент групи ЗКН-19-1
СВІРСЬКИЙ Віталій Борисович

Керівник роботи:
ЛЕВКІВСЬКИЙ Віталій Леонідович

Рецензент:
КУЗЬМЕНКО Олександр Вікторович

Житомир – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

Марина ГРАФ

«05» січня 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу освітнього ступеня «бакалавр»
за спеціальністю 122 «Комп'ютерні науки»
(освітня програма «Комп'ютерна графіка та розробка ігор»)

Здобувач вищої освіти: **СВІРСЬКИЙ Віталій Борисович**

Керівник роботи: **ЛЕВКІВСЬКИЙ Віталій Леонідович**

Тема роботи: **«Система для моніторингу стану здоров'я»,**

затверджена наказом закладу вищої освіти від **«05» січня 2023 р., №11/с**

Термін здачі закінченої роботи: **«10» червня 2023 р.**

Вихідні дані роботи (зазначається предмет і об'єкт дослідження): об'єкт дослідження – веб-орієнтовані інформаційні системи та бази даних для збору, аналізу та надання інформації про стан здоров'я людини. Предмет дослідження – операції внесення, зберігання та обробки даних про стан здоров'я людини та особливості їх реалізації у веб-системі.

Консультанти випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Дата	
		Завдання видав	Завдання прийняв

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка задачі. Пошук, огляд та аналіз аналогічних розробок. Формулювання технічного завдання. Опрацювання літературних джерел.	2 лютого 2023 – 5 березня 2023	Виконано
2	Проектування структури системи	6 березня 2023 – 25 березня 2023	Виконано
3	Написання програмного коду	26 березня 2023 – 26 квітня 2023	Виконано
4	Тестування системи	27 квітня 2023 – 14 травня 2023	Виконано
5	Оформлення пояснювальної записки	15 травня 2023 – 1 червня 2023	
6	Попередній захист	1 червня 2023	Виконано

Здобувач вищої освіти **Віталій СВІРСЬКИЙ**

Керівник **Віталій ЛЕВКІВСЬКИЙ**

РЕФЕРАТ

Кваліфікаційна робота складається з вихідного програмного коду та пояснювальної записки обсягом 65 сторінок. В роботі присутні 20 зображень, 10 таблиць. Список використаних джерел налічує 15 найменувань.

Метою розробки веб-системи моніторингу здоров'я є створення комплексної та надійної системи з веб-інтерфейсом, яка може контролювати та відстежувати показники здоров'я людини. Дана система використовує технологію для збору та аналізу даних, пов'язаних із фізіологічними параметрами людини, рівнями активності та іншою важливою інформацією про здоров'я.

В роботі визначено цілі та актуальність, сформульовано задачу та описано функціональні можливості системи. Наведено технічне завдання, визначено етапи проектування та розробки, здійснено проектування UML-діаграм та реляційної бази даних веб-системи. Написано вихідний код системи: сконфігуровано середовище, створені відповідні моделі даних, контролери та шаблони веб-інтерфейсу користувача.

Проведено функціональне тестування, тестування помилок, навантаження, швидкодії, зручності користування, адаптивності. Розроблено інструкції користування та розгортання системи.

КЛЮЧОВІ СЛОВА: МОНІТОРИНГ СТАНУ ЗДОРОВ'Я, КРОВ'ЯНИЙ ТИСК, ПУЛЬС, ТЕМПЕРАТУРА, РІВЕНЬ ЦУКРУ.

					КН.КР.Б – 122 – 23 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Система для моніторингу стану здоров'я Пояснювальна записка	Літ.	Арк.	Аркушів
Розроб.		В.Б. СВИРСЬКИЙ					3	64
Керівник		В.Л. ЛЕВКІВСЬКИЙ						
Рецензент		О.В. КУЗЬМЕНКО						
Н. контр.								
Зав. каф.		М.С. ГРАФ				Житомирська політехніка, група ЗКН-19-1		

ABSTRACT

The bachelor degree work consists of source code and an explanatory note of 65 pages. There are 20 images and 10 tables in the work. The list of used sources includes 15 names.

The goal of health monitoring system development is to create a comprehensive and reliable system that can monitor and track various aspects of an individual's health. This system utilizes technology to collect and analyze data related to a person's physiological parameters, activity levels, and other relevant health information.

The work defines the goals and relevance, formulates the task and describes the functional capabilities of the system. The technical task is given, the stages of design and development are determined, the design of UML diagrams and the relational database of the web system is carried out. The source code of the system is written: the environment is configured, the corresponding data models, controllers and web user interface templates are created.

Functional testing, error testing, load testing, performance testing, ease of use, adaptability testing were carried out. Instructions for using and deploying the system have been developed.

KEY WORDS: HEALTH MONITORING, BLOOD PRESSURE, PULSE, TEMPERATURE, SUGAR LEVEL.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

ЗМІСТ

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ТЕХНОЛОГІЙ ДЛЯ ПРОЕКТУВАННЯ І РОЗРОБКИ ВЕБ-СИСТЕМ МОНІТОРИНГУ СТАНУ ЗДОРОВ'Я	13
1.1. Постановка задачі	13
1.2. Огляд існуючих веб-систем моніторингу стану здоров'я	17
1.3. Вибір та обґрунтування засобів та інструментів розробки	22
Висновки до розділу 1	24
РОЗДІЛ 2. ПРОЕКТУВАННЯ І ЕТАПИ РОЗРОБКИ СИСТЕМИ МОНІТОРИНГУ СТАНУ ЗДОРОВ'Я	25
2.1. Варіанти використання та функціональні вимоги системи	25
2.2. Проектування UML-діаграм	28
2.3. Проектування бази даних веб-системи	33
2.4. Хід розробка системи	37
Висновки до розділу 2	43
РОЗДІЛ 3. КЕРІВНИЦТВО РОЗГОРТАННЯ І ВИКОРИСТАННЯ. ОПИС РЕЗУЛЬТАТІВ ТЕСТУВАННЯ	44
3.1. Керівництво розгортання системи	44
3.2. Інструкція користувача	46
3.3. Особливості тестування системи моніторингу стану здоров'я	52
Висновки до розділу 3	55
ВИСНОВКИ	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	58
ДОДАТКИ	60

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

BPM: Beats Per Minute – ударів за хвилину – показник частоти серцевих скорочень

АТ: артеріальний тиск - сила тиску крові на стінки артерій

ЕКГ/ЕКГ: Електрокардіограма - тест, який записує електричну активність серця

SpO2: Peripheral Oxygen Saturation – Периферична сатурація киснем - показник рівня насичення крові киснем

HRV: Heart Rate Variability – варіабельність серцевого ритму - зміна часових інтервалів між ударами серця

ІМТ: Body Mass Index – індекс маси тіла - міра жиру в організмі на основі зросту та ваги

RHR: Resting Heart Rate – частота серцевих скорочень у спокої - частота серцевих скорочень, коли тіло перебуває в спокої

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. Сучасні системи, призначені для моніторингу стану різних аспектів здоров'я людини, надають цінні дані та інформацію як для окремих осіб, так і для медичних працівників. Серед їх важливих цілей – відстеження особистого здоров'я, керування процесом лікування, історією розвитку захворювань, дистанційних моніторинг пацієнтів, раннє виявлення та профілактика тощо. Розглянемо декілька ключових тез, що підкреслюють актуальність систем моніторингу здоров'я:

Відстеження особистого здоров'я. Системи моніторингу здоров'я дозволяють людям відстежувати та контролювати параметри свого здоров'я, такі як частота серцевих скорочень, артеріальний тиск, режим сну, рівень фізичної активності тощо, що допоможе краще зрозуміти свій загальний добробут і приймати обґрунтовані рішення щодо свого способу життя та звичок.

Керування процесом лікування та історією розвитку захворювань. Системи моніторингу здоров'я відіграють вирішальну роль у лікуванні хронічних захворювань, таких як діабет, гіпертонія та серцево-судинні захворювання. Ці системи забезпечують постійний моніторинг життєво важливих показників, рівнів глюкози, дотримання режиму лікування та відстеження симптомів. Надаючи дані в реальному часі, вони допомагають пацієнтам і медичним працівникам визначати закономірності, керувати ліками та відповідно коригувати плани лікування.

Дистанційний моніторинг пацієнтів. Системи моніторингу здоров'я стали особливо важливими в контексті дистанційного моніторингу пацієнтів. Вони дозволяють лікарям дистанційно контролювати стан здоров'я пацієнтів, зменшуючи потребу в частих відвідуваннях лікарень і покращуючи доступ до медичної допомоги, особливо для людей, які живуть у віддалених районах. Ця технологія особливо цінна для людей похилого віку, людей з хронічними захворюваннями та післяопераційних пацієнтів.

Раннє виявлення та профілактика. Системи моніторингу стану здоров'я полегшують раннє виявлення потенційних проблем зі здоров'ям. Завдяки

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

безперервному моніторингу життєво важливих показників, режиму сну та інших відповідних даних ці системи можуть попереджати людей і медичних працівників про будь-які відхилення від норми або тривожні ознаки. Раннє виявлення дозволяє вчасно втручатися та вживати профілактичних заходів, покращуючи результати та потенційно рятуючи життя.

Технологія переносних пристроїв. Широке використання таких пристроїв, як розумні годинники, браслети для фітнесу та інші датчики здоров'я, сприяло популярності та актуальності систем моніторингу здоров'я. Ці переносні пристрої можуть легко збирати та передавати дані на смартфони або спеціальні програми, надаючи користувачам інформацію про здоров'я в режимі реального часу. Вони стають все більш досконалими, пропонуючи такі функції, як електрокардіограма (ЕКГ), моніторинг рівня кисню в крові, відстеження стресу тощо.

Охорона здоров'я, керована даними. Системи моніторингу стану здоров'я генерують величезні обсяги даних, які можна використовувати для охорони здоров'я. Зведені та анонімні дані від кількох осіб можуть допомогти визначити тенденції щодо стану здоров'я, фактори ризику та покращити ініціативи у сфері охорони здоров'я. Крім того, сучасні алгоритми аналізу даних можуть генерувати персоналізовану інформацію та прогнози, допомагаючи в постановці діагнозу та прийнятті рішень щодо лікування.

Розширення людських можливостей. Системи моніторингу стану здоров'я дають людям можливість брати активну участь у керуванні своїм здоров'ям. Надаючи доступ до даних про здоров'я в режимі реального часу та персоналізованих рекомендацій, такі системи заохочують до самообслуговування та проактивного управління здоров'ям. Вони сприяють розширенню можливостей і відповідальності, сприяють здоровому способу життя та знижують витрати на охорону здоров'я.

Отже, актуальність систем моніторингу здоров'я полягає в їх здатності відстежувати та контролювати індивідуальні параметри здоров'я, сприяти лікуванню захворювань, уможливлювати віддалений моніторинг пацієнтів,

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

допомагати в ранньому виявленні та профілактиці, використовувати технології мобільних пристроїв та гаджетів, сприяти охороні здоров'я, керованої даними, та розширювати людські можливості щодо відповідальності за свій стан здоров'я та в цілому добробут. Дані системи стають все більш поширеними, змінюючи наш підхід до охорони здоров'я та пропагуючи більш проактивний та персоналізований підхід до управління охороною здоров'я.

Метою роботи є розробка комплексної та надійної веб-орієнтованої системи для моніторингу здоров'я, яка може контролювати та відстежувати показники здоров'я людини. Дана система використовує технологію для збору та аналізу даних, пов'язаних із фізіологічними параметрами людини, рівнями активності та іншою важливою інформацією про здоров'я.

Завданнями випускної роботи є:

- визначити цілі та актуальність розробки веб-системи для моніторингу стану здоров'я;
- здійснити постановку задачі, визначити функціональні можливості та варіанти використання системи;
- розробити технічне завдання, спланувати етапи проектування та розробки;
- здійснити проектування UML-діаграм: варіантів використання, діяльності, послідовності та класів;
- здійснити проектування реляційної бази даних веб-системи;
- написати програмний код системи: розробити моделі даних, контролери та шаблони веб-інтерфейсу користувача, систему авторизації на основі ролей та дозволів;
- провести функціональне тестування, тестування помилок, тестування навантаження, швидкодії, зручності користування, адаптивності;
- розробити інструкції користувача та розгортання системи;
- написати пояснювальну записку до роботи;
- здійснити розгортання системи в робочому режимі.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Об'єктом дослідження було визначено веб-орієнтовані інформаційні системи та бази даних для збору, аналізу та надання інформації про стан здоров'я людини.

Предмет дослідження – операції внесення, зберігання та обробки даних про стан здоров'я людини та особливості їх реалізації у веб-системі

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ТЕХНОЛОГІЙ ДЛЯ ПРОЕКТУВАННЯ І РОЗРОБКИ ВЕБ-СИСТЕМ МОНІТОРИНГУ СТАНУ ЗДОРОВ'Я

1.1. Постановка задачі

Для формулювання основної задачі нашої роботи визначимо завдання сучасних систем моніторингу стану здоров'я.

Внесення даних показників стану здоров'я. Дане завдання можна реалізувати шляхом ручного внесення або інтеграцією різних датчиків та пристроїв, здатних вимірювати життєво важливі показники, такі як частота серцевих скорочень, артеріальний тиск, температура тіла, насичення киснем і фізична активність.

Зберігання даних – розробка надійного механізму для постійного, безпечного та масштабованого рішення зберігання структурованої інформації про зроблені вимірювання показників стану здоров'я, забезпечуючи конфіденційність і відповідність відповідним нормам.

Моніторинг життєво важливих показників і параметрів здоров'я у режимі реального часу через інтуїтивно зрозумілий інтерфейс користувача – забезпечення миттєвого зворотного зв'язку і сповіщення про відхилення від норм, дозволяючи користувачам вжити відповідних заходів або звернутися за медичною допомогою, якщо це необхідно.

Аналіз даних – впровадження алгоритмів та інструментів аналітики для аналізу зібраних даних і отримання значущої інформації, розробка інтелектуальних алгоритмів для виявлення закономірностей, тенденцій і потенційних ризиків для здоров'я, надаючи користувачам можливість проактивно керувати своїм здоров'ям.

Інтуїтивно зрозумілий і зручний інтерфейс для мобільних і веб-платформ: забезпечення простоти навігації, візуальної чіткості і доступності для користувачів із різним рівнем технічного досвіду.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

Конфіденційність і безпека – застосування надійних заходів безпеки для захисту даних користувачів, дотримуючись правил конфіденційності, таких як HIPAA або GDPR, використання методів шифрування, автентифікації користувачів і анонімізації даних для забезпечення конфіденційності і цілісності даних.

Масштабованість – розробка системи з урахуванням зростання бази користувачів і технологій, що розвиваються, планування майбутнього розширення, врахування взаємодії з новими медичними технологіями або переносними пристроями.

Тестування та валідація – проведення комплексного тестування, включаючи модульне тестування, інтеграційне тестування та тестування доступу користувача для надійності, точності та зручності використання системи, перевірка ефективності системи за пробними впровадженнями або відгуками користувачів.

Формулювання задачі. Основною задачею роботи є проектування та розробка багатокористувацької веб-орієнтованої системи моніторингу здоров'я, як може зберігати та відображати дані показників стану здоров'я. Система має забезпечувати моніторинг у реальному часі, збір даних та аналіз, щоб допомогти людям керувати своїм благополуччям, а медичним працівникам – приймати обґрунтовані рішення.

Результатами роботи при розробці системи очікуються:

- Розроблена інтерактивна система моніторингу стану здоров'я з можливостями внесення та отримання даних про стан здоров'я в реальному часі.
- Рішення для зберігання даних із заходами конфіденційності та безпеки.
- Наявність веб-інтерфейсу для взаємодії з користувачем і керування.
- Документація щодо розгортання та використання системи.
- Звіт про результати тестування.

Отже, метою роботи є розробка комплексної та надійної веб-орієнтованої системи для моніторингу здоров'я, яка може контролювати та відстежувати

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

показники здоров'я людини. Дана система використовує технологію для збору та аналізу даних, пов'язаних із фізіологічними параметрами людини, рівнями активності та іншою важливою інформацією про здоров'я. Система повинна враховувати конфіденційність та безпеку, володіти зручним веб-інтерфейсом, супроводжуватись документацією щодо розгортання та використання.

1.2. Огляд існуючих веб-систем моніторингу стану здоров'я

Здійснимо огляд та порівняння сучасних веб-орієнтованих систем моніторингу стану здоров'я: Fitbit Health Solutions, MyFitnessPal, AdaHealth, Livongo.

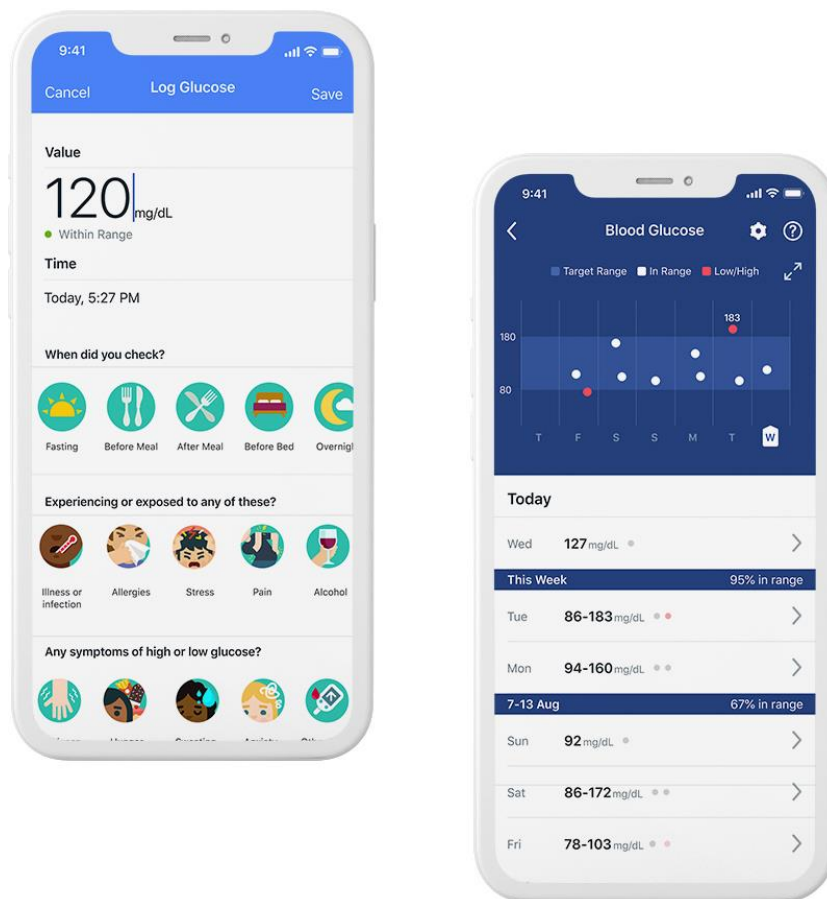


Рисунок 1.1 – Інтерфейс системи Fitbit Health Solutions

Fitbit Health Solutions пропонує комплексну веб-орієнтовану систему моніторингу здоров'я, розроблену як для окремих користувачів, так і для корпоративних програм оздоровлення. Він інтегрується з переносними

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

пристроями Fitbit, забезпечуючи відстеження життєво важливих показників, фізичної активності, режиму сну та харчування в реальному часі. Система пропонує інтерактивні інформаційні панелі та персоналізовану інформацію, що дозволяє користувачам встановлювати цілі, відстежувати прогрес і отримувати мотивацію. Fitbit Health Solutions також пропонує інтеграцію з системами електронних медичних записів (EHR), що забезпечує безперебійний обмін інформацією з постачальниками медичних послуг.

MyFitnessPal — це популярна веб-система моніторингу здоров'я, яка зосереджується на харчуванні та контролі ваги. Це дозволяє користувачам відстежувати щоденне споживання їжі, встановлювати цілі щодо споживання калорій і контролювати баланс поживних речовин. Система пропонує велику базу даних продуктів харчування, можливості сканування штрих-кодів і персоналізовані рекомендації щодо їжі. Користувачі також можуть відстежувати фізичну активність і синхронізувати дані з різних фітнес-додатків і пристроїв. MyFitnessPal надає докладні звіти, діаграми та графіки для візуалізації прогресу та пропонує соціальні функції для підтримки та мотивації.

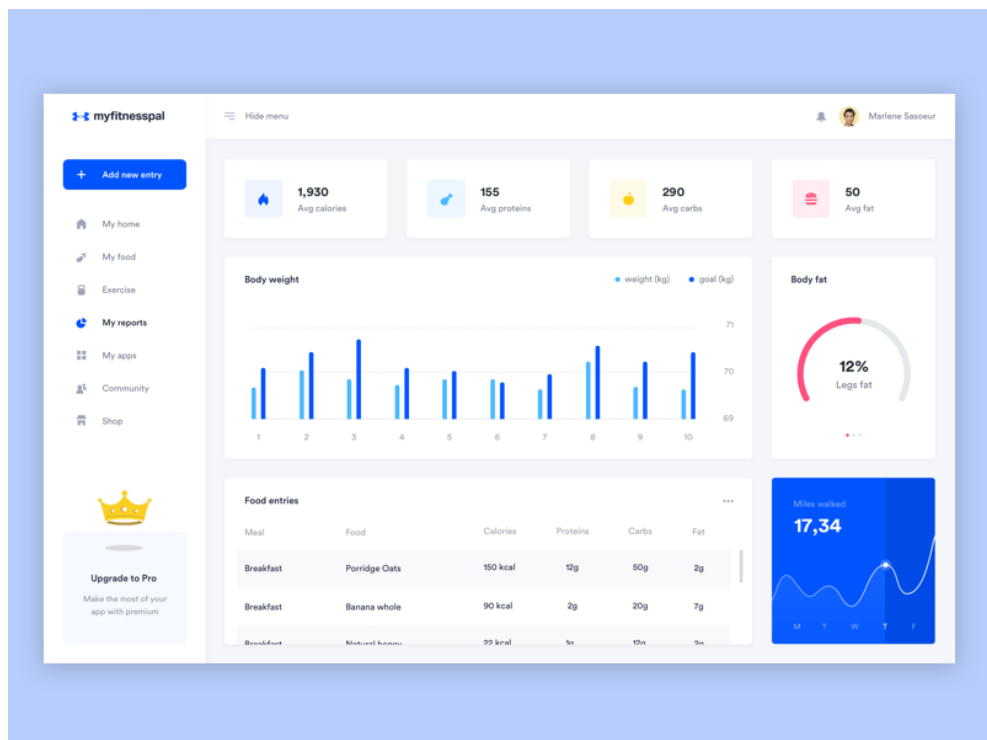


Рисунок 1.2 – Інтерфейс системи MyFitnessPal

Ada Health — це веб-орієнтована система моніторингу стану здоров'я на основі ШІ, яка забезпечує персоналізовану оцінку симптомів і сортування. Користувачі можуть вводити свої симптоми, а система використовує розширені алгоритми для створення можливих діагнозів і рекомендацій. Ada Health пропонує зручний інтерфейс, який направляє користувачів через низку запитань, щоб зібрати відповідну інформацію. Він також надає освітні ресурси та зв'язує користувачів із медичними працівниками для подальшого керівництва та догляду.

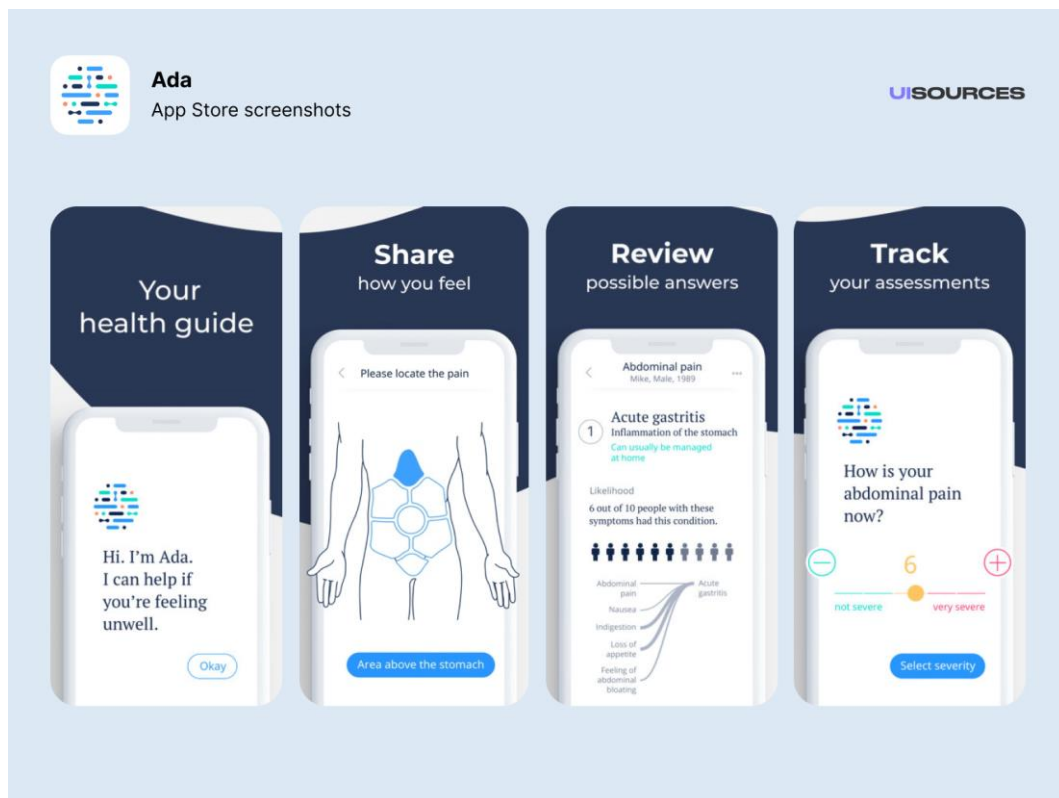


Рисунок 1.3 – Інтерфейс системи Ada Health

Livongo — це комплексна веб-орієнтована система моніторингу здоров'я, спрямована на лікування хронічних захворювань. Вона обслуговує людей із такими захворюваннями, як діабет, гіпертонія та потреби в контролі ваги. Система інтегрується з підключеними пристроями, такими як глюкометри та монітори артеріального тиску, для відстеження життєво важливих показників і надання зворотного зв'язку в реальному часі. Livongo пропонує персоналізований коучинг, освітні ресурси та інтерактивні завдання для

сприяння здоровій поведінці. Це також полегшує спілкування з постачальниками медичних послуг і дозволяє обмінюватися даними з системами EHR.

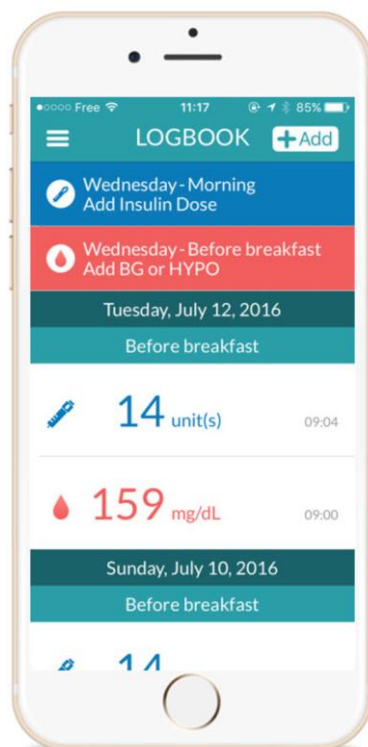


Рисунок 1.4 – Інтерфейс системи Livongo

Для порівняння даних систем розглянемо такі критерії: інтеграція з пристроями, сфера застосування, персоналізація та аналітика, інтеграція з охороною здоров'я, інтерфейс користувача.

Інтеграція з пристроями. Fitbit Health Solutions і Livongo пропонують бездоганну інтеграцію з певними переносними пристроями та обладнанням для моніторингу стану здоров'я, тоді як MyFitnessPal більше зосереджується на ручному введенні даних. Ada Health не вимагає інтеграції пристрою, оскільки вона зосереджена на оцінці симптомів.

Сфера застосування. Fitbit Health Solutions і Livongo надають ширший спектр функцій моніторингу здоров'я, включаючи життєво важливі показники, фізичну активність, сон, харчування та управління хронічними захворюваннями. MyFitnessPal в основному зосереджується на харчуванні та контролі ваги, тоді як Ada Health спеціалізується на оцінці симптомів і сортуванні.

Персоналізація та аналітика. Fitbit Health Solutions, MyFitnessPal і Livongo надають персоналізовану статистику, встановлення цілей і відстеження

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

прогресу. Ada Health генерує персоналізовані рекомендації на основі симптомів. Fitbit Health Solutions і MyFitnessPal пропонують соціальні функції для підтримки та мотивації.

Інтеграція з охороною здоров'я. Fitbit Health Solutions і Livongo пропонують інтеграцію з системами постачальників медичних послуг і платформами EHR, полегшуючи спілкування та обмін даними. Ada Health зв'язує користувачів із медичними працівниками для отримання подальших вказівок.

Інтерфейс користувача. усі чотири системи пропонують зручний веб-інтерфейс, але дизайн і компонування можуть відрізнятися. Fitbit Health Solutions і Livongo також надають інтерфейси мобільних додатків для додаткової зручності.

Здійснивши огляд декількох аналогів, можна виділити основні функції системи моніторингу здоров'я, які можуть відрізнятися в залежності від конкретного призначення системи та її цільової аудиторії. Наведемо загальні функції систем моніторингу здоров'я:

Моніторинг життєво важливих показників. Система повинна контролювати та відстежувати життєво важливі показники, такі як частота серцевих скорочень, артеріальний тиск, температура тіла, частота дихання та рівень насичення киснем. Це можна зробити за допомогою інтегрованих датчиків, переносних пристроїв або ручного введення даних.

Відстеження активності. Системи моніторингу здоров'я часто включають функції для відстеження фізичної активності, зокрема зроблених кроків, пройденої відстані та спалених калорій. Це може бути корисно для людей, які прагнуть вести активний спосіб життя, або для медичних працівників для моніторингу рівня активності пацієнта.

Моніторинг сну. Багато систем моніторингу здоров'я забезпечують функцію відстеження сну, яка вимірює тривалість сну, якість сну та шаблони стадій сну. Ця інформація допомагає користувачам зрозуміти їхні звички сну та визначити сфери, які потрібно покращити.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

Нагадування про ліки та стан здоров'я. Система може надсилати нагадування про розклад прийому ліків, запис до лікаря або завдання, пов'язані зі здоров'ям: пиття води, перерви чи виконання певних вправ. Ці нагадування допомагають користувачам дотримуватися призначеного лікування та підтримувати здорові звички.

Аналіз даних про стан здоров'я. Система моніторингу стану здоров'я часто містить можливості аналізу даних для обробки зібраної інформації та надання значущої інформації. Це може включати створення звітів, діаграм і графіків для візуалізації тенденцій, визначення закономірностей і відстеження прогресу з часом.

Персоналізовані рекомендації щодо здоров'я. Системи пропонують персоналізовані рекомендації на основі даних про здоров'я користувача. Це може включати пропозиції щодо розпорядку фізичних вправ, зміни дієти, методів боротьби зі стресом або інших дій, пов'язаних із здоров'ям, адаптованих до конкретних потреб і цілей людини.

Візуалізація даних і інформаційні панелі. Системи моніторингу працездатності зазвичай забезпечують зручний інтерфейс із інтерактивними інформаційними панелями та візуалізаціями. Ці інтерфейси дозволяють користувачам легко переглядати та інтерпретувати дані про своє здоров'я, що робить їх більш доступними та зручними.

Інтеграція з постачальниками медичних послуг. Системи дозволяють інтегруватися з системами електронних медичних записів (EHR) або іншими платформами постачальників медичних послуг. Це полегшує обмін інформацією між користувачами та медичними працівниками, забезпечуючи повне уявлення про стан здоров'я людини та дозволяючи приймати більш обґрунтовані медичні рішення.

Сповіщення. Системи моніторингу стану здоров'я можуть надсилати сповіщення про критичні події, відхилення від норми або надзвичайні ситуації. Ці своєчасні сповіщення допомагають користувачам вжити відповідних заходів або звернутися за медичною допомогою, коли це необхідно.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

Безпека та конфіденційність. Забезпечення безпеки та конфіденційності даних про здоров'я має вирішальне значення. Системи моніторингу стану здоров'я повинні впроваджувати надійні заходи безпеки, такі як шифрування даних, автентифікація користувачів і дотримання відповідних норм конфіденційності (наприклад, HIPAA, GDPR).

1.3. Вибір та обґрунтування засобів та інструментів розробки

Для розробки системи будемо використовувати стек LAMP (Linux, Apache, MySQL, PHP) та PHP-фреймворк Yii2. Вибір даних технологій для розробки системи моніторингу здоров'я може бути виправданий на основі кількох переваг:

Відкритий код і економічна ефективність. Linux, Apache, PHP, MySQL і Yii2 — це технології з відкритим кодом, що означає, що вони доступні у вільному доступі та не потребують дорогих ліцензій. Це робить їх економічно ефективним вибором, особливо для стартапів або організацій з обмеженим бюджетом.

Стабільність і надійність. операційна система Linux відома своєю стабільністю та надійністю. Вона має міцну репутацію надійності, безпеки та постійного вдосконалення завдяки розвитку, що керується великою спільнотою. Apache, як широко використовуваний веб-сервер, також відомий своєю стабільністю та продуктивністю.

PHP для веб-розробки. PHP — це популярна серверна мова сценаріїв, розроблена для веб-розробки. Вона має широку підтримку, велику спільноту та велику кількість доступних фреймворків і бібліотек. PHP добре підходить для створення динамічних веб-застосунків, що робить його відповідним вибором для розробки веб-орієнтованої системи моніторингу працездатності.

MySQL для керування базами даних. MySQL — це широко поширена система керування реляційними базами даних (RDBMS) з відкритим кодом. Він пропонує хорошу продуктивність, масштабованість і надійність, що робить його придатним для обробки даних щодо зберігання та керування вимогами системи моніторингу працездатності.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

Yii2 Framework для швидкої розробки. Yii2 — це високопродуктивний PHP-фреймворк, спеціально розроблений для розробки веб-додатків. Він відповідає архітектурному шаблону Model-View-Controller (MVC), який сприяє організації коду, повторному використанню та зручності обслуговування. Yii2 пропонує широкий спектр функцій, включаючи кешування даних, механізми безпеки та розробку RESTful API, які можуть бути корисними для створення надійної системи моніторингу працездатності.

Підтримка спільноти та документація. Linux, Apache, PHP, MySQL та Yii2 мають широку підтримку та документацію спільноти. Це означає, що розробники можуть знайти ресурси, навчальні посібники та форуми, які допоможуть їм у створенні та усуненні несправностей системи моніторингу працездатності. Активні спільноти також сприяють розвитку та вдосконаленню цих технологій.

Гнучкість і масштабованість. Вибраний стек забезпечує гнучкість і масштабованість, дозволяючи системі моніторингу працездатності враховувати зростаючі бази користувачів, обробляти зростаючі обсяги даних і адаптуватися до мінливих вимог. Linux, Apache, PHP, MySQL і Yii2 здатні впоратися з високим навантаженням трафіку і можуть бути легко масштабовані шляхом додавання додаткових серверів або оптимізації архітектури програми.

Важливо відзначити, що вибір технології має узгоджуватися з конкретними вимогами, досвідом команди розробників і наявними ресурсами. Хоча згадані технології мають свої переваги, інші альтернативи також можуть підійти залежно від унікальних потреб і обмежень проекту.

Висновки до розділу 1

В першому розділі визначено основні типові завдання системи моніторингу стану здоров'я, такі як забезпечення внесення і зберігання даних показників стану здоров'я, здійснення моніторингу життєво важливих показників і параметрів здоров'я у режимі реального часу, аналіз даних, наявність інтуїтивно зрозумілого і зручного інтерфейсу, забезпечення конфіденційності даних і

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

безпеки, масштабованість системи. На основі визначених завдань здійснено формулювання задачі роботи по розробці веб-орієнтованої системи для моніторингу стану здоров'я.

Розглянуто декілька сучасних веб-систем для моніторингу стану здоров'я та здійснено їх порівняння по таким критеріям, як інтеграція з пристроями, сфера застосування, персоналізація та аналітика, інтеграція з охороною здоров'я, інтерфейс користувача. На основі даного розгляду виділено ряд типових функцій для систем моніторингу здоров'я.

Обґрунтовано вибір стеку LAMP та PHP-фреймворка Yii2 для розробки системи, враховуючи такі вимоги як економічна ефективність, безпека, надійність, швидкість розробки, гнучкість, масштабованість, документація та активна підтримка спільноти розробників.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЕКТУВАННЯ І ЕТАПИ РОЗРОБКИ СИСТЕМИ МОНІТОРИНГУ СТАНУ ЗДОРОВ'Я

2.1. Варіанти використання та функціональні вимоги системи

Для незареєстрованих користувачів та для таких, що не здійснили вхід, доступні функції:

- реєстрації;
- входу;
- відновлення паролю.

Для зареєстрованих користувачів системи передбачено дві ролі: Admin, User:

Головний адміністратор (Admin):

- може реєструвати інших користувачів
- може редагувати профілі всіх користувачів (в т.ч. і тих, хто зареєструвався самостійно)
- може переглядати і вносити дані про стан здоров'я всіх користувачів

Зареєстрований користувач (User). Дана роль надається користувачу автоматично одразу після реєстрації:

- може формувати список своєї групи, надсилаючи запит іншим користувачам
- може вносити дані про стан здоров'я членів своєї групи
- може редагувати власний профіль
- може приймати або відхиляти (видаляти) запит іншого користувача на додавання до його групи
- може перебувати в декількох групах

Здійснимо проектування та наведемо ілюстрацію діаграми варіантів використання системи з позицій двох акторів: адміністратора та користувача:

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

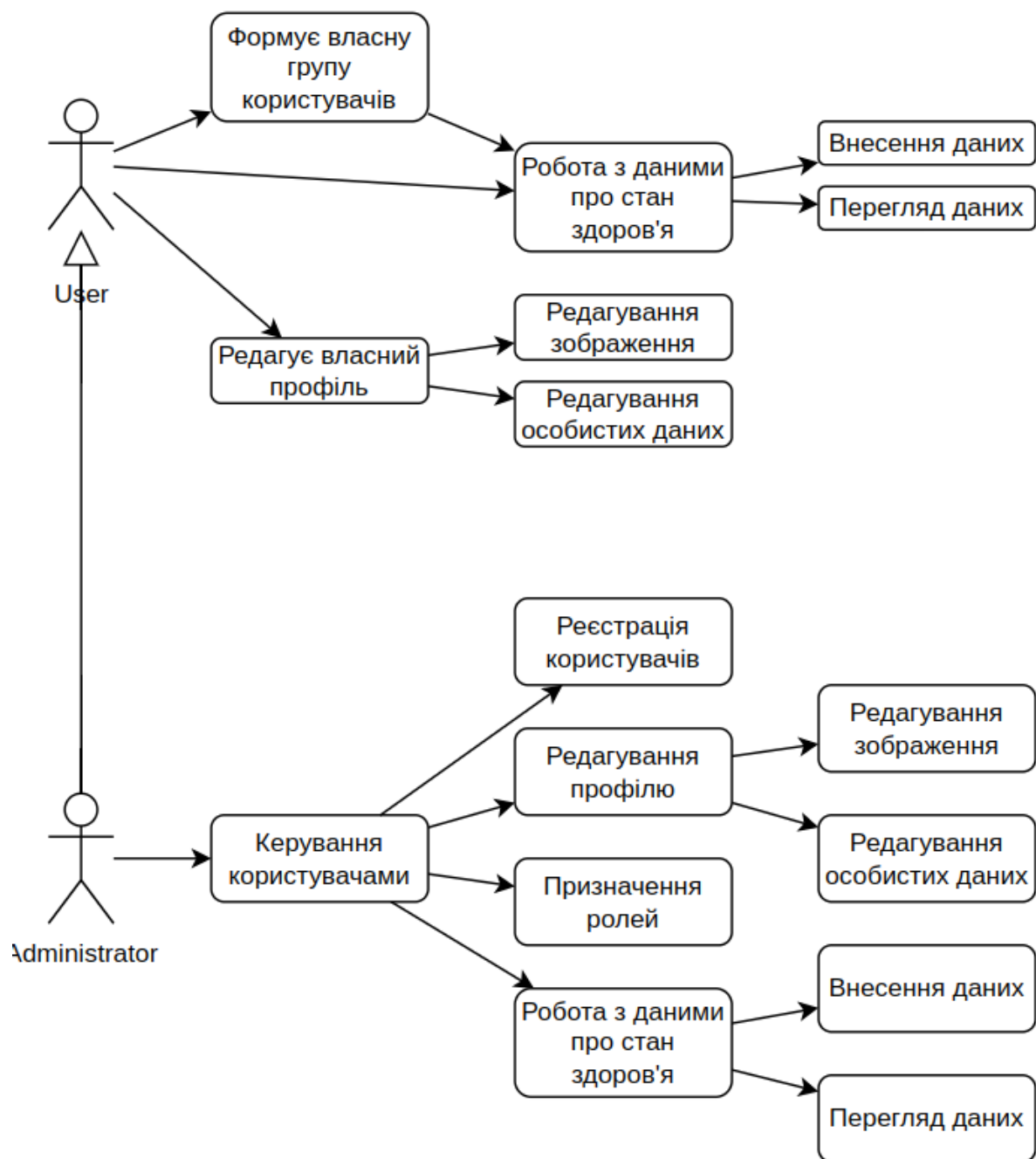


Рисунок. 2.1. Діаграма варіантів використання веб-системи моніторингу стану здоров'я

Дана UseCase-діаграма ілюструє високорівневі функції, доступні для двох акторів: адміністратора і користувача:

- *Адміністратор*: адміністратор відповідає за керування системою та обліковими записами користувачів. Користувачі з роллю *Admin* можуть виконувати такі дії, як керування користувачами, створення або зміна

облікових записів користувачів, надання або скасування ролей, а також внесення та перегляд даних про стан здоров'я всіх користувачів;

- *Користувач*: Актор користувача представляє осіб, які зареєстровані або адміністратором або самостійно. Користувачі можуть отримувати доступ до своїх особистих даних про здоров'я, оновлювати їх. Також користувачі можуть формувати власну групу серед інших користувачів (наприклад, членів родини), таким чином отримуючи можливість керувати даними про стан здоров'я членів своєї групи. Також користувачі мають можливість підтверджувати або скасовувати запити на додавання їх до групи іншого користувача, який відправив запит.

2.2. Проектування UML-діаграм

Для спрощення сприйняття ілюстрації різних процесів у системі, розділимо активності на три діаграми: перші дві стосуються ролі користувача, третя – адміністратора.

На першій діаграмі активності (рис. 2.3) користувач виконує такі дії:

- Користувач, який бажає додати іншого користувача у власну групу, натискає на кнопку “Додати користувача у мою групу”.
- Система відображає форму з внесенням логіну іншого користувача. Даний користувач вводить потрібний логін і натискає кнопку “Запросити”;
- Користувач, якого запросили, в списку “Мої запрошення”, знаходить потрібно запрошення та приймає або скасовує його;
- Якщо запрошення прийняте, користувач, що запросив, може вносити та переглядати дані про стан здоров'я іншого користувача.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

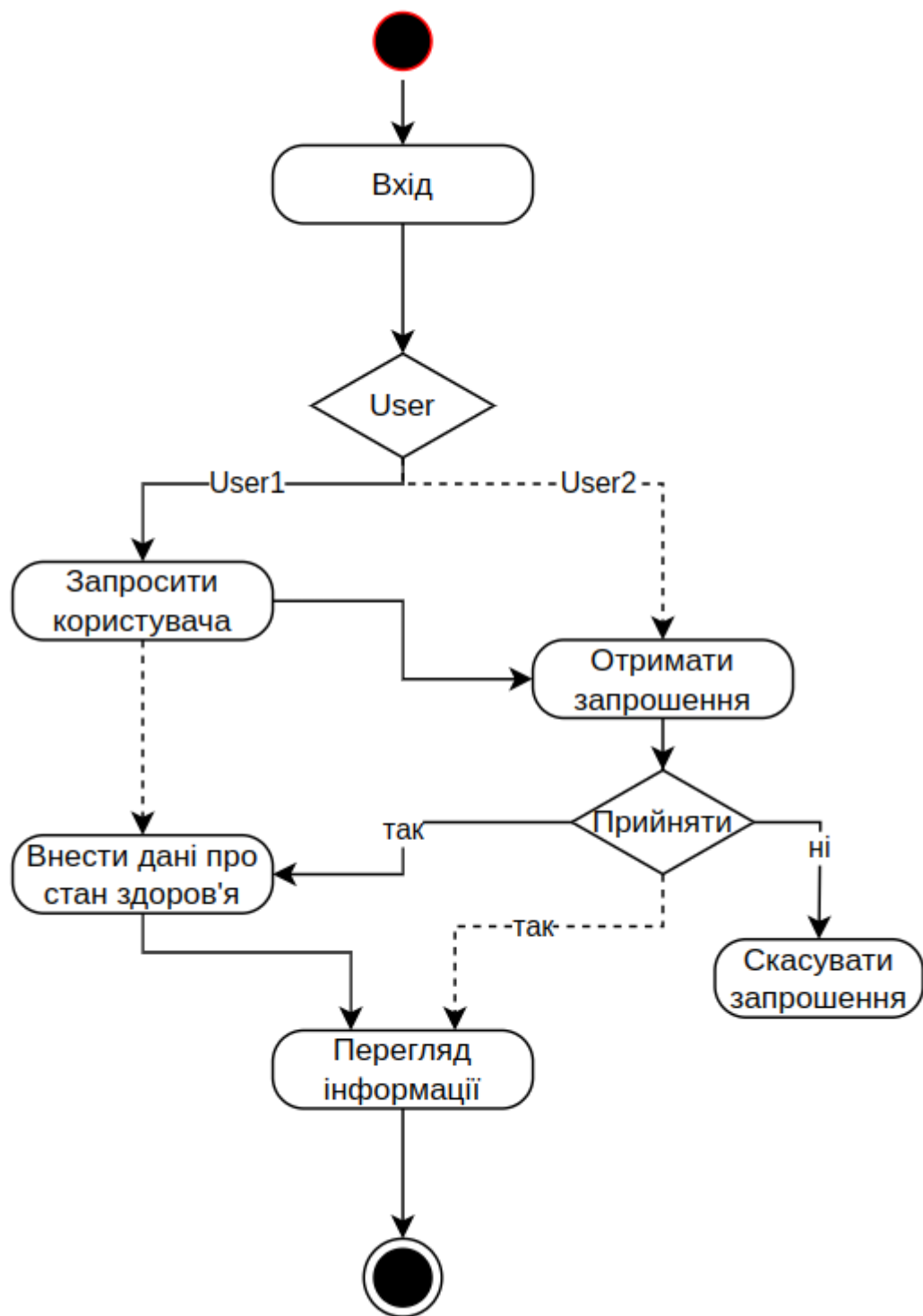


Рисунок. 2.3. Діаграма активності, що ілюструє процес додавання одним користувачем іншого до своєї групи для моніторингу стану здоров'я

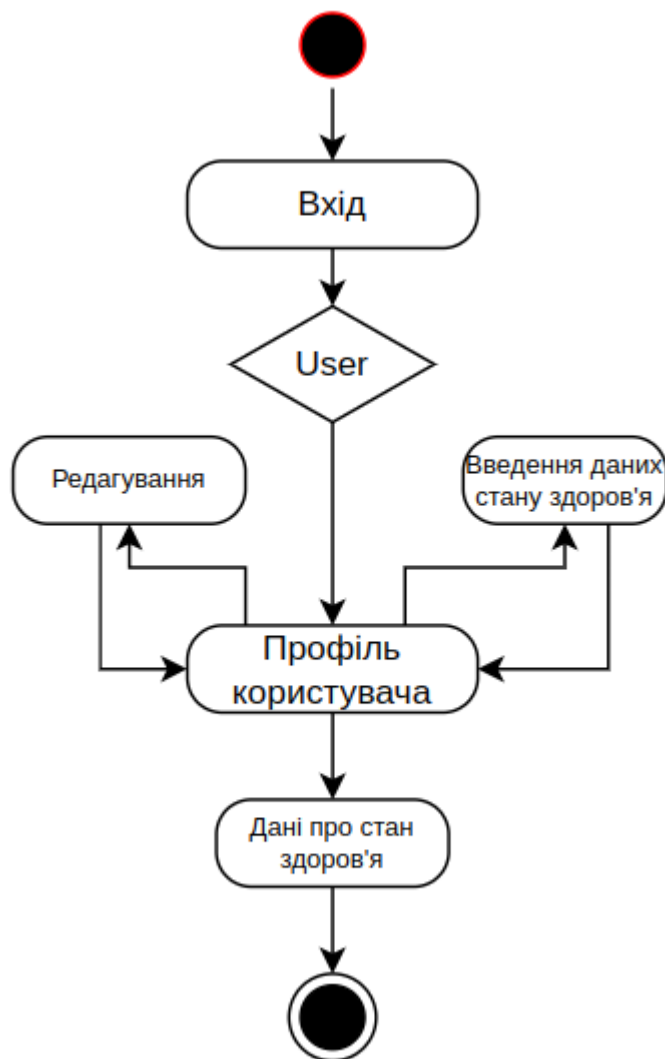


Рисунок. 2.4. Діаграма активності, що ілюструє процес роботи користувача з власним профілем та власними даними про стан здоров'я

На другій діаграмі активності (рис. 2.4) користувачу доступні такі дії:

- Користувач починає з вибору опції «Мій профіль» у системі управління здоров'ям.
- Система відображає особисту інформацію користувача, включаючи такі деталі, як ім'я, вік, контактна інформація та будь-яка інша відповідна інформація.
- Користувач вибирає опцію «Редагувати профіль», щоб внести зміни до своєї особистої інформації. Система надає користувачеві інтерфейс для редагування бажаних полів: прізвище, ім'я, дата народження, фото. Після внесення необхідних змін користувач вибирає опцію «Зберегти зміни», щоб зберегти оновлену інформацію. Система підтверджує, що зміни

успішно збережені, і відображає повідомлення про те, що особисту інформацію оновлено.

- Користувач вибирає опцію «Введення даних стану здоров'я». При цьому він має можливість ввести тиск, пульс, рівень цукру в крові, температуру. Система надає різні форми для внесення даних.
- Користувач вибирає опцію «Перегляд даних про стан здоров'я». Система надає користувачу можливості щодо перегляду історії різних показників стану здоров'я з можливістю сортування, пошуку і фільтрації даних

Третя діаграма активностей (рис. 2.5) ілюструє процес роботи адміністратора з профілями користувачів:

- Після входу, якщо користувач має роль адміністратора, система надає йому можливості працювати з усіма користувачами системи:
- Адміністратор може реєструвати користувачів, редагувати їх профілі, вносити та переглядати дані про стан їх здоров'я;
- На діаграмі стрілками проілюстровано можливі ланцюжки виконання запитів адміністратора по роботі з користувачами.

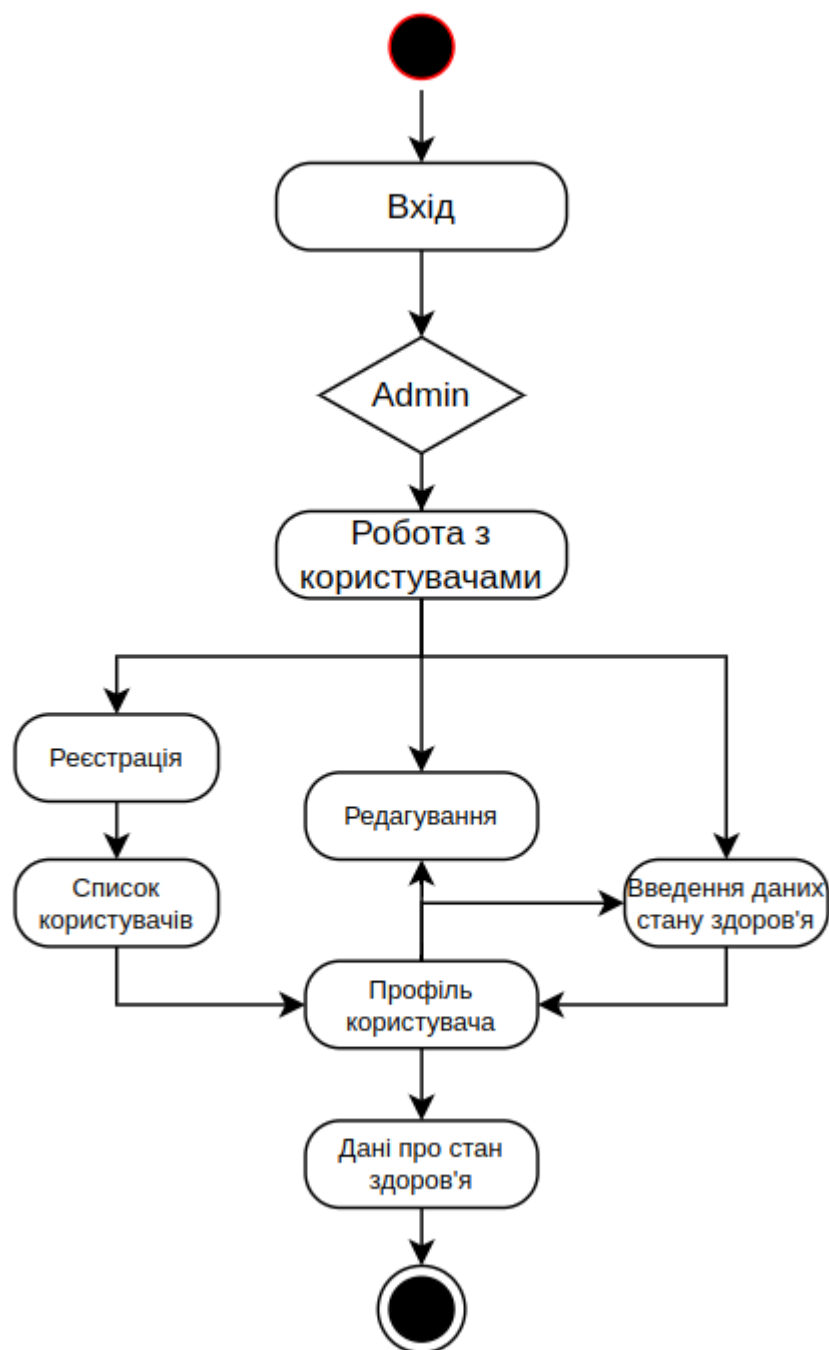


Рисунок. 2.5. Діаграма активності, що ілюструє процес роботи адміністратора з профілями користувачів та моніторингом стану їх здоров'я

Розробимо UML-діаграму послідовностей. Дана діаграма ілюструє процес взаємодії користувачів із системою при додаванні користувача в групу іншого користувача для здійснення моніторингу стану здоров'я (наприклад, користувач може формувати групу з членів своєї сім'ї).

На рис. 2.5 зображена діаграма послідовностей, що ілюструє даний процес:

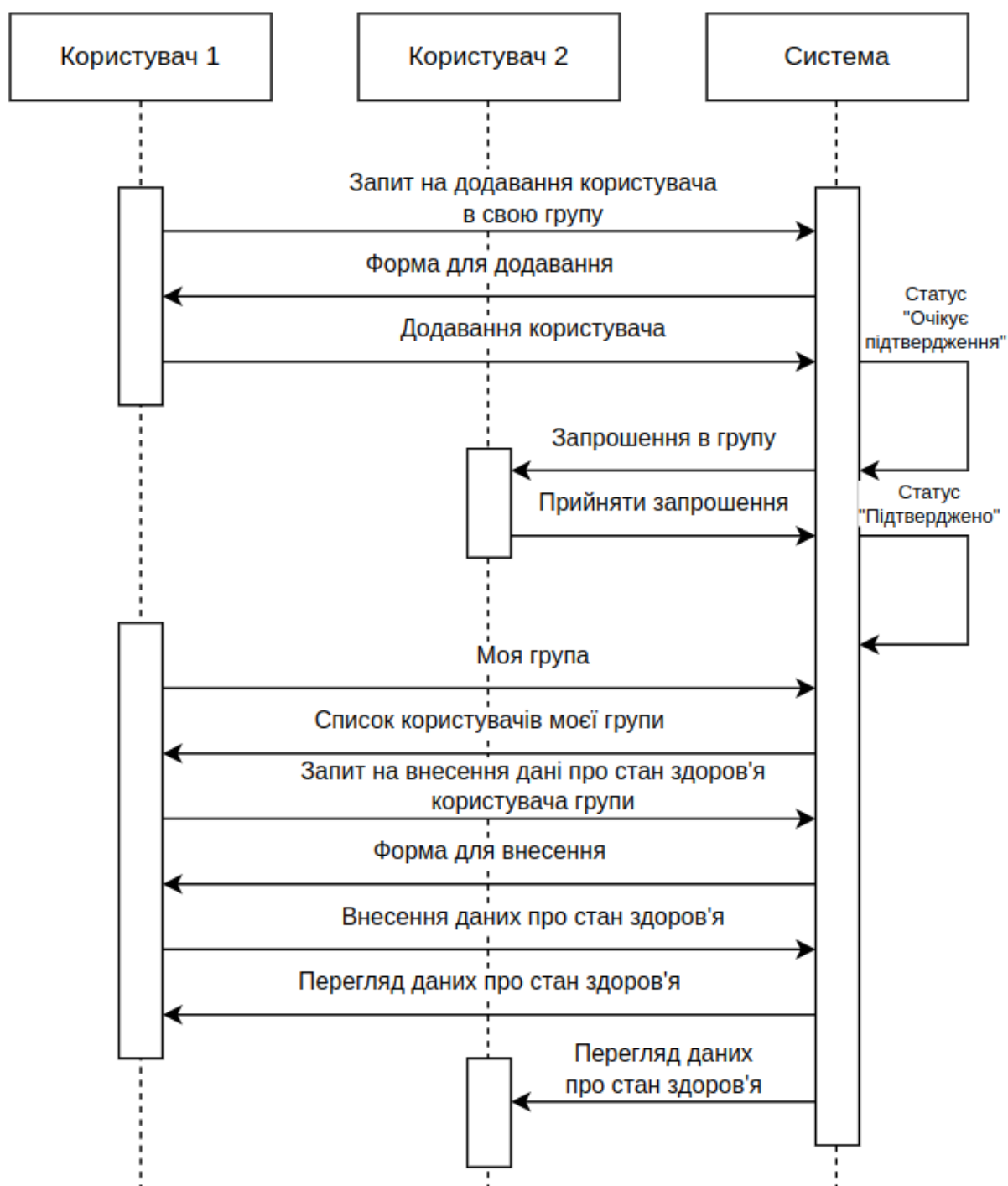


Рисунок. 2.5. Діаграма послідовностей для додавання іншого користувача у власну групу для можливості моніторингу стану здоров'я

- 1) Користувач 1 надсилає запит на додавання користувача 2 в свою групу;
- 2) Система, отримавши запит, надсилає користувачу 1 форму для додавання користувача 2 в групу;
- 3) Користувач 1 вводить логін користувача 2 і натискає кнопку “Відправити запрошення”;

Змн.	Арк.	№ докум.	Підпис	Дата

- 4) Система отримує запит та додає його в БД зі статусом “Очікує підтвердження”;
- 5) Користувач 2 отримує запрошення на вступ до групи у вкладці “Мої запрошення”;
- 6) Користувач 2 підтверджує або скасовує запрошення. В подальшому він може вийти з групи;
- 7) Користувач 1 переглядає список своєї групи, може вносити дані різних показників стану здоров’я для користувачів своєї групи;
- 8) Користувач 1 і користувач 2 можуть переглядати історію різних показників стану здоров’я користувача 2.

Наступним етапом нашої роботи – проектування діаграми класів системи (рис. 2.5). Діаграма складається із основних сутностей та їх відношень: користувачі, показники стану здоров’я: артеріальний тиск, пульс, рівень цукру тощо.

Клас User – працює з даними зареєстрованих користувачів: реєстрація, автентифікація, вхід, робота з профілем користувача;

Клас Admin – наслідується від User. Має можливості щодо перегляду/редагування/видалення профілів та роботою з даними про стан здоров’я усіх користувачів.

Клас GroupUser – призначений для роботи з даними про групи користувачів. Кожен користувач може формувати власну групу інших користувачів, надсилаючи їм запрошення. Таким чином, він стає менеджером цих користувачів з можливостями перегляду їх профілю та роботи з даними про стан їх здоров’я.

Клас HealthData - базовий клас для зберігання спільних даних про стан здоров’я: пацієнт (для кого ввели дані), менеджер (хто ввів дані), дата і час вимірювання.

Класи *BloodPressure*, *BloodSugar*, *Temperature* – дані про конкретні показники стану здоров’я. На основі цих даних формується історія та можна генерувати статистику та звітну інформацію.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

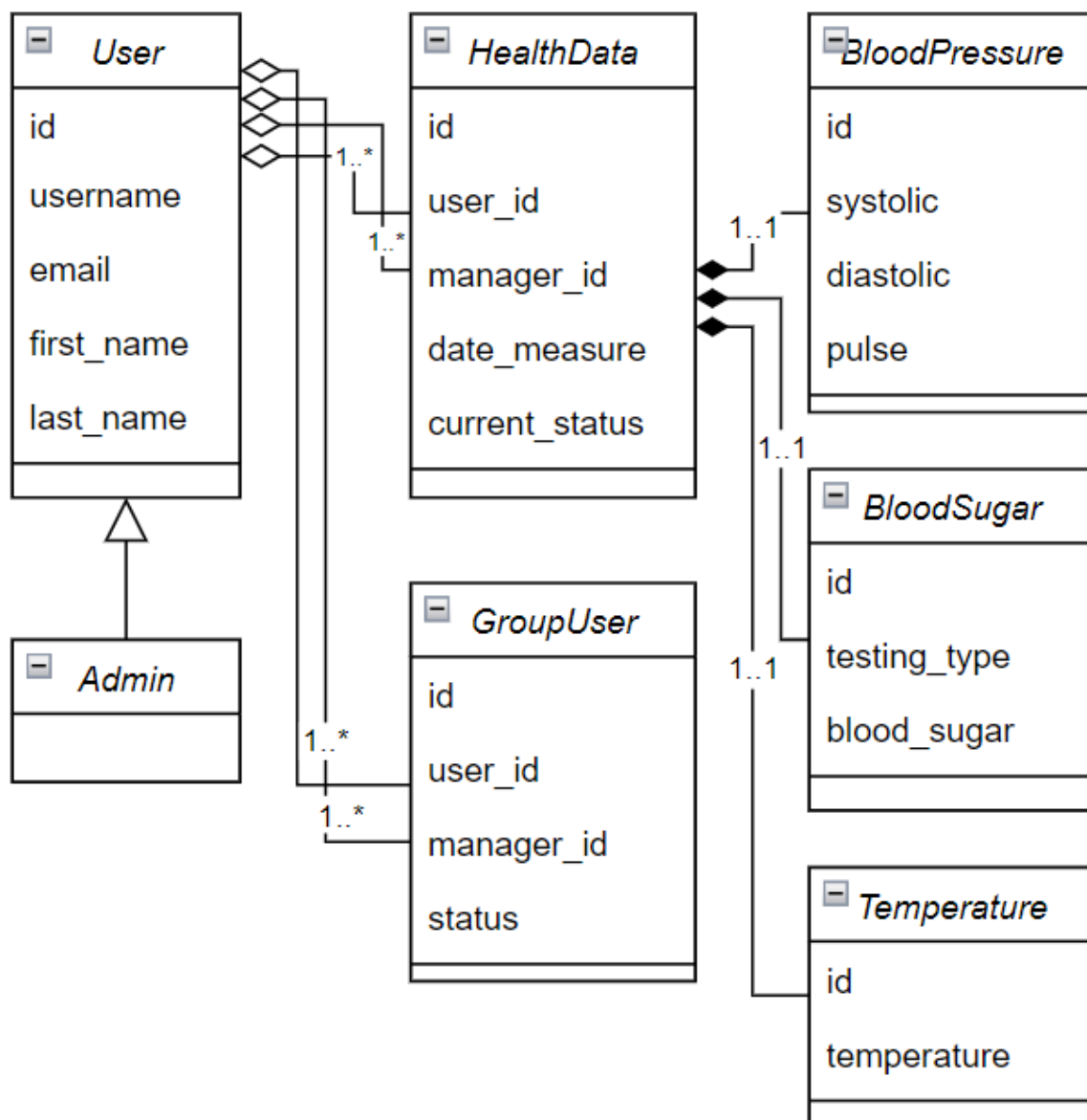


Рисунок. 2.5. Діаграма класів системи моніторингу стану здоров'я.

2.3. Проектування бази даних веб-системи

Наведемо перелік таблиць реляційної бази даних:

Таблиця 2.1.

Перелік таблиць реляційної бази даних

Таблиця	Дані
user	Користувачі системи
group_user	Групи користувачів по керівниках (наприклад, мої

	члени родини, мої пацієнти тощо)
data_blood_pressure	Дані показників вимірювання артеріального тиску
data_blood_sugar	Дані вимірювання рівня цукру в крові
data_body_temperature	Дані вимірювання температури тіла
auth_item	Дані системи авторизації: ролі та дозволи
auth_assignment	Призначення користувачам ролей або дозволів
auth_item_child	Ієрархія ролей та дозволів
auth_rule	Правила, що перевіряють та обмежують доступ

На рис. 2.6 наведена загальна структура бази даних: таблиці, поля, зв'язки. Здійснимо опис структури всіх таблиць.

Таблиця 2.2

Таблиця *user* – зареєстровані користувачі.

Поле	Тип поля	Індекс	Пояснення
id	int(11)	Primary Key	ID користувача
username	varchar(255)	Unique	Логін
email	varchar(255)	Unique	Email
first_name	varchar(255)		Ім'я користувача
last_name	varchar(255)		Прізвище користувача
image	varchar(255)		Фото користувача

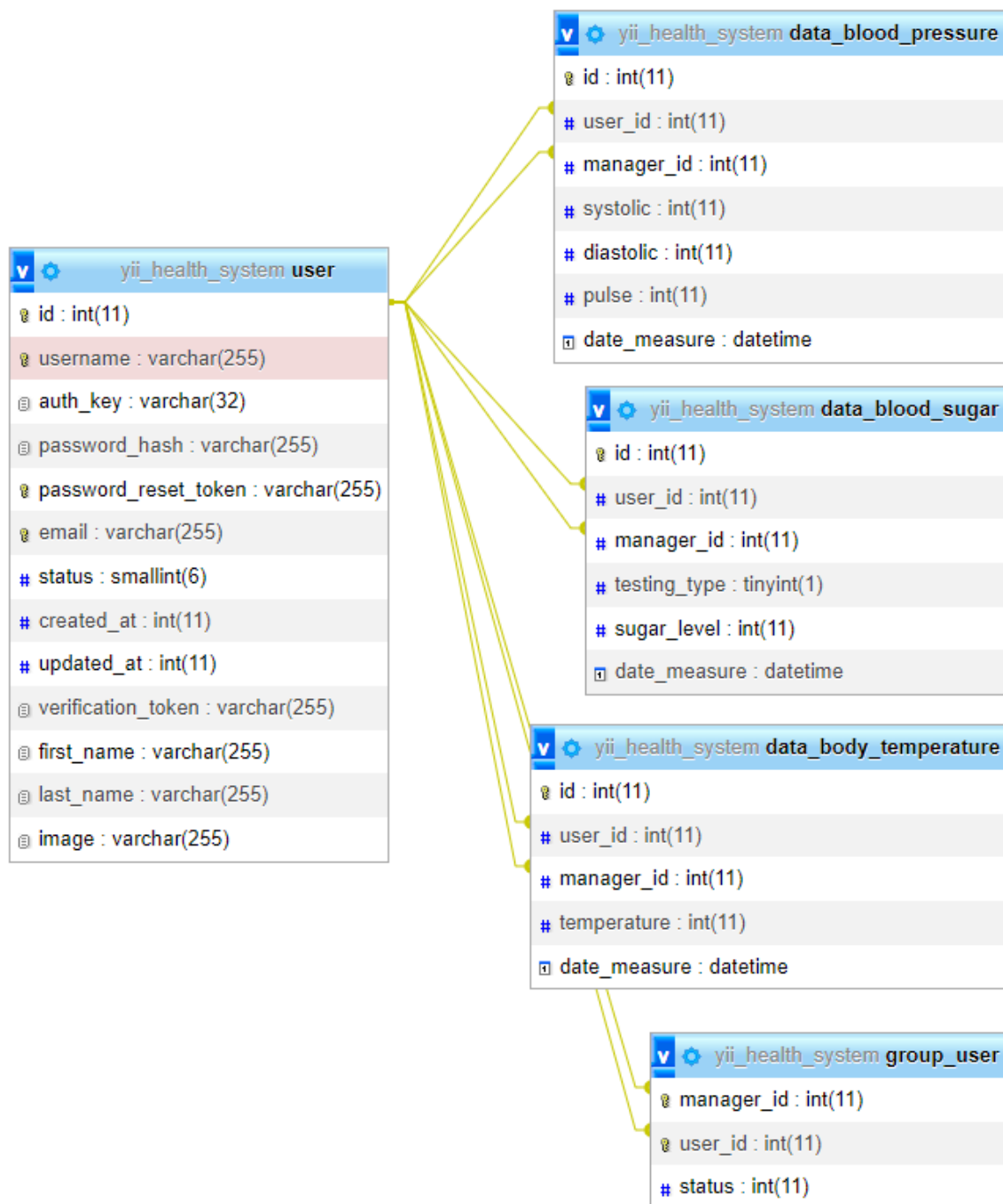


Рисунок 2.6. Схема бази даних системи для моніторингу стану здоров'я

Таблиця 2.3

Таблиця *group_user* – групування користувачів по керівниках.

Поле	Тип поля	Індекс	Пояснення
manager_id	int(11)	Unique	ID керівника групи
user_id	int(11)		ID користувача групи
status	tinyint(1)	Index	Статус ("Очікує", "Підтверджено")

Таблиця 2.4

Таблиця *data_blood_pressure* – введені показники даних артеріального тиску.

Поле	Тип поля	Індекс	Пояснення
id	int(11)	Primary Key	ID запису
user_id	int(11)	Index	ID користувача (пацієнт)
manager_id	int(11)	Index	ID користувача (менеджер)
systolic	int(11)		Систолічний тиск
diastolic	int(11)		Діастолічний тиск
pulse	int(11)		Пульс
date_measure	datetime		Дата і час вимірювання

Таблиця 2.5

Таблиця *data_blood_sugar* – введені показники даних цукру в крові.

Поле	Тип поля	Індекс	Пояснення
id	int(11)	Primary Key	ID запису
user_id	int(11)	Index	ID користувача (пацієнт)
manager_id	int(11)	Index	ID користувача (менеджер)
testing_type	tinyint(1)		Тип тесту (PPBS, FBS)
sugar_level	int(11)		Рівень цукру, мг/дл
date_measure	datetime		Дата і час вимірювання

Таблиця 2.6

Таблиця *data_body_temperature* – введені показники температури тіла.

Поле	Тип поля	Індекс	Пояснення
id	int(11)	Primary Key	ID запису
user_id	int(11)	Index	ID користувача (пацієнт)
manager_id	int(11)	Index	ID користувача (менеджер)

temperature	int(11)	float	Значення температури, °C
date_measure	datetime		Дата і час вимірювання

2.4. Хід розробки системи

Наведемо приклад розробки системи ролей та дозволів для керування користувачами: доступ до перегляду профілів, управління та моніторингу станом здоров'я. В системі весь доступ розподіляється на дві ролі: адміністратора (Admin) та користувача (User). Згідно визначених функціональних вимог, необхідно забезпечити та обмежити доступ до виконання таких дій:

- *перегляд профілю користувача, внесення та моніторинг даних про стан здоров'я.* Адміністратор має доступ на кожного користувача, користувач – на тих, хто належить до його групи, а також на власний профіль;
- *редагування профілю користувача.* Адміністратор має доступ на кожного користувача;
- *видалення профілю користувача.* Має доступ лише адміністратор.

Згідно такого розподілу доступу, розробимо систему ролей та дозволів:

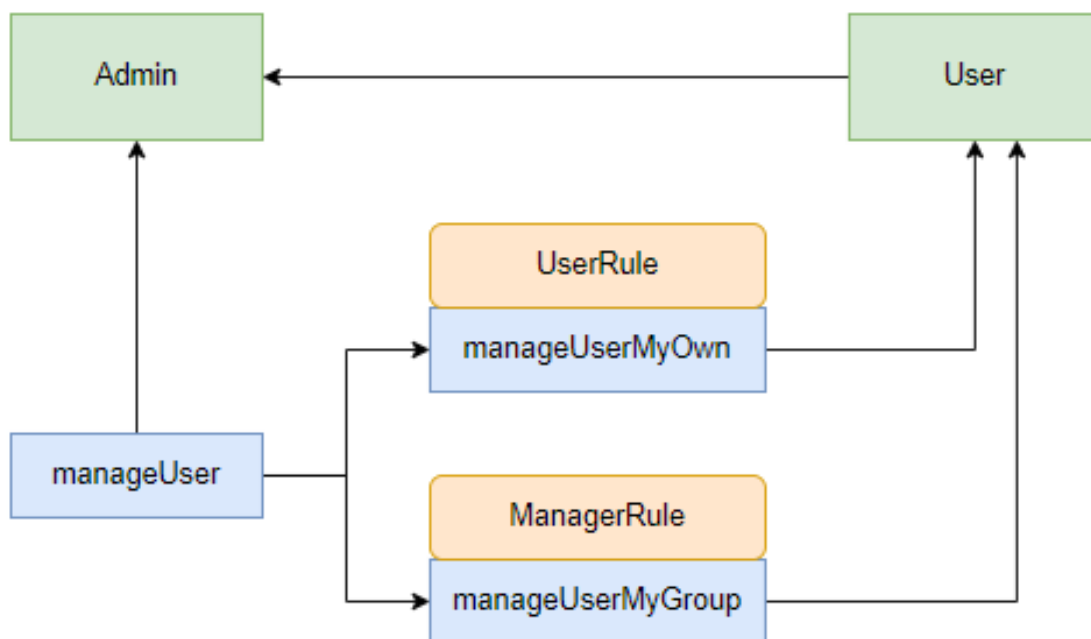


Рисунок 2.7. Розподіл дозволів для перегляду профілю користувача, внесення та моніторинг даних про стан здоров'я

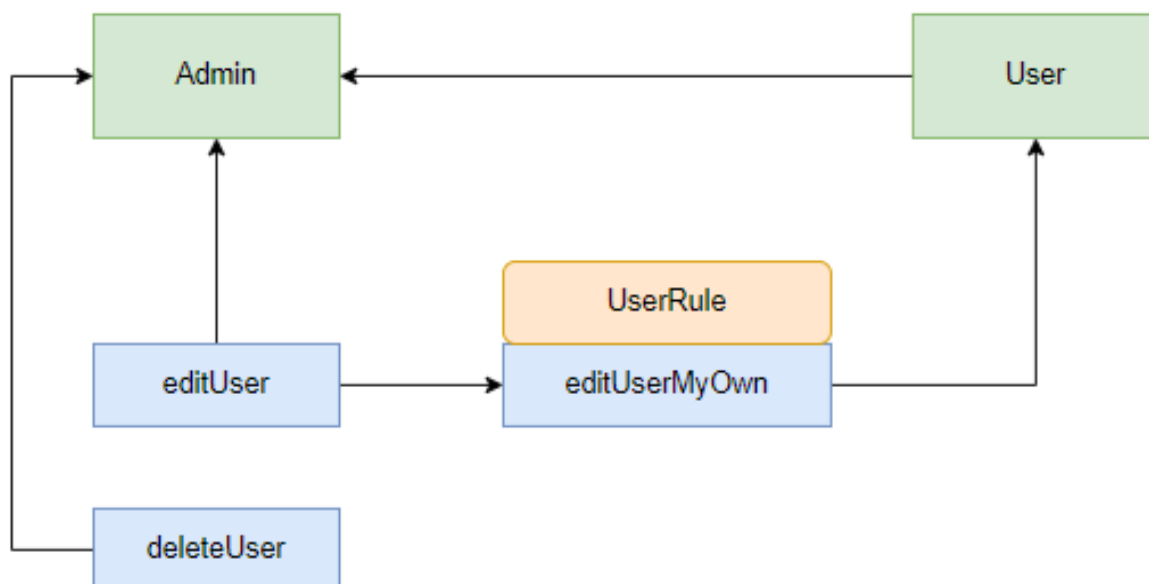


Рисунок 2.8. Розподіл дозволів для редагування та видалення профілю користувача

Згідно побудованої схеми, в системі потрібно створити:

- дві ролі: *Admin* та *User*. Роль *User* необхідно призначити всім користувачам, хто щойно зареєструвався, по замовчуванню.
- два правила: *ManagerRule* та *UserRule*. Правило *ManagerRule* перевірятиме чи відноситься деякий користувач до групи даного автентифікованого користувача. Правило *UserRule* перевіряти, чи є деякий користувач фактично даним автентифікованим користувачем.
- дозволи *manageUser*, *editUser*, *deleteUser*. Дозвіл *manageUser* буде використовуватися в діях на отримання профілю користувача, внесення та перегляд даних про стан його здоров'я. Дозвіл *editUser* перевірятиметься в дії на редагування профілю користувача. Дозвіл *deleteUser* перевірятиметься в дії на видалення профілю користувача.
- Дані дозволи *manageUser*, *editUser*, *deleteUser* віднесемо до ролі *Admin*.
- дозволи *manageUserMyOwn*, *manageUserMyGroup*, *editUserMyOwn*. До дозволів *manageUserMyOwn* та *manageUserMyGroup* будуть додані правила відповідно *UserRule* та *ManagerRule*. До дозволу *editUserMyOwn* буде додане правило *UserRule*.

- Дозвіл *manageUser* підпорядкуємо дозволам *manageUserMyOwn* та *manageUserMyGroup*. Дозвіл *editUser* підпорядкуємо дозволу *editUserMyOwn*.
- Дані дозволи *manageUserMyOwn*, *manageUserMyGroup*, *editUserMyOwn* віднесемо до ролі *User*.

Етап 1. Конфігурація менеджера авторизації. Створення таблиць БД по авторизації.

Крок 1. Налаштуємо менеджер авторизації. Для цього у файл *config/web.php* та *config/console.php* необхідно додати в масив *components* налаштування для *authManager*:

```
'components' => [
    'authManager' => [
        'class' => 'yii\rbac\DbManager',
    ],
],
```

Крок 2. Створимо таблиці авторизації. Для цього виконаємо команду міграції:

```
yii migrate --migrationPath=@yii/rbac/migrations
```

Дана команда створить таблиці *auth_item*, *auth_item_child*, *auth_assignment*, *auth_rule*, необхідні для збереження даних RBAC.

Етап 2. Створимо консольний клас *RbacController*. В ньому будуть команди для покрокового створення системи ролей та дозволів.

Крок 1. Створимо клас *app\commands\RbacController* (файл *commands/RbacController.php*):

```
<?php
namespace app\commands;

use Yii;
use yii\console\Controller;

class RbacController extends Controller
{
```

```
}
```

*Крок 2. Створимо команду init (метод *actionInit*):*

```
public function actionInit()  
{  
    $auth = Yii::$app->authManager;  
    $auth->removeAll();  
  
    $user = $auth->createRole('User');  
    $admin = $auth->createRole('Admin');  
  
    $auth->add($user);  
    $auth->add($admin);  
  
    $auth->addChild($admin, $user);  
  
    $auth->assign($admin, 1);  
    $auth->assign($user, 2);  
    $auth->assign($user, 3);  
    $auth->assign($user, 4);  
    $auth->assign($user, 5);  
    $auth->assign($user, 6);  
}
```

Даний метод створює дві ролі: User і Admin, підпорядковує роль User ролі Admin та призначає дані ролі відповідним користувачам.

*Крок 3. Забезпечимо автоматичне призначення ролі User користувачам, які реєструються або яких реєструють. Для цього в моделі *app\models\User* додамо метод *afterSave()*:*

```
public function afterSave($insert, $changedAttributes)  
{  
    if ($insert) {  
        $auth = Yii::$app->authManager;  
        $user = $auth->getRole("User");  
        $auth->assign($user, $this->getId());  
    }  
  
    parent::afterSave($insert, $changedAttributes);  
}
```

В даному методі ми отримуємо змінну \$auth як вказівник на компонент менеджера авторизації, отримуємо роль User та призначаємо її щойно зареєстрованому користувачу.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

Крок 4. Створимо два правила *app\rbac\ManagerRule* (файл *rbac/ManagerRule.php*) та *app\rbac\UserRule* (файл *rbac/UserRule.php*):

```
<?php

namespace app\rbac;

use app\models\GroupUser;
use yii\rbac\Rule;

/**
 * Checks if given user ($params['user']) is in group of current user
 */
class ManagerRule extends Rule
{
    public $name = 'ManagerRule';

    public function execute($user, $item, $params)
    {
        if ((GroupUser::findOne([
            'manager_id' => $user,
            'user_id' => $params['user']->user_id,
            'status' => GroupUser::STATUS_APPROVED
        ])) !== null) {
            return true;
        }
        return false;
    }
}
```

```
<?php

namespace app\rbac;

use yii\rbac\Rule;

/**
 * Checks if give user matches current user
 */
class UserRule extends Rule
{
    public $name = 'UserRule';
```

```

public function execute($user, $item, $params)
{
    return isset($params['user']) ? $params['user']->id ==
$user : false;
}
}

```

Ці правила будуть обмежувати доступ користувачам з роллю User до дій, пов'язаних з іншими користувачами. Їх ми віднесемо до потрібних дозволів.

Крок 5. Створимо команду *permissions* в *app\commands\RbacController* для додавання дозволів *manageUser*, *editUser*, *deleteUser*.

```

public function actionPermissions()
{
    $auth = Yii::$app->authManager;

    $manageUser = $auth->createPermission("manageUser");
    $editUser = $auth->createPermission("editUser");
    $deleteUser = $auth->createPermission("deleteUser");

    $auth->add($manageUser);
    $auth->add($editUser);
    $auth->add($deleteUser);

    $admin = $auth->getRole("Admin");
    $auth->addChild($admin, $manageUser);
    $auth->addChild($admin, $editUser);
    $auth->addChild($admin, $deleteUser);
}

```

Дана команда створить та збереже у БД три дозволи і підпорядкує їх ролі *Admin*.

Крок 6. Створимо команду *user-permissions* для додавання дозволів *manageUserMyOwn*, *manageUserMyGroup*, *editUserMyOwn*. з прив'язуванням до них правил *UserRule* та *ManagerRule*.

```

public function actionUserPermissions()
{
    $auth = Yii::$app->authManager;

    //Створюємо та зберігаємо екземпляри правил в БД
    $managerRule = new \app\rbac\ManagerRule();
    $userRule = new \app\rbac\UserRule();
    $auth->add($managerRule);
    $auth->add($userRule);
}

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

```

//Створюємо дозволи для користувача
$manageUserMyOwn = $auth->createPermission("manageUserMyOwn");
$manageUserMyGroup = $auth-
>createPermission("manageUserMyGroup");
$editUserMyOwn = $auth->createPermission("editUserMyOwn");

//Додаємо правила до створених дозволів
$manageUserMyOwn->ruleName = $userRule->name;
$manageUserMyGroup->ruleName = $managerRule->name;
$editUserMyOwn->ruleName = $userRule->name;

//Зберігаємо дозволи в БД
$auth->add($manageUserMyOwn);
$auth->add($manageUserMyGroup);
$auth->add($editUserMyOwn);

//Підпорядковуємо наявні дозволи щойно створеним
$manageUser = $auth->getPermission("manageUser");
$editUser = $auth->getPermission("editUser");
$auth->addChild($manageUserMyOwn, $manageUser);
$auth->addChild($manageUserMyGroup, $manageUser);
$auth->addChild($editUserMyOwn, $editUser);

//Відносимо створені дозволи до ролі User
$user = $auth->getRole("User");
$auth->addChild($user, $manageUserMyOwn);
$auth->addChild($user, $manageUserMyGroup);
$auth->addChild($user, $editUserMyOwn);
}

```

Дана команда створює екземпляри правил *UserRule* та *ManagerRule* та дозволи *manageUserMyOwn*, *manageUserMyGroup*, *editUserMyOwn*. До дозволів *manageUserMyOwn* та *manageUserMyGroup* додаються правила відповідно *UserRule* та *ManagerRule*. До дозволу *editUserMyOwn* додається правило *UserRule*. Дозвіл *manageUser* підпорядковується дозволам *manageUserMyOwn* та *manageUserMyGroup*. Дозвіл *editUser* підпорядковується дозволу *editUserMyOwn*. Дані дозволи *manageUserMyOwn*, *manageUserMyGroup*, *editUserMyOwn* віднесемо до ролі *User*.

Крок 7. Виконаємо по черзі створені команди:

```

php yii rbac/init
php yii rbac/permissions
php yii rbac/user-permissions

```

Після чого можна переглянути вміст таблиць системи авторизації та перевірити наявність всіх необхідних ролей та дозволів, їх відношень та призначень.

Висновки до розділу 2

На етапі проектування системи визначено варіанти використання для адміністраторів та зареєстрованих користувачів: наведено перелік основних функціональних можливостей, розподілених по відповідних ролях. Здійснено опис та проілюстровано діаграми варіантів використання, активностей, послідовностей та класів.

Розроблено структуру реляційної БД системи. Наведено опис всіх таблиць та їх відношень. Наведено приклад підготовки до розробки та безпосередньої розробки механізму авторизації на основі системи ролей та дозволів (RBAC): створення ролей і дозволів, формування їх відношень примінення правил та призначення ролей користувачам.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. КЕРІВНИЦТВО РОЗГОРТАННЯ І ВИКОРИСТАННЯ. ОПИС РЕЗУЛЬТАТІВ ТЕСТУВАННЯ

3.1. Керівництво розгортання системи

Для розгортання системи в робочому режимі виконаємо такі кроки:

Крок 1. Клонування проекту на локальний комп'ютер. Для цього необхідно встановити систему Git, створити папку для проекту, наприклад, /home/projects/health-monitoring-system, та в ній виконати команду:

- `git clone https://gitlab.com/<user>/task-employee-management-system`

Крок 2. Встановимо всі залежності. Для цього використаємо команду:

- `composer update`

Крок 3. Зареєструємо доменне ім'я та хостинг, наприклад на Хостинг Україна.

Крок 4. Створимо сайт. Для цього у вкладці панелі керування хостингом “Хостинг” перейдемо в розділ “Мої сайти” та створимо новий сайт:

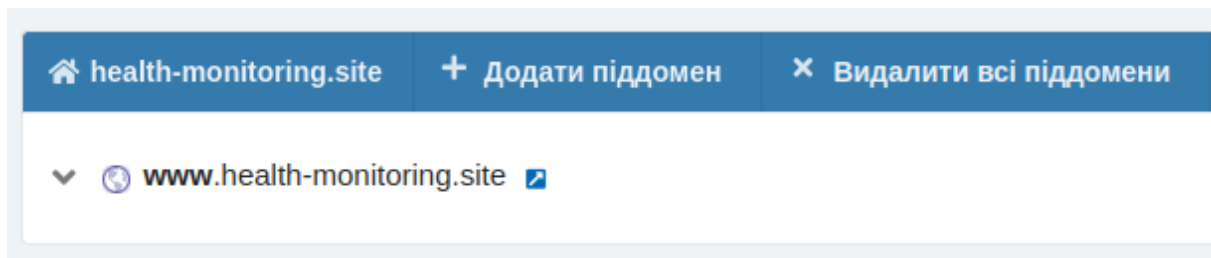


Рисунок 3.1. Створений веб-сайт у списку сайтів на панелі керування

ХОСТИНГОМ

Крок 5. На панелі керування хостингом створимо базу даних. Для цього в розділі MySQL перейдемо на сторінку *Бази даних* і натиснемо *Створити базу даних*:

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

Створення бази даних

×

Ім'я бази даних:

healthmonitor

10/23

Кодування:

utf8_general_ci

☒ Створити користувача з таким же ім'ям та повним доступом до бази даних

Створити

Ім'я бази даних може містити тільки символи латиниці і цифри.

Рисунок 3.2. Створення БД на сервері

Крок 6. Виконаємо імпорт бази даних. Для цього відкриємо інструмент РНРМуAdmin та скористаємось функцією імпорту з файлу, вибравши файл з дампом бази даних *yii_health_system.sql*. Результатом імпорту є 10 таблиць з демо-даними:

☐ auth_assignment

☐ auth_item

☐ auth_item_child

☐ auth_rule

☐ data_blood_pressure

☐ data_blood_sugar

☐ data_body_temperature

☐ group_user

☐ migration

☐ user

Рисунок 3.3. Таблиці БД після імпорту

Крок 7. Використовуючи файл-менеджер (FTP – Файл-менеджер) завантажимо zip-архів з вихідним кодом в папку */home/<username>/health-monitoring.site/www*. Після чого потрібно розпакувати архів:

account

/home/health-monitoring.site/www/

↑

↶

↷

🔍

🔖

>

🌀

☐	Ім'я	Тип	Розмір	Дата	Права
	..				
☐	assets		—	22.05.2023	rwXr-Xr-X
☐	commands		—	21.06.2023	rwXr-Xr-X
☐	config		—	23.06.2023	rwXr-Xr-X
☐	controllers		—	23.06.2023	rwXr-Xr-X
☐	mail		—	22.05.2023	rwXr-Xr-X
☐	models		—	23.06.2023	rwXr-Xr-X
☐	rbac		—	23.06.2023	rwXr-Xr-X
☐	runtime		—	17.06.2023	rwXr-Xr-X
☐	tests		—	22.05.2023	rwXr-Xr-X

Рисунок 3.4. Файли сайту після завантаження і розпакування архіву з вихідним кодом

Крок 8. Конфігурування з'єднання з БД. Відкриємо файл config/db.php та сконфігуруємо підключення до бази даних, вказавши *host*, *dbname*, *username* та *password* у відповідних налаштуваннях:

```
return [
    'class' => 'yii\db\Connection',
    'dsn' => 'mysql:host=<user>.mysql.tools;dbname=<user>_healthmonitor',
    'username' => '<user>_healthmonitor',
    'password' => '<password>',
    'charset' => 'utf8',
];
```

Лістинг 3.1. Конфігурація БД

Крок 9. Для того, щоб позбутися наявності папки web в URL, в корені проекту створимо файл .htaccess з наступним вмістом:

```
Options +SymLinksIfOwnerMatch
IndexIgnore */*
RewriteEngine On

RewriteCond %{REQUEST_URI} !^(web)
RewriteRule ^assets/(.*)$ /web/assets/$1 [L]
RewriteRule ^css/(.*)$ web/css/$1 [L]
RewriteRule ^js/(.*)$ web/js/$1 [L]
RewriteRule ^images/(.*)$ web/images/$1 [L]
RewriteRule (.*?) /web/$1
```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

```

RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /web/index.php

```

Лістинг 3.2. Конфігурація .htaccess

Таким чином, ми розгорнули систему в робочому режимі. Тепер вона доступна за доменною адресою *health-monitoring.site*:

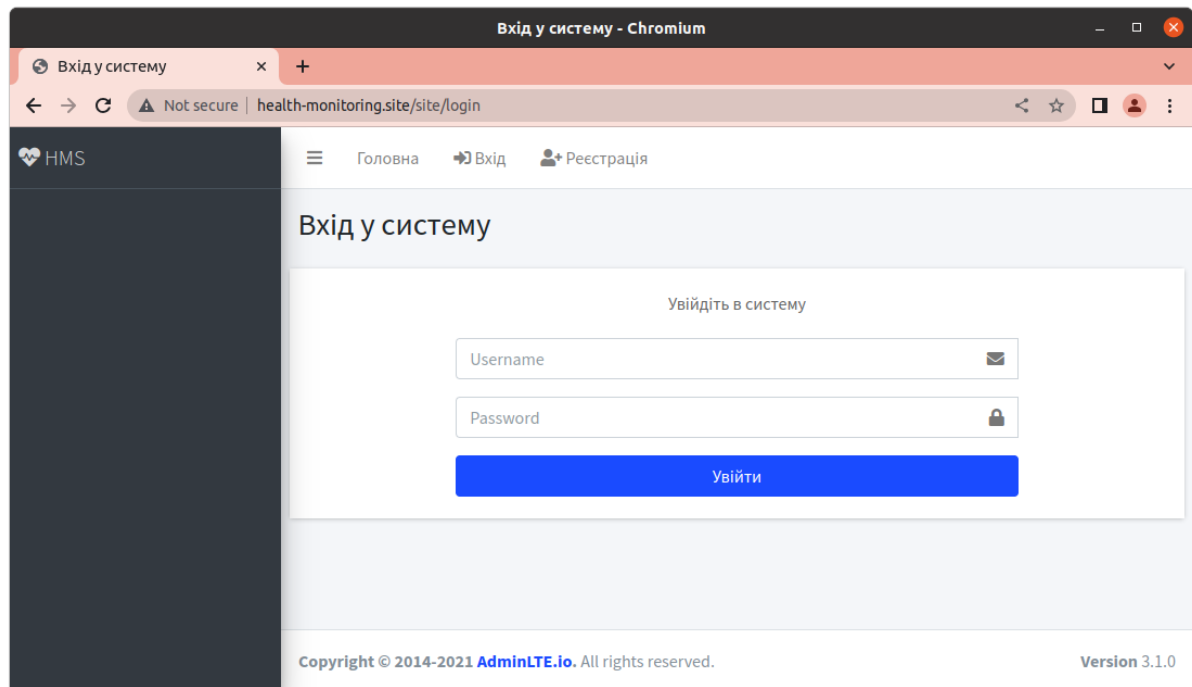


Рисунок 3.5. Встановлена система, готова до використання

3.2. Інструкція користувача

Сторінка реєстрації. Доступна для гостей системи. Кожен користувач може створити обліковий запис, ввівши логін, email та пароль:

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

Рисунок 3.6. Сторінка реєстрації

Навігація для користувача. Після реєстрації та входу для користувача доступні такі посилання: Головна, Мій профіль, Моя група, Запросити, Мої менеджери, Запрошення для мене:

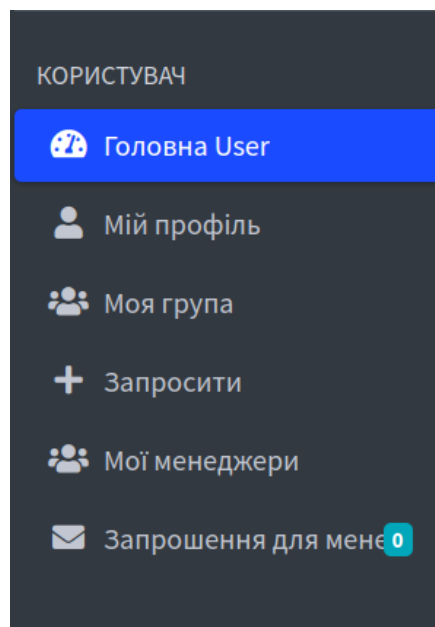


Рисунок 3.7. Навігація користувача

Головна сторінка користувача. Виводить статистику та зведену інформацію про стан здоров'я: кількості вимірювань показників та останні вимірювання.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

Мій профіль. Сторінка містить загальну інформацію про користувача: прізвище, ім'я, електронна адреса, фото, а також кнопки з посиланнями на редагування профілю та внесення даних про стан здоров'я:

Андрій Мельник Home / Люди / Андрій Мельник

Тиск і пульс Цукор Температура

Username	andrew
Email	andrew@gmail.com
Full Name	Андрій Мельник

Рисунок 3.8. Мій профіль

Показники артеріального тиску та пульсу. Список введених значень вимірювання артеріального тиску та пульсу. Дані можуть вводитись самим користувачем, його менеджером, або адміністратором сайту:

Покази артеріального тиску та пульсу Home / andrew / Покази артеріального тиску та пульсу

Користувач: andrew

Додати покази

Showing 1-3 of 3 items.

#	Ким введено	Систолічний тиск	Діастолічний тиск	Пульс	Дата і час вимірювання	
1	nicolas	120	70	72	2023-06-23 14:35:57	🗑️
2	nicolas	120	75	75	2023-06-23 22:55:34	🗑️
3	andrew	120	75	75	2023-06-23 23:25:32	🗑️

Рисунок 3.9. Показники АТ та пульсу

Внесення даних про стан здоров'я. На кожній сторінці показників здоров'я є кнопка для внесення даних про стан здоров'я, причому система контролює доступ користувачів до виконання відповідної дії.

Додати показники артеріального тиску [Home](#) / [nicolas](#) / [Історія AT](#) / Додати показники артеріального тиску

Користувач: nicolas

Систолічний тиск

120

Діастолічний тиск

70

Пульс

72

Дата і час вимірювання

📅

✕

2023-6-7 09:45:21

✓

Зберегти

Рисунок 3.10. Внесення значень артеріального тиску та пульсу

Моя група. Функціонал, що дає можливість користувачам керувати даними та здійснювати моніторинг стану інших користувачів, наприклад членів своєї сім'ї. Для формування власної групи, користувачу необхідно відправити запрошення, а інший користувач повинен або підтвердити або скасувати запрошення на вступ до групи:

Моя група [Home](#) / [Моя група](#)

Запросити користувача в групу

Showing 1-2 of 2 items.

#	User ID	Status	
1	andrew	Підтверджено	
2	peter	Підтверджено	

Рисунок 3.11. Список групи даного користувача

Користувачі, що сформували групу, мають доступ до перегляду профілів користувачів групи, внесення перегляду та видалення даних про їх стан здоров'я.

Навігація адміністратора. В системі передбачено роль Адміністратора – суперкористувача з логіном admin, що має доступ до перегляду, редагування, створення облікових записів користувачів, керування та моніторингу станом їх здоров'я. Інші користувачі теж можуть бути адміністраторами, що забезпечується механізмом системи ролей та дозволів, що зберігається в БД системи.

3.3. Особливості тестування системи моніторингу стану здоров'я

Тестування системи моніторингу стану здоров'я включає в себе декілька етапів для забезпечення її точності, надійності та ефективності. Наведемо загальний процес тестування системи:

Визначення цілей та завдань процесу тестування: які аспекти системи потрібно оцінити, наприклад перевірка доступу на основі ролей, точність вимірювань, надійність передачі даних, зручність використання інтерфейсу користувача тощо.

Визначення різних сценаріїв та умов, за яких перевірятиметься система, що може включати різні сценарії використання, умови середовища та потенційні сценарії збою.

Розробка тестів – створення конкретних тестів на основі визначених сценаріїв тестування, які охоплюють широкий діапазон ситуацій. Кожен тестовий приклад повинен мати чітку мету, вхідні дані та очікувані результати.

Підготовка тестового середовища: налаштування необхідної інфраструктури для створення тестового середовища.

Виконання тестових випадків: систематичний запуск тестових випадків, дотримуючись визначених процедур, документування результатів кожного тесту, включаючи будь-які спостереження, відхилення або проблеми, які виникли під час тестування.

Оцінка зручності використання та інтерфейсу користувача системи моніторингу здоров'я. Це можна зробити за допомогою відгуків користувачів, опитувань або проведення сеансів тестування зручності використання з представниками користувачів.

Стрес-тестування – піддавання системи моніторингу працездатності стрес-тестам для оцінки її продуктивності за великих навантажень або екстремальних умов. Це може передбачати симуляцію високого трафіку користувачів, тривалих періодів моніторингу або ресурсомістких сценаріїв для перевірки стабільності та стійкості системи.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

Перевірка безпеки та конфіденційності для виявлення вразливих місць та забезпечення відповідності відповідним нормам і стандартам.

Важливо зазначити, що конкретний процес тестування може відрізнятися залежно від типу системи моніторингу здоров'я, передбачуваного використання та будь-яких нормативних вимог, яких необхідно дотримуватися. Подумайте про залучення експертів у галузі, медичних працівників і кінцевих користувачів до процесу тестування, щоб отримати цінні відгуки та інформацію.

Визначимо наступну послідовність запитів для тестування системи:

Гість системи:

- Всі запити повинні перенаправлятися на сторінку із входом

Адміністратор системи:

- Перегляд списку користувачів, створення нових користувачів, редагування профілю, видалення.
- Внесення та редагування даних про стан здоров'я кожного користувача.

Користувач системи:

- Перегляд лише власного профілю, а також профілів користувачів власної групи; заборонено доступ до перегляду профілю користувачів інших груп;
- Керування даними про стан здоров'я лише власного профілю, а також профілів користувачів власної групи. Заборонено доступ до керування станом здоров'я користувачів інших груп.

В таблиці 3.1 наведемо послідовність запитів визначеного сценарію, очікувану відповідь та результат тесту. Попередньо в системі зареєстровано одного користувача, який є адміністратором:

Таблиця 3.1

Процес виконання тестового сценарію

Запит	Очікуваний результат запиту	Результат тесту
Вхід під адміністратором	Сторінка з блоком навігації для адміністратора	Пройдено
Створення користувача 1	Сторінка з формою, запит на додавання користувача в БД	Пройдено

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

Перегляд списку користувачів	Сторінка зі списком користувачів	Пройдено
Редагування профілю користувача 1	Сторінка з формою, оновлення даних в БД	Пройдено
Внесення даних про стан здоров'я користувача 1	Сторінка з формою, внесення даних в БД	Пройдено
Видалення даних про стан здоров'я користувача 1	Діалогове вікно підтвердження, безпосередньо видалення з БД	Пройдено
Самостійна реєстрація користувача 2	Форма реєстрації, додавання користувача в БД	Пройдено
Вхід користувача 2	Аутентифікація в системі	Пройдено
Перегляд власного профілю	Сторінка з профілем	Пройдено
Перегляд профілю користувача 1	Сторінка зі статусам 403 (заборона доступу)	Пройдено
Внесення даних про свій стан здоров'я	Сторінка з формою, додавання даних в БД	Пройдено
Внесення даних про стан здоров'я користувача 1	Сторінка зі статусам 403 (заборона доступу)	Пройдено
Надсилання запиту на додавання в свою групу користувача 1	Сторінка "Моя група" з інформацією про надіслане запрошення	Пройдено
Вхід під користувачем 1. Підтвердження запрошення	Кнопка для підтвердження запрошення. Оновлення статусу запрошення в БД	Пройдено
Вхід під користувачем 2. Перегляд профілю користувача 1	Сторінка з профілем	Пройдено
Внесення даних про стан здоров'я користувача 1	Форма для внесення даних. Збереження даних в БД.	Пройдено

Таким чином, даний сценарій дозволить здійснити перевірку доступу до дій та їх виконання авторизованим користувачам. Крім функціонального тестування були також проведені такі типи тестування: зручності використання, продуктивності, тестування безпеки та швидкодії, регресійне тестування. Оскільки система знаходиться в стані активної розробки з додаванням нових функцій, постійним виправленням помилок, в даний процес необхідно інтегрувати етапи періодичного тестування для забезпечення безперебійної роботи системи, її функціональності, надійності та ефективності.

Висновки до розділу 3

Для розгортання і користування системою моніторингу стану здоров'я було розроблено і наведено відповідні інструкції. Система має вихідний код та дампи бази даних, що міститься на репозиторії за адресою <https://gitlab.com/<username>/health-monitoring-system>, що дає можливість розгорнути її власноруч у робочому режимі або режимі розробки. Також користувач має можливість користуватись системою за адресою <https://health-monitoring.site>.

Наведено результати тестування сценарію для перевірки доступу до дій та їх виконання авторизованим користувачам з ролями Admin та User. Крім функціонального тестування були також проведені такі типи тестування: зручності використання, продуктивності, тестування безпеки та швидкодії, регресійне тестування. Оскільки система знаходиться в стані активної розробки з додаванням нових функцій, постійним виправленням помилок, в даний процес необхідно інтегрувати етапи періодичного тестування для забезпечення безперебійної роботи системи, її функціональності, надійності та ефективності.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

ВИСНОВКИ

В випускній кваліфікаційній роботі бакалавра визначено основні типові завдання системи моніторингу стану здоров'я, базуючись на яких здійснено формулювання задачі роботи по розробці веб-орієнтованої системи для моніторингу стану здоров'я. Розглянуто декілька сучасних веб-систем для моніторингу стану здоров'я та здійснено їх порівняння по таким критеріям, як інтеграція з пристроями, сфера застосування, персоналізація та аналітика, інтеграція з охороною здоров'я, інтерфейс користувача. Виділено ряд типових функцій для систем моніторингу здоров'я. Обґрунтовано вибір стеку технологій для розробки системи, враховуючи такі вимоги як економічна ефективність, безпека, надійність, швидкість розробки, гнучкість, масштабованість, документація та активна підтримка спільноти розробників.

На етапі проектування системи визначено варіанти використання для адміністраторів та зареєстрованих користувачів: наведено перелік основних функціональних можливостей, розподілених по відповідних ролях. Здійснено опис та проілюстровано діаграми варіантів використання, активностей, послідовностей та класів.

Розроблено структуру реляційної БД системи. Наведено опис всіх таблиць та їх відношень. Наведено приклад підготовки до розробки та безпосередньої розробки механізму авторизації на основі системи ролей та дозволів (RBAC): створення ролей і дозволів, формування їх відношень примінення правил та призначення ролей користувачам.

Для розгортання і користування системою моніторингу стану здоров'я було розроблено і наведено відповідні інструкції. Система має вихідний код та дампи бази даних, а також робочу версію за адресою <https://health-monitoring.site>.

Наведено результати тестування сценарію для перевірки доступу до дій та їх виконання авторизованим користувачам. Крім функціонального тестування були також проведені такі типи тестування: зручності використання, продуктивності, тестування безпеки та швидкодії, регресійне тестування. На

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

даний час система знаходиться в стані активної розробки з додаванням нових функцій, постійним виправленням помилок, тому в даний процес необхідно інтегрувати етапи періодичного тестування для забезпечення безперебійної роботи системи, її функціональності, надійності та ефективності.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Yii2 framework. Керівництво користувача [Електронний ресурс] - Режим доступу до ресурсу: <https://www.yiiframework.com>
2. Документація мови програмування PHP [Електронний ресурс] - Режим доступу до ресурсу: <http://php.net>
3. Gamma, E., Helm, R., Johnson, R., Vlissides, J. M. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional. ISBN: 0201633612
4. Chavan, Megha & Pardeshi, Prajakta & Khoje, Suchitra & Patil, Manasvi. (2018). Study of Health Monitoring System. 1779-1784. 10.1109/ICCONS.2018.8663063.
5. Kang M, Park E, Cho BH, Lee KS. Recent Patient Health Monitoring Platforms Incorporating Internet of Things-Enabled Smart Devices. Int Neurourol J. 2018 Jul;22(Suppl 2):S76-82. doi: 10.5213/inj.1836144.072. Epub 2018 Jul 31. Erratum in: Int Neurourol J. 2018 Dec;22(4):313. PMID: 30068069; PMCID: PMC6077937.
6. Kalid N, Zaidan AA, Zaidan BB, Salman OH, Hashim M, Muzammil H. Based real time remote health monitoring systems: a review on patients prioritization and related “big data” using body sensors information and communication technology. J Med Syst. 2017;42:30.
7. Ключко О.М. Медична інформаційна система моніторингу стану здоров'я населення із захистом персональних даних. Медична інформатика та інженерія, К, 2020, V.49. – №1 – С. 17-28.
8. Ключко О.М. Інформаційно-комп'ютерні технології в біології та медицині. К: НАУ-друк. 2008, 252 с.
9. Saleh, Shawki & Cherradi, Bouchaib & el Gannour, Oussama & Gouiza, Nissrine & Bouattane, Omar. (2022). Healthcare monitoring system for automatic database management using mobile application in IoT environment. 14. 10.11591/eei.v12i2.4282.
10. Saleh, Shawki & Cherradi, Bouchaib & Laghmati, Sara & el Gannour, Oussama & Hamida, Soufiane & Bouattane, Omar. (2023). Healthcare Embedded System for

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

Predicting Parkinson's Disease Based on AI of Things.
10.1109/IRASET57153.2023.10153040.

11.Alzboon, Mowafaq. (2022). Survey on Patient Health Monitoring System Based on Internet of Things. Information Sciences Letters. 11. 1371-1384.
10.18576/isl/110418.

12.Yi, Won-Jae & Saniie, Jafar. (2016). Patient Centered Real-Time Mobile Health Monitoring System. E-Health Telecommunication Systems and Networks. 05. 75-94.
10.4236/etsn.2016.54007.

13.Michał Romańczuk. Solid PHP - SOLID principles in PHP [Електронний ресурс] - Режим доступу до ресурсу: <https://accesto.com/blog/solid-php-solid-principles-in-php/>

14.Kanimozhi, M & Yamuna, I & Srimathi, B & Rajendran, Senthil Kumaran. (2021). Development of Health Monitoring Vest using Velostat. 1-5.
10.1109/ICSCAN53069.2021.9526519.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг класу контролера DataBloodSugar для обробки запитів по керуванню
даними артеріального тиску та пульсу

```
<?php

namespace app\controllers;

use app\models\DataBloodSugar;
use app\models\DataBloodSugarSearch;
use Yii;
use yii\web\Controller;
use yii\web\ForbiddenHttpException;
use yii\web\NotFoundHttpException;
use yii\filters\VerbFilter;

/**
 * DataBloodSugarController implements the CRUD actions for DataBloodSugar model.
 */
class DataBloodSugarController extends Controller
{
    /**
     * @inheritdoc
     */
    public function behaviors()
    {
        return array_merge(
            parent::behaviors(),
            [
                'verbs' => [
                    'class' => VerbFilter::className(),
                    'actions' => [
                        'delete' => ['POST'],
                    ],
                ],
            ]
        );
    }

    /**
     * Lists all DataBloodSugar models.
     */
}
```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

```

* @return string
*/
public function actionIndex($user_id)
{
    if (!Yii::$app->user->can('manageUser', ['user_id' => $user_id])) {
        throw new ForbiddenHttpException("You can't manage this user");
    }

    $searchModel = new DataBloodSugarSearch();
    $dataProvider = $searchModel->search($this->request->queryParams);

    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
    ]);
}

/**
 * Creates a new DataBloodSugar model.
 * If creation is successful, the browser will be redirected to the 'view' page.
 * @return string|\yii\web\Response
 */
public function actionCreate()
{
    $model = new DataBloodSugar();

    if ($this->request->isPost) {
        if ($model->load($this->request->post()) && $model->save()) {
            return $this->redirect(['view', 'id' => $model->id]);
        }
    } else {
        $model->loadDefaultValues();
    }

    return $this->render('create', [
        'model' => $model,
    ]);
}

/**
 * Deletes an existing DataBloodSugar model.
 * If deletion is successful, the browser will be redirected to the 'index' page.

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

```

* @param int $id ID
* @return \yii\web\Response
* @throws NotFoundHttpException if the model cannot be found
*/
public function actionDelete($id)
{
    $this->findModel($id)->delete();

    return $this->redirect(['index']);
}

/**
 * Finds the DataBloodSugar model based on its primary key value.
 * If the model is not found, a 404 HTTP exception will be thrown.
 * @param int $id ID
 * @return DataBloodSugar the loaded model
 * @throws NotFoundHttpException if the model cannot be found
 */
protected function findModel($id)
{
    if (($model = DataBloodSugar::findOne(['id' => $id])) !== null) {
        return $model;
    }

    throw new NotFoundHttpException('The requested page does not exist.');
```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК В

Лістинг моделі DataBloodSugar для логіки роботи з даними артеріального тиску та пульсу

```
<?php

namespace app\models;

use Yii;

/**
 * This is the model class for table "data_blood_sugar".
 *
 * @property int $id
 * @property int $user_id
 * @property int $manager_id
 * @property int $testing_type
 * @property int $sugar_level
 * @property string $date_measure
 *
 * @property User $manager
 * @property User $user
 */
class DataBloodSugar extends \yii\db\ActiveRecord
{
    /**
     * {@inheritdoc}
     */
    public static function tableName()
    {
        return 'data_blood_sugar';
    }

    /**
     * {@inheritdoc}
     */
    public function rules()
    {
        return [
            [['user_id', 'manager_id', 'testing_type', 'sugar_level', 'date_measure'],
                'required'],
            [['user_id', 'manager_id', 'testing_type', 'sugar_level'], 'integer'],
            [['date_measure'], 'safe'],
            [['user_id'], 'exist', 'skipOnError' => true, 'targetClass' => User::class,
                'targetAttribute' => ['user_id' => 'id']],
            [['manager_id'], 'exist', 'skipOnError' => true, 'targetClass' => User::class,
                'targetAttribute' => ['manager_id' => 'id']],
        ];
    }

    /**
     * {@inheritdoc}
     */
    public function attributeLabels()
    {
        return [

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

```

'id' => 'ID',
'user_id' => 'User ID',
'manager_id' => 'Manager ID',
'testing_type' => 'Testing Type',
'sugar_level' => 'Sugar Level',
'date_measure' => 'Date Measure',
];
}

/**
 * Gets query for [[Manager]].
 *
 * @return \yii\db\ActiveQuery
 */
public function getManager()
{
    return $this->hasOne(User::class, ['id' => 'manager_id']);
}

/**
 * Gets query for [[User]].
 *
 * @return \yii\db\ActiveQuery
 */
public function getUser()
{
    return $this->hasOne(User::class, ['id' => 'user_id']);
}
}

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64

Лістинг шаблону сторінки з формою для додавання даних кров'яного тиску та пульсу

```
<?php

use kartik\date\DatePicker;
use kartik\datetime\DateTimePicker;
use yii\bootstrap5\ActiveForm;
use yii\helpers\Html;

/** @var yii\web\View $this */
/** @var app\models\DataBloodPressure $model */

$this->title = 'Додати показники артеріального тиску';
$this->params['breadcrumbs'][] = ['label' => $user->username, 'url' =>
['user/view', 'id' => $user->id]];
$this->params['breadcrumbs'][] = ['label' => 'Історія АТ', 'url' => ['index',
'user_id' => $user->id]];
$this->params['breadcrumbs'][] = $this->title;
?>
<div class="data-blood-pressure-create">

<h1>Користувач: <?= $user->username ?></h1>
<div class="data-blood-pressure-form">

<?php $form = ActiveForm::begin(); ?>

<?= $form->field($model, 'systolic')->textInput() ?>

<?= $form->field($model, 'diastolic')->textInput() ?>

<?= $form->field($model, 'pulse')->textInput() ?>

<?= $form->field($model, 'date_measure')->widget(DateTimePicker::classname(), [
'options' => ['placeholder' => 'Введіть дату і час отримання даних ...'],
'pluginOptions' => [
'autoclose' => true,
'format' => 'yyyy-m-d hh:ii:ss'
]
]); ?>

<div class="form-group">
<?= Html::submitButton('Зберегти', ['class' => 'btn btn-success']) ?>
</div>

<?php ActiveForm::end(); ?>

</div>

</div>
```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		