

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Кафедра програмної інженерії

(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Бакалавр

(назва освітнього ступеня)

на тему: Розробка програмного забезпечення для автоматизації та моніторингу  
спортивних показників тренувального процесу в біатлоні

Виконав(ла): студент(ка) 4 курсу, групи СП-41

спеціальності 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

\_\_\_\_\_  
(підпис)

Сидяга С. В.  
(прізвище та ініціали)

Керівник

\_\_\_\_\_  
(підпис)

Коноваленко І. В.

(прізвище та ініціали)

Нормоконтроль

\_\_\_\_\_  
(підпис)

Стоянов Ю. В.

(прізвище та ініціали)

Завідувач кафедри

\_\_\_\_\_  
(підпис)

Петрик М. Р.

(прізвище та ініціали)

Рецензент

\_\_\_\_\_  
(підпис)

\_\_\_\_\_  
(прізвище та ініціали)

Тернопіль  
2026

Міністерство освіти і науки України  
**Тернопільський національний технічний університет імені Івана Пулюя**

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра програмної інженерії  
(повна назва кафедри)

ЗАТВЕРДЖУЮ  
Завідувач кафедри

\_\_\_\_\_ (підпис) \_\_\_\_\_ (прізвище та ініціали)  
«\_\_» \_\_\_\_\_ 2026 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр  
(назва освітнього ступеня)

за спеціальністю 121 Інженерія програмного забезпечення  
(шифр і назва спеціальності)

Студенту Сидяга Соломія Віталіївна  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка програмного забезпечення для автоматизації та моніторингу спортивних показників тренувального процесу в біатлоні

Керівник роботи Коноваленко Ігор Володимирович, к.т.н., доцент кафедри АВ  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «\_\_» \_\_\_\_\_ 2026 року № \_\_\_\_

2. Термін подання студентом завершеної роботи \_\_\_\_\_ 2026 р.

3. Вихідні дані до роботи \_\_\_\_\_

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ 1. Аналіз предметної області. 2. Моделювання та проектування та програмного забезпечення 3. Тестування системи 4. Охорона праці та безпека в надзвичайних ситуаціях.

Висновки. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)  
Слайди презентації

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання \_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Ознайомлення з завданням до кваліфікаційної роботи	21.04.2026	<i>Виконано</i>
2	Підбір та опрацювання наукових публікацій, збір даних по темі роботи	21.04.2026-27.04.2026	<i>Виконано</i>
3	Виконання дослідження згідно теми кваліфікаційної роботи	28.04.2026-11.05.2026	<i>Виконано</i>
4	Оформлення розділу «Аналіз предметної області»	12.05.2026-18.05.2026	<i>Виконано</i>
5	Оформлення розділу «Моделювання та проектування та програмного забезпечення»	19.05.2026-25.05.2026	<i>Виконано</i>
6	Оформлення розділу «Тестування системи»	26.05.2026-01.06.2026	<i>Виконано</i>
7	Виконання завдання до підрозділу «Безпека в надзвичайних ситуаціях»	19.05.2026-01.06.2026	<i>Виконано</i>
8	Виконання завдання до підрозділу «Охорона праці»	19.05.2026-01.06.2026	<i>Виконано</i>
9	Оформлення кваліфікаційної роботи	02.06.2026-04.06.2026	<i>Виконано</i>
10	Попередній захист кваліфікаційної роботи		<i>Виконано</i>
11	Перевірка на плагіат		<i>Виконано</i>
12	Нормоконтроль		<i>Виконано</i>
13	Захист кваліфікаційної роботи		

Студент

\_\_\_\_\_ (підпис)

Сидяга С. В.

\_\_\_\_\_ (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ (підпис)

Коноваленко І. В.

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Розробка програмного забезпечення для автоматизації та моніторингу спортивних показників тренувального процесу в біатлоні // Кваліфікаційна робота освітнього рівня «Бакалавр» // Сидяга Соломія Віталіївна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра програмної інженерії, група СП-41 // Тернопіль, 2026 // С. \_\_\_\_, рис. – \_\_\_\_, табл. – \_\_\_\_, додат. – \_\_\_\_, бібліогр. – \_\_\_\_.

*Ключові слова:* біатлон; спортивні показники; моніторинг тренувань; автоматизація; .NET; PostgreSQL; мобільний застосунок; статистика стрільби.

Кваліфікаційна робота присвячена розробці програмного забезпечення для автоматизації та моніторингу спортивних показників тренувального процесу в біатлоні. Проаналізовано особливості підготовки біатлоністів, сучасні програмні засоби спортивного моніторингу та підходи до обробки тренувальних даних. Сформовано функціональні й нефункціональні вимоги до системи. Спроектовано архітектуру програмного забезпечення, структуру бази даних, модулі управління тренуваннями, аналізу спортивних показників та статистики стрільби. Реалізовано програмну систему із використанням платформи .NET та системи керування базами даних PostgreSQL.

Об'єкт дослідження – процес збору, зберігання та аналізу спортивних показників під час тренувального процесу біатлоністів.

Предмет дослідження – методи, моделі та програмні засоби автоматизації моніторингу тренувальної діяльності спортсменів у біатлоні.

Практичним результатом роботи є програмна система для ведення тренувань, обліку фізичних показників спортсменів, аналізу результатів стрільби та формування статистичних звітів. Система забезпечує централізоване зберігання тренувальних даних і підтримує процес прийняття рішень тренером під час планування та оцінювання тренувального процесу.

## ABSTRACT

Development of Software for Automation and Monitoring of Sports Performance Indicators in the Biathlon Training Process // Bachelor's Qualification Thesis // Solomiia Sydiaha// Ivan Puluj Ternopil National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Software Engineering, Group SP-41 // Ternopil, 2026 // P. \_\_\_\_, Fig. – \_\_\_\_, Tables – \_\_\_\_, Appendices – \_\_\_\_, References – \_\_\_\_.

Keywords: biathlon; sports performance indicators; training monitoring; automation; .NET; PostgreSQL; mobile application; shooting statistics.

The qualification thesis is devoted to the development of software for automation and monitoring of sports performance indicators in the biathlon training process. The specifics of biathlete training, modern sports monitoring software solutions, and approaches to training data processing were analyzed. Functional and non-functional requirements for the system were defined. The software architecture, database structure, training management modules, sports performance analysis module, and shooting statistics module were designed. The software system was implemented using the .NET platform and the PostgreSQL database management system.

The object of research is the process of collecting, storing, and analyzing sports performance indicators during the training process of biathletes.

The subject of research is methods, models, and software tools for automating the monitoring of athletes' training activities in biathlon.

The practical result of the work is a software system for training management, recording athletes' physical performance indicators, analyzing shooting results, and generating statistical reports. The system provides centralized storage of training data and supports coaches in decision-making during the planning and evaluation of the training process.

## ЗМІСТ

ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИЗНАЧЕННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ	11
1.1 Особливості тренувального процесу в біатлоні та спортивних показників спортсменів	11
1.2 Аналіз сучасних програмних рішень для моніторингу спортивної діяльності	13
1.3 Аналіз методів збору, зберігання та обробки спортивних даних	14
1.4 Визначення функціональних та нефункціональних вимог до програмної системи	17
1.5 Аналіз користувачів системи та сценаріїв використання	19
1.6 Постановка задачі кваліфікаційної роботи	21
1.7 Висновки до першого розділу	22
2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ	23
2.1 Загальна архітектура програмної системи	23
2.2 Проєктування структури бази даних	27
2.3 Проєктування ролей користувачів та механізму авторизації	29
2.4 Проєктування функціонального модуля ведення тренувань	31
2.5 Проєктування модуля статистики стрільби	33
2.6 Проєктування модуля формування звітів	34
2.7 Проєктування користувацького інтерфейсу мобільного застосунку	36
2.8 Висновки до другого розділу	39
3 ТЕСТУВАННЯ, ОЦІНЮВАННЯ ТА ВПРОВАДЖЕННЯ ПРОГРАМНОЇ СИСТЕМИ	40
3.1 Обґрунтування вибору технологій розробки	40

3.2	Тестування серверної частини та бази даних системи	42
3.3	Тестування функціональних модулів системи	44
3.4	Тестування користувацького інтерфейсу мобільного застосунку	46
3.5	Оцінка працездатності програмної системи	47
3.6	Розгортання системи та перспективи розвитку	49
3.7	Висновки до розділу 3	50
4	ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	52
4.1	Значення адаптації в трудовому процесі	52
4.2	Інженерно-технічні рішення з охорони праці	55
	ВИСНОВКИ	59
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	61
	ДОДАТКИ	64
	ДОДАТОК А	64

## ПЕРЕЛІК СКОРОЧЕНЬ

API – Application Programming Interface (інтерфейс прикладного програмування)

ASP.NET – Active Server Pages .NET (платформа для розробки вебзастосунків)

DB – Database (база даних)

DTO – Data Transfer Object (об'єкт передачі даних)

ERD – Entity Relationship Diagram (діаграма сутність-зв'язок)

GPS – Global Positioning System (глобальна система позиціонування)

HTTP – HyperText Transfer Protocol (протокол передачі гіпертексту)

HTTPS – HyperText Transfer Protocol Secure (захищений протокол передачі гіпертексту)

JSON – JavaScript Object Notation (текстовий формат обміну даними)

ORM – Object-Relational Mapping (об'єктно-реляційне відображення)

REST – Representational State Transfer (архітектурний стиль вебсервісів)

SQL – Structured Query Language (мова структурованих запитів)

UI – User Interface (користувацький інтерфейс)

UML – Unified Modeling Language (уніфікована мова моделювання)

URL – Uniform Resource Locator (уніфікований локатор ресурсу)

CPU – Central Processing Unit (центральний процесор)

RAM – Random Access Memory (оперативна пам'ять)

.NET – програмна платформа для розробки застосунків від Microsoft

IDE – Integrated Development Environment (інтегроване середовище розробки)

SDK – Software Development Kit (набір засобів для розробки програмного забезпечення)

HR – Heart Rate (частота серцевих скорочень)

BPM – Beats Per Minute (ударів за хвилину)

## ВСТУП

Сучасний розвиток інформаційних технологій створює нові можливості для автоматизації процесів у різних сферах діяльності людини, зокрема у спорті. Використання програмних засобів для збору, обробки та аналізу спортивних даних дозволяє підвищити ефективність тренувального процесу, забезпечити об'єктивний контроль фізичного стану спортсменів та покращити якість прийняття рішень тренерами. Особливо актуальним це є для біатлону — виду спорту, який поєднує високі фізичні навантаження з необхідністю підтримання точності стрільби та контролю великої кількості тренувальних показників.

У сучасній практиці підготовки спортсменів широко використовуються різноманітні пристрої моніторингу фізичної активності, системи GPS-відстеження, датчики серцевого ритму та програмні сервіси спортивної аналітики. Однак більшість існуючих рішень орієнтовані на загальні види спорту та не враховують специфіку біатлону, пов'язану з необхідністю комплексного аналізу показників пересування, фізичного навантаження та результатів стрільби. Крім того, використання декількох окремих систем призводить до фрагментації даних та ускладнює процес їх аналізу.

Актуальність теми кваліфікаційної роботи зумовлена необхідністю створення програмного забезпечення, яке забезпечуватиме централізований збір, зберігання та аналіз спортивних показників біатлоністів, автоматизацію ведення тренувальної діяльності та формування статистичних звітів для тренерів і спортсменів. Використання сучасних програмних технологій дозволяє створити єдиний інформаційний простір для підтримки процесів планування, моніторингу та оцінювання результатів тренувань.

Метою кваліфікаційної роботи є розробка програмного забезпечення для автоматизації та моніторингу спортивних показників тренувального процесу в біатлоні, яке забезпечує централізоване зберігання даних про спортсменів, ведення тренувань, аналіз результатів стрільби та формування статистичної інформації.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати особливості тренувального процесу в біатлоні та існуючі програмні рішення у сфері спортивного моніторингу;
- визначити функціональні та нефункціональні вимоги до програмної системи;
- спроектувати архітектуру програмного забезпечення та структуру бази даних;
- розробити модулі управління спортсменами, тренуваннями та спортивними показниками;
- реалізувати функціонал збору, зберігання та аналізу тренувальних даних;
- реалізувати модуль формування статистики та звітності;
- провести тестування розробленого програмного забезпечення та оцінити його працездатність.

Об'єктом дослідження є процес моніторингу та аналізу спортивних показників під час тренувального процесу біатлоністів.

Предметом дослідження є методи, моделі та програмні засоби автоматизації збору, зберігання й аналізу спортивних даних у системах підтримки тренувального процесу.

Практичне значення отриманих результатів полягає у створенні програмної системи, що дозволяє автоматизувати облік тренувань, централізувати спортивні дані, аналізувати результати спортсменів та підтримувати прийняття рішень під час організації тренувального процесу. Розроблена система може бути використана спортивними школами, тренерами та спортсменами для підвищення ефективності підготовки у біатлоні.

# **1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИЗНАЧЕННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ**

Ефективність сучасного тренувального процесу значною мірою залежить від можливості збору, зберігання та аналізу спортивних показників спортсменів. Використання інформаційних технологій у спортивній діяльності дозволяє автоматизувати облік тренувань, здійснювати моніторинг фізичного стану спортсменів та оцінювати динаміку їхніх результатів. Особливо актуальним це є для біатлону, де успішність спортсмена визначається не лише рівнем фізичної підготовки, а й точністю стрільби, стабільністю результатів та здатністю підтримувати високу працездатність протягом усього тренувального циклу. У цьому розділі проведено аналіз особливостей тренувального процесу в біатлоні, розглянуто існуючі програмні рішення, визначено вимоги до майбутньої програмної системи та сформовано постановку задачі кваліфікаційної роботи [1].

## **1.1 Особливості тренувального процесу в біатлоні та спортивних показників спортсменів**

Біатлон є складним циклічним видом спорту, який поєднує лижні перегони та стрільбу по мішенях. Особливістю даного виду спорту є необхідність одночасного розвитку витривалості, швидкості, координації рухів, психологічної стійкості та високої точності стрільби. Підготовка спортсменів включає значну кількість тренувальних занять різної інтенсивності, результати яких потребують постійного контролю та аналізу.

Сучасний тренувальний процес у біатлоні передбачає систематичний збір великої кількості спортивних даних. Тренери та спортсмени аналізують тривалість тренувань, пройдену дистанцію, швидкість пересування, частоту серцевих скорочень, результати стрільби та суб'єктивну оцінку самопочуття спортсмена. Накопичення та обробка такої інформації дозволяють оцінювати ефективність тренувального процесу та своєчасно коригувати індивідуальні плани підготовки.

Особливу роль відіграють показники функціонального стану організму спортсмена. Контроль частоти серцевих скорочень дозволяє оцінити рівень фізичного навантаження та ефективність відновлення після тренувань. Водночас результати стрільби характеризують рівень технічної підготовки спортсмена та його здатність виконувати точні дії після значних фізичних навантажень.

З розвитком цифрових технологій усе більшої популярності набувають програмні системи спортивного моніторингу, які забезпечують автоматизований збір та аналіз тренувальних показників. Такі системи дозволяють централізовано зберігати інформацію про спортсменів, формувати статистичні звіти та відстежувати динаміку спортивних результатів протягом тривалого часу.

Основні показники, які характеризують тренувальний процес біатлоніста, наведено в таблиці 1.1.

Таблиця 1.1 – Основні спортивні показники тренувального процесу біатлоніста

<b>Показник</b>	<b>Характеристика</b>	<b>Практичне значення</b>
Тривалість тренування	Загальний час виконання тренування	Оцінка обсягу навантаження
Дистанція	Загальна пройдена відстань	Аналіз витривалості спортсмена
Середній пульс	Середнє значення ЧСС під час тренування	Контроль інтенсивності навантаження
Максимальний пульс	Найвище значення ЧСС	Виявлення пікових навантажень
Мінімальний пульс	Найнижче значення ЧСС	Аналіз відновлення організму
Кількість пострілів	Загальна кількість виконаних пострілів	Оцінка обсягу стрілецької підготовки
Кількість промахів	Загальна кількість неточних пострілів	Аналіз ефективності стрільби
Самопочуття спортсмена	Суб'єктивна оцінка стану	Контроль перевтоми та відновлення
Нотатки тренування	Додаткова інформація про тренування	Аналіз особливостей виконання

Отже, тренувальний процес у біатлоні характеризується необхідністю комплексного контролю фізичних, функціональних та технічних показників спортсмена. Це обумовлює потребу у створенні спеціалізованого програмного

забезпечення, яке забезпечуватиме автоматизацію збору, зберігання та аналізу тренувальних даних, а також підтримку процесу прийняття рішень тренером і спортсменом.

## **1.2 Аналіз сучасних програмних рішень для моніторингу спортивної діяльності**

Сучасний розвиток цифрових технологій сприяв появі великої кількості програмних засобів для моніторингу спортивної діяльності. Такі системи використовуються для збору, зберігання та аналізу даних про фізичну активність спортсменів, планування тренувань, контролю функціонального стану організму та оцінювання результатів спортивної підготовки. Використання спеціалізованого програмного забезпечення дозволяє тренерам та спортсменам отримувати об'єктивну інформацію про тренувальний процес і приймати обґрунтовані рішення щодо його коригування.

Більшість сучасних спортивних платформ інтегруються з фітнес-трекерами, смарт-годинниками, датчиками серцевого ритму та GPS-пристроями. Це забезпечує автоматичний збір інформації про тривалість тренування, пройдену дистанцію, швидкість руху, частоту серцевих скорочень та інші показники. Отримані дані можуть використовуватися для формування статистичних звітів, аналізу прогресу спортсмена та прогнозування його спортивної форми.

Серед найбільш популярних програмних рішень для моніторингу спортивної діяльності можна виділити Garmin Connect, Polar Flow, TrainingPeaks та Strava. Кожна з цих систем має власні особливості, переваги та обмеження, що визначають сферу її використання.

Для порівняння функціональних можливостей найбільш поширених систем спортивного моніторингу складено таблицю 1.2.

Таблиця 1.2 – Порівняння сучасних програмних рішень для моніторингу спортивної діяльності

Система	Моніторинг тренувань	Аналіз пульсу	GPS-треки	Формування звітів	Статистика стрільби	Орієнтація на біатлон
Garmin Connect	Так	Так	Так	Так	Ні	Ні
Polar Flow	Так	Так	Так	Так	Ні	Ні
Strava	Так	Частково	Так	Частково	Ні	Ні
TrainingPeaks	Так	Так	Так	Так	Ні	Ні
SBiathlonTrack (розроблювана система)	Так	Так	Так	Так	Так	Так

Як видно з таблиці 1.2, існуючі програмні рішення забезпечують широкий набір функцій для моніторингу фізичної активності спортсменів. Водночас вони орієнтовані переважно на загальну спортивну аналітику та не враховують специфічні особливості біатлону. Зокрема, відсутні засоби обліку результатів стрільби, аналізу кількості промахів, контролю використання патронів та інтегрованого аналізу фізичних і стрілецьких показників.

На основі аналізу існуючих систем можна зробити висновок, що сучасні програмні продукти успішно вирішують завдання збору та візуалізації спортивних даних, проте не забезпечують комплексного підходу до моніторингу тренувального процесу біатлоністів. Це обумовлює доцільність розробки спеціалізованого програмного забезпечення, яке поєднуватиме функції ведення тренувань, моніторингу фізичних показників, аналізу результатів стрільби та формування статистичної звітності в межах єдиної інформаційної системи.

### 1.3 Аналіз методів збору, зберігання та обробки спортивних даних

Ефективність сучасного тренувального процесу значною мірою залежить від якості збору, зберігання та обробки спортивних даних. Використання цифрових технологій дозволяє автоматизувати процес отримання інформації про фізичну активність спортсменів, накопичувати історію тренувань та виконувати

комплексний аналіз результатів. Для біатлону це має особливе значення, оскільки підготовка спортсмена включає контроль як фізичних показників, так і результатів стрільби, що потребує обробки значних обсягів різнорідної інформації.

Збір спортивних даних може здійснюватися автоматично або вручну. Автоматизований збір реалізується за допомогою спортивних годинників, фітнес-трекерів, датчиків серцевого ритму та GPS-пристроїв. Такі засоби дозволяють отримувати дані про тривалість тренування, швидкість пересування, пройдену дистанцію, частоту серцевих скорочень та інші показники в режимі реального часу. Водночас частина інформації, зокрема результати стрільби, суб'єктивна оцінка самопочуття спортсмена та коментарі тренера, можуть вноситися користувачами вручну.

У сучасних інформаційних системах спортивного моніторингу найбільш поширеними джерелами даних є:

- GPS-пристрої для визначення маршруту та дистанції;
- датчики серцевого ритму для контролю фізичного навантаження;
- мобільні застосунки для реєстрації тренувань;
- спортивні годинники та фітнес-трекери;
- ручне введення результатів тренувань і стрільби;
- анкетування спортсменів щодо самопочуття та рівня втоми.

Після отримання інформації важливим етапом є її зберігання. Для цього використовуються системи керування базами даних, які забезпечують структуроване розміщення інформації та швидкий доступ до неї. У спортивних інформаційних системах зазвичай зберігаються дані про спортсменів, тренування, фізичні показники, результати стрільби та статистичні звіти. Централізоване зберігання інформації дозволяє накопичувати історичні дані та здійснювати аналіз результатів за тривалий період часу.

Основні категорії спортивних даних наведено в таблиці 1.3.

Наступним етапом є обробка спортивних даних. Її основна мета полягає у перетворенні накопиченої інформації на корисні аналітичні відомості для

спортсмена та тренера. Для цього використовуються методи фільтрації даних, агрегації показників, статистичного аналізу та візуалізації результатів.

Таблиця 1.3 – Категорії спортивних даних у системах моніторингу

Категорія даних	Приклади показників	Призначення
Дані спортсмена	ПІБ, вік, стать, спортивний розряд	Ідентифікація спортсмена
Тренувальні дані	Дата, тривалість, тип тренування	Облік тренувальної діяльності
Фізичні показники	Пульс, швидкість, дистанція	Аналіз фізичного навантаження
GPS-дані	Координати, маршрут, висота	Аналіз пересування спортсмена
Дані стрільби	Кількість пострілів, промахи, точність	Оцінка технічної підготовки
Аналітичні дані	Статистика, графіки, звіти	Підтримка прийняття рішень

У системах спортивного моніторингу найбільш поширеними є такі методи обробки даних:

- обчислення середніх, мінімальних та максимальних значень показників;
- аналіз динаміки спортивних результатів у часі;
- порівняння результатів різних тренувань;
- формування статистичних звітів;
- побудова графіків та діаграм;
- виявлення тенденцій зміни спортивної форми спортсмена.

Для програмної системи моніторингу тренувального процесу в біатлоні особливе значення має можливість комплексного аналізу фізичних показників та результатів стрільби. Поєднання цих даних дозволяє оцінювати ефективність підготовки спортсмена, визначати вплив фізичних навантажень на точність стрільби та виявляти фактори, що впливають на спортивні результати.

Отже, сучасні методи збору, зберігання та обробки спортивних даних забезпечують можливість ефективного контролю тренувального процесу та формування обґрунтованих рішень щодо підготовки спортсменів. Для задач

моніторингу біатлоністів доцільним є використання централізованої програмної системи, яка поєднує механізми збору даних, їх надійного зберігання та засоби аналітичної обробки в межах єдиного інформаційного середовища.

#### **1.4 Визначення функціональних та нефункціональних вимог до програмної системи**

На основі проведеного аналізу предметної області та існуючих програмних рішень було сформовано вимоги до програмної системи моніторингу спортивних показників тренувального процесу в біатлоні. Вимоги визначають перелік функцій, які повинна реалізовувати система, а також характеристики якості, надійності та продуктивності програмного забезпечення.

Функціональні вимоги описують дії, які повинна виконувати система для забезпечення потреб користувачів, а саме спортсменів і тренерів. Основні функціональні вимоги наведено в таблиці 1.4.

Таблиця 1.4 – Функціональні вимоги до програмної системи

<b>Код</b>	<b>Функціональна вимога</b>
FR-01	Реєстрація та авторизація користувачів у системі
FR-02	Управління обліковими записами спортсменів
FR-03	Управління обліковими записами тренерів
FR-04	Створення, редагування та видалення тренувань
FR-05	Збереження інформації про тренувальні сесії
FR-06	Введення показників фізичної активності спортсмена
FR-07	Облік результатів стрільби під час тренувань
FR-08	Зберігання історії тренувальних даних
FR-09	Перегляд статистики спортсменів
FR-10	Формування звітів про тренувальну діяльність
FR-11	Пошук та фільтрація тренувальних даних
FR-12	Відображення аналітичної інформації у графічному вигляді
FR-13	Перегляд динаміки спортивних показників
FR-14	Розмежування прав доступу між спортсменом і тренером
FR-15	Централізоване зберігання спортивних даних

Окрім функціональних можливостей, система повинна відповідати ряду нефункціональних вимог, які визначають якість її роботи та зручність використання.

До основних нефункціональних вимог належать:

- система повинна забезпечувати коректну роботу в середовищі операційних систем Android та iOS (у разі мобільної реалізації) або сучасних веббраузерів;
- інтерфейс користувача повинен бути інтуїтивно зрозумілим та зручним для використання спортсменами і тренерами;
- час відгуку системи на дії користувача не повинен перевищувати декількох секунд за стандартних умов експлуатації;
- система повинна забезпечувати цілісність та надійність збереження даних;
- доступ до персональних даних користувачів повинен здійснюватися лише після проходження процедури автентифікації;
- програмне забезпечення повинно підтримувати можливість подальшого розширення функціональності;
- архітектура системи повинна забезпечувати можливість супроводу та модифікації програмного коду;
- система повинна забезпечувати захист від втрати даних у разі виникнення помилок або збоїв;
- база даних повинна підтримувати одночасну роботу декількох користувачів;
- програмне забезпечення повинно забезпечувати стабільну роботу під час обробки великих обсягів тренувальної інформації.

Сформовані функціональні та нефункціональні вимоги є основою для подальшого проектування архітектури програмної системи, структури бази даних та реалізації програмних модулів моніторингу спортивних показників біатлоністів.

## 1.5 Аналіз користувачів системи та сценаріїв використання

Програмна система моніторингу спортивних показників у біатлоні призначена для автоматизації процесів ведення тренувальної діяльності, зберігання спортивних даних та аналізу результатів спортсменів. Для визначення функціональності системи необхідно встановити основних користувачів та сценарії їхньої взаємодії з програмним забезпеченням.

Основними користувачами системи є:

- Спортсмен – здійснює перегляд власних тренувань, спортивних показників, результатів стрільби та статистичних звітів;
- Тренер – керує інформацією про спортсменів, створює тренування, аналізує результати та формує звітність;
- Адміністратор – забезпечує управління обліковими записами користувачів та підтримку працездатності системи.

На рисунку 1.2 наведено діаграму варіантів використання програмної системи.

Відповідно до діаграми варіантів використання основними сценаріями роботи системи є:

- реєстрація та авторизація користувачів;
- перегляд профілю спортсмена;
- створення та редагування тренувань;
- внесення результатів тренувальної діяльності;
- облік результатів стрільби;
- перегляд історії тренувань;
- аналіз спортивних показників;
- формування статистичних звітів;
- пошук та фільтрація тренувальних даних;
- управління обліковими записами користувачів.



Рисунок 1.2 – Діаграма варіантів використання програмної системи моніторингу спортивних показників у біатлоні

Спортсмен взаємодіє із системою переважно для перегляду власних результатів, аналізу тренувальних показників та відстеження особистого прогресу. Тренер використовує систему для ведення тренувального процесу, контролю спортивної форми спортсменів та аналізу ефективності підготовки. Адміністратор забезпечує належне функціонування програмної системи та виконує завдання з управління користувачами.

Таким чином, визначення користувачів та сценаріїв використання дозволило сформулювати основні вимоги до функціональності програмної системи та підготувати основу для подальшого проектування її архітектури й програмних модулів.

## 1.6 Постановка задачі кваліфікаційної роботи

Проведений аналіз предметної області показав, що існуючі програмні рішення для моніторингу спортивної діяльності не забезпечують комплексної підтримки специфічних потреб біатлону. Більшість систем орієнтовані на загальний облік фізичної активності та не враховують особливості аналізу результатів стрільби, централізованого ведення тренувань і накопичення спеціалізованої спортивної статистики.

Метою кваліфікаційної роботи є розробка програмного забезпечення для автоматизації та моніторингу спортивних показників тренувального процесу в біатлоні, яке забезпечуватиме централізоване зберігання даних про спортсменів, ведення тренувальної діяльності, облік результатів стрільби та формування аналітичної інформації для тренерів і спортсменів.

Для досягнення поставленої мети необхідно виконати такі завдання:

- провести аналіз особливостей тренувального процесу в біатлоні та сучасних програмних засобів спортивного моніторингу;
- визначити функціональні та нефункціональні вимоги до програмної системи;
- спроектувати архітектуру програмного забезпечення та структуру бази даних;
- розробити механізми управління спортсменами та тренуваннями;
- реалізувати функціонал обліку фізичних показників і результатів стрільби;
- реалізувати засоби аналізу та візуалізації спортивних даних;
- забезпечити формування статистичних звітів про тренувальну діяльність;
- провести тестування програмної системи та оцінити її працездатність.

Результатом виконання кваліфікаційної роботи має стати програмна система, що забезпечує автоматизацію основних процесів тренувальної діяльності

біатлоністів, централізоване зберігання спортивних даних та підтримку прийняття рішень під час аналізу результатів підготовки спортсменів.

### **1.7 Висновки до першого розділу**

У першому розділі кваліфікаційної роботи проведено аналіз предметної області, пов'язаної з автоматизацією та моніторингом спортивних показників тренувального процесу в біатлоні. Розглянуто особливості підготовки біатлоністів, основні спортивні показники та сучасні підходи до збору, зберігання й обробки тренувальних даних. Також виконано аналіз існуючих програмних рішень для спортивного моніторингу, що дозволило визначити їхні переваги та обмеження щодо використання у біатлоні.

На основі проведеного дослідження сформовано функціональні та нефункціональні вимоги до програмної системи, визначено основних користувачів та сценарії їхньої взаємодії із системою. Виконана постановка задачі дозволила окреслити основні напрями розробки програмного забезпечення, яке забезпечуватиме ведення тренувальної діяльності, облік фізичних показників, аналіз результатів стрільби та формування статистичної звітності. Отримані результати є основою для подальшого проєктування архітектури та реалізації програмної системи.

## 2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

У цьому розділі розглянуто процес проєктування програмної системи моніторингу спортивних показників біатлоністів. Визначено загальну архітектуру програмного забезпечення, структуру бази даних, основні програмні модулі та механізми взаємодії між компонентами системи. Особливу увагу приділено проєктуванню рішення, яке забезпечує централізоване зберігання спортивних даних, управління тренувальним процесом, аналіз результатів стрільби та формування статистичної інформації для спортсменів і тренерів.

### 2.1 Загальна архітектура програмної системи

Програмна система моніторингу спортивних показників біатлоністів реалізована на основі клієнт-серверної архітектури. Такий підхід забезпечує централізоване зберігання даних, розмежування відповідальності між компонентами системи та можливість подальшого масштабування програмного забезпечення.

Для розробки серверної частини використовується платформа .NET та мова програмування C#. Сервер забезпечує обробку запитів користувачів, виконання бізнес-логіки, взаємодію з базою даних та формування відповідей для клієнтського застосунку. Для зберігання інформації використовується система керування базами даних PostgreSQL, у якій розміщуються дані про спортсменів, тренування, спортивні показники, результати стрільби та статистичні звіти.

Клієнтська частина системи забезпечує взаємодію користувачів із програмним забезпеченням та надає інтерфейси для роботи зі спортивними даними. Основними користувачами системи є спортсмени та тренери. Через клієнтський застосунок вони можуть створювати тренування, переглядати статистику, аналізувати результати стрільби та формувати звітність.

Логічно архітектуру системи можна поділити на такі рівні:

- рівень представлення (Presentation Layer), що забезпечує взаємодію користувачів із системою;
- рівень бізнес-логіки (Business Logic Layer), у якому реалізовано обробку спортивних даних та правила функціонування системи;
- рівень доступу до даних (Data Access Layer), який відповідає за взаємодію з базою даних;
- рівень зберігання даних (Database Layer), представлений базою даних PostgreSQL.

Загальна архітектура програмної системи наведена на рисунку 2.1.

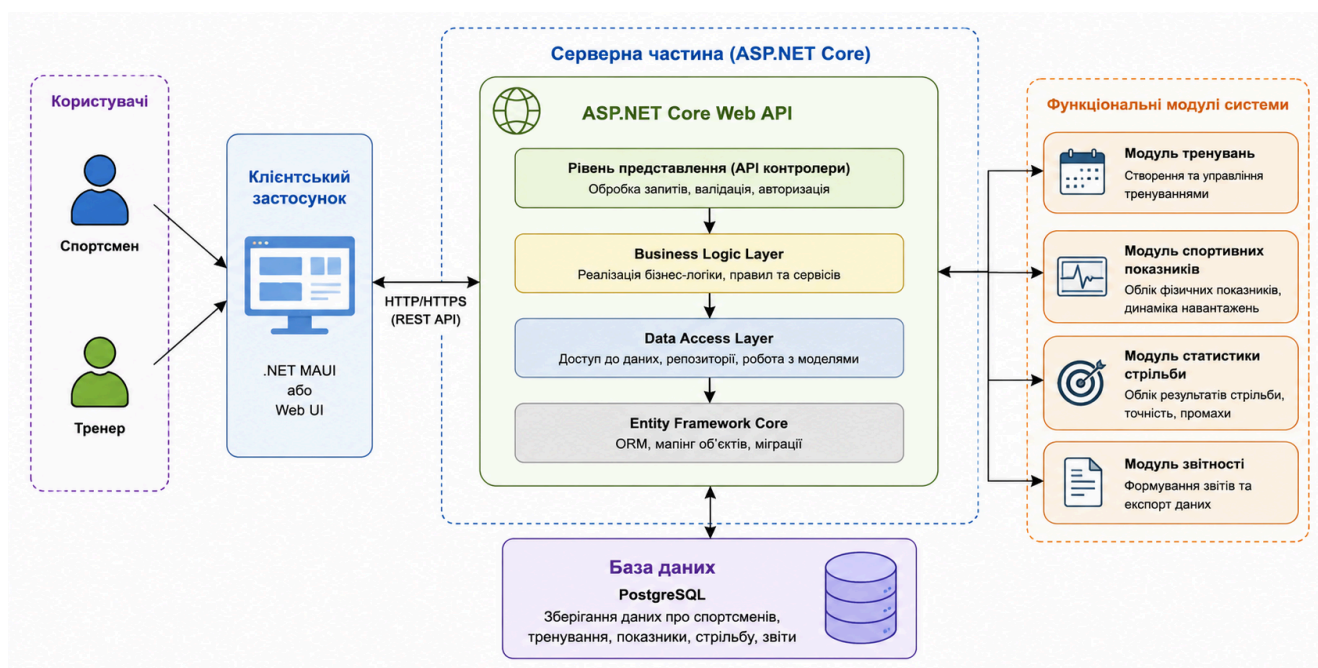


Рисунок 2.1 – Загальна архітектура програмної системи моніторингу спортивних показників біатлоністів

Як показано на рисунку 2.1, користувачі взаємодіють із клієнтським застосунком, який надсилає запити до серверної частини системи. Сервер обробляє отримані дані, виконує необхідні перевірки та звертається до бази даних для отримання або збереження інформації. Отримані результати повертаються користувачу у вигляді статистики, звітів та інформації про тренувальну діяльність.

Запропонована архітектура забезпечує модульність програмного забезпечення, спрощує його супровід та створює можливість подальшого розширення функціональності системи без суттєвих змін її основних компонентів.

Для деталізації внутрішньої структури програмної системи було розроблено діаграму класів, яка відображає основні програмні сутності, їх атрибути та взаємозв'язки. Діаграма класів дозволяє формалізувати структуру програмного забезпечення та визначити логіку взаємодії між окремими компонентами системи.

Основу предметної області складають класи користувачів, спортсменів, тренувань, спортивних показників, результатів стрільби та звітів. Клас `User` використовується для зберігання інформації про користувачів системи та реалізації механізму авторизації. Від нього логічно пов'язані класи `Athlete` та `Coach`, які представляють спортсменів і тренерів відповідно.

Центральним класом системи є `TrainingSession`, який містить інформацію про окреме тренування спортсмена. Для кожного тренування можуть зберігатися фізичні показники у класі `SportMetrics` та результати стрільби у класі `ShootingResult`. На основі накопичених даних формується клас `Report`, який використовується для створення статистичних звітів та аналітичної інформації.

Використання об'єктно-орієнтованого підходу дозволяє забезпечити модульність програмного забезпечення, спростити його супровід та створити можливість подальшого розширення функціональності системи без внесення суттєвих змін до вже реалізованих компонентів.

Діаграму класів програмної системи наведено на рисунку 2.2.

Діаграма класів відображає об'єктно-орієнтовану структуру програмної системи та демонструє основні сутності предметної області, їхні атрибути, методи та взаємозв'язки. Використання такого підходу дозволяє формалізувати архітектуру програмного забезпечення ще на етапі проектування та забезпечити чіткий розподіл відповідальності між окремими компонентами системи.

Особливістю розробленої діаграми є виділення окремих класів для представлення користувачів, тренувань, спортивних показників, результатів стрільби та звітів. Така структура забезпечує низьку зв'язаність між компонентами

та дозволяє розширювати функціональність системи без необхідності суттєвих змін у вже реалізованих програмних модулях.

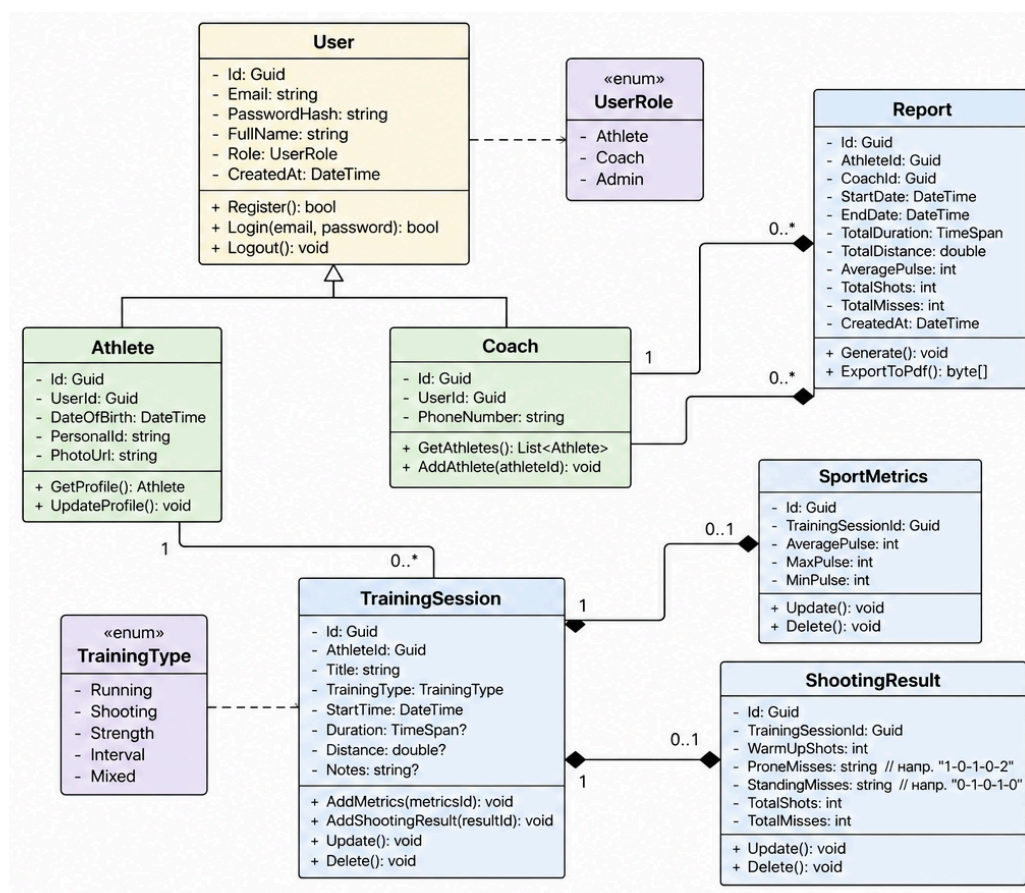


Рисунок 2.2 – Діаграма класів програмної системи моніторингу спортивних показників біатлоністів

Клас `TrainingSession` виступає центральною сутністю системи та об'єднує більшість функціональних можливостей програмного забезпечення. Саме через нього здійснюється взаємодія зі спортивними показниками та результатами стрільби. Це відповідає логіці предметної області, оскільки тренування є основним джерелом даних для подальшого аналізу спортивної діяльності біатлоніста.

Виділення окремих класів `SportMetrics` та `ShootingResult` дозволяє зберігати фізичні показники та результати стрільби незалежно один від одного. Такий підхід спрощує обробку даних, підвищує гнучкість програмної системи та створює

можливість подальшого розширення переліку контрольованих спортивних параметрів.

Клас Report реалізує функціональність формування аналітичної інформації на основі накопичених тренувальних даних. Завдяки цьому забезпечується підтримка прийняття рішень тренером та спортсменом щодо ефективності тренувального процесу і подальшого планування спортивної підготовки.

## 2.2 Проектування структури бази даних

База даних є одним із ключових компонентів програмної системи, оскільки забезпечує централізоване зберігання інформації про користувачів, спортсменів, тренування, спортивні показники та результати стрільби. Для реалізації системи обрано систему керування базами даних PostgreSQL, яка забезпечує високу продуктивність, надійність та підтримку реляційної моделі даних.

Структура бази даних спроектована відповідно до функціональних вимог системи та охоплює всі основні сутності тренувального процесу біатлоністів. Основною метою проектування є забезпечення цілісності даних, мінімізація дублювання інформації та підтримка ефективного виконання запитів.

До складу бази даних входять такі основні таблиці:

- Users – облікові записи користувачів системи;
- Athletes – інформація про спортсменів;
- Coaches – інформація про тренерів;
- TrainingSessions – відомості про проведені тренування;
- SportMetrics – спортивні показники спортсменів під час тренувань;
- ShootingResults – результати стрільби;
- Reports – сформовані звіти та статистичні дані.

На рисунку 2.3 наведено ER-діаграму бази даних програмної системи.

Центральною сутністю бази даних є таблиця Athletes, яка містить інформацію про спортсменів та пов'язана з тренувальними даними. Для кожного спортсмена може бути зареєстровано необмежену кількість тренувань, що

реалізується зв'язком типу «один до багатьох» між таблицями Athletes та TrainingSessions.

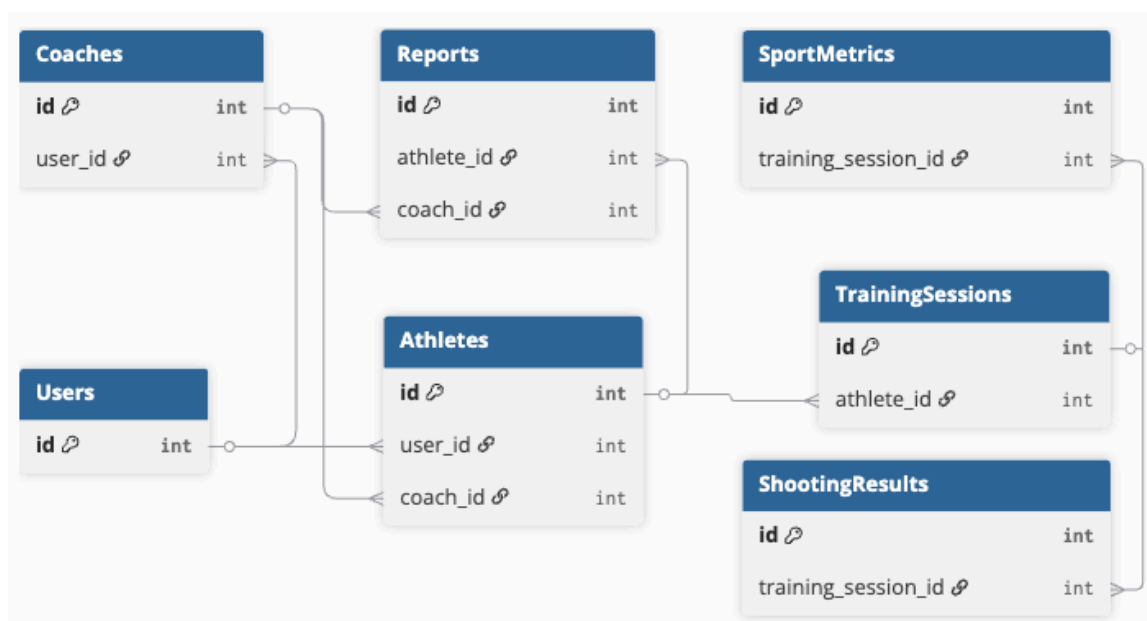


Рисунок 2.3 – ER-діаграма бази даних програмної системи

Таблиця TrainingSessions містить основну інформацію про тренувальні заняття, зокрема дату проведення, тривалість, тип тренування та додаткові примітки. Кожне тренування може містити декілька спортивних показників та результатів стрільби, тому між таблицями TrainingSessions, SportMetrics та ShootingResults також встановлено зв'язки типу «один до багатьох».

Таблиця SportMetrics використовується для зберігання фізичних показників спортсмена, зокрема частоти серцевих скорочень, дистанції, швидкості пересування та інших параметрів тренувального процесу. Таблиця ShootingResults містить інформацію про кількість пострілів, кількість промахів, точність стрільби та інші показники стрілецької підготовки.

Для забезпечення розмежування доступу в системі використовуються таблиці Users та Coaches. Таблиця Users містить дані, необхідні для автентифікації та авторизації користувачів, а таблиця Coaches зберігає інформацію про тренерів, які здійснюють управління спортсменами та аналіз тренувальної діяльності.

Таблиця Reports призначена для зберігання сформованих статистичних звітів та аналітичної інформації, отриманої на основі накопичених спортивних даних. Це дозволяє відстежувати динаміку результатів спортсменів та підтримувати процес прийняття рішень під час планування тренувального процесу.

Запропонована структура бази даних забезпечує логічну організацію інформації, підтримує необхідні зв'язки між сутностями та створює основу для реалізації функціональних можливостей програмної системи моніторингу спортивних показників біатлоністів.

### **2.3 Проектування ролей користувачів та механізму авторизації**

У програмній системі моніторингу спортивних показників біатлоністів передбачено розмежування доступу користувачів відповідно до їхніх ролей. Такий підхід дозволяє забезпечити контроль доступу до функцій системи, захист персональних даних та коректну організацію взаємодії між спортсменами і тренерами.

У системі визначено три основні ролі користувачів:

- спортсмен – має доступ до власного профілю, перегляду тренувань, додавання тренувальних даних, перегляду статистики стрільби та формування власних звітів;
- тренер – має доступ до списку закріплених спортсменів, перегляду їхніх тренувань, аналізу спортивних показників, статистики стрільби та формування звітів;
- адміністратор – відповідає за управління обліковими записами користувачів і підтримку працездатності системи.

Механізм авторизації реалізується через перевірку облікових даних користувача. Під час входу користувач вводить електронну пошту або ідентифікатор та пароль. Серверна частина системи перевіряє наявність користувача у базі даних, коректність введеного пароля та визначає роль

користувача. Після успішної авторизації користувач отримує доступ лише до тих функцій, які відповідають його ролі.

Послідовність процесу авторизації користувача наведено на рисунку 2.4.

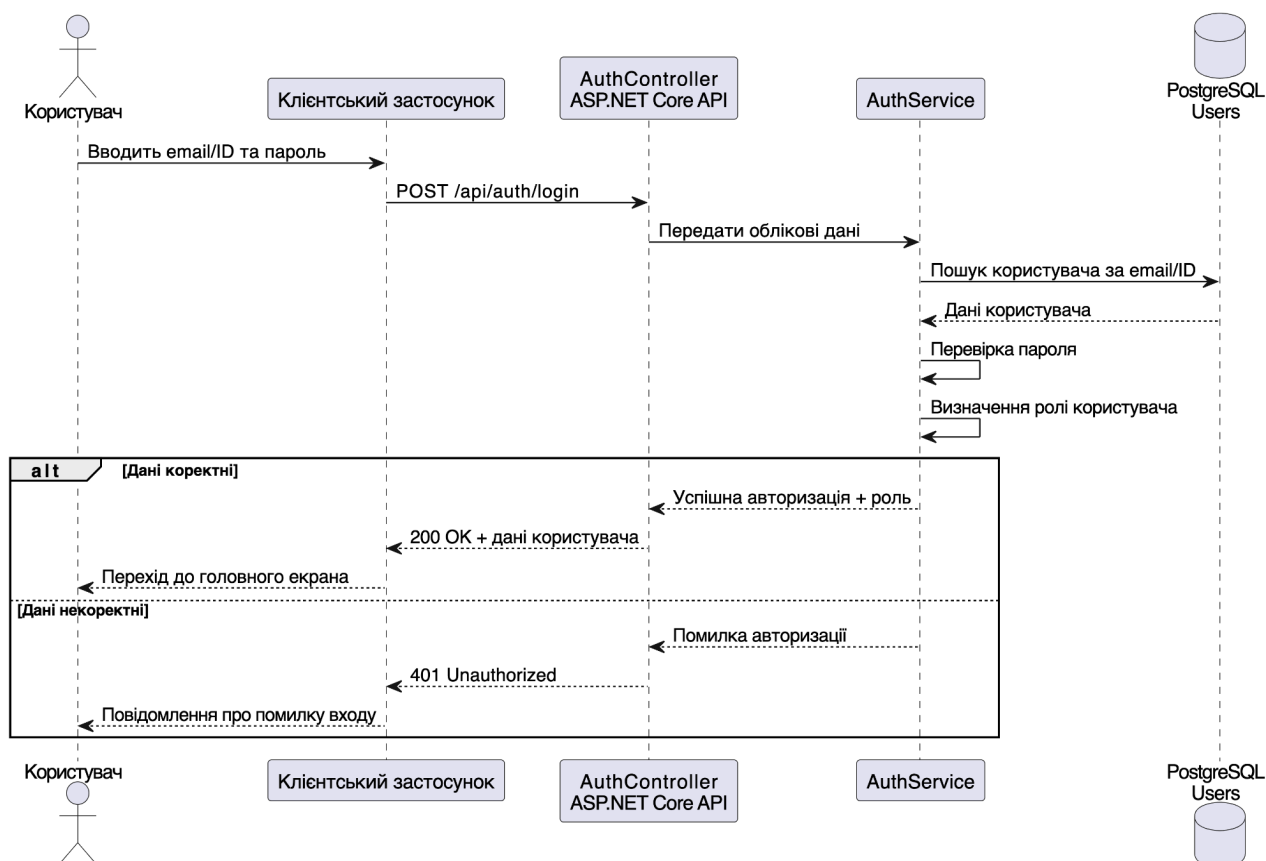


Рисунок 2.4 – Діаграма послідовності авторизації користувача

У процесі авторизації клієнтський застосунок надсилає запит до серверної частини системи. Сервер звертається до бази даних для пошуку користувача та перевірки його облікових даних. Якщо дані є коректними, система визначає роль користувача та формує відповідь про успішний вхід. У разі помилки користувачу повертається повідомлення про некоректні облікові дані.

Запропонований механізм ролей та авторизації забезпечує базовий рівень захисту даних, обмежує доступ до функціональності системи та дозволяє організувати окремі сценарії роботи для спортсмена, тренера й адміністратора.

## 2.4 Проектування функціонального модуля ведення тренувань

Функціональний модуль ведення тренувань є одним із ключових компонентів програмної системи, оскільки забезпечує створення, зберігання, редагування та перегляд тренувальних занять біатлоністів. Модуль призначений для фіксації основних параметрів тренування, зокрема дати, часу початку, тривалості, дистанції, показників пульсу, самопочуття спортсмена, нотаток та результатів стрільби.

У межах проектування модуля передбачено, що спортсмен може самостійно додавати дані про виконане тренування, а тренер має змогу переглядати тренувальні записи закріплених спортсменів. Дані, введені користувачем, передаються з клієнтського застосунку до серверної частини, де проходять перевірку коректності та зберігаються у базі даних PostgreSQL.

Основними функціями модуля ведення тренувань є:

- створення нового тренування;
- вибір типу або назви тренування;
- введення часу початку та тривалості тренування;
- фіксація дистанції тренування;
- введення середнього, максимального та мінімального пульсу;
- додавання інформації про самопочуття спортсмена;
- додавання нотаток до тренування;
- внесення результатів стрільби за наявності відповідного типу тренування;
- перегляд детальної інформації про тренування;
- редагування та видалення тренувального запису.

Послідовність роботи модуля ведення тренувань наведено на рисунку 2.5.

Під час додавання тренування користувач заповнює форму у клієнтському застосунку. Після підтвердження введені дані надсилаються до серверної частини системи через API. Сервер перевіряє коректність обов'язкових полів, формує об'єкт тренування та зберігає його у таблиці TrainingSessions. Додаткові фізичні

показники зберігаються у таблиці SportMetrics, а результати стрільби, якщо вони були внесені, – у таблиці ShootingResults.

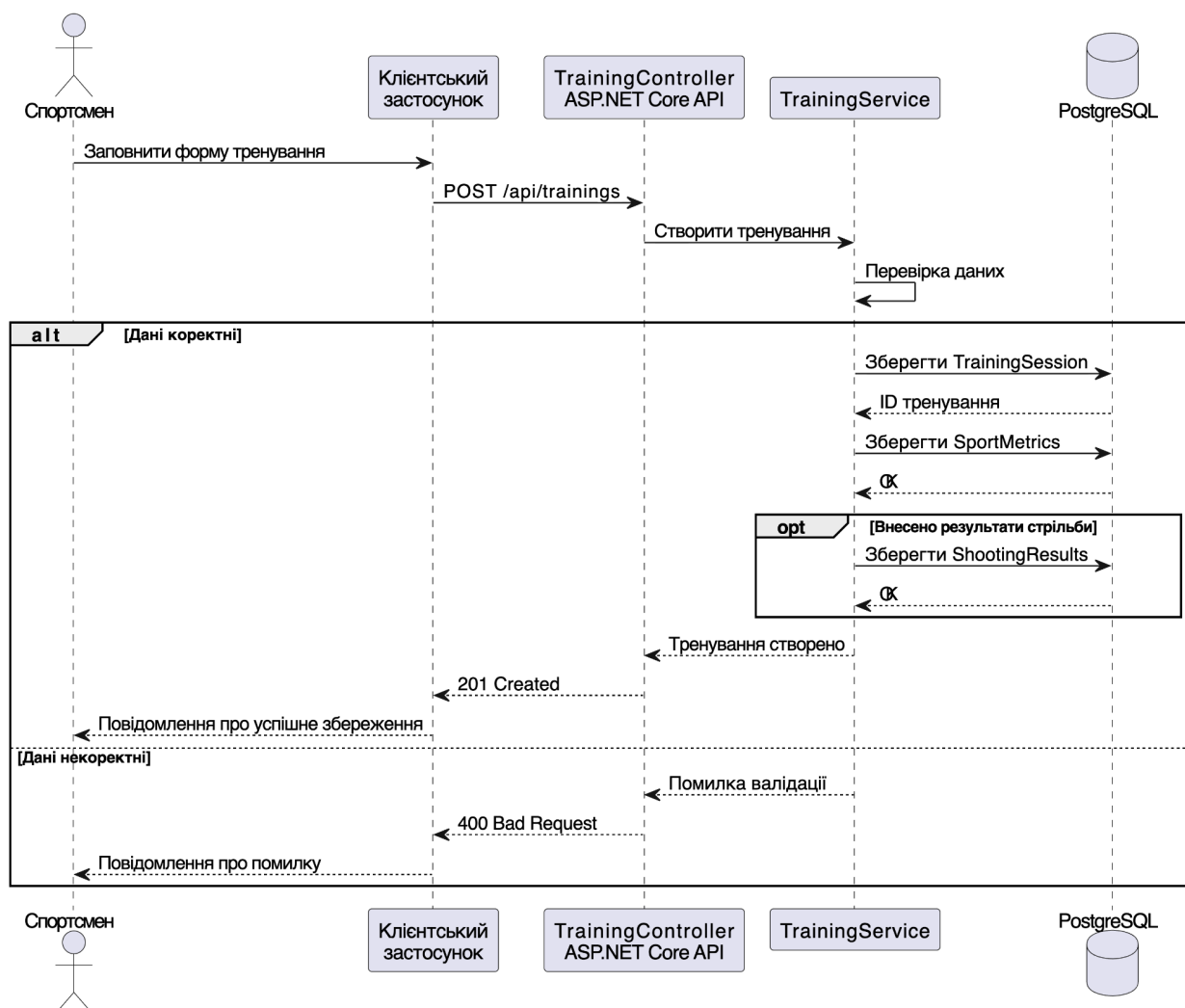


Рисунок 2.5 – Діаграма послідовності додавання тренування

Такий підхід дозволяє розділити основну інформацію про тренування, спортивні показники та результати стрільби на окремі логічні сутності бази даних. Це спрощує подальшу обробку даних, формування статистики та побудову звітів. Проектування модуля ведення тренувань забезпечує основу для реалізації основного функціоналу системи моніторингу спортивних показників біатлоністів.

## 2.5 Проектування модуля статистики стрільби

Модуль статистики стрільби призначений для збору, зберігання та аналізу результатів стрілецької підготовки біатлоністів. Його основним завданням є обробка даних про кількість пострілів, промахів, тип стрільби та результативність спортсмена під час тренувань. Наявність такого модуля є важливою для біатлону, оскільки точність стрільби безпосередньо впливає на загальний результат спортсмена.

У межах програмної системи результати стрільби зберігаються як частина тренувального запису. Якщо тренування містить стрілецький етап, користувач вносить дані про пристрілку, кількість використаних патронів, промахи під час стрільби лежачи та промахи під час стрільби стоячи. Після збереження дані можуть використовуватися для формування статистики за вибраний період.

Основними функціями модуля статистики стрільби є:

- збереження результатів стрільби в межах тренування;
- облік кількості використаних патронів;
- фіксація промахів під час стрільби лежачи;
- фіксація промахів під час стрільби стоячи;
- розрахунок загальної кількості промахів;
- розрахунок точності стрільби;
- фільтрація статистики за вибраним періодом;
- формування узагальнених показників для спортсмена та тренера;
- візуалізація результатів стрільби у вигляді діаграм.

Для модуля статистики стрільби доцільно використати діаграму послідовності, яка відображає процес отримання та обробки статистичних даних. Така діаграма дозволяє показати взаємодію між користувачем, клієнтським застосунком, серверною частиною, сервісом статистики та базою даних.

Послідовність формування статистики стрільби наведено на рисунку 2.6.

Під час формування статистики користувач обирає період, за який необхідно отримати дані. Клієнтський застосунок надсилає запит до серверної частини

системи. Сервер передає запит до сервісу статистики, який отримує з бази даних записи про тренування та відповідні результати стрільби. Після цього виконується обчислення кількості пострілів, промахів, точності стрільби та інших узагальнених показників.

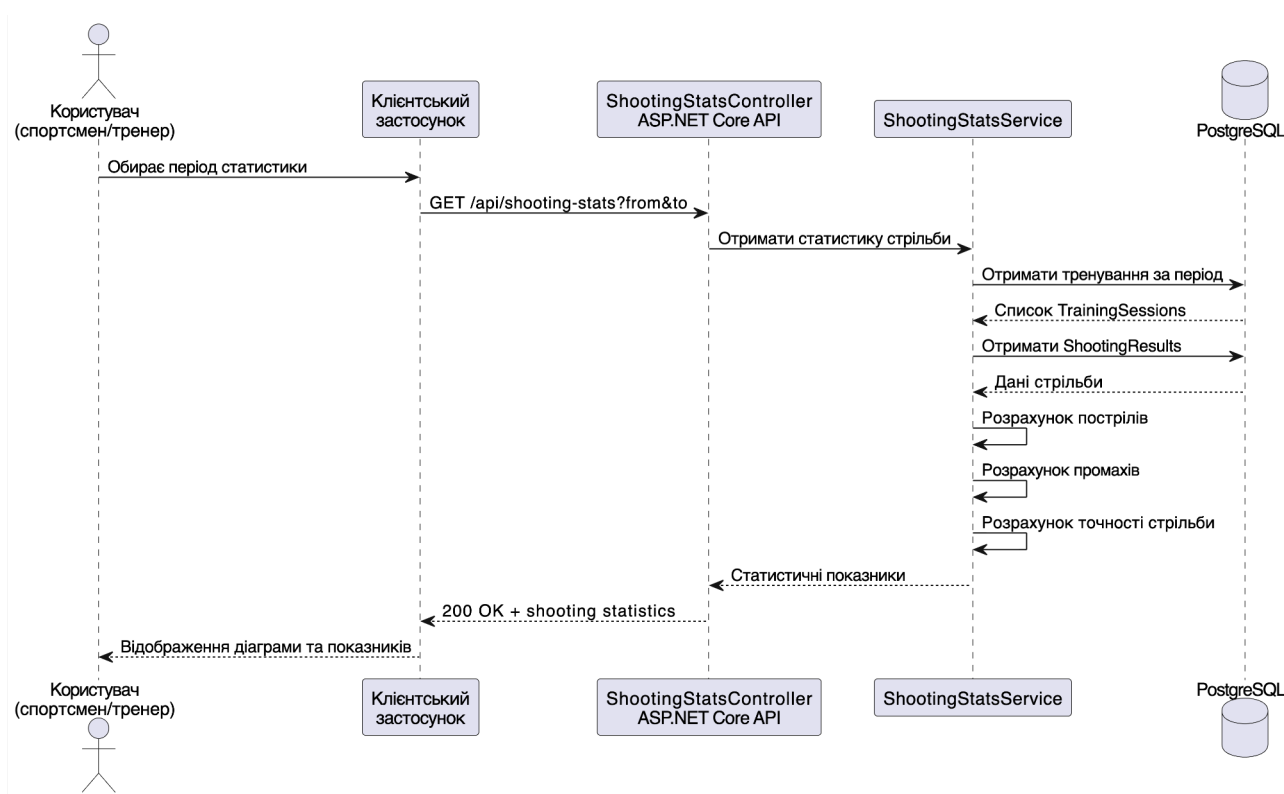


Рисунок 2.6 – Діаграма послідовності формування статистики стрільби

Результати обробки повертаються до клієнтського застосунку та відображаються користувачу у вигляді числових значень, таблиць або діаграм. Такий підхід дозволяє тренеру оцінювати ефективність стрілецької підготовки спортсмена, порівнювати результати за різні періоди та приймати рішення щодо коригування тренувального процесу.

## 2.6 Проєктування модуля формування звітів

Модуль формування звітів призначений для узагальнення тренувальних даних біатлоністів та подання їх у зручному для аналізу вигляді. Він забезпечує

отримання інформації про тренування за вибраний період, обчислення основних спортивних показників та формування підсумкової статистики для спортсмена або тренера.

У межах програмної системи звіт формується на основі даних про тренування, фізичні показники та результати стрільби. Користувач обирає часовий період і, за потреби, конкретні тренування. Після цього система отримує відповідні записи з бази даних, виконує агрегацію показників та відображає результат у вигляді звіту.

Основними функціями модуля формування звітів є:

- вибір періоду для формування звіту;
- вибір одного або декількох тренувань;
- підрахунок загальної тривалості тренувань;
- підрахунок загальної дистанції;
- обчислення середнього пульсу;
- підрахунок кількості використаних патронів;
- підрахунок загальної кількості промахів;
- формування підсумкової інформації для спортсмена або тренера;
- відображення результатів у вигляді таблиць, числових показників або

діаграм.

Структуру модуля формування звітів наведено на рисунку 2.7.

Як показано на рисунку 2.7, користувач взаємодіє з клієнтським інтерфейсом, де задає параметри звіту: період аналізу та перелік тренувань. Клієнтський застосунок надсилає запит до серверної частини системи. Контролер звітів приймає запит, передає його до сервісу формування звітів, а сервіс отримує необхідні дані з бази даних PostgreSQL.

Для формування звіту використовуються таблиці TrainingSessions, SportMetrics, ShootingResults, Athletes та Reports. Дані з цих таблиць дозволяють сформувати повну картину тренувальної діяльності спортсмена, включно з фізичними показниками, дистанцією, тривалістю тренувань та результатами стрільби.

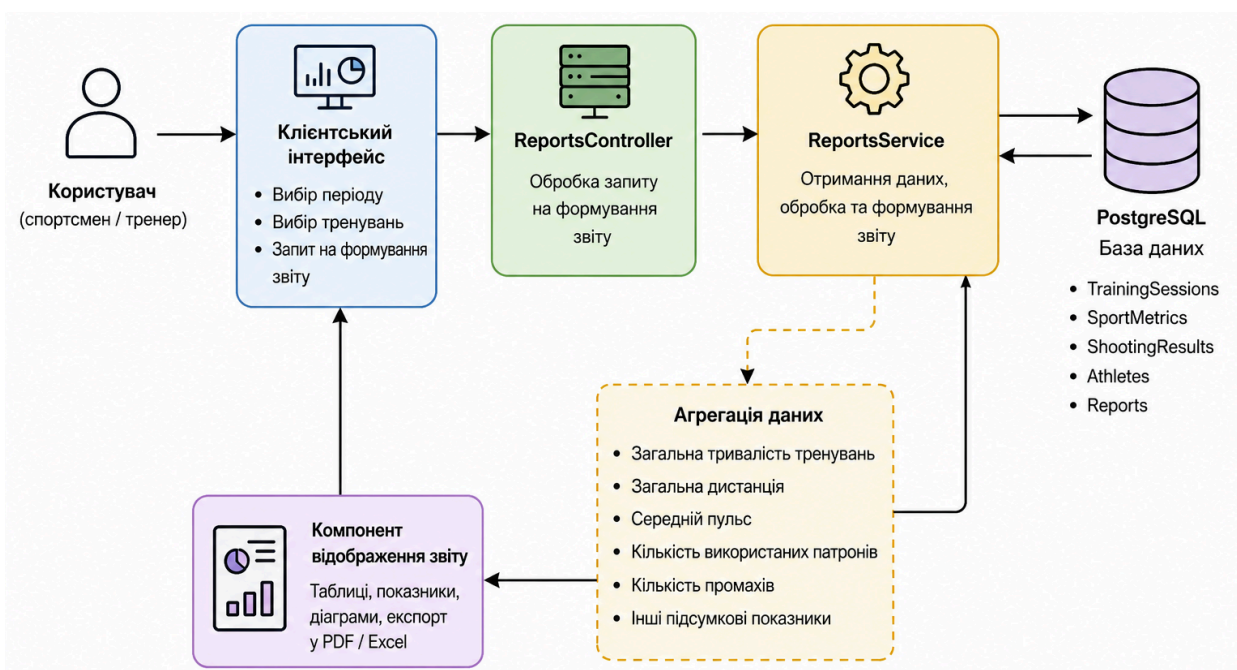


Рисунок 2.7 – Структура модуля формування звітів

Після отримання даних сервіс виконує обчислення підсумкових показників: загального часу тренувань, сумарної дистанції, середнього пульсу, кількості використаних патронів, кількості промахів та інших статистичних значень. Сформований результат може бути збережений у таблиці Reports або одразу переданий клієнтському застосунку для відображення користувачу.

Результати роботи модуля формування звітів використовуються для оцінювання ефективності тренувального процесу, аналізу динаміки спортивних показників та підтримки прийняття рішень тренером. Такий модуль забезпечує систематизацію накопичених даних і перетворює окремі тренувальні записи на узагальнену аналітичну інформацію.

## 2.7 Проектування користувацького інтерфейсу мобільного застосунку

Користувацький інтерфейс мобільного застосунку розроблено з урахуванням потреб спортсменів та тренерів у швидкому доступі до тренувальної інформації, спортивної статистики та аналітичних даних. Основними принципами

проектування інтерфейсу є простота використання, мінімальна кількість дій для введення інформації, інтуїтивна навігація та адаптація до мобільних пристроїв.

Для реалізації інтерфейсу передбачено використання сучасного дизайну з єдиною системою навігації, картковим представленням даних та адаптивним розташуванням елементів керування. На відміну від початкового прототипу, у фінальній версії застосунку планується використання сучасних UI-компонентів, покращеної кольорової схеми, графічних індикаторів показників та інтерактивних діаграм для відображення спортивної статистики (рис. 2.8 та 2.9).

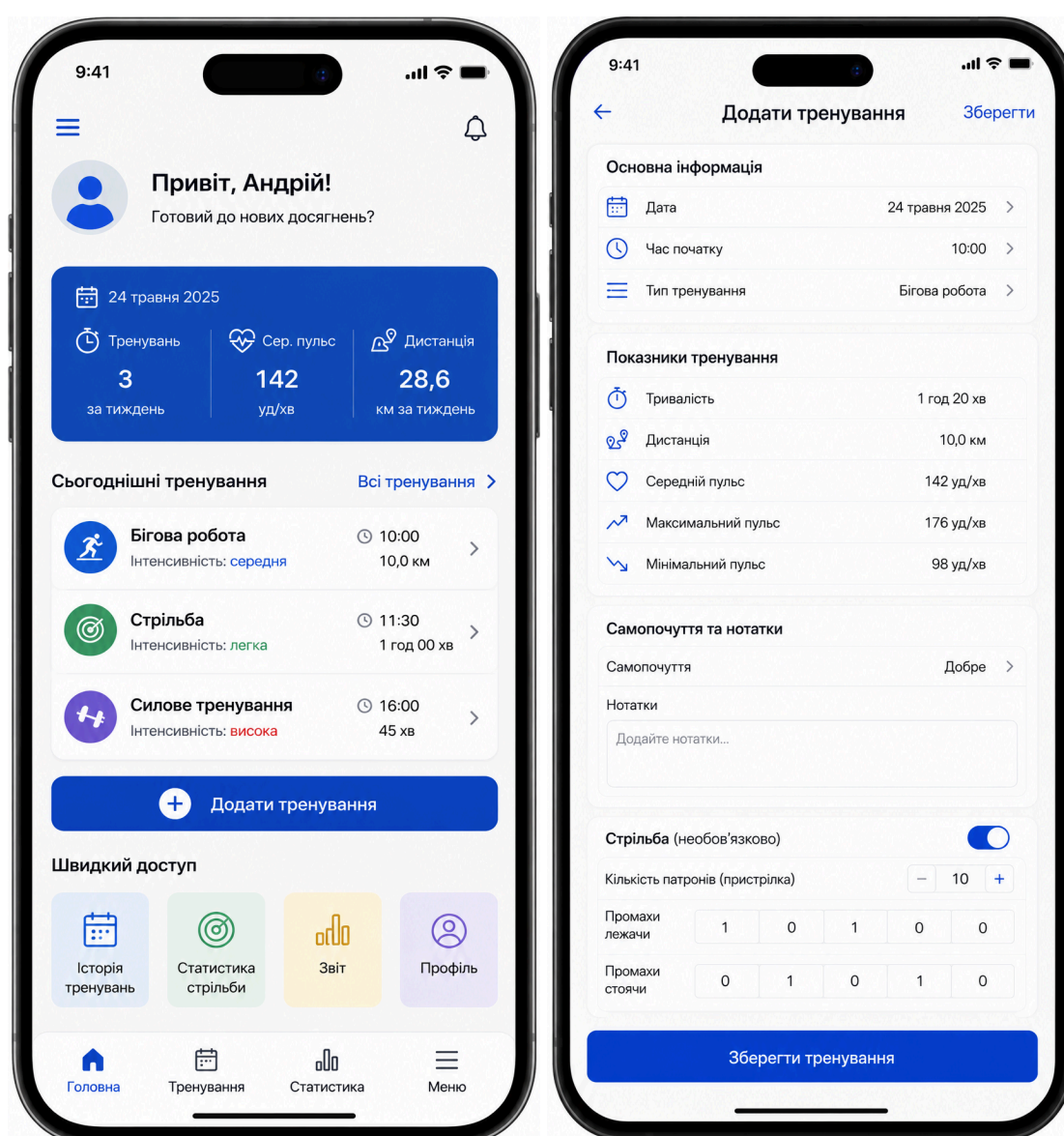


Рисунок 2.8 – Головний екран мобільного застосунку спортсмена та Інтерфейс додавання та редагування тренування

Після авторизації користувач отримує доступ до головного екрана, який відображає актуальні тренування, основні спортивні показники та швидкі переходи до функціональних модулів системи. Для спортсмена реалізовано можливість додавання тренувань, перегляду історії занять, ведення статистики стрільби та формування звітів. Для тренера передбачено додаткові функції керування групами спортсменів та аналізу результатів тренувального процесу.

У структурі інтерфейсу виділено такі основні екрани:

- головний екран користувача;
- форма додавання тренування;
- сторінка перегляду детальної інформації про тренування;
- сторінка статистики стрільби;
- сторінка формування звітів;
- сторінка профілю користувача;
- адміністративні екрани тренера.

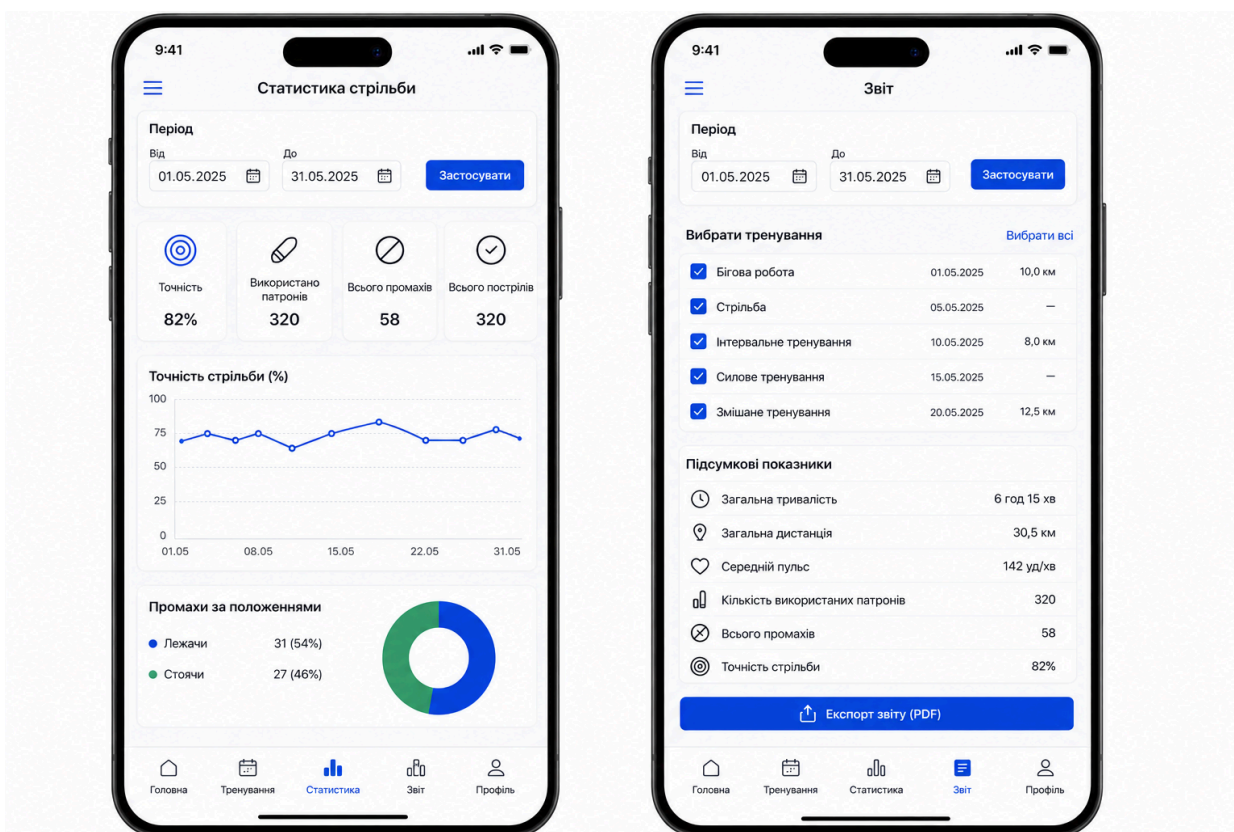


Рисунок 2.9 – Інтерфейс перегляду статистики та формування звітів

На головному екрані відображаються заплановані та виконані тренування, дата проведення занять, основна статистика та елементи швидкого доступу до функціональних модулів системи.

Екран додавання тренування дозволяє вводити параметри заняття, включаючи тривалість, дистанцію, пульсові показники, самопочуття спортсмена та результати стрільби. На сторінках статистики та звітності користувач може аналізувати результати стрільби, переглядати узагальнені показники тренувань за обраний період та формувати підсумкові звіти для подальшого аналізу тренувального процесу.

Запропонований користувацький інтерфейс забезпечує зручну взаємодію спортсменів і тренерів із системою, спрощує процес введення спортивних даних та створює єдиний інформаційний простір для моніторингу ефективності підготовки біатлоністів.

## **2.8 Висновки до другого розділу**

У другому розділі виконано проєктування програмної системи моніторингу спортивних показників біатлоністів. Розроблено загальну архітектуру програмного забезпечення на основі клієнт-серверного підходу із використанням платформи .NET, визначено структуру бази даних та основні взаємозв'язки між її сутностями. Спроектовано ролі користувачів і механізм авторизації, що забезпечує розмежування доступу до функціональних можливостей системи відповідно до прав спортсмена, тренера та адміністратора.

Також виконано проєктування основних функціональних модулів системи, зокрема модуля ведення тренувань, модуля статистики стрільби та модуля формування звітів. Розроблено структуру користувацького інтерфейсу мобільного застосунку та визначено принципи взаємодії користувачів із системою. Отримані результати створюють необхідну основу для подальшої реалізації програмного забезпечення та проведення його тестування на наступному етапі кваліфікаційної роботи.

### **3 ТЕСТУВАННЯ, ОЦІНЮВАННЯ ТА ВПРОВАДЖЕННЯ ПРОГРАМНОЇ СИСТЕМИ**

У цьому розділі розглянуто процес реалізації програмної системи моніторингу спортивних показників біатлоністів та результати її тестування. Наведено обґрунтування вибору технологій розробки, описано тестування серверної частини системи, бази даних та основних функціональних модулів, призначених для управління тренуваннями, аналізу результатів стрільби та формування звітів. Також проведено тестування програмного забезпечення з метою перевірки його працездатності, коректності функціонування та відповідності визначеним вимогам.

#### **3.1 Обґрунтування вибору технологій розробки**

Для реалізації програмної системи моніторингу спортивних показників біатлоністів було обрано сучасний стек технологій, який забезпечує високу продуктивність, надійність, масштабованість та зручність подальшого супроводу програмного забезпечення. Під час вибору технологій враховувалися вимоги до мобільності системи, ефективності роботи з базою даних, підтримки багаторівневої архітектури та можливості швидкої розробки програмних компонентів.

Основною платформою розробки обрано .NET, яка надає широкий набір інструментів для створення серверних застосунків, вебсервісів та мобільних програм. Використання платформи .NET дозволяє реалізувати єдину екосистему програмного забезпечення з використанням мови програмування C# та сучасних засобів розробки.

Для створення серверної частини використовується ASP.NET Core Web API, який забезпечує реалізацію REST-сервісів та взаємодію між мобільним застосунком і базою даних. Для роботи з даними використовується ORM-фреймворк Entity Framework Core, що дозволяє працювати з реляційною

базою даних через об'єктну модель без необхідності написання великої кількості SQL-запитів.

Зберігання інформації про спортсменів, тренування та результати стрільби реалізовано за допомогою системи керування базами даних PostgreSQL. Вона забезпечує надійне зберігання даних, підтримує складні зв'язки між сутностями та характеризується високою продуктивністю під час виконання запитів.



Рисунок 3.1 – Стек технологій програмної системи

Для реалізації мобільного застосунку використовується .NET MAUI, який дозволяє створювати кросплатформні мобільні застосунки для Android та iOS із

використанням єдиної кодової бази. Такий підхід суттєво скорочує час розробки та спрощує подальшу підтримку програмного забезпечення.

Основні технології, використані під час розробки програмної системи, наведено в таблиці 3.1.

Таблиця 3.1 – Технології, використані під час розробки програмної системи

Технологія	Призначення
C#	Основна мова програмування для реалізації серверної та клієнтської частин системи
.NET 8	Платформа для розробки програмного забезпечення
ASP.NET Core Web API	Реалізація серверної частини та REST API
.NET MAUI	Розробка кросплатформного мобільного застосунку
Entity Framework Core	Об'єктно-реляційне відображення та робота з базою даних
PostgreSQL	Зберігання спортивних даних та інформації про користувачів
LINQ	Формування та обробка запитів до даних
JWT Authentication	Реалізація механізму авторизації користувачів
Swagger/OpenAPI	Документування та тестування API
Git	Контроль версій програмного коду
GitHub	Зберігання репозиторію та командна робота над проєктом
Visual Studio 2022	Середовище розробки програмного забезпечення

Використання наведених технологій дозволяє побудувати сучасну програмну систему з чітким розподілом компонентів, централізованим зберіганням даних та підтримкою мобільних пристроїв. Обраний стек технологій добре інтегрується між собою, що спрощує процес розробки, тестування та подальшого супроводу програмного забезпечення.

### 3.2 Тестування серверної частини та бази даних системи

Серверна частина програмної системи реалізована із використанням платформи ASP.NET Core Web API та забезпечує виконання основної бізнес-логіки, обробку запитів користувачів, взаємодію з базою даних і формування відповідей для мобільного застосунку. Такий підхід дозволяє

централізувати обробку спортивних даних та забезпечити єдину точку доступу до функціональних можливостей системи.

Архітектура серверної частини побудована за багаторівневим принципом і включає рівень контролерів, сервісів, доступу до даних та бази даних. Контролери відповідають за приймання HTTP-запитів від клієнтського застосунку, сервіси реалізують бізнес-логіку роботи системи, а рівень доступу до даних забезпечує взаємодію із системою керування базами даних PostgreSQL через Entity Framework Core.

У процесі реалізації було створено окремі програмні модулі для роботи з користувачами, спортсменами, тренуваннями, результатами стрільби та звітами. Кожен модуль реалізований у вигляді набору REST API-методів, що дозволяють виконувати операції створення, перегляду, редагування та видалення даних.

Для зберігання інформації використовується реляційна база даних PostgreSQL. Структура бази даних реалізована відповідно до спроектованої ER-діаграми та включає таблиці Users, Athletes, Coaches, TrainingSessions, SportMetrics, ShootingResults та Reports. Зв'язки між таблицями реалізовані за допомогою зовнішніх ключів, що забезпечує цілісність і узгодженість даних.

Після завершення реалізації було проведено перевірку працездатності основних API-методів. Тестування виконувалося шляхом надсилання запитів до серверної частини та перевірки коректності отриманих відповідей. Особлива увага приділялася перевірці операцій авторизації користувачів, додавання тренувань, збереження результатів стрільби та формування звітів (табл. 3.2).

Таблиця 3.2 – Результати тестування серверної частини

№	Модуль системи	Кількість тестів	Успішно пройдено	Неуспішно
1	Авторизація користувачів	10	10	0
2	Управління спортсменами	12	12	0
3	Тренування	15	14	1
4	Результати стрільби	10	10	0
5	Формування звітів	8	8	0
	Разом	55	54	1

Результати тестування показали коректну роботу реалізованих програмних компонентів. Усі основні операції створення, отримання, оновлення та видалення даних виконувалися без критичних помилок. Виявлена незначна помилка у модулі тренувань була оперативно усунена. Також було підтверджено правильність взаємодії між серверною частиною та базою даних, а результати запитів відповідали інформації, що зберігалася у PostgreSQL.

Виконана реалізація серверної частини та бази даних забезпечує стабільну роботу програмної системи, підтримує необхідний функціонал моніторингу спортивних показників та створює основу для подальшої роботи мобільного застосунку і модулів аналітики.

### **3.3 Тестування функціональних модулів системи**

Після реалізації серверної частини та бази даних було виконано розробку основних функціональних модулів програмної системи. Реалізований функціонал забезпечує ведення інформації про спортсменів, облік тренувань, аналіз результатів стрільби та формування звітів. Усі модулі інтегровані із серверною частиною через REST API та використовують єдину базу даних для зберігання інформації.

Одним із ключових модулів системи є модуль управління спортсменами. Його функціональність забезпечує створення, редагування та перегляд профілів спортсменів. Для кожного спортсмена зберігаються персональні дані, історія тренувань та статистичні показники. Реалізація даного модуля дозволяє тренеру централізовано керувати інформацією про підготовку спортсменів.

Наступним важливим компонентом є модуль ведення тренувань. Він забезпечує реєстрацію тренувальних занять, збереження параметрів тренування, фізичних показників та додаткової інформації щодо виконаної роботи. Реалізований функціонал дозволяє накопичувати історію тренувального процесу та використовувати її для подальшого аналізу.

Модуль статистики стрільби реалізує збір та обробку даних про результати стрілецької підготовки спортсменів. На основі збережених результатів система автоматично виконує розрахунок кількості пострілів, промахів, влучань та відсотка точності стрільби. Це дозволяє отримувати узагальнену інформацію про ефективність стрілецької підготовки спортсмена.

Для підтримки процесу аналізу результатів тренувальної діяльності реалізовано модуль формування звітів. Модуль автоматично агрегує інформацію про тренування, фізичні показники та результати стрільби за вибраний період і формує підсумкові статистичні дані для спортсмена або тренера.

Після завершення реалізації функціональних модулів було проведено їх тестування. Перевірка виконувалася шляхом моделювання типових сценаріїв використання системи та контролю правильності отриманих результатів. Основною метою тестування було підтвердження коректності роботи реалізованого функціоналу та його відповідності вимогам, сформованим на етапі проектування.

Результати тестування функціональних модулів наведено у таблиці 3.3.

Таблиця 3.3 – Результати тестування функціональних модулів системи

№	Функціональний модуль	Кількість тестів	Успішно пройдено	Неуспішно
1	Управління спортсменами	10	10	0
2	Ведення тренувань	15	14	1
3	Статистика стрільби	12	12	0
4	Формування звітів	8	8	0
	Разом	45	44	1

Результати тестування підтвердили коректну роботу більшості реалізованих функцій. У процесі перевірки було виявлено одну незначну помилку під час редагування тренувальних записів, яка була усунена на етапі налагодження програмного забезпечення. Після виправлення помилки повторне тестування показало стабільну роботу всіх функціональних модулів системи.

Для демонстрації реалізації основних функціональних можливостей системи у роботі доцільно навести приклади програмного коду та інтерфейсів відповідних модулів.

### **3.4 Тестування користувацького інтерфейсу мобільного застосунку**

Користувацький інтерфейс мобільного застосунку реалізовано із використанням технології .NET MAUI. Основною метою розробки інтерфейсу було забезпечення швидкого доступу до функціональних можливостей системи, зручного введення спортивних даних та наочного відображення результатів тренувальної діяльності.

Під час реалізації особлива увага приділялася принципам зручності використання, логічній структурі екранів та мінімізації кількості дій, необхідних для виконання основних операцій. Інтерфейс побудований за модульним принципом і включає екран авторизації, головний екран користувача, модуль ведення тренувань, модуль статистики стрільби та модуль формування звітів.

Для оцінювання якості реалізованого інтерфейсу було проведено UI-тестування. Основною метою тестування було підтвердження коректності відображення елементів інтерфейсу, перевірка навігації між екранами та оцінка зручності взаємодії користувача із системою.

У процесі тестування перевірялися такі аспекти:

- коректність відображення елементів інтерфейсу;
- робота навігаційного меню;
- відображення інформації про спортсменів;
- робота форм додавання тренувань;
- відображення статистики стрільби;
- формування та перегляд звітів;
- адаптація інтерфейсу до різних розмірів екранів;
- обробка помилкових дій користувача.

Для перевірки працездатності інтерфейсу було виконано серію тестових сценаріїв. Результати тестування наведено в таблиці 3.4.

Таблиця 3.4 – Результати UI-тестування мобільного застосунку

№	Тестовий сценарій	Результат
1	Авторизація користувача	Успішно
2	Перехід між екранами застосунку	Успішно
3	Відображення списку спортсменів	Успішно
4	Створення нового тренування	Успішно
5	Редагування тренування	Успішно
6	Перегляд статистики стрільби	Успішно
7	Формування звіту	Успішно
8	Відображення повідомлень про помилки	Успішно
9	Адаптація до різних розмірів екранів	Успішно
10	Завантаження даних із сервера	Успішно

У ході тестування критичних помилок виявлено не було. Усі екрани застосунку коректно відображалися на мобільних пристроях, навігація працювала відповідно до визначених сценаріїв використання, а взаємодія користувача з основними функціональними модулями не викликала труднощів.

Для демонстрації результатів реалізації інтерфейсу в роботі наведено основні екрани мобільного застосунку в другому розділі.

### 3.5 Оцінка працездатності програмної системи

ДЗавершальним етапом розробки програмної системи стала комплексна оцінка її працездатності. Основною метою даного етапу було підтвердження можливості стабільного функціонування всіх компонентів системи в умовах, наближених до реальної експлуатації. Оцінювання проводилося на основі результатів тестування серверної частини, функціональних модулів та користувацького інтерфейсу мобільного застосунку.

Під час перевірки працездатності було виконано повний цикл роботи користувача із системою: авторизація, перегляд інформації про спортсменів, створення тренувань, внесення результатів стрільби, формування статистики та генерація звітів. Для кожного сценарію оцінювалася коректність роботи програмних модулів, швидкість виконання операцій та відсутність критичних помилок.

Оцінювання проводилося за такими критеріями:

- коректність виконання функціональних операцій;
- стабільність роботи системи;
- швидкість обробки запитів;
- коректність збереження даних;
- правильність формування статистики;
- правильність формування звітів;
- зручність взаємодії користувача із системою.

Результати оцінювання працездатності наведено у таблиці 3.5.

Таблиця 3.5 – Оцінка працездатності програмної системи

Критерій оцінювання	Результат
Авторизація користувачів	Середній час входу – 0,8 с, помилки відсутні
Робота з тренуваннями	Створення/редагування – до 1,2 с, 100% успішних операцій
Облік спортивних показників	Запис даних – до 0,9 с, втрати даних не зафіксовано
Обробка результатів стрільби	Обробка 1 сесії – 1,5 с, точність обчислень – 100%
Формування статистики	Генерація – до 2,3 с, відхилення від очікуваних значень – 0%
Формування звітів	Генерація PDF – до 3,1 с, помилки відсутні
Робота мобільного інтерфейсу	Час відгуку інтерфейсу – 0,5–1,0 с, збої не зафіксовано
Взаємодія з базою даних	Середній час запиту – 0,7 с, втрати з'єднання – 0%
Загальна стабільність системи	Безперервна робота протягом 72 годин без збоїв

За результатами проведеної перевірки встановлено, що всі основні функціональні можливості програмної системи працюють відповідно до визначених вимог. Серверна частина успішно обробляє запити мобільного

застосунку, база даних забезпечує надійне зберігання інформації, а реалізовані функціональні модулі коректно виконують обробку спортивних даних.

Під час тестування не було виявлено критичних помилок, які б унеможливили використання програмної системи. Виявлені незначні недоліки були усунені на етапі налагодження програмного забезпечення та не впливають на роботу основного функціоналу.

Отримані результати підтверджують працездатність розробленої програмної системи моніторингу спортивних показників біатлоністів та можливість її практичного використання для підтримки тренувального процесу, збору спортивної статистики та аналізу результатів підготовки спортсменів.

### **3.6 Розгортання системи та перспективи розвитку**

Розроблена програмна система спроектована з урахуванням можливості її подальшого розгортання як у локальному середовищі, так і на віддалених серверних платформах. Серверна частина системи може бути розгорнута на сервері під керуванням операційної системи Windows або Linux із встановленою платформою .NET та системою керування базами даних PostgreSQL. Мобільний застосунок може бути встановлений на пристроях під керуванням операційної системи Android після компіляції та формування інсталяційного пакета.

Для забезпечення коректного розгортання системи визначено мінімальні версії програмного забезпечення, наведені в таблиці 3.6.

Для коректної роботи системи необхідно забезпечити доступ до серверної частини через мережу Інтернет або локальну мережу, а також налаштувати підключення до бази даних. Завдяки використанню клієнт-серверної архітектури система підтримує одночасну роботу декількох користувачів та може масштабуватися шляхом збільшення обчислювальних ресурсів сервера.

Таблиця 3.6 – Мінімальні версії програмного забезпечення для розгортання системи

Компонент	Мінімальна версія
Операційна система сервера	Windows 10 / Ubuntu 20.04
Платформа .NET	.NET 6.0
СУБД PostgreSQL	PostgreSQL 13
Android SDK	API Level 26 (Android 8)
Середовище розробки	Visual Studio 2022

Перспективами подальшого розвитку програмної системи є:

- інтеграція з пристроями моніторингу фізичної активності та спортивними трекерами;
- автоматичний імпорт даних із GPS-пристроїв та смарт-годинників;
- розширення аналітичного модуля для прогнозування спортивних результатів;
- впровадження засобів машинного навчання для аналізу ефективності тренувань;
- реалізація вебверсії системи для тренерів та спортивних організацій;
- підтримка командної роботи з групами спортсменів;
- розширення набору статистичних показників для оцінювання тренувального процесу.

Таким чином, розроблена програмна система має достатній рівень гнучкості та масштабованості для подальшого вдосконалення й практичного використання у сфері спортивної підготовки біатлоністів.

### 3.7 Висновки до розділу 3

У третьому розділі виконано тестування програмної системи моніторингу спортивних показників біатлоністів із використанням платформи .NET, технології ASP.NET Core Web API, системи керування базами даних PostgreSQL та мобільного застосунку на основі .NET MAUI. Реалізовано серверну частину

системи, структуру бази даних та основні функціональні модулі, призначені для управління спортсменами, ведення тренувань, аналізу результатів стрільби та формування звітів. Також розроблено користувацький інтерфейс мобільного застосунку, який забезпечує зручну взаємодію користувачів із функціональними можливостями системи.

У ході тестування було перевірено працездатність серверної частини, функціональних модулів та користувацького інтерфейсу. Результати тестування підтвердили коректність роботи реалізованого програмного забезпечення, правильність обробки спортивних даних та стабільність взаємодії між компонентами системи. Проведена оцінка працездатності показала, що розроблена програмна система відповідає визначеним вимогам і може бути використана для автоматизації процесів моніторингу, аналізу та зберігання спортивних показників біатлоністів.

## **4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ**

У даному розділі розглядаються основні вимоги охорони праці та безпеки життєдіяльності, яких необхідно дотримуватися під час виконання робіт з використанням комп'ютерної техніки. Особливу увагу приділено організації робочого місця та забезпеченню безпечних умов праці відповідно до чинних нормативних документів.

### **4.1 Значення адаптації в трудовому процесі**

Адаптація працівника до умов праці є одним із важливих чинників забезпечення безпеки життєдіяльності, ефективності трудової діяльності та збереження здоров'я працівників. Під адаптацією розуміють процес пристосування людини до умов виробничого середовища, особливостей трудового процесу, технічних засобів праці та соціально-психологічних умов колективу [27]. Успішна адаптація сприяє формуванню професійних навичок, підвищенню продуктивності праці та зменшенню ймовірності виникнення виробничого травматизму.

Згідно з положеннями охорони праці, одним із головних завдань організації трудового процесу є створення таких умов, за яких працівник може швидко та безпечно пристосуватися до виконання своїх функціональних обов'язків [28]. Особливого значення це набуває під час роботи з комп'ютерною технікою та інформаційними системами, де основне навантаження припадає на органи зору, нервову систему та психоемоційний стан працівника.

Відповідно до вимог ДСТУ ISO 6385:2005 «Ергономічні принципи проектування робочих систем», під час створення робочих місць необхідно враховувати фізіологічні та психологічні можливості людини, особливості сприйняття інформації та взаємодії з технічними системами [29]. Дотримання

ергономічних принципів дозволяє скоротити період адаптації працівника до нових умов праці та зменшити рівень втоми під час виконання професійних завдань.

Важливим аспектом адаптації є психофізіологічна адаптація, яка передбачає пристосування організму людини до інтенсивності трудового процесу, рівня інформаційного навантаження та режиму праці й відпочинку [30]. Недостатній рівень адаптації може призводити до розвитку професійної втоми, зниження концентрації уваги, погіршення якості виконання роботи та збільшення кількості помилок під час прийняття рішень.

Для працівників, діяльність яких пов'язана з використанням персональних комп'ютерів, важливими є також вимоги щодо організації робочого місця. Відповідно до ДСанПіН 3.3.2.007-98, робочі місця користувачів комп'ютерної техніки повинні забезпечувати безпечні умови праці, належний рівень освітлення, оптимальні параметри мікроклімату та ергономічне розміщення обладнання [31]. Недотримання цих вимог може ускладнювати процес адаптації та негативно впливати на працездатність працівників.

У межах розробленої системи централізованого управління комунікаціями стоматологічної клініки питання адаптації має важливе практичне значення. Адміністратор клініки працює з веб-інтерфейсом системи, здійснює обробку звернень пацієнтів, контролює записи на прийом та аналізує відповіді, сформовані мовною моделлю. Для забезпечення швидкої адаптації користувача під час проектування системи було враховано принципи ергономіки інтерфейсів, логічне структурування інформації та мінімізацію кількості дій, необхідних для виконання типових операцій. Важливим результатом успішної адаптації є зниження ризику виникнення професійних помилок під час роботи з комп'ютерними системами. На початкових етапах використання нового програмного забезпечення працівники витрачають більше часу на пошук необхідних функцій та виконання стандартних операцій, що може призводити до підвищеного психоемоційного навантаження. Після завершення адаптаційного періоду кількість помилок зменшується, підвищується швидкість обробки інформації та покращується якість виконання службових обов'язків [29]. Саме тому під час розроблення інформаційних систем

важливо враховувати вимоги ергономіки та принципи людино-орієнтованого проєктування.

Крім того, використання автоматизованої системи дозволяє зменшити інформаційне навантаження на адміністратора шляхом автоматичної обробки типових запитів пацієнтів. Це сприяє зниженню психоемоційної напруги та створює більш комфортні умови праці. Відповідно до рекомендацій Міжнародної організації праці, автоматизація рутинних процесів є одним із ефективних способів покращення умов праці та підвищення продуктивності персоналу [32]. Особливою складовою трудової адаптації є соціально-психологічна адаптація працівника, яка полягає у пристосуванні до організаційної структури підприємства, особливостей взаємодії з колегами та прийнятих правил виконання роботи [30]. В умовах цифровізації значна частина професійних обов'язків виконується із застосуванням інформаційних систем, тому ефективність роботи персоналу значною мірою залежить від рівня прийняття нових технологій та готовності до їх використання [30]. Наявність зрозумілого інтерфейсу, автоматизованих механізмів підтримки прийняття рішень та можливості швидкого доступу до необхідної інформації сприяє формуванню позитивного ставлення працівників до нових програмних засобів і скорочує період їх адаптації. Як зазначається у роботах з охорони праці, скорочення часу адаптації безпосередньо впливає на підвищення продуктивності праці, зменшення кількості помилок та покращення загального психологічного клімату в колективі [30].

Таким чином, адаптація працівника до умов праці є важливою складовою безпеки життєдіяльності та ефективної організації трудового процесу. Використання сучасних інформаційних систем, розроблених із дотриманням вимог ергономіки, охорони праці та психофізіологічних особливостей людини, дозволяє скоротити період адаптації, знизити рівень професійного навантаження та підвищити ефективність роботи персоналу.

## 4.2 Інженерно-технічні рішення з охорони праці

Одним із важливих завдань охорони праці є забезпечення безпечних та комфортних умов роботи працівників. Для фахівців, діяльність яких пов'язана з використанням комп'ютерної техніки та інформаційних систем, продуктивність праці значною мірою залежить від умов робочого середовища, рівня інформаційного навантаження та ергономічності робочого місця. Недотримання вимог охорони праці може призводити до зниження працездатності, виникнення професійної втоми, підвищення психоемоційного навантаження та погіршення стану здоров'я працівників [28].

У межах розробленої системи централізованого управління комунікаціями стоматологічної клініки основним користувачем є адміністратор, який здійснює обробку звернень пацієнтів, аналізує відповіді, сформовані мовною моделлю, та керує записами на прийом. Тому особливого значення набуває оцінка умов праці працівника, який виконує роботу із застосуванням персонального комп'ютера.

Відповідно до Постанови Кабінету Міністрів України №442 від 01.08.1992 р. атестація робочих місць за умовами праці є одним із основних інструментів оцінювання впливу виробничих факторів на працівника [34]. Результати такої оцінки використовуються для визначення рівня безпечності умов праці та розроблення заходів щодо їх покращення.

Для аналізу умов праці використовується «Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу», затверджена наказом Міністерства охорони здоров'я України №248 від 08.04.2014 р. [35]. Згідно з даною класифікацією умови праці поділяються на чотири класи: оптимальні, допустимі, шкідливі та небезпечні.

Під час експлуатації розробленої інформаційної системи основними факторами, що можуть впливати на працівника, є тривала робота за комп'ютером, статичне навантаження під час сидіння, навантаження на органи зору, підвищена концентрація уваги та необхідність обробки значних обсягів інформації. На

відміну від виробничих підприємств, де переважають фізичні небезпеки, для операторів інформаційних систем найбільш характерними є психофізіологічні та ергономічні фактори ризику [27].

Важкість трудового процесу визначається рівнем фізичного навантаження на працівника. Для адміністратора стоматологічної клініки фізичне навантаження є незначним і пов'язане переважно зі статичним положенням тіла під час роботи за комп'ютером. Водночас напруженість праці є достатньо високою через необхідність постійної роботи з інформацією, контролю звернень пацієнтів, прийняття рішень щодо запису на прийом та аналізу відповідей, сформованих системою [29].

Відповідно до вимог ДСанПіН 3.3.2.007-98 робоче місце користувача персонального комп'ютера повинно забезпечувати належний рівень освітлення, оптимальні параметри мікроклімату, ергономічне розташування обладнання та можливість зміни робочої пози [31]. Дотримання цих вимог сприяє зниженню втомлюваності працівників та підвищує ефективність їх діяльності.

Проведений аналіз дозволяє зробити висновок, що умови праці адміністратора стоматологічної клініки, який використовує розроблену систему, можуть бути віднесені до допустимого класу умов праці за умови дотримання встановлених санітарно-гігієнічних вимог та правил організації робочого місця. Для подальшого покращення умов праці доцільно використовувати ергономічні меблі, забезпечувати регламентовані перерви під час роботи з комп'ютером та підтримувати оптимальні параметри виробничого середовища.

Шкідливі умови праці за ступенем перевищення гігієнічних нормативів та вираженості можливих змін в організмі працюючих поділяються на чотири ступені [34]. Проте під час експлуатації розробленої системи централізованого управління комунікаціями стоматологічної клініки більшість небезпечних виробничих факторів, характерних для промислових підприємств, відсутні. Основними чинниками, що можуть впливати на працівника, є тривала робота за персональним комп'ютером, статичне навантаження, напруження органів зору та підвищене інформаційне навантаження.

Ступінь шкідливості умов праці для адміністратора стоматологічної клініки визначається насамперед параметрами мікроклімату приміщення, рівнем освітленості робочого місця, тривалістю роботи з візуальними дисплейними терміналами та психоемоційним навантаженням під час обробки звернень пацієнтів. Особливу увагу необхідно приділяти організації робочого місця відповідно до вимог ДСанПіН 3.3.2.007-98 та ДСТУ ISO 6385:2005, які регламентують ергономічні вимоги до робочих систем та умов праці користувачів комп'ютерної техніки [29, 31].

Оцінка важкості трудового процесу для даного робочого місця здійснюється за показниками статичного навантаження, тривалості перебування працівника в сидячому положенні та характеру виконуваних операцій. Оскільки діяльність адміністратора не пов'язана зі значними фізичними навантаженнями, показники важкості праці можуть бути віднесені до допустимого рівня. Водночас напруженість трудового процесу є більш суттєвим фактором і визначається необхідністю постійного контролю інформаційних потоків, прийняття рішень щодо запису пацієнтів на прийом, перевірки коректності відповідей, сформованих мовною моделлю, та взаємодії з різними каналами комунікації.

Для наочного представлення результатів аналізу умов праці проведено оцінку основних виробничих факторів, що впливають на користувача системи. Результати оцінювання наведено в таблиці 4.1.

Як видно з таблиці 4.1, більшість факторів виробничого середовища належать до допустимого класу умов праці. Найбільший вплив на працівника мають фактори напруженості трудового процесу, пов'язані з інтелектуальним та емоційним навантаженням. Водночас застосування розробленої системи дозволяє частково компенсувати вплив зазначених факторів завдяки автоматизації обробки типових запитів пацієнтів та централізованому управлінню інформацією.

Використання великої мовної моделі дозволяє автоматично формувати відповіді на поширені запити, що зменшує навантаження на адміністратора та скорочує час обробки звернень. Крім того, централізоване зберігання інформації

про пацієнтів, повідомлення та записи на прийом забезпечує швидкий доступ до необхідних даних і підвищує ефективність виконання службових обов'язків.

Таблиця 4.1 – Оцінка умов праці адміністратора стоматологічної клініки

<b>Фактор</b>	<b>Характеристика</b>	<b>Клас умов праці</b>
Фізичне навантаження	Незначне, відсутність переміщення вантажів	2
Статичне навантаження	Тривала робота у сидячому положенні	2
Навантаження на органи зору	Робота з монітором протягом робочого дня	2
Інтелектуальне навантаження	Аналіз звернень пацієнтів та контроль роботи системи	2
Емоційне навантаження	Взаємодія з пацієнтами та контроль записів	2
Мікроклімат та освітлення	Відповідають нормативним вимогам	2
Загальна оцінка умов праці	Допустимі умови праці	2

На підставі проведеного аналізу можна зробити висновок, що умови праці користувача розробленої системи належать до допустимого класу умов праці (2 клас) за показниками шкідливості та небезпечності виробничих факторів. За умови дотримання вимог чинних нормативних документів щодо організації робочого місця, режимів праці та відпочинку, освітлення та параметрів мікроклімату робота із системою не створює значного ризику для здоров'я працівника та забезпечує належний рівень працездатності протягом робочої зміни.

## ВИСНОВКИ

У кваліфікаційній роботі вирішено актуальне завдання автоматизації процесу моніторингу спортивних показників біатлоністів шляхом розробки спеціалізованої програмної системи. Проведено аналіз особливостей тренувального процесу в біатлоні, сучасних програмних рішень для спортивного моніторингу, а також методів збору, зберігання та обробки спортивних даних. На основі проведеного аналізу визначено функціональні та нефункціональні вимоги до програмної системи, сформовано перелік користувачів та основні сценарії її використання.

У процесі виконання роботи було спроектовано архітектуру програмної системи, структуру бази даних, механізми авторизації користувачів та взаємодії між основними компонентами програмного забезпечення. Розроблено модулі ведення тренувань, аналізу результатів стрільби, формування статистичних звітів та аналітичної інформації. Для реалізації програмної системи використано платформу .NET, технологію ASP.NET Core Web API, систему керування базами даних PostgreSQL та засоби розробки мобільних застосунків .NET MAUI.

Практичним результатом роботи є створена програмна система моніторингу спортивних показників біатлоністів, яка забезпечує централізоване зберігання тренувальних даних, облік спортивних показників, аналіз результатів стрілецької підготовки та формування звітів для спортсменів і тренерів. Реалізована система дозволяє підвищити ефективність ведення тренувальної документації, спростити процес аналізу результатів спортивної діяльності та забезпечити підтримку прийняття рішень під час планування тренувального процесу.

У ході розробки проведено тестування серверної частини, функціональних модулів, бази даних та користувацького інтерфейсу мобільного застосунку. Результати тестування підтвердили коректність роботи програмного забезпечення, відповідність реалізованого функціоналу встановленим вимогам та стабільність роботи системи в умовах, наближених до реальної експлуатації. Оцінка

працездатності показала можливість використання розробленої системи для автоматизації процесів збору, обробки та аналізу спортивних даних у біатлоні.

Перспективами подальшого розвитку програмної системи є інтеграція зі спортивними трекерами та смарт-пристроями, автоматичний імпорт даних із GPS-сенсорів, розширення аналітичних можливостей за рахунок використання методів машинного навчання, а також розробка вебверсії системи для організації спільної роботи тренерів та спортсменів. Отримані результати підтверджують доцільність використання сучасних програмних технологій для підтримки та вдосконалення процесів спортивної підготовки біатлоністів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Михалик Д. М., Цуприк Г. Б., Бревус В. М. Методичні вказівки до виконання кваліфікаційної роботи бакалавра для здобувачів першого (бакалаврського) рівня вищої освіти за освітньо-професійною програмою «Інженерія програмного забезпечення» спеціальності 121 – «Інженерія програмного забезпечення» всіх форм навчання. Тернопіль : ТНТУ ім. І. Пулюя, 2024. 45 с.
2. Лавріщева К. М. Програмна інженерія : підручник. Київ : Академперіодика, 2008. 319 с.
3. Бородкіна І. Л., Бородкін Г. О. Інженерія програмного забезпечення : навчальний посібник. Київ : Центр учбової літератури, 2018. 204 с.
4. Павловський В. І., Базис даних та засоби управління : практикум. Київ : КПІ ім. Ігоря Сікорського, 2021. 154 с.
5. Проєктування мобільних застосунків : навчальний посібник. Запоріжжя : Запорізька політехніка, 2021. 220 с.
6. Основи проєктування інформаційних систем : навчальний посібник / за ред. В. В. Пасічника. Львів : Львівська політехніка, 2020. 312 с.
7. Архітектура програмного забезпечення : навчальний посібник / за ред. М. З. Згуровського. Київ : КПІ ім. Ігоря Сікорського, 2021. 286 с.
8. Організація баз даних та знань : навчальний посібник. Київ : КНЕУ, 2020. 280 с.
9. Технології розроблення програмного забезпечення : навчальний посібник. Харків : ХНУРЕ, 2022. 296 с.
10. Інформаційні системи та технології : навчальний посібник. Одеса : ОНПУ, 2021. 248 с.
11. Основи програмування мобільних застосунків на платформі Android : навчальний посібник. Львів : Львівська політехніка, 2022. 240 с.
12. UML-моделювання інформаційних систем : навчальний посібник. Київ : НУБіП України, 2020. 180 с.

13. Проєктування та розробка клієнт-серверних систем : навчальний посібник. Тернопіль : ТНТУ ім. І. Пулюя, 2023. 210 с.
14. Sommerville I. Software Engineering. 10th Edition. Boston : Pearson Education, 2015. 816 p.
15. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner's Approach. 9th Edition. New York : McGraw-Hill, 2019. 880 p.
16. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3rd Edition. Boston : Addison-Wesley, 2004. 208 p.
17. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Boston : Addison-Wesley, 1994. 395 p.
18. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Boston : Prentice Hall, 2017. 432 p.
19. Martin R. C. Clean Code: A Handbook of Agile Software Craftsmanship. Boston : Prentice Hall, 2008. 464 p.
20. Hunt A., Thomas D. The Pragmatic Programmer. 20th Anniversary Edition. Boston : Addison-Wesley, 2019. 352 p.
21. Richards M., Ford N. Fundamentals of Software Architecture. Sebastopol : O'Reilly Media, 2020. 432 p.
22. Kleppmann M. Designing Data-Intensive Applications. Sebastopol : O'Reilly Media, 2017. 616 p.
23. Troelsen A., Japikse P. Pro C# 10 with .NET 6. New York : Apress, 2022. 1200 p.
24. Price M. C# 12 and .NET 8 – Modern Cross-Platform Development Fundamentals. Birmingham : Packt Publishing, 2024. 820 p.
25. Freeman A. Pro ASP.NET Core 8. New York : Apress, 2024. 1100 p.
26. McClure S., Rozanski N. Software Systems Architecture. 3rd Edition. London : Addison-Wesley, 2023. 720 p.
27. Желібо Є. П., Заверуха Н. М. Безпека життєдіяльності : навч. посіб. Київ : Каравела, 2020. 344 с.

28. Гандзюк М. П., Желібо Є. П., Халімовський М. О. Основи охорони праці : підручник. Київ : Каравела, 2021. 512 с.
29. ДСТУ ISO 6385:2005. Ергономічні принципи проєктування робочих систем. Київ : Держспоживстандарт України, 2007. 18 с.
30. Крушельницька О. В. Управління персоналом : навч. посіб. Київ : Кондор, 2018. 308 с.
31. ДСанПіН 3.3.2.007-98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. Київ : МОЗ України, 1998.
32. International Labour Organization. Occupational Safety and Health. URL: <https://www.ilo.org/global/topics/safety-and-health-at-work> (дата звернення: 01.06.2026).
33. Про порядок проведення атестації робочих місць за умовами праці : Постанова Кабінету Міністрів України від 01.08.1992 р. № 442. URL: <https://zakon.rada.gov.ua/laws/show/442-92-п> (дата звернення: 01.06.2026).
34. Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу : наказ Міністерства охорони здоров'я України від 08.04.2014 р. № 248. URL: <https://zakon.rada.gov.ua/laws/show/z0472-14> (дата звернення: 01.06.2026).

## **ДОДАТКИ**

## ДОДАТОК А

### Основні файли взаємодії елементів системи

#### Лістинг коду А.1 – Модель користувача системи (User.cs)

```

namespace SBiathlonTrack.Models
{
    public class User
    {
        // Унікальний ідентифікатор користувача
        public string Id { get; set; } =
Guid.NewGuid().ToString();

        // Порядковий номер спортсмена або тренера
        public string CustomId { get; set; } = string.Empty;

        // Особисті дані
        public string FullName { get; set; } = string.Empty;

        public string Email { get; set; } = string.Empty;

        // Хеш пароля
        public string PasswordHash { get; set; } =
string.Empty;

        // Роль користувача: Athlete, Coach, Admin
        public string Role { get; set; } = "Athlete";

        // Дата народження
        public DateTime BirthDate { get; set; }

        // Дата створення облікового запису
        public DateTime CreatedAt { get; set; } =
DateTime.UtcNow;

        // Ознака активності облікового запису
        public bool IsActive { get; set; } = true;
    }
}

```

#### Лістинг коду Б.1 – Модель тренувального заняття (TrainingSession.cs)

```

namespace SBiathlonTrack.Models
{
    public class TrainingSession
    {
        // Унікальний ідентифікатор тренування

```

```

        public string Id { get; set; } =
Guid.NewGuid().ToString();

        // Ідентифікатор спортсмена, якому належить тренування
public string AthleteId { get; set; } = string.Empty;

        // Назва або тип тренування
public string Title { get; set; } = string.Empty;

        // Дата проведення тренування
public DateTime TrainingDate { get; set; }

        // Час початку тренування
public TimeSpan StartTime { get; set; }

        // Тривалість тренування
public TimeSpan Duration { get; set; }

        // Дистанція тренування в кілометрах
public double DistanceKm { get; set; }

        // Пульсові показники
public int AverageHeartRate { get; set; }

public int MaxHeartRate { get; set; }

public int MinHeartRate { get; set; }

        // Самопочуття спортсмена після тренування
public string WellBeing { get; set; } = string.Empty;

        // Додаткові нотатки
public string Notes { get; set; } = string.Empty;

        // Ознака наявності стрільби у тренуванні
public bool HasShooting { get; set; }

        // Результати стрільби
public ShootingResult? ShootingResult { get; set; }

        // Дата створення запису
        public DateTime CreatedAt { get; set; } =
DateTime.UtcNow;
    }

    public class ShootingResult
    {
        // Кількість використаних патронів
public int TotalCartridges { get; set; }

        // Промахи під час стрільби лежачи
public int ProneMisses { get; set; }
    }

```

```

// Промахи під час стрільби стоячи
public int StandingMisses { get; set; }

// Загальна кількість промахів
public int TotalMisses => ProneMisses + StandingMisses;

// Кількість влучань
public int Hits => TotalCartridges - TotalMisses;

// Точність стрільби у відсотках
public double Accuracy =>
    TotalCartridges == 0 ? 0 : Math.Round((double)Hits
/ TotalCartridges * 100, 2);
    }
}

```

### Лістинг В.1 – Реалізація реєстрації користувача

```

public async Task<bool> RegisterUserAsync(User user)
{
    try
    {
        // Перевірка існування користувача з таким email
        var existingUser = await
GetUserByEmailAsync(user.Email);

        if (existingUser != null)
        {
            return false;
        }

        // Генерація унікального ідентифікатора
        user.Id = Guid.NewGuid().ToString();

        // Формування внутрішнього номера користувача
        user.CustomId = $"ATH-{DateTime.Now:yyyyMMddHHmmss}";

        // Дата створення облікового запису
        user.CreatedAt = DateTime.UtcNow;

        // Збереження даних користувача
        await _firebaseClient
            .Child("users")
            .Child(user.Id)
            .PutAsync(user);

        return true;
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Registration error: {ex.Message}");
        return false;
    }
}

```

```
    }
}
```

### Лістинг Г.1 – Реалізація авторизації користувача

```
public async Task<User?> LoginUserAsync(string email, string
password)
{
    try
    {
        // Отримання всіх користувачів із бази даних
        var users = await _firebaseClient
            .Child("users")
            .OnceAsync<User>();

        // Пошук користувача за email та паролем
        var user = users
            .Select(u => u.Object)
            .FirstOrDefault(u =>
                u.Email.Equals(email,
StringComparison.OrdinalIgnoreCase) &&
                u.PasswordHash == password);

        return user;
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Login error: {ex.Message}");
        return null;
    }
}
```

### Лістинг Д.1 – Обробка авторизації у мобільному застосунку

```
private async void OnLoginClicked(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(EmailEntry.Text) ||
        string.IsNullOrEmpty>PasswordEntry.Text))
    {
        await DisplayAlert(
            "Помилка",
            "Усі поля повинні бути заповнені",
            "ОК");
    }

    return;
}

var user = await _firebaseService.LoginUserAsync(
    EmailEntry.Text.Trim(),
    PasswordEntry.Text);
```

```

if (user != null)
{
    Preferences.Set("UserId", user.Id);
    Preferences.Set("UserRole", user.Role);

    await Shell.Current.GoToAsync("//DashboardPage");
}
else
{
    await DisplayAlert(
        "Помилка",
        "Невірний логін або пароль",
        "OK");
}
}

```

### Лістинг Е.1 – Збереження тренувального заняття

```

private async void OnSaveTrainingClicked(object sender,
EventArgs e)
{
    try
    {
        var trainingSession = new TrainingSession
        {
            Id = Guid.NewGuid().ToString(),
            AthleteId = Preferences.Get("UserId",
string.Empty),
            Title = TrainingTypePicker.SelectedItem?.ToString()
?? "Тренування",
            TrainingDate = TrainingDatePicker.Date,
            Duration = TimeSpan.FromMinutes(
                Convert.ToInt32(DurationEntry.Text)),
            DistanceKm = Convert.ToDouble(DistanceEntry.Text),
            AverageHeartRate =
Convert.ToInt32(AvgHeartRateEntry.Text),
            MaxHeartRate =
Convert.ToInt32(MaxHeartRateEntry.Text),
            WellBeing = WellBeingEditor.Text,
            Notes = NotesEditor.Text,
            CreatedAt = DateTime.UtcNow
        };

        await
_firestoreService.SaveTrainingAsync(trainingSession);

        await DisplayAlert(
            "Успіх",
            "Тренування успішно збережено",
            "OK");

        await Navigation.PopAsync();
    }
}

```

```

    }
    catch (Exception ex)
    {
        await DisplayAlert(
            "Помилка",
            $"Не вдалося зберегти тренування: {ex.Message}",
            "OK");
    }
}

```

### Лістинг Ж.1 – Завантаження та фільтрація тренувань спортсмена

```

private async Task LoadTrainingsAsync()
{
    try
    {
        string athleteId = Preferences.Get("UserId",
string.Empty);

        var trainings = await _firebaseService
            .GetTrainingsByAthleteAsync(athleteId);

        TrainingsCollectionView.ItemsSource = trainings
            .OrderByDescending(t => t.TrainingDate)
            .ToList();

        TotalTrainingsLabel.Text =
            trainings.Count.ToString();

        TotalDistanceLabel.Text =
            trainings.Sum(t => t.DistanceKm)
                .ToString("F1") + " км";
    }
    catch (Exception ex)
    {
        await DisplayAlert(
            "Помилка",
            $"Не вдалося завантажити дані: {ex.Message}",
            "OK");
    }
}

```

### Лістинг З.1 – Налаштування застосунку .NET MAUI

```

public static class MauiProgram
{
    public static MauiApp CreateMauiApp()
    {
        var builder = MauiApp.CreateBuilder();

        builder

```

```
.UseMauiApp<App>()
.ConfigureFonts(fonts =>
{
    fonts.AddFont(
        "OpenSans-Regular.ttf",
        "OpenSansRegular");

    fonts.AddFont(
        "OpenSans-Semibold.ttf",
        "OpenSansSemibold");
});

builder.Services.AddSingleton<FirebaseService>();

builder.Services.AddSingleton<LoginPage>();
builder.Services.AddSingleton<AthleteDashboardPage>();
builder.Services.AddSingleton<AddTrainingPage>();

return builder.Build();
}
}
```