

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

Бакалаврська кваліфікаційна робота

на тему: «Розробка мобільного застосунку для формування рекомендацій щодо
занять у тренажерному залі»

Виконав: студент IV курсу
групи 5ПІ-216
спеціальності

121 – Інженерія програмного забезпечення

(шифр і назва напрямку підготовки, спеціальності)

Донець В. А.

(прізвище та ініціали)

Керівник: к.т.н., доц. каф. ПЗ Ткаченко О. М.

(прізвище та ініціали)

Рецензент: к.т.н., доц. каф. ОТ Кадук О. В.

(прізвище та ініціали)

Допущено до захисту

Зав. кафедри ПЗ Романюк О. Н.

09 » 06 2025 р.

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Рівень вищої освіти перший (бакалаврський)
Галузь знань 12 - «Інформаційні технології»
Спеціальність 121 - «Інженерія програмного забезпечення»
Освітньо-професійна програма - «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
д.т.н., професор
О. Н. Романюк
«21» березня 2025 р.

З А В Д А Н Н Я НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Донцю Владиславу Андрійовичу

1. Тема роботи - розробка мобільного застосунку для формування рекомендацій щодо занять у тренажерному залі.

Керівник роботи: Ткаченко Олександр Миколайович, к.т.н., доцент кафедри ПЗ, затверджені наказом вищого навчального закладу від 20 березня 2025 року № 97.

2. Строк подання студентом роботи 2 червня 2025.

3. Вихідні дані до роботи: середовище розробки Android Studio, мова розробки – Kotlin, операційна система – Android.

4. Зміст текстової частини: вступ; аналіз розвитку технологій мобільних систем для формування рекомендацій щодо занять у тренажерному залі та постановка задачі дослідження; розробка структури та алгоритмів мобільного застосунку; розробка програмних модулів для реалізації створення індивідуальних тренувальних планів; тестування програми; висновки; список використаних джерел; додатки.

5. Перелік графічного матеріалу: титульний слайд; актуальність теми; мета, об'єкт та предмет дослідження; новизна, практична цінність отриманих результатів, аналіз сучасних систем, структура мобільної системи, розробка графічного інтерфейсу, тестування застосунку, апробація результатів роботи і публікації, висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1-4	Ткаченко Олександр Миколайович	24.03.25 <i>Міс</i>	01.06.25 <i>Міс</i>

7. Дата видачі завдання 24 березня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз розвитку технологій мобільних систем для формування рекомендацій щодо занять в тренажерному залі та постановка задачі дослідження	25.03.25 - 07.04.25	<i>Вик.</i>
2	Розробка структури та алгоритмів мобільного застосунку	08.04.25 - 21.04.25	<i>Вик.</i>
3	Розробка програмних модулів для реалізації створення індивідуальних тренувальних планів	22.04.25 - 01.05.25	<i>Вик.</i>
4	Тестування програми	02.05.25 - 14.05.25	<i>Вик.</i>
5	Оформлення матеріалів до захисту БКР	15.05.25 - 30.05.25	<i>Вик.</i>

Студент

(підпис)

В.А. Донець

(прізвище та ініціали)

Керівник бакалаврської кваліфікаційної роботи

(підпис)

О.М. Ткаченко

(прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська кваліфікаційна робота складається з 73 сторінок формату А4, на яких представлено 15 рисунків, 2 таблиці та список використаних джерел, що налічує 20 найменувань.

У бакалаврській кваліфікаційній роботі було проведено аналіз сучасних мобільних систем, орієнтованих на фітнес та тренування. Розглянуто наявні аналоги, визначено їхні переваги та недоліки, що дозволило обґрунтувати доцільність розробки власного мобільного застосунку.

Розроблений мобільний застосунок покликаний покращити процес формування індивідуальних рекомендацій щодо занять у тренажерному залі. Система надає користувачеві можливість зручно переглядати вправи за групами м'язів, формувати персоналізований тижневий план тренувань, а також відстежувати свій прогрес (наприклад, зміну ваги або кількість підходів).

Мобільний застосунок реалізовано за допомогою мови програмування Kotlin із використанням Jetpack Compose для побудови інтерфейсу користувача, Room для локального зберігання даних, а також Coroutines і Kotlin Flows для обробки асинхронних операцій. Середовищем розробки було обрано Android Studio.

У результаті виконання бакалаврської кваліфікаційної роботи створено мобільний застосунок, що підвищує ефективність та зручність тренувального процесу завдяки автоматизованому формуванню рекомендацій. Отримані результати можуть бути використані для подальшого вдосконалення цифрових рішень у сфері фітнесу.

Ключові слова: застосунок, фітнес, тренування, рекомендації.

ANNOTATION

The bachelor's thesis consists of 73 A4 pages, featuring 15 figures, 2 tables, and a list of 20 references.

The thesis includes an analysis of modern mobile systems focused on fitness and training. Existing analogues were reviewed, their advantages and disadvantages were identified, which justified the feasibility of developing a custom mobile application.

The developed mobile application is aimed at improving the process of generating individual workout recommendations for gym sessions. The system allows users to conveniently view exercises by muscle groups, create a personalized weekly workout plan, and track their progress (e.g., changes in weight or number of sets).

The mobile application was implemented using the Kotlin programming language with Jetpack Compose for building the user interface, Room for local data storage, and Coroutines and Kotlin Flows for handling asynchronous operations. Android Studio was chosen as the development environment.

As a result of the bachelor's thesis, a mobile application was created that enhances the efficiency and convenience of the workout process through automated recommendation generation. The obtained results can be used to further improve digital solutions in the fitness domain.

Keywords: application, fitness, training, recommendations.

ЗМІСТ

ВСТУП.....	3
1 АНАЛІЗ РОЗВИТКУ ТЕХНОЛОГІЙ МОБІЛЬНИХ СИСТЕМ ДЛЯ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ ЩОДО ЗАНЯТЬ У ТРЕНАЖЕРНОМУ ЗАЛІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ.....	7
1.1 Аналіз стану технологій формування тренувальних рекомендацій на основі мобільних застосунків у сфері фітнесу.....	7
1.2 Порівняльний аналіз аналогів.....	9
1.3 Аналіз методів розв’язання задачі.....	13
1.4 Постановка задач бакалаврської кваліфікаційної роботи.....	15
1.5 Висновки	17
2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМІВ МОБІЛЬНОГО ЗАСТОСУНКУ .	18
2.1 Розробка архітектури мобільного застосунку.....	18
2.2 Розробка інтерфейсу програмного додатку.....	21
2.3 Розробка алгоритму роботи системи	25
2.4 Розробка моделі системи.....	27
2.5 Розробка методу формування індивідуального плану тренувань.....	28
2.6 Висновки	29
3. РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ ДЛЯ РЕАЛІЗАЦІЇ СТВОРЕННЯ ІНДИВІДУАЛЬНИХ ТРЕНУВАЛЬНИХ ПЛАНІВ.	30
3.1 Варіантний аналіз і обґрунтування вибору засобів для реалізації програмного забезпечення	30
3.2 Реалізація модулів обробки даних користувача у середовищі Android	31
3.3 Розробка головного вікна програми.....	34
3.4 Висновки	38
4 ТЕСТУВАННЯ ПРОГРАМИ	39
4.1 Тестування системи	40
4.2 Розробка інструкції користувача	50
4.3 Висновки	52
ВИСНОВКИ.....	54
ЛІТЕРАТУРА.....	55
ДОДАТКИ.....	57
Додаток А. – Технічне завдання	58
Додаток Б. – Перевірка на плагіат.....	61
Додаток В. – Лістинг програми	62
Додаток Г. – Графічна частина	69

ВСТУП

Обґрунтування вибору теми дослідження. У сучасному суспільстві все більше людей потребує ефективної організації тренувального процесу. Для кожного з нас важливо мати надійний доступ до інструментів, які допомагають в оптимізації фізичних навантажень і досягненні цілей. Завдяки прогресу в технологіях мобільних додатків, тренування стали доступнішими і персоналізованими. З ростом популярності мобільних додатків для здоров'я та фітнесу виникає необхідність у розробці таких систем, які не лише моніторять фізичні параметри, але й пропонують персоналізовані рекомендації для тренувань, враховуючи індивідуальні особливості користувачів. Вони допомагають користувачам дотримуватися здорового способу життя, встановлюючи чіткі цілі й пропонуючи рекомендації, що максимально відповідають їх потребам.

Одним із важливих аспектів таких додатків є збір даних про користувача, таких як вага, зріст, рівень фізичної підготовки. Це дозволяє створювати точні рекомендації для кожного користувача, оптимізуючи тренування для досягнення найкращих результатів. Інтеграція таких інструментів, як Jetpack Compose для створення графічного інтерфейсу та Room для збереження даних, дозволяє розробляти зручні та ефективні рішення, які задовольняють вимоги користувачів.

Один з основних аспектів розробки таких додатків — це забезпечення асинхронного виконання операцій і зберігання даних без переривання роботи користувача. Використання Coroutines та Kotlin Flows дозволяє забезпечити безперебійну роботу програми, навіть коли кількість даних значна.

Сучасні технології також дозволяють здійснювати інтеграцію з різноманітними носимими пристроями та фітнес-трекерами, що дозволяє більш точно вимірювати фізичні параметри користувача в реальному часі. Завдяки цьому дані про фізичний стан користувача постійно оновлюються, що дає змогу створювати ще більш персоналізовані рекомендації. Це, в свою чергу, підвищує ефективність тренувань та знижує ризик травм.

Мобільні застосунки значно підвищують ефективність тренувань,

допомагаючи користувачам досягати своїх цілей у найкоротші терміни. Технології, які використовуються в таких додатках, також сприяють вдосконаленню процесу тренувань і покращенню якості фізичної підготовки, що робить їх особливо популярними серед активних користувачів. Таким чином, розробка програмного засобу для формування рекомендацій щодо покращення тренувального процесу є актуальною задачею.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалася згідно плану виконання наукових досліджень на кафедрі 5 програмного забезпечення.

Мета та завдання дослідження. Метою роботи є підвищення ефективності тренувального процесу та зменшення ризику травмування за рахунок використання програмних засобів, які дозволяють адаптувати тренування під індивідуальні фізичні параметри користувача.

Основними задачами дослідження є:

- аналіз стану технологій формування тренувальних рекомендацій на основі мобільних застосунків у сфері фітнесу;
- порівняльний аналіз аналогів;
- аналіз методів розв'язання задачі;
- постановка задач бакалаврської кваліфікаційної роботи;
- аналіз вхідних даних системи;
- розробка інтерфейсу програмного додатку;
- розробка моделі системи;
- розробка алгоритму роботи системи;
- варіантний аналіз і обґрунтування вибору засобів для реалізації програмного забезпечення;
- реалізація модулів обробки даних користувача у середовищі Android;
- тестування системи;
- розробка інструкції користувача;
- розробка алгоритму роботи системи;
- висновки

Об'єкт дослідження – процес тренувань у тренажерному залі.

Предмет дослідження – методи та засоби створення мобільного застосунку для моніторингу та формування рекомендацій щодо процесу тренувань.

Методи дослідження. У процесі дослідження використовувались такі методи: порівняльний аналіз функціональних характеристик програм-аналогів з метою виявлення їхніх недоліків і формулювання завдань розробки; використання теоретичних засад алгоритміки для побудови та візуалізації алгоритмів функціонування програмного забезпечення; комп'ютерне моделювання для перевірки ефективності розроблених рішень і підтвердження теоретичних положень.

Наукова новизна отриманих результатів:

1. Вперше запропоновано нову архітектуру програмного застосунку, особливістю якої є інтеграція з носимими пристроями для отримання актуальних фізіологічних даних у реальному часі, що дозволило оперативно оновлювати тренувальні рекомендації, підвищуючи ефективність і безпечність тренувань.

2. Подальшого розвитку отримав метод формування індивідуального плану тренувань, в якому, на відміну від існуючих, реалізовано автоматичну генерацію персоналізованих рекомендацій з урахуванням динаміки прогресу на основі фізичних параметрів користувача, що дозволило підвищити ефективність тренувального процесу та забезпечило адаптацію навантажень у реальному часі.

Практична цінність отриманих результатів. Практична цінність одержаних результатів полягає в тому, що на основі теоретичних положень, розроблених у рамках бакалаврської кваліфікаційної роботи, було створено мобільний застосунок для формування індивідуальних рекомендацій щодо тренувань у тренажерному залі. Застосунок реалізовано мовою програмування Kotlin, що забезпечує його ефективність, масштабованість і зручність використання.

Особистий внесок здобувача.

Усі результати, викладені у роботі, отримані автором особисто. В роботі [1], опублікованій у співавторстві, автору належить розробка архітектури застосунку.

Апробація матеріалів бакалаврської кваліфікаційної роботи. Результати роботи було представлено на Міжнародній науково-практичній інтернет-конференції Молодь в науці: дослідження, проблеми, перспективи (МН-2025) (Вінниця, 2025 р.).

Публікації. Результати роботи опубліковано в матеріалах конференції Молодь в науці: дослідження, проблеми, перспективи (МН-2025) [1].

1 АНАЛІЗ РОЗВИТКУ ТЕХНОЛОГІЙ МОБІЛЬНИХ СИСТЕМ ДЛЯ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ ЩОДО ЗАНЯТЬ У ТРЕНАЖЕРНОМУ ЗАЛІ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз стану технологій формування тренувальних рекомендацій на основі мобільних застосунків у сфері фітнесу

У сучасному світі мобільні додатки відіграють важливу роль у сфері фітнесу та здорового способу життя. З розвитком цифрових технологій усе більше людей обирають персоналізовані цифрові рішення для покращення фізичної форми, контролю прогресу та підвищення мотивації до регулярних занять спортом. Такий підхід дозволяє не лише зручно організувати тренувальний процес, а й враховувати індивідуальні особливості кожного користувача, що суттєво підвищує ефективність досягнення поставлених цілей.

Значення мобільних фітнес-додатків

Мобільні фітнес-додатки стали незамінним інструментом для користувачів різного віку та рівня підготовки — від початківців до професійних спортсменів. Вони надають доступ до великої кількості вправ, відеоуроків, рекомендацій щодо харчування, відстеження активності, сну, пульсу, кількості пройдених кроків тощо. За допомогою вбудованих або підключених сенсорів, таких як фітнес-браслети чи смарт-годинники, додатки можуть отримувати точні фізіологічні дані користувача, що дозволяє формувати ще більш адаптовані тренувальні програми.

Переваги мобільних фітнес-додатків

1. Доступність та зручність використання: більшість додатків доступні безкоштовно або за символічну плату, встановлюються на будь-який сучасний смартфон і можуть використовуватись у будь-якому місці — вдома, у спортзалі, на вулиці.
2. Персоналізація тренувань: користувач має можливість обирати цілі (схуднення, набір м'язової маси, покращення витривалості), рівень складності вправ, час тренування та інші параметри.

3. Наочне відстеження прогресу: завдяки візуалізації результатів (графіки, звіти, досягнення) користувач бачить свої успіхи й мотивується не зупинятись.

4. Інтеграція з фітнес-трекерами та розумними пристроями: автоматичне зчитування даних дозволяє уникнути ручного введення показників та підвищує точність розрахунків.

5. Мотиваційні механізми: нагадування, челенджі, рейтинги, значки за досягнення сприяють формуванню звички до регулярних тренувань.

Недоліки мобільних фітнес-додатків

6. Обмежена точність рекомендацій без якісних вхідних даних: без достовірної інформації про стан здоров'я користувача, його фізичну підготовку та обмеження, ризик дати шкідливу рекомендацію зростає.

7. Залежність від стабільної роботи пристрою: збій у роботі додатка або втрата підключення до мережі може завадити тренуванню чи зіпсувати відстеження прогресу.

8. Наявність платного функціоналу: більшість популярних додатків обмежують безкоштовний доступ до повноцінних тренувальних програм, дієт чи персональних рекомендацій.

9. Відсутність єдиного стандарту побудови програм тренувань: різні додатки ґрунтуються на різних методиках, що ускладнює вибір оптимального варіанту для користувача.

10. Недостатня адаптація під специфічні потреби: деякі додатки не враховують обмеження, пов'язані з травмами, хронічними захворюваннями чи іншими особливостями користувача.

Актуальні виклики та проблеми

Основною проблемою залишається відсутність доступних та гнучких рішень, які б одночасно поєднували індивідуалізацію, простоту використання, сучасний інтерфейс та науково обґрунтовані методики формування тренувальних програм. Більшість існуючих додатків мають закриту структуру, що унеможливорює

модифікацію логіки або інтерфейсу під конкретні запити користувача. Крім того, відсутність єдиного підходу до аналізу даних і формування рекомендацій часто призводить до неточностей або неефективних порад.

Напрями розвитку фітнес-додатків

У майбутньому розвиток мобільних фітнес-рішень буде зосереджений на наступних аспектах:

1. Штучний інтелект і машинне навчання: для автоматичного аналізу фізичних показників та динамічного формування індивідуальних планів занять.
2. Глибша інтеграція з медичними системами: для забезпечення безпечного використання додатків особами з хронічними захворюваннями або після реабілітації.
3. Соціальні функції: можливість тренуватись у групах, ділитись результатами, проходити челенджі разом із друзями або тренерами.
4. Гейміфікація: використання ігрових механік для підвищення мотивації — рівні, досягнення, нагороди, звання тощо.
5. AR/VR технології: доповнена або віртуальна реальність, що дозволяє створювати віртуальні зали, тренуватись з анімованими тренерами тощо.

Фітнес-додатки є потужним інструментом у формуванні здорового способу життя, але їхня ефективність напряму залежить від якості реалізації, точності аналізу даних і рівня персоналізації. Тому розробка нових мобільних рішень має бути спрямована на усунення існуючих обмежень, підвищення адаптивності, а також забезпечення балансу між простотою використання, функціональністю та науковою обґрунтованістю тренувальних рекомендацій [2].

1.2 Порівняльний аналіз аналогів

Проведемо порівняльний аналіз додатків для фітнесу, що формують індивідуальні рекомендації щодо занять у тренажерному залі. Розглянемо популярні рішення: Nike Training Club, Adidas Training, Fitify та Freeletics.

Nike Training Club — це один із найбільш відомих фітнес-додатків, який пропонує велику бібліотеку тренувань, що охоплюють різні аспекти загальної фізичної підготовки: кардіо, сила, йога, мобільність тощо. Додаток вирізняється доступністю — більшість його контенту є безкоштовним, включаючи відеоуроки з чіткою візуалізацією вправ та рекомендаціями від професійних тренерів. Інтерфейс простий і зручний у використанні, що робить його особливо привабливим для новачків. Однак, система персоналізації в Nike Training Club є досить базовою: користувач обирає ціль тренувань, рівень складності та тривалість, але фізіологічні показники або зміни у формі не враховуються. Це обмежує глибину адаптації планів під конкретного користувача, що може бути недоліком для тих, хто шукає справді індивідуалізовані рішення.

Adidas Training, раніше відомий як Runtastic, також пропонує персоналізовані фітнес-програми, орієнтовані на досягнення конкретних цілей: схуднення, набір м'язової маси або покращення загального тону. На відміну від Nike Training Club, він містить преміальний функціонал, який відкриває доступ до розширених програм, спеціалізованих тренувань і більш глибокої персоналізації. Додаток враховує базові параметри користувача, як-от стать, вік, частота тренувань, однак також має обмеження у динамічній адаптації. Деякі функції доступні лише за передплатою, що може бути бар'єром для частини користувачів. Тим не менш, візуальна складова, якість матеріалів та брендова підтримка роблять Adidas Training надійним вибором для широкої аудиторії.

Fitify вирізняється акцентом на варіативність у виборі обладнання — користувач може проходити тренування як з власною вагою, так і з гантелями, медболами, еспандерами та іншими інструментами. Це робить застосунок гнучким у використанні як удома, так і в тренажерному залі. Персоналізація тут проявляється через вибір мети, наявного обладнання, рівня складності та тривалості сесій. Але, попри адаптивність тренувань, Fitify недостатньо аналізує історію тренувань, не враховує медичні обмеження чи зміни в стані здоров'я. Відсутність активного моніторингу результатів або фітнес-аналітики також обмежує можливість глибокої персоналізації. Проте додаток є зручним,

функціональним і добре підходить для користувачів, які бажають гнучких рішень без надмірного ускладнення.

Freeletics позиціонується як застосунок для інтенсивних тренувань за методом високої інтенсивності (НІТ). Його особливістю є наявність потужного алгоритму персоналізації, який формує тренування на основі даних, отриманих з початкового тестування та опитування користувача. Додаток використовує штучний інтелект для адаптації навантаження, з урахуванням ефективності попередніх сесій. Програма може змінювати інтенсивність, тривалість або склад вправ відповідно до відгуків користувача. Однак, попри високий рівень технологічності, більшість функціоналу, включаючи повноцінний доступ до персонального тренера та адаптивних планів, доступна лише за передплатою. Крім того, не всі фізіологічні показники враховуються — наприклад, інтеграція з трекерами або можливість відстеження змін пульсу є обмеженою. Це може бути критичним для користувачів з особливими потребами або тим, хто прагне точного контролю за навантаженням.

Розглянемо можливості програм з готовими рішеннями (див. таблицю 1.1) та порівняємо їх за кількома ключовими критеріями:

- Можливість зміни профілю занять за розкладом — дозволяє користувачу адаптувати тренування відповідно до свого щоденного графіка, змінюючи інтенсивність або тип вправ.

- Врахування зовнішніх умов — можливість коригувати тренувальний план з урахуванням стану здоров'я, рівня стресу або навантаження протягом дня.

- Контроль фактичного прогресу — відстеження змін у фізичному стані користувача, що дозволяє коригувати план тренувань для досягнення кращих результатів.

- Інтеграція з іншими фітнес-сервісами — забезпечує об'єднання даних з різних джерел для точнішої оцінки стану користувача та кращої адаптації тренувань.

- Сумісність з додатковим обладнанням — підтримка фітнес-браслетів, пульсометрів та інших пристроїв для збору фізіологічних даних.

- Програмування тренувального плану — дозволяє користувачам створювати індивідуальні тренування відповідно до своїх цілей та рівня підготовки.

Таблиця 1.1 – Порівняльні характеристики фітнес-додатків

Критерії	Nike Training Club	Adidas Training	Fitify	Freeletics	Власна розробка
Персоналізація тренувань	+	+	+	+	+
Врахування цілей користувача	+	+	+	+	+
Врахування фізіологічних даних	-	+	-	+	+
Безкоштовна функціональність	+	-	+	-	+
Можливість офлайн-режиму	+	+	+	+	+
Ведення статистики/прогресу	+	+	-	+	+
Загальна оцінка	80%	80%	60%	80%	100%

Отже, порівняльний аналіз виявив, що наявні рішення мають як сильні сторони, так і суттєві обмеження. Жоден з них повністю не задовольняє потреби користувачів, які хочуть отримати повністю персоналізовані рекомендації з

урахуванням фізіологічних показників, історії тренувань, прогресу та стану здоров'я. Розробка власного мобільного додатку є доцільною, адже дозволить створити більш адаптивну, доступну та ефективну систему, яка відповідатиме індивідуальним потребам користувача.

1.3 Аналіз методів розв'язання задачі

Розробка мобільного застосунку для формування персоналізованих рекомендацій щодо занять у тренажерному залі вимагає комплексного підходу до аналізу фізичних параметрів користувача, рівня його підготовки та цілей тренувань. Існує низка підходів до створення таких систем, кожен з яких має свої переваги й обмеження. У цьому розділі проаналізовано найбільш ефективні методи побудови рекомендаційних систем для фітнес-застосунків.

Одним із базових варіантів є створення шаблонних програм тренувань на основі вибору цілей (схуднення, набір м'язової маси, загальна фізична форма). Такий підхід простий у реалізації, дозволяє швидко запускати додаток у роботу, не потребує складної логіки й аналізу даних. Однак він не враховує індивідуальні особливості користувача — антропометричні дані, фізичні обмеження, рівень прогресу, наявність травм тощо. Це знижує ефективність тренувань і може навіть призвести до зниження мотивації або погіршення фізичного стану користувача в разі некоректно підібраного навантаження.

Більш просунутим методом є використання анкетування в поєднанні з алгоритмами логіки прийняття рішень (rule-based systems). Система збирає дані про зріст, вагу, вік, стать, рівень фізичної підготовки, досвід занять, наявність медичних обмежень, бажану інтенсивність і тривалість тренувань. На основі цих даних формуються умовні правила, які обирають відповідну програму. Такий підхід забезпечує початкову персоналізацію, але він залишається статичним — не враховує зміни у формі користувача, його самопочуття чи прогрес у динаміці. Крім того, з часом користувачі можуть втратити інтерес до незмінного набору вправ, навіть якщо ті спочатку відповідали їхнім потребам.

Перспективним і найбільш технологічно розвиненим є застосування

алгоритмів машинного навчання. Такі системи аналізують великі обсяги вхідних даних: історію тренувань, кількість підходів, повторень, частоту серцевих скорочень, витрачені калорії, час відновлення, оцінки самопочуття, результати фітнес-тестів тощо. Вони здатні адаптувати план тренувань у реальному часі, пропонуючи зміни на основі аналізу ефективності попередніх занять. Це дозволяє досягти високого рівня персоналізації, зменшити ризики перенавантаження чи недостатньої інтенсивності, покращити мотивацію завдяки чіткому відчуттю прогресу. Водночас такі системи потребують ретельно підготовлених датасетів, налаштування моделей, оцінки точності прогнозів, а також постійного моніторингу результатів роботи для уникнення помилок і некоректних рекомендацій.

Ще один важливий аспект — використання зовнішніх джерел даних, зокрема трекерів активності (фітнес-браслетів, смарт-годинників, смартфонів з вбудованими сенсорами). Ці пристрої надають точну інформацію про пульс, кількість кроків, підйомів сходами, рівень активності протягом дня, якість сну, а також можуть вимірювати рівень кисню в крові чи варіабельність серцевого ритму. Інтеграція таких даних до системи персоналізованих рекомендацій значно підвищує точність аналізу фізичного навантаження та дозволяє коригувати плани залежно від фактичного стану користувача, а не лише від запланованих показників.

З огляду на переваги та недоліки кожного з підходів, у межах даної роботи було обрано комбінований метод — поєднання початкового анкетування з динамічною адаптацією планів тренувань на основі даних користувача та інтеграції з трекерами активності. Такий підхід забезпечує гнучкість, адаптивність і точність рекомендацій. Застосунок може змінювати інтенсивність, види вправ або частоту тренувань залежно від змін у фізіологічних показниках, рівні навантаження, кількості виконаних занять або відгуків користувача. Наприклад, у разі високої втомлюваності або погіршення показників пульсу система може зменшити навантаження або запропонувати день відновлення.

Крім того, у межах реалізації додатку передбачено можливість ручного налаштування програм для просунутих користувачів, які бажають самостійно коригувати тренування, а також модуль рекомендацій щодо харчування та

гідратації на основі добового навантаження.

Таким чином, реалізований підхід поєднує інтуїтивність та простоту для новачків із гнучкістю та потужним функціоналом для досвідчених користувачів. Така концепція дозволяє створити конкурентоспроможний продукт із високою користувацькою цінністю, який адаптується до потреб конкретної аудиторії та сприяє досягненню фітнес-цілей ефективніше, ніж традиційні підходи.

1.4 Постановка задач бакалаврської кваліфікаційної роботи

Після аналізу стану програм для формування індивідуальних тренувальних планів та порівняння існуючих аналогів було визначено основні завдання, які необхідно реалізувати для створення ефективного, надійного та зручного у користуванні мобільного застосунку. Основна мета полягає в розробці інструменту, що дозволить користувачам отримувати персоналізовані тренувальні програми, відслідковувати свій прогрес та підтримувати мотивацію до досягнення спортивних цілей.

До ключових завдань, які слід реалізувати у процесі створення застосунку, належать:

1) Розробка модуля збору даних користувача.

Необхідно створити функціонал для введення користувачем своїх фізичних параметрів (зріст, вага, вік, стать), рівня фізичної підготовки, особистих фітнес-цілей (набір м'язової маси, схуднення, підтримка форми) та інших характеристик, що впливають на підбір оптимальної програми тренувань. Також передбачається можливість оновлення даних у міру зміни фізичних показників користувача.

2) Реалізація модуля обробки даних та формування рекомендацій.

Цей модуль відповідатиме за аналіз зібраної інформації, історії тренувань та прогресу користувача. На основі цих даних застосунок генеруватиме персоналізовані плани тренувань, адаптуючи інтенсивність, обсяг навантажень та тип вправ відповідно до обраної мети та змін у фізичному стані користувача. Модуль має передбачати динамічне оновлення рекомендацій залежно від результатів користувача.

3) Створення бази даних для зберігання профілів користувачів та історії тренувань.

Для забезпечення надійного збереження всієї інформації про користувачів буде реалізовано локальну базу даних із використанням бібліотеки Room. У базі будуть зберігатися особисті профілі користувачів, параметри, історія тренувань, статистика виконаних вправ та інші важливі дані. Особливу увагу буде приділено безпеці та цілісності даних.

4) Реалізація інтерфейсу налаштувань програми.

Користувачі повинні мати можливість налаштовувати роботу застосунку під свої індивідуальні потреби. Для цього буде створено інтуїтивно зрозумілий інтерфейс налаштувань, де користувач зможе змінювати параметри профілю, обирати цілі, коригувати план тренувань, налаштовувати повідомлення про заплановані заняття, а також вмикати або вимикати різні функції програми відповідно до своїх уподобань.

5) Впровадження модуля авторизації та контролю доступу до персональних даних.

Для захисту персональної інформації буде реалізовано систему авторизації користувача. Передбачатиметься можливість реєстрації нового профілю, входу в існуючий профіль за допомогою облікових даних (логін/пароль) або через інтеграцію з популярними сервісами аутентифікації. Крім того, застосунок повинен забезпечувати контроль доступу до особистої інформації, використовуючи шифрування даних і, за можливості, біометричну авторизацію (наприклад, через відбиток пальця або розпізнавання обличчя).

6) Проведення тестування застосунку для перевірки коректності роботи всіх модулів.

На завершальному етапі розробки буде проведено комплексне тестування застосунку. Воно включатиме функціональне тестування окремих модулів, інтеграційне тестування взаємодії між компонентами системи, а також тестування користувацького інтерфейсу для оцінки зручності та інтуїтивності роботи програми. Особливу увагу буде приділено перевірці стабільності роботи під

навантаженням, швидкості відгуку інтерфейсу та захищеності персональних даних.

Реалізація всіх вищезазначених завдань дозволить створити конкурентоспроможний мобільний застосунок, який не лише задовольнятиме потреби користувачів у формуванні індивідуальних тренувальних планів, а й сприятиме підвищенню їх мотивації та досягненню кращих результатів у фізичному розвитку.

1.5 Висновки

У цьому розділі було проаналізовано актуальність проблеми автоматизованого формування тренувальних програм у мобільному форматі. Проведено детальний огляд існуючих рішень, визначено їхні ключові переваги та недоліки. Зокрема, встановлено, що більшість сучасних мобільних застосунків пропонують користувачам стандартні програми тренувань, які не враховують індивідуальні особливості, такі як рівень фізичної підготовки, стан здоров'я, поставлені цілі чи наявність обмежень. Крім того, більшість рішень не адаптуються до змін фізичної форми користувача в процесі тренувань і лише обмежено інтегруються з фітнес-трекерами та іншими пристроями збору фізіологічних даних.

На основі отриманих висновків обґрунтовано доцільність розробки нового мобільного застосунку, орієнтованого на глибоку персоналізацію тренувальних планів. Передбачається реалізація механізму динамічного оновлення рекомендацій відповідно до поточного прогресу користувача, змін у його фізичному стані та нових цілей. Особливу увагу буде приділено тісній інтеграції із сучасними фітнес-трекерами, розумними годинниками та іншими пристроями моніторингу активності для автоматичного збору даних.

У межах цієї роботи сформульовано основні завдання, серед яких — розробка функціональних модулів для побудови персоналізованих тренувальних програм, впровадження алгоритмів адаптації планів на основі аналізу отриманих фізіологічних даних, а також проведення комплексного тестування застосунку для виявлення та усунення можливих недоліків на різних етапах його використання.

2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ

2.1 Розробка архітектури мобільного застосунку

Для реалізації програмного продукту для формування рекомендацій щодо занять у тренажерному залі буде використано поєднання мови програмування Kotlin з фреймворком Jetpack Compose для створення сучасного, адаптивного інтерфейсу, а також бібліотек Room, Coroutines та Kotlin Flows для роботи з локальними даними та асинхронними потоками [3].

Kotlin — це сучасна, безпечна та зручна мова програмування, яка активно використовується для розробки Android-застосунків завдяки своїй лаконічності, підтримці функціонального програмування, сумісності з Java та активному розвитку екосистеми [4].

Jetpack Compose — це новий декларативний інструмент для побудови інтерфейсів користувача, який дозволяє значно спростити процес розробки UI шляхом винесення всієї логіки побудови екранів у кодову базу без необхідності використання XML-розмітки. Завдяки цьому підходу забезпечується більша гнучкість у створенні адаптивних та динамічних інтерфейсів [5].

Room — це ORM-бібліотека, яка абстрагує роботу з локальною базою даних SQLite, дозволяючи розробнику працювати з базою через об'єктну модель замість безпосереднього написання SQL-запитів. Це забезпечує надійне збереження даних про тренування, прогрес користувача, його фізичні параметри та інші важливі характеристики [6].

Coroutines і Kotlin Flows використовуються для реалізації асинхронної обробки даних, що дозволяє забезпечити плавність роботи застосунку, уникнути блокування основного потоку та ефективно управляти обробкою великих обсягів даних у реальному часі [7].

Розробка мобільного застосунку для тренажерного залу включає декілька основних етапів.

Першим етапом є створення інтерфейсного модуля, який відповідатиме за взаємодію користувача з додатком. У межах цього модуля буде реалізовано простий, інтуїтивно зрозумілий і привабливий дизайн, створений відповідно до сучасних стандартів UI/UX. Застосунок передбачає головний екран для навігації між основними функціональними модулями, екран для відображення та вибору тренувань, профіль користувача для управління персональними даними, а також модуль аналітики, де користувач зможе переглядати прогрес у вигляді графіків і статистичних звітів [8].

Наступним етапом стане розробка модуля збору та збереження даних. Він забезпечуватиме можливість введення та редагування фізичних параметрів користувача, таких як зріст, вага, вік, стать, рівень фізичної підготовки, наявність обмежень або травм, а також його фітнес-цілей (набір маси, схуднення, підтримка форми). Додатково передбачається можливість фіксації виконаних тренувань, ваги снарядів, кількості підходів і повторів. Уся ця інформація буде надійно зберігатися в локальній базі даних, створеній за допомогою бібліотеки Room.

Далі буде реалізовано модуль генерації рекомендацій. Він базуватиметься на аналізі введених користувачем даних, його історії тренувань, динаміки змін фізичних параметрів і обраної мети тренувань. На основі цих даних система формуватиме персоналізовані пропозиції щодо тренувань, що включатимуть вибір вправ, кількість підходів, повторень, рекомендовану вагу та частоту занять. Рекомендації будуть адаптивними, тобто змінюватимуться відповідно до успіхів користувача та прогресу, що дозволить підтримувати оптимальний рівень навантаження та уникати перетренованості або застою результатів.

Особливу увагу буде приділено мотиваційним елементам: реалізується система нагород за досягнення цілей, нагадування про заплановані тренування, поради щодо відновлення та харчування. У майбутньому також можливе розширення функціоналу за рахунок впровадження інтелектуальної обробки даних для ще точнішої персоналізації рекомендацій та інтеграції із зовнішніми пристроями — наприклад, фітнес-браслетами для збору додаткових метрик активності та здоров'я.

Таким чином, розробка мобільного застосунку охоплюватиме повний цикл роботи з користувачем — від збору початкової інформації та формування персоналізованих програм тренувань до аналізу результатів і підтримки мотивації на шляху до досягнення поставлених цілей.

Для реалізації всіх цих функцій буде використано Android Studio як основне середовище розробки, а також сторонні бібліотеки для побудови графіків, інтеграції з Google Fit, та реалізації темної/світлої теми інтерфейсу.

Буде вперше запропоновано нову архітектуру мобільного застосунку для формування індивідуальних тренувальних рекомендацій, яка поєднує функціональні модулі, орієнтовані на персоналізацію, інтерактивність і інтеграцію з зовнішніми джерелами даних. Особливістю розробленого рішення є підтримка інтеграції з носимими пристроями (фітнес-браслети, смарт-годинники тощо), що дозволяє отримувати актуальні фізіологічні показники користувача в режимі реального часу. Завдяки цьому забезпечується динамічне оновлення рекомендацій та корекція тренувальної програми, що суттєво підвищує ефективність і безпечність занять.

Архітектура застосунку (див. рисунок 2.1) складається з таких основних модулів:

- Інтерфейс користувача: забезпечує авторизацію, керування профілем, налаштування сповіщень і відображення статистичних діаграм. Цей модуль відповідає за взаємодію користувача із системою та візуалізацію результатів.
- Модуль збору й аналізу даних: реалізує механізми відстеження активності користувача, у тому числі шляхом обробки даних з носимих пристроїв. Отримані показники передаються для подальшої аналітики та адаптації тренувань.
- Модуль генерації програми тренувань: відповідає за формування індивідуального плану тренувань на основі вхідних даних, а також за його корекцію з урахуванням результатів і зміни фізичного стану користувача.
- Модуль підключення сторонніх додатків: забезпечує можливість інтеграції із зовнішніми сервісами та платформами, що дозволяє розширити

функціональність застосунку.



Рисунок 2.1 – Архитектура застосунку

Усі модулі взаємодіють між собою через чітко визначені інтерфейси обміну даними, що забезпечує гнучкість і масштабованість архітектури.

Таким чином, розробка мобільного додатку для формування тренувальних рекомендацій є комплексною роботою, що включає проєктування та реалізацію кількох взаємопов'язаних модулів. Завдяки правильному використанню сучасних інструментів Android-розробки, додаток буде не тільки зручним і функціональним, але й здатним забезпечити індивідуальний підхід до кожного користувача на основі його фізичних даних та цілей.

2.2 Розробка інтерфейсу програмного додатку

При розробці інтерфейсу мобільного застосунку було приділено особливу увагу його простоті, зрозумілості та зручності для користувача, що є ключовими аспектами у сфері фітнесу.

Основним кольором інтерфейсу обрано білий, оскільки він асоціюється з простотою, чистотою та легкістю у сприйнятті. Допоміжними кольорами

виступають чорний та оранжевий, які забезпечують добрий візуальний контраст, акцентують увагу користувача на важливих елементах та підкреслюють енергійний характер застосунку. Така кольорова палітра сприяє зручному сприйняттю інформації та комфортній взаємодії з інтерфейсом.

Після запуску застосунку користувач бачить екран завантаження з логотипом і назвою системи (див. рисунок 2.2), що створює перше враження про застосунок та забезпечує впізнаваність бренду.



Рисунок 2.2 – Екран завантаження

Оскільки програмний продукт передбачає взаємодію користувача з персоналізованими тренувальними даними та рекомендаціями, було вирішено створити зручне мобільне робоче середовище з чітким розмежуванням між основними функціональними блоками — тренуваннями, аналітикою, профілем користувача та налаштуваннями.

Для полегшення навігації та взаємодії з основним функціоналом було реалізовано головне меню застосунку у вигляді нижньої навігаційної панелі на рисунку 2.3. Меню дозволяє швидко перемикатися між основними розділами, забезпечуючи інтуїтивно зрозумілий користувацький досвід.



Рисунок 2.3 – Меню програмного застосунку

На рисунку 2.4 зображено початковий екран застосунку, який відображається після авторизації. Він містить стислу інформацію про поточну ціль користувача, короткий огляд прогресу. Такий підхід сприяє підвищенню мотивації користувача та покращує загальну імерсивність під час взаємодії із застосунком.

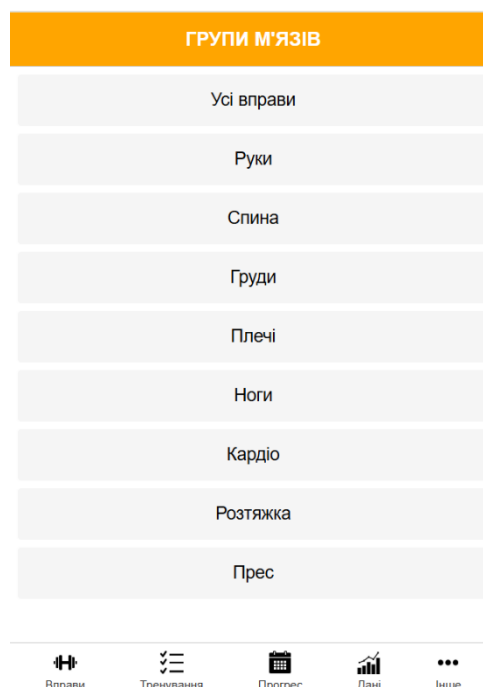


Рисунок 2.4 – Початковий екран мобільного застосунку

Після заповнення користувачем первинних даних (рівень фізичної підготовки, цілі, доступне обладнання тощо), він може перейти до автоматизованого формування тренувального плану, натиснувши на відповідний пункт меню. Після цього інтерфейс застосунку зміниться (див. рисунок 2.4) та відобразить персоналізовану програму тренувань, сформовану на основі введених параметрів.

Цей екран також містить візуалізацію структури тренування, включаючи

перелік вправ, кількість підходів і повторень для кожної з них, а також короткі рекомендації та поради щодо техніки виконання. Додатково можуть відображатися важливі акценти, такі як цільові м'язові групи, тривалість відпочинку між підходами та можливі варіації вправ залежно від рівня підготовки користувача. Такий підхід дозволяє користувачу одразу розпочати тренування згідно з індивідуальними рекомендаціями, мінімізуючи час на самостійну підготовку і планування занять, що, у свою чергу, значно покращує ефективність використання застосунку та сприяє досягненню поставлених фітнес-цілей.

ПЛАН ТРЕНУВАНЬ
Понеділок: Груді та біцепс
Вівторок: Ноги та прес
Середа: Спина та трицепс
Четвер: Кардіо + Розтяжка
П'ятниця: Плечі та руки
Субота: Загальне тренування
Неділя: Відпочинок

 Вправи	 Тренування	 Прогрес	 Дані	 Інше
--	--	---	---	--

Рисунок 2.5 – Персональна програма тренування

Розробка графічного інтерфейсу мобільного застосунку для формування індивідуальних тренувальних планів була проведена у середовищі Android Studio з використанням технології Jetpack Compose та мови програмування Kotlin. Такий підхід дозволив створити сучасний, зручний та адаптивний інтерфейс, який забезпечує комфортну взаємодію користувача із функціоналом застосунку.

2.3 Розробка алгоритму роботи системи

Для розробки програмного застосунку для створення індивідуальних тренувальних планів, необхідно розробити загальний алгоритм, згідно якого працюватиме програмний модуль (див. рисунок 2.7).

Згідно цього алгоритму, першим етапом є завантаження робочого середовища програми, яке надає користувачу доступ до всіх функцій застосунку. Далі користувач має ввести свої дані, такі як зріст, вага, вік, рівень фізичної підготовки та інші параметри, що впливають на тренувальний процес.

Після введення даних, програмний застосунок проводить аналіз введених параметрів. Якщо введені дані неповні або невірні (наприклад, неправильний формат або значення), програма надасть користувачеві відповідне повідомлення про помилки. Якщо дані коректні, програма автоматично переходить до етапу створення тренувального плану.

Програмний застосунок використовує алгоритм для створення індивідуальних тренувальних планів, враховуючи фізичні параметри користувача та його цілі (наприклад, збільшення сили, зниження ваги або поліпшення витривалості). Програма створює набір вправ для кожного дня тренувального циклу, визначаючи кількість підходів, повторів та відпочинку між підходами.

Наступним кроком буде генерування рекомендацій щодо виконання кожної вправи, з урахуванням її техніки, правильного виконання та можливих помилок. Користувач отримує візуалізацію кожної вправи разом із рекомендаціями щодо навантаження .

Для покращення зручності користування, програма надає можливість змінювати тренувальний план. Користувач може редагувати вправи, змінювати їх порядок або відкоригувати навантаження відповідно до свого самопочуття та досягнутого прогресу.

Додатковий алгоритм для модулю тренувального плану:

1. Завантаження інтерфейсу програми.
2. Введення даних користувача.
3. Перевірка введених даних на коректність.

4. Аналіз введених параметрів для визначення індивідуальних тренувальних цілей.
5. Генерація індивідуального тренувального плану.
6. Визначення вправ, підходів, повторів та відпочинку.
7. Генерація рекомендацій по техніці виконання вправ.
8. Надання користувачеві можливості редагувати план тренувань.

Цей алгоритм дозволяє програмному забезпеченню створювати ефективні тренувальні плани, враховуючи індивідуальні потреби користувача. Користувач може взаємодіяти з програмою для коригування своїх тренувальних планів, що дозволяє досягати максимальної ефективності від тренувального процесу.

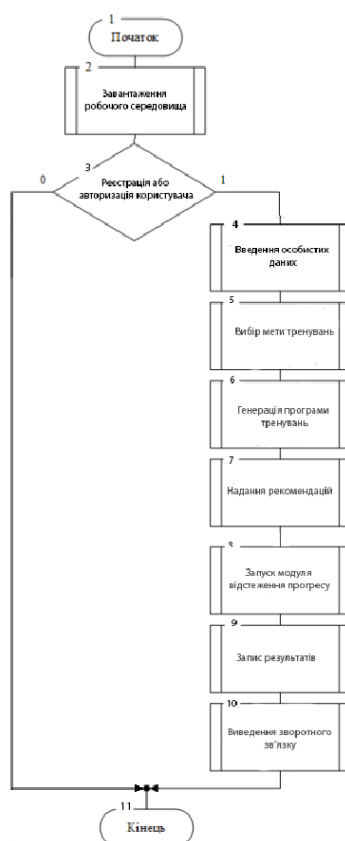


Рисунок 2.7 – Алгоритм роботи програмного застосунку

2.4 Розробка моделі системи

Для кращого розуміння логіки роботи модулів мобільного застосунку для формування індивідуальних тренувальних планів було вирішено розробити модель

роботи системи на основі UML діаграми діяльності (див. рисунок 2.6). Дана діаграма ілюструє основні етапи взаємодії користувача із застосунком та внутрішні процеси обробки даних [8].

На початковому етапі користувач запускає застосунок і переходить до екрану введення особистих даних, де необхідно вказати ключові параметри: вік, вагу, зріст, рівень фізичної підготовки та цілі тренувань (набір м'язової маси, схуднення, підтримка фізичної форми тощо). Якщо всі дані заповнені коректно, система зберігає їх у локальній базі даних та переходить до формування початкового плану тренувань. У разі виявлення помилок у введенні (наприклад, пропущені обов'язкові поля або некоректні значення) застосунок автоматично відображає повідомлення про помилку із зазначенням полів, які потрібно виправити.

Після успішного введення даних система пропонує користувачеві початковий варіант тренувального плану, сформованого на основі аналізу вхідної інформації. Далі користувач має можливість переглянути запропоновані вправи, відредагувати їх (замінити вправу, змінити кількість підходів/повторень або додати нові вправи за власним бажанням). Таким чином, забезпечується високий рівень персоналізації тренувальної програми.

Фіналізований план автоматично генерується програмою з урахуванням індивідуальних особливостей користувача, його поточного рівня фізичної форми та поставлених цілей. У складі плану чітко зазначаються типи вправ, кількість підходів і повторень, оптимальні періоди відпочинку, а також розклад тренувань за днями тижня.

Особливою функцією застосунку є візуалізація вправ за допомогою інтерактивної 3D-моделі. Під час перегляду вправ користувач бачить, які саме м'язові групи залучені під час виконання конкретної вправи. Модель дозволяє змінювати кут огляду, масштабувати зображення та взаємодіяти з окремими елементами (наприклад, вибирати м'язи для детальнішого ознайомлення з їхньою роботою). Це значно покращує розуміння техніки виконання вправ та допомагає уникнути типових помилок під час тренувань [9].

Після кожного завершеного тренування система автоматично фіксує дані про

виконання вправ: час тренування, кількість виконаних підходів, вага снарядів, оцінка суб'єктивної складності тощо. На основі зібраної інформації застосунок аналізує прогрес користувача та формує рекомендації щодо подальших тренувань: пропонує збільшити навантаження, змінити програму або додати нові елементи для більш ефективного досягнення поставлених цілей.

Для зручності користувач має можливість експортувати свій тренувальний план у форматі PDF. Це дозволяє зберігати плани для подальшого використання, друкувати їх або ділитися з тренером чи друзями. Експортований документ містить усі необхідні деталі: перелік вправ, кількість підходів і повторень, план тренувань за днями та короткі поради з техніки виконання.

Завдяки побудові чіткої структури роботи програми та використанню UML-діаграми діяльності забезпечується прозоре розуміння логіки роботи застосунку на всіх етапах взаємодії з користувачем.

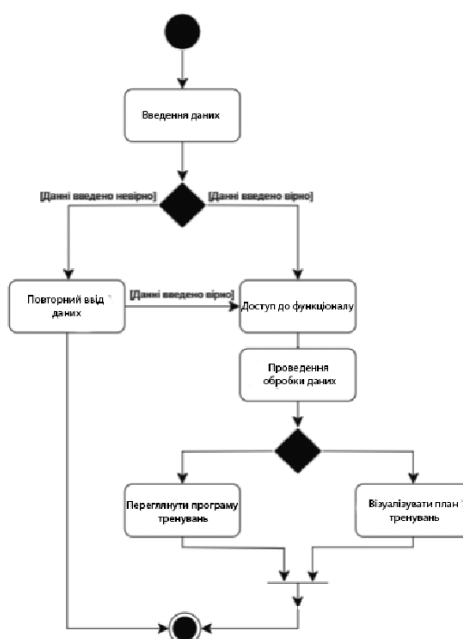


Рисунок 2.6 – Модель системи у вигляді діаграми діяльності програмних засобів діяльності

2.5 Розробка методу формування індивідуального плану тренувань

Для підвищення ефективності занять у тренажерному залі в мобільному застосунку реалізовано метод формування індивідуального плану тренувань, що

враховує персональні характеристики користувача та його цілі. Основною метою методу є генерація адаптивної програми тренувань, яка відповідає фізичному стану, потребам та рівню підготовки кожного окремого користувача.

На першому етапі відбувається збір вхідних даних, які користувач вводить через інтерфейс: вік, стать, рівень фізичної підготовки, бажані цілі (наприклад, зниження ваги, набір м'язової маси, підвищення витривалості), а також доступні дні для тренувань. Ці параметри зберігаються у внутрішній базі даних застосунку.

Далі за допомогою алгоритму логічного відбору здійснюється вибір відповідних типів тренувань (кардіо, силові, змішані), визначається інтенсивність навантаження, тривалість кожного заняття та кількість днів відпочинку. Метод ґрунтується на базових принципах спортивної періодизації, адаптованих до мобільного використання.

Після обробки даних формується персоналізований тижневий план тренувань, який користувач може переглядати у додатку. Усі вправи супроводжуються описами, кількістю повторів, підходів і рекомендаціями щодо техніки виконання. При необхідності користувач може змінювати або оновлювати вхідні параметри, що автоматично оновлює сформований план відповідно до нових умов.

Такий підхід дозволяє забезпечити високий рівень персоналізації та ефективності тренувального процесу, зменшуючи ризик перевантаження чи неефективних занять. Метод легко масштабується та може бути доповнений новими параметрами у майбутніх версіях застосунку.

2.6 Висновки

У другому розділі був проведений аналіз вхідних і вихідних даних програмного застосунку для створення індивідуальних тренувальних планів. Описано процес розробки графічного інтерфейсу, алгоритм роботи програмних засобів та алгоритм визначення тренувальних параметрів. Також була створена модель роботи програмного продукту за допомогою UML діаграм, що демонструють взаємодію компонентів системи.

3. РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ ДЛЯ РЕАЛІЗАЦІЇ СТВОРЕННЯ ІНДИВІДУАЛЬНИХ ТРЕНУВАЛЬНИХ ПЛАНІВ З ВИКОРИСТАННЯМ ТРИВИМІРНОГО МОДЕЛЮВАННЯ.

3.1 Варіантний аналіз і обґрунтування вибору засобів для реалізації програмного забезпечення

Розглянемо найпоширеніші мови програмування та інструменти для розробки мобільних додатків.

Kotlin — мова програмування високого рівня, яка використовується для створення Android-застосунків. Розроблена компанією JetBrains у 2011 році. Kotlin підтримує об'єктно-орієнтоване програмування, функціональні можливості та інтероперабельність з Java. Завдяки своїй простоті та інтеграції з Android Studio, Kotlin є ідеальним для розробки мобільних додатків для тренажерних залів [10].

Java — об'єктно-орієнтована мова програмування, що широко використовується для розробки Android-додатків. Розроблена компанією Sun Microsystems в 1991 році, вона надає потужні інструменти для створення складних додатків. Java має велику екосистему бібліотек і фреймворків, що дозволяє легко реалізувати функціонал для мобільних додатків, зокрема для тренувальних програм [11].

Flutter (Dart) — фреймворк для крос-платформової розробки додатків, який використовує мову Dart. Розроблений компанією Google, Flutter дозволяє створювати мобільні додатки як для Android, так і для iOS. Цей фреймворк може бути корисним для створення інтерактивних тренувальних додатків з візуалізацією, оскільки має вбудовані можливості для розробки анімацій та інтерактивних елементів [12].

Android Studio — інтегроване середовище розробки, яке підтримує Kotlin та Java, призначене для створення Android-додатків. Це стандартне середовище для розробки додатків під ОС Android, яке включає в себе всі необхідні інструменти для створення, тестування та оптимізації додатків для тренажерних залів [13].

Room Database — бібліотека для Android, яка забезпечує простий доступ до

бази даних SQLite. Вона дозволяє зберігати інформацію про тренування, плани тренувань та прогрес користувача. Завдяки Room можна ефективно зберігати та маніпулювати даними про антропометричні параметри та тренувальні плани [14].

Для реалізації мобільного додатку для тренажерного залу було обрано Kotlin через його інтеграцію з Android Studio, зручність та високу продуктивність, що є важливими для реалізації функціоналу, який вимагає швидкої обробки даних та інтерактивності [15].

3.2 Реалізація модулів обробки даних користувача у середовищі Android

В ході розробки програмного продукту було створено програмний код на мові Kotlin, використовуючи бібліотеки Android для взаємодії з сенсорами та алгоритмами обчислення антропометричних параметрів. Ключовими файлами вихідного коду є «MeasurementWorker.kt» та «MeasurementActivity.kt». Перший файл визначає клас MeasurementWorker, призначений для управління потоком обробки даних сенсорів і вимірювання параметрів. Відповідно, файл «MeasurementActivity.kt» реалізує головну активність, в якій запускаються необхідні модулі і обробляються вхідні дані.

Одним із ключових аспектів розробленої мобільної системи для формування рекомендацій щодо занять у тренажерному залі є забезпечення точного та індивідуалізованого розрахунку інтенсивності тренувань. Адекватна інтенсивність фізичного навантаження має вирішальне значення для досягнення бажаних результатів, запобігання перевантаженням і забезпечення безпеки користувача під час виконання вправ.

З цією метою у програмі реалізовано спеціальну функцію, яка виконує розрахунок на основі сукупності індивідуальних антропометричних і фізіологічних параметрів. Враховується вага та зріст користувача, що дозволяє оцінити його загальний фізичний стан, зокрема індекс маси тіла. Вік також відіграє важливу роль, оскільки фізіологічні особливості організму змінюються з часом і потребують адаптації навантаження. Стать користувача впливає на розподіл сили, витривалості та швидкість відновлення після фізичних зусиль, що також відображається в

алгоритмах розрахунку. Крім того, система враховує рівень фізичної підготовленості, який визначається на основі початкового опитування або аналізу попередніх результатів тренувань.

Завдяки комплексному підходу до обробки індивідуальних даних користувача, система здатна формувати високоточні та максимально персоналізовані програми тренувань, які повністю відповідають фізіологічним можливостям конкретної особи. У процесі генерації таких програм враховується широкий спектр параметрів, включаючи поточний стан здоров'я, рівень підготовки, попередній тренувальний досвід, цілі (наприклад, зниження ваги, набір м'язової маси або підвищення витривалості), а також динаміка змін, зафіксована в результаті попередніх занять.

Такий підхід дозволяє не лише значно підвищити загальну ефективність тренувального процесу, адаптуючи навантаження до оптимального рівня для кожного користувача, але й забезпечує високий рівень безпеки, мінімізуючи ризик травм або перенапруження організму. Крім того, регулярне оновлення рекомендацій на основі нових даних сприяє підтриманню актуальності тренувального плану, що особливо важливо при зміні фізичної форми або цілей користувача.

Окремо варто підкреслити, що індивідуалізація тренувань позитивно впливає на психологічний стан користувача, адже він відчуває, що система враховує його унікальні потреби. Це значно підвищує рівень залученості до тренувального процесу та сприяє довготривалому збереженню мотивації, що є одним із ключових факторів у досягненні стабільних і стійких результатів. Таким чином, застосунок не просто пропонує вправи, а виконує функцію інтелектуального тренера, який веде користувача до цілі з урахуванням усіх особливостей його організму та способу життя.

Функція приймає введені користувачем параметри, такі як вік, рівень фізичної підготовки, цілі тренувань, наявність медичних протипоказань та інші індивідуальні характеристики, і на їх основі здійснює комплексний розрахунок оптимальної інтенсивності тренування. Цей розрахунок враховує не тільки

фізіологічні особливості користувача, але й рекомендації з галузі спортивної науки, щоб забезпечити максимально ефективний та безпечний режим навантажень.

На основі отриманої інтенсивності тренувань система формує персоналізований набір рекомендацій, що відповідає потребам і можливостям конкретного користувача. Ці рекомендації включають детальний опис рекомендованих типів вправ — зокрема, кардіонавантаження для покращення роботи серцево-судинної системи, силові тренування для розвитку м'язової маси та витривалості, а також вправи на розтяжку і гнучкість.

Крім того, система визначає оптимальну тривалість кожного тренувального блоку, враховуючи рівень підготовки користувача та поставлені цілі. Важливою частиною рекомендацій є і планування інтервалів відпочинку між вправами та підходами, що сприяє відновленню сил, зниженню ризику травм і покращенню загальної продуктивності тренування.

Таким чином, запропонований підхід дозволяє створити індивідуальну програму тренувань, що максимально адаптована до фізичних можливостей і цілей користувача, сприяє підвищенню ефективності занять у тренажерному залі та підтримує мотивацію до регулярних тренувань.

Функція формує рекомендації, враховуючи рівень фізичної підготовленості, який визначається за допомогою опитувальника або базових даних користувача (наприклад, початковий рівень підготовки чи досвід у тренуваннях):

```

1 fun generateTrainingRecommendations(intensity: Double, experienceLevel: String): String {
2     return when {
3         intensity > 1.0 -> {
4             when (experienceLevel) {
5                 "Beginner" -> "Легкі кардіонавантаження 30 хвилин, 3 рази на тиждень."
6                 "Intermediate" -> "Змішане кардіо і силові тренування, 40 хвилин, 4 рази на тиждень."
7                 "Advanced" -> "Інтенсивні тренування з високим навантаженням, 60 хвилин, 5 разів на тиждень."
8                 else -> "Загальні рекомендації: тренування 3 рази на тиждень."
9             }
10        }
11        else -> {
12            "Легкі кардіонавантаження або розтягування, 2 рази на тиждень."
13        }
14    }
15 }

```

Рисунок 3.1 – Фрагмент коду функції «generateTrainingRecommendations»

Ця функція допомагає генерувати індивідуальні рекомендації на основі фізичних характеристик і рівня досвіду користувача, що дозволяє створити більш ефективну та безпечну програму тренувань.

Ще одним важливим аспектом є обчислення калорій, які спалюються під час тренування. Для цього була розроблена функція, яка враховує час тренування, тип вправ і фізичні дані користувача. Вона дозволяє користувачам відстежувати кількість витрачених калорій і коригувати свій раціон відповідно до результатів.

Ця функція дозволяє точно розраховувати кількість калорій, що витрачаються під час тренування, і дає можливість користувачеві оптимізувати свій режим харчування.

3.3 Розробка головного вікна програми

У цьому пункті описується процес створення головного вікна мобільного додатку, який слугує стартовою точкою взаємодії користувача з системою формування персоналізованих рекомендацій щодо тренувань. Це вікно виконує функцію навігаційного центру, що дозволяє швидко перейти до основних функцій додатку.

Головне вікно містить інтерактивні кнопки, за допомогою яких користувач може вибрати свій рівень фізичної підготовки. Інтерфейс реалізовано таким чином, щоб процес вибору був максимально простим, зрозумілим і не потребував додаткових пояснень. Після натискання на відповідну кнопку система миттєво реагує, викликаючи функцію, що відповідає за формування тренувальних рекомендацій на основі введеного користувачем рівня підготовки.

Ця функція проводить аналіз обраного параметра і генерує набір вправ, адаптованих до індивідуальних можливостей користувача. Рекомендації включають детальну інформацію щодо типів фізичних навантажень, їх тривалості, інтенсивності, кількості підходів, а також інтервалів відпочинку між вправами. Такий підхід дозволяє створити ефективний та збалансований план тренувань, який відповідає реальному рівню фізичної форми користувача та сприяє досягненню поставлених цілей.

Завдяки інтуїтивно зрозумілому дизайну головного вікна та зручній структурі навігації, користувач може швидко налаштувати початкові параметри, що забезпечує комфортне користування додатком та підвищує ефективність подальшої роботи з персональними тренувальними програмами.

Головне вікно застосунку (див. рисунок 3.2) використовує клас `WorkoutRecommendation`, який відповідає за автоматичне генерування списку вправ на основі індивідуальних параметрів користувача. Для кожної вправи визначається оптимальна кількість підходів, повторень та рекомендований час відпочинку між ними. Інтерфейс головного вікна створено з використанням стандартних елементів Android, таких як кнопки (`Button`) для навігації та виконання дій, а також текстові поля (`TextView`) для відображення сформованих рекомендацій. Окрім використання базових елементів інтерфейсу, в мобільному застосунку реалізовано адаптивне компонування, яке забезпечує коректне та зручне відображення вмісту на різних типах пристроїв, включаючи як смартфони з меншими екранами, так і планшети з більшими діагоналями. Такий підхід дозволяє гарантувати, що незалежно від розміру або орієнтації екрана, всі елементи інтерфейсу залишаються доступними, добре масштабованими та логічно впорядкованими.

Адаптивне компонування реалізовано з урахуванням принципів *responsive design*, що дозволяє динамічно змінювати розміщення елементів, їх розміри, відступи та співвідношення залежно від роздільної здатності екрана. Це особливо важливо для забезпечення комфортного користування функціоналом застосунку в умовах різноманіття сучасних Android-пристроїв.

Інтерфейс автоматично підлаштовується під умови експлуатації: наприклад, на компактних екранах інформація розміщується у вертикальному форматі з можливістю прокручування, тоді як на великих екранах планшетів дані можуть бути представлені у вигляді багатопанельного макету з одночасним відображенням кількох секцій. Такий підхід сприяє кращій організації простору та більш ефективному використанню доступної площі екрану.

Реалізація адаптивного дизайну не лише підвищує загальну зручність роботи

із застосунком, а й робить інтерфейс більш інтуїтивно зрозумілим і дружнім до користувача, що позитивно впливає на користувацький досвід. Завдяки цьому застосунок однаково ефективно працює на будь-якому пристрої, забезпечуючи стабільність, гнучкість і якість взаємодії з користувачем [18].

```

1  class MainActivity : AppCompatActivity() {
2
3      private lateinit var workoutRecommendation: WorkoutRecommendation
4
5      override fun onCreate(savedInstanceState: Bundle?) {
6          super.onCreate(savedInstanceState)
7          setContentView(R.layout.activity_main)
8
9          workoutRecommendation = WorkoutRecommendation()
10
11
12          val beginnerButton = findViewById<Button>(R.id.button_beginner)
13          beginnerButton.setOnClickListener {
14              workoutRecommendation.recommendBeginnerWorkout()
15          }
16
17
18          val intermediateButton = findViewById<Button>(R.id.button_intermediate)
19          intermediateButton.setOnClickListener {
20              workoutRecommendation.recommendIntermediateWorkout()
21          }
22
23          |
24          val advancedButton = findViewById<Button>(R.id.button_advanced)
25          advancedButton.setOnClickListener {
26              workoutRecommendation.recommendAdvancedWorkout()
27          }
28      }
29  }

```

Рисунок 3.2 – Код для головного вікна програми

Було розроблено «Додати персоналізовані рекомендації» (див. рисунок 3.3).

У цьому фрагменті коду реалізовано спеціальне вікно, яке призначене для введення користувачем своїх персональних параметрів, що є критично важливими для формування індивідуальної програми тренувань. Зокрема, користувач має можливість вказати свій рівень фізичної підготовки — від початкового до просунутого, поставити конкретні цілі тренувань, такі як схуднення, набір м'язової маси або загальне покращення фізичної форми, а також вказати свій вік і стать. Ці

дані збираються для того, щоб система могла створити максимально точні, адаптовані до особистих особливостей користувача рекомендації щодо тренувального процесу.

Для збору та введення цих даних у застосунку використано стандартні та перевірені часом елементи інтерфейсу Android. Зокрема, застосовуються текстові поля (EditText), які дозволяють користувачу вводити числові або текстові значення, наприклад, свій вік чи інші параметри. Для вибору зі списку заздалегідь визначених варіантів, таких як рівень фізичної підготовки або типи цілей, застосовуються випадаючі списки (Spinner), які забезпечують швидкий та зручний вибір. Крім того, для вибору статі або інших двійкових параметрів використані перемикачі (RadioButton), що гарантують інтуїтивно зрозумілу взаємодію та зручність у використанні.

Завдяки такому підходу інтерфейс користувача є простим, зрозумілим і функціональним, що забезпечує комфортну взаємодію із застосунком. Усі елементи управління логічно розміщені, мають інтуїтивно зрозуміле призначення, що значно знижує ймовірність помилок під час введення персональних даних.

Застосування знайомих користувачу елементів — таких як текстові поля, перемикачі та списки — сприяє швидкому та точному заповненню форми, що, у свою чергу, дозволяє системі формувати більш точні та індивідуалізовані тренувальні рекомендації.

Це безпосередньо впливає на якість сервісу та загальний рівень задоволеності користувача. Інтерфейс сприймається як зручний та продуманий, що створює позитивний досвід використання та стимулює регулярну взаємодію із застосунком.

```

1  class AddCustomWorkoutActivity : AppCompatActivity() {
2
3      private lateinit var workoutRecommendation: WorkoutRecommendation
4
5      override fun onCreate(savedInstanceState: Bundle?) {
6          super.onCreate(savedInstanceState)
7          setContentView(R.layout.activity_add_custom_workout)
8
9          workoutRecommendation = WorkoutRecommendation()
10
11         val levelInput = findViewById<EditText>(R.id.editText_level)
12
13         val goalInput = findViewById<EditText>(R.id.editText_goal)
14
15
16         val generateButton = findViewById<Button>(R.id.button_generate)
17         generateButton.setOnClickListener {
18             val level = levelInput.text.toString()
19             val goal = goalInput.text.toString()
20
21
22             workoutRecommendation.recommendCustomWorkout(level, goal)
23         }
24     }
25 }

```

Рисунок 3.3 – Код для вікна «Додати персоналізовані рекомендації»

3.4 Висновки

У третьому розділі було обґрунтовано вибір технологій та інструментів, що використовувались для розробки програмних модулів. Виконавши аналіз програмного продукту та його задач, було обрано мову програмування Kotlin. Для зручності розробки графічного інтерфейсу користувача було обрано Android Studio з підтримкою Jetpack Compose. Для роботи з базою даних було використано бібліотеку Room, а для асинхронного оброблення даних — Kotlin Coroutines та Kotlin Flows. Також для збору антропометричних параметрів було розроблено алгоритми вимірювання відстаней між точками на тривимірних моделях. У розділі також було описано розробку основних програмних модулів для визначення позицій точок антропометрії та вимірювання антропометричних параметрів на моделях користувачів мобільної системи [19].

4 ТЕСТУВАННЯ ПРОГРАМИ

4.1 Тестування системи

У залежності від рівня доступу розробника тестів до вихідного коду мобільного застосунку розрізняють тестування «білого ящика» та тестування «чорного ящика».

Тестування «білого ящика»

Цей тип тестування застосовується, коли розробник має повний доступ до вихідного коду програми. У контексті мобільного застосунку для тренажерного залу це дозволяє створювати модульні тести для окремих компонентів, таких як:

- функції генерації індивідуальних програм тренувань;
- обробка введених користувачем даних;
- взаємодія з локальною базою даних Room;
- логіка обчислення прогресу користувача.

Таке тестування дозволяє перевірити правильність реалізації кожного модуля, незалежно від решти системи. Застосовуються такі методи, як покриття коду або мутаційне тестування, що дозволяють виявити слабкі місця в логіці застосунку.

Тестування «чорного ящика»

У цьому випадку тестувальник не має доступу до внутрішнього коду і взаємодіє із застосунком виключно через користувацький інтерфейс — тобто так, як це робив би звичайний користувач.

Тестування «чорного ящика» було використано для перевірки:

- коректної поведінки при реєстрації користувача,
- відповідності відображуваних рекомендацій заданим параметрам (мета занять, досвід, стать, вага),
- валідності форм введення,
- реакції на неправильні або неповні дані.

Під час цього типу тестування застосовувалися тестові сценарії на основі вимог до функціональності, не враховуючи внутрішню реалізацію коду. Це

дозволяє оцінити застосунок з точки зору кінцевого користувача.

Фактично, у таких тестах ми знаємо очікуваний результат на вхідний набір даних (наприклад, введення користувача з певною метою та характеристиками) і перевіряємо, чи відповідає вихідний результат (генерована програма) специфікації.

Для тестування додатку було обрано стратегію «чорного ящика».

Перевірка розташування блоків для головної сторінки на рисунку 4.1

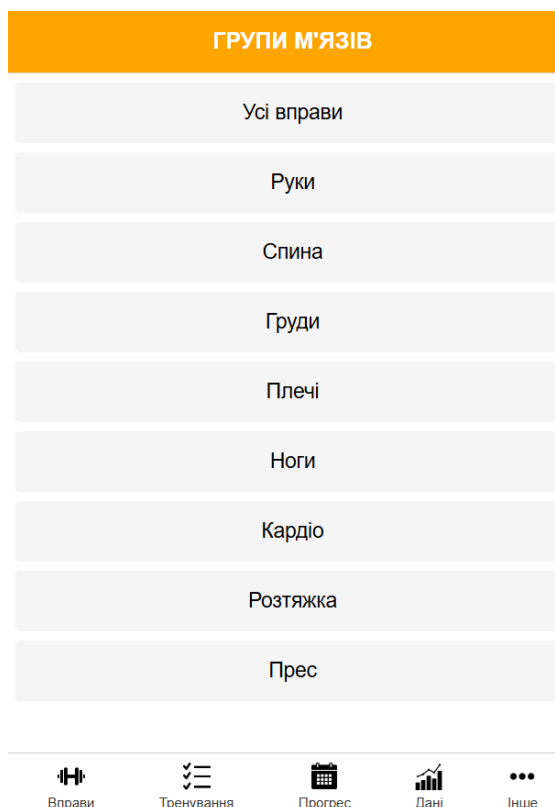


Рисунок 4.1 – Головна сторінка

Модуль реєстрації та авторизації є критично важливою складовою мобільного застосунку для формування індивідуального плану тренувань, оскільки забезпечує захищений доступ користувачів до персоналізованих функцій, збереження їхніх даних і відстеження прогресу.

На рисунку 4.2 представлено результати тестування модуля реєстрації та авторизації. Перевірка проводилась згідно з наступними критеріями:

1. Правильність введення даних при реєстрації:

- Введення дійсної електронної пошти;
- Введення пароля, що відповідає вимогам безпеки (мінімальна довжина, наявність великих/малих літер, цифр);
- Повторне введення пароля для підтвердження.

2. Реакція на помилкові або неповні дані:

- Виведення повідомлення про помилку при спробі зареєструватися без заповнення обов'язкових полів;
- Попередження при невідповідності паролів;
- Повідомлення про вже існуючий акаунт при повторній реєстрації.

3. Функціональність авторизації:

- Успішний вхід з коректними обліковими даними;
- Повідомлення про невірний пароль або email;
- Відображення головного екрану після успішної авторизації.

4. Безпека:

- Дані передаються через захищене з'єднання (HTTPS);
- Зберігання токенів авторизації в безпечному сховищі (SharedPreferences з режимом PRIVATE або EncryptedSharedPreferences).

5. UI/UX тестування:

- Перевірено розміщення елементів інтерфейсу на екранах різних розмірів;
- Відповідність стилю оформлення (основний колір – білий, допоміжні – чорний та оранжевий);
- Наявність кнопок «Реєстрація», з очікуваною поведінкою.

Таким чином, модуль пройшов усі етапи функціонального та інтеграційного тестування, демонструючи стабільну роботу та відповідність вимогам до зручності і безпеки. Результати тестування засвідчують готовність модуля до використання в основному потоці застосунку.

Реєстрація

Ім'я користувача

Електронна пошта

Пароль

Зареєструватися

Рисунок 4.2 – Екран входу/реєстрації

На рисунку 4.3 зображено приклад сформованого індивідуального тренувального плану для першого дня тижня, створеного модулем рекомендацій відповідно до заданих параметрів користувача. Цей елемент є одним із ключових у структурі застосунку, оскільки він забезпечує персоналізований підхід до організації тренувального процесу. Формування плану відбувається з урахуванням основної мети користувача, такої як набір м'язової маси, зниження ваги або підтримка фізичної форми. Окрім цього, система бере до уваги рівень фізичної підготовки, стать, вік, а також наявне обладнання, доступне для тренувань.

Завдяки комплексному аналізу вхідних даних, модуль рекомендацій генерує оптимізований план занять, який відповідає індивідуальним особливостям користувача. Такий підхід сприяє підвищенню ефективності тренувань, зменшує ризик перенавантаження або травм, а також підтримує високу мотивацію до регулярних фізичних занять. У результаті користувач отримує чіткий, структурований та адаптивний тренувальний план, який відповідає його поточному стану та цілям.

Основні цілі тестування:

1. Перевірка коректності логіки формування плану:

- Чи відповідає набір вправ обраній цілі (наприклад, переважання базових вправ для набору маси);
- Чи враховується попередній досвід користувача (для початківців — менше складних вправ і нижче навантаження);
- Чи генеруються плани з розподілом за днями (спліт-система).

2. Адаптивність рекомендацій:

- Система правильно змінює план при зміні параметрів (наприклад, обравши домашні тренування без обладнання);
- Вправи оновлюються у відповідності до введених змін без необхідності перезапуску застосунку.

3. UI-тестування:

- Коректне відображення вправ, підходів та повторень (наприклад, "4 x 15");
- Тренування згруповані за днями — зручна навігація;
Відповідність дизайну: білий фон — основний, оранжевий — для акцентів (активні вкладки), чорний — для тексту.

4. Юзабіліті-тестування:

- Зручність взаємодії з блоком (швидкість генерації, доступність функції повторного формування);
- Користувач розуміє, що і як виконувати — чітка структура (назва вправи + кількість підходів і повторень).

5. Функціональна стійкість:

- Відсутність збоїв при генерації плану;
- План не дублює одні й ті самі вправи при кожному запуску, якщо це не передбачено логікою.

Проведене тестування засвідчило працездатність модуля рекомендацій. Він стабільно генерує індивідуальні тренування згідно заданих критеріїв користувача, забезпечуючи зручне відображення та відповідність очікуванням. Модуль повністю готовий до використання у виробничому середовищі.

ПЛАН ТРЕНУВАНЬ
Понеділок: Груді та біцепс
Вівторок: Ноги та прес
Середа: Спина та трицепс
Четвер: Кардіо + Розтяжка
П'ятниця: Плечі та руки
Субота: Загальне тренування
Неділя: Відпочинок

				
Вправи	Тренування	Прогрес	Дані	Інше

Рисунок 4.3 – Рекомендації тренувань

Перевірка інтерфейсу прогресу користувача (див. рисунок 4.4).

У рамках реалізації мобільного застосунку, основною метою якого є формування персоналізованих тренувальних планів, важливим етапом стало тестування інтерфейсу, що відповідає за відображення прогресу користувача. Цей інтерфейс відіграє ключову роль у взаємодії користувача із застосунком, оскільки дозволяє візуально оцінити досягнення, результати виконаних тренувань та загальну динаміку покращення фізичної форми.

Під час перевірки було звернено увагу на правильність відображення даних про виконані вправи, кількість підходів і повторень, дати занять, а також загальний рівень відповідності інформації тому, що зберігається в базі даних. Інтерфейс надає можливість користувачу переглядати історію своїх тренувань, а також порівнювати результати за різні періоди часу, що сприяє підвищенню мотивації. Крім того, була перевірена інтерактивність елементів інтерфейсу: при натисканні на певну дату відкривається список вправ, які були виконані цього дня, а також індикатори завершеності тренування.

Окрему увагу приділено графічному оформленню модуля. Основним кольором інтерфейсу виступає білий, що забезпечує відчуття чистоти, простоти та мінімалізму. Чорний використовується для текстових елементів та контурів, а яскравий оранжевий додає акценти в індикаторах прогресу, кнопках завершення тренування та інших ключових елементах. Такий підхід дозволяє ефективно спрямовувати увагу користувача до найбільш важливої інформації.

Завдяки тісній інтеграції цього інтерфейсу з іншими модулями додатку — такими як модуль створення плану, реєстрації тренувань і персональних налаштувань — було досягнуто високої узгодженості у відображенні даних. Після завершення кожного тренування інформація про нього автоматично оновлюється в інтерфейсі прогресу, що забезпечує актуальність і достовірність аналітики.

Таким чином, проведене тестування підтвердило ефективність, зручність та функціональність інтерфейсу відображення прогресу. Його реалізація повністю відповідає меті застосунку — допомогти користувачу бачити результати своєї роботи та зберігати мотивацію до систематичних занять.

Прогрес

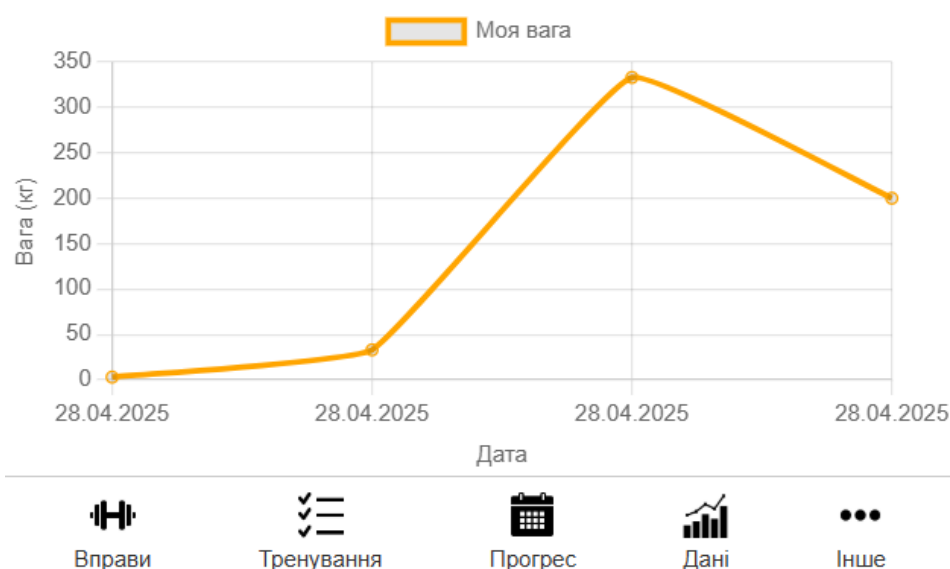


Рисунок 4.4 – Інтерфейс прогресу користувача

Перевірка модуля налаштувань користувача (див. рисунок 4.5)

Одним із основних компонентів мобільного застосунку є модуль налаштувань користувача, який надає можливість адаптувати функціональність програми відповідно до індивідуальних потреб та цілей користувача. У процесі розробки було реалізовано набір функцій, що дозволяють користувачам змінювати персональні параметри, включаючи рівень фізичної підготовки, тип фітнес-цілі (набір м'язової маси, зниження ваги, підтримка форми), бажану інтенсивність тренувань, доступні дні для занять, а також вибір м'язових груп, на які користувач хоче зробити акцент.

Модуль налаштувань дає користувачеві гнучкість у формуванні свого індивідуального тренувального плану, що забезпечує високий рівень персоналізації та дозволяє адаптувати програму до конкретних цілей та фізичних можливостей користувача. Важливим аспектом є можливість зміни параметрів на будь-якому етапі використання програми, що дозволяє коригувати тренувальний процес залежно від прогресу або зміни цілей. Це забезпечує інтерактивний підхід до тренувань і дозволяє підтримувати мотивацію користувача.

У процесі тестування модуля налаштувань особлива увага приділялася зручності інтерфейсу та коректності збереження і обробки введених даних. Перевірялися функціональні можливості системи з точки зору збереження введених параметрів у тимчасовій пам'яті (під час активного сеансу) та постійній пам'яті (за допомогою таких механізмів, як Room або SharedPreferences). Це забезпечує збереження налаштувань навіть після перезапуску додатку або відключення пристрою, гарантуючи стабільність і збереження даних користувача.

Важливим аспектом тестування була перевірка функціональності зміни параметрів та їх впливу на подальшу генерацію тренувальних планів. Наприклад, якщо користувач змінює фітнес-мету з «підтримки форми» на «набір м'язової маси», наступного разу при формуванні тренувального плану система повинна врахувати нові параметри. Це може включати зміну підходів до навантаження, кількості повторень, тривалості відпочинку між підходами тощо, щоб план тренувань відповідав новим цілям користувача. Перевірка цього аспекту дозволила

переконатися, що система коректно відображає зміни і адаптує тренування в реальному часі, без необхідності повторного введення даних.

Окрім персональних налаштувань, також були протестовані додаткові функції, що підвищують зручність використання програми. Це включає вибір мови інтерфейсу, можливість перемикання між темним і світлим режимами оформлення, налаштування сповіщень про тренування, а також інтеграцію з календарем пристрою для автоматичного планування тренувань та нагадувань. Ці функції дозволяють зробити використання додатку ще більш персоналізованим та комфортним для кожного користувача, покращуючи загальний досвід взаємодії з програмою.

Інтерфейс модуля налаштувань реалізований в єдиному стилі з загальним дизайном застосунку. Основним кольором є білий, що символізує чистоту, нейтральність та створює візуально приємне середовище для введення та редагування даних. Допоміжні елементи, такі як текстові поля та кнопки для навігації, виконані в чорному кольорі, що дозволяє чітко структурувати інтерфейс і забезпечити зручну навігацію. Важливі кнопки підтвердження або зміни налаштувань акцентовані оранжевим кольором, що фокусує увагу користувача на основних діях, роблячи інтерфейс інтуїтивно зрозумілим і зручним у використанні.

В процесі тестування було виявлено, що модуль налаштувань ефективно зберігає введені дані в різних типах пам'яті, забезпечуючи коректну синхронізацію між усіма компонентами програми. Модуль взаємодіє з іншими частинами системи через ViewModel, що дозволяє миттєво оновлювати дані в усіх частинах програми без необхідності повторного введення параметрів користувачем. Завдяки цьому забезпечується висока ефективність та зручність у використанні програми.

Таким чином, тестування модуля налаштувань користувача показало його високу функціональність, зручність у використанні та повну інтеграцію в загальну логіку персоналізованого підходу до формування тренувальних планів. Модуль забезпечує коректну обробку введених даних, адаптацію плану тренувань під зміни в налаштуваннях, а також взаємодію з іншими частинами додатку, що робить користувацький досвід максимально комфортним та ефективним.



ДАНІ	
Введіть свої дані	
Зріст (см):	190
Вага (кг):	90
Відсоток жиру:	19
Шия (см):	58
Руки (см):	42
Ноги (см):	68
Груди (см):	120
Зберегти	

Рисунок 4.5 – Налаштування користувача

Результати перевірки підтвердили правильну роботу всіх модулів мобільного застосунку, що вказує на високий рівень стабільності та ефективності розробленої системи. Всі основні функціональні компоненти програми працюють як належить: дані користувача обробляються коректно, що забезпечує точність і консистентність у відображенні індивідуальних налаштувань та рекомендацій. Вхідні дані, незалежно від їхнього формату чи типу, правильно валідуються та зберігаються, що гарантує цілісність інформації навіть при переході між різними сесіями використання додатку.

Що стосується інтерфейсу користувача, він продемонстрував високу зручність у використанні. Всі елементи інтерфейсу мають чітку структуру, не перевантажують екран зайвою інформацією і дозволяють користувачеві без труднощів здійснювати необхідні дії. Особлива увага приділялася забезпеченню оптимального користувацького досвіду: використання контрастних кольорів для важливих елементів, зручна навігація, адаптивність інтерфейсу для різних розмірів екранів і пристроїв — усе це сприяє підвищенню рівня задоволення від взаємодії з додатком.

Особливо важливим результатом є коректне формування рекомендованих тренувань згідно з персональними налаштуваннями користувача. Програма здатна адаптувати тренувальні плани до індивідуальних фізичних можливостей та цілей користувача, забезпечуючи оптимальні навантаження та прогрес. Наприклад, у випадку зміни мети (з підтримки форми на набір м'язової маси) програма коректно підлаштовує всі параметри тренувань: кількість підходів, повторень, тривалість відпочинку, що значно підвищує ефективність тренувального процесу.

Перевірка також показала, що система здатна адекватно реагувати на різноманітні сценарії користування, зокрема на ситуації з некоректним або неповним введенням даних. Це важлива складова надійності та стабільності програми, оскільки вона гарантує, що користувачі не зіштовхнуться з помилками чи збоями при невірно введених даних.

Завдяки правильно спроектованій архітектурі застосунку, включаючи використання таких технологій, як Room для збереження даних і ViewModel для управління станом, додаток продовжує функціонувати стабільно навіть після перезапуску, що додає додатковий рівень надійності. Всі зміни, що вносяться в налаштування користувача, миттєво оновлюються в системі, що забезпечує безперервність і актуальність даних.

У результаті тестування було встановлено, що мобільний застосунок відповідає усім заявленим вимогам, а також додатково показує високу ступінь адаптивності, забезпечуючи персоналізований підхід до кожного користувача. Ці результати підтверджують успішне досягнення основної мети роботи — створення функціонального, зручного та адаптивного інструменту, який не лише сприяє підтримці фізичної активності користувачів, але й допомагає досягати їхніх фітнес-цілей ефективно і з мінімальними зусиллями.

Тому можна стверджувати, що розроблений мобільний застосунок є надійним, ефективним та зручним інструментом для користувачів, які прагнуть підтримувати свою фізичну форму або досягати нових фітнес-цілей.

4.2 Розробка інструкції користувача

Інструкція користувача передбачає визначення технічних вимог для запуску мобільного застосунку на пристрої користувача. Деталі щодо мінімальної та рекомендованої конфігурації наведено в таблиці 4.1.

Таблиця 4.1 – Вимоги до апаратного забезпечення

Вимоги до конфігурування	Рекомендовано	Мінімально
Процесор	Qualcomm Snapdragon 860 або вище	Qualcomm Snapdragon 450 або аналогічний
Оперативна пам'ять	6 ГБ RAM	2 ГБ RAM
Вільний простір на пристрої	500 МБ	150 МБ
Дисплей	Full HD (1920x1080), 6.0" або більше	HD (1280x720), 5.0" або більше
Операційна система	Android 11 або новіші	Android 8
Інтернет-з'єднання	Wi-Fi або 4G для повноцінної роботи	Тимчасове з'єднання для авторизації

Після встановлення та запуску мобільного застосунку користувача зустрічає вітальний екран із логотипом програми, назвою та короткою інструкцією щодо подальших дій. Екран оформлений у сучасному стилі, що одразу створює позитивне перше враження та сприяє комфортному початку роботи з додатком. Після короткої затримки або за натисканням кнопки «Продовжити» відбувається автоматичне завантаження головного інтерфейсу програми.

Головний інтерфейс містить зручне нижнє меню навігації, яке забезпечує швидкий доступ до основних розділів застосунку, а також панель для введення особистих даних. На цьому етапі користувачу пропонується створити або оновити

власний профіль. Інтерфейс введення даних інтуїтивно зрозумілий і складається з послідовних полів для заповнення таких параметрів, як: вік, стать, зріст, вага, рівень фізичної підготовки (початковий, середній, просунутий) та основна мета тренувань (набір м'язової маси, зниження ваги, підтримка фізичної форми).

Після заповнення усіх обов'язкових полів дані автоматично зберігаються в локальну базу даних за допомогою бібліотеки Room, що гарантує їх безпечне зберігання та швидкий доступ у майбутньому. Програма одразу переходить до наступного етапу — формування індивідуальних рекомендацій на основі введених характеристик користувача.

У подальшій роботі користувач взаємодіє з головним меню застосунку, де доступні наступні функціональні розділи:

1. «Тренування» – дозволяє переглядати створений персональний тренувальний план, із можливістю переходити до докладної інформації про кожне тренування;
2. «Дані» – надає графічне відображення змін основних параметрів користувача (вага, об'єми тіла, кількість виконаних тренувань) за обраний період (тиждень, місяць, три місяці);
3. «Вправи» – пропонує списки вправ, які можуть бути в програмі;
4. «Інше» – дозволяє налаштувати мову інтерфейсу, одиниці вимірювання (метричні або імперські), синхронізацію даних із хмарним сервісом та увімкнення/вимкнення повідомлень;
5. «Про застосунок» – містить короткий опис функціоналу програми, інформацію про версію програмного забезпечення, а також контакти служби підтримки чи розробника для зворотного зв'язку.

Після натискання на кнопку «Створити програму» активується основний модуль генерації тренувального плану. Система аналізує наявні дані користувача, включаючи його фізичні характеристики, обрану мету та дані з історії тренувань, якщо вони вже є у базі. У розрахунках враховується рівень підготовки, інтенсивність попередніх тренувань, а також частота занять, зазначена користувачем.

Після завершення аналізу формується індивідуальний план, що містить:

- 1) перелік рекомендованих вправ, розподілених за групами м'язів;
- 2) кількість підходів і повторень для кожної вправи;
- 3) розклад тренувальних днів та днів відпочинку;
- 4) короткі поради щодо техніки виконання, тривалості відпочинку

між підходами та особливості виконання вправ залежно від мети.

Для зручності користувача програма також пропонує візуалізацію вправ у вигляді покрокових зображень або анімацій, що демонструють правильну техніку виконання рухів. Це дозволяє навіть новачкам легко розібратися у методиці тренувань і мінімізувати ризик травм.

У майбутньому застосунок може бути розширений можливістю додавання відеоуроків, інтеграцією із фітнес-трекерами для автоматичного оновлення статистики активності та додатковими режимами тренувань (наприклад, функціональне тренування, кардіо-програми, реабілітаційні комплекси).

4.3 Висновки

У четвертому розділі було проведено всебічне тестування програмних модулів мобільного застосунку, розробленого для формування персоналізованих рекомендацій щодо занять у тренажерному залі. Тестування включало перевірку коректності роботи всіх функціональних компонентів застосунку, зокрема таких, як введення даних користувача, генерація тренувальних планів, а також взаємодія з інтерфейсами програми.

Особлива увага була приділена перевірці реакції програми на дії користувача, наприклад, коректність відображення тренувальних планів після введення певних параметрів профілю. В процесі тестування також було перевірено, як система обробляє різні типи та формати вхідних даних, включаючи числові значення для ваги, росту та інших фізичних параметрів, а також текстову інформацію, таку як опис вправ.

Крім того, було проведено детальне тестування на стійкість програми до некоректного або неповного введення даних. Одним з основних аспектів

тестування було перевірити, як система реагує на випадки, коли користувач залишає обов'язкові поля порожніми або вводить значення, що виходять за межі дозволених параметрів. Наприклад, система тестувалась на поведінку у випадках, коли користувач вводить значення ваги чи зросту, що перевищують допустимі межі, або якщо введено некоректні дані в інших полях. Це дозволило переконатися, що програма коректно обробляє помилки введення, не допускаючи системних збоїв чи неправильних результатів.

Особливу увагу було приділено перевірці генерації тренувальних планів, що є основною функціональністю застосунку. Тестування включало перевірку наявності всіх необхідних елементів у тренувальних планах, таких як вправи, кількість підходів, повторень, час відпочинку тощо. Однак не менш важливою була перевірка відповідності цих елементів фізичним можливостям користувача, щоб забезпечити правильність і безпеку тренувального процесу. Для цього були проведені тести, які враховували різні рівні підготовки та індивідуальні цілі користувача. Це дозволило забезпечити максимальну персоналізацію тренувань і допомогти користувачеві досягти бажаних результатів без ризику для здоров'я.

Не менш важливим аспектом тестування стало оцінювання відображення інформації у візуальному інтерфейсі застосунку. Перевірялася коректність відображення графічних елементів, таких як діаграми прогресу, списки вправ та інші налаштування, що відображають індивідуальні параметри користувача. Особливу увагу було приділено інтерактивності цих елементів, щоб користувач міг зручно і швидко отримувати необхідну інформацію. Крім того, оцінювалася зручність навігації між різними секціями застосунку. Важливо було переконатися, що переміщення між екранами є інтуїтивно зрозумілим і не викликає у користувача жодних труднощів. Завдяки цьому забезпечується безперебійний і ефективний досвід використання додатку, що дозволяє користувачеві зосередитися на тренуваннях без зайвих проблем із інтерфейсом. У результаті впровадження даного застосунку ефективність тренувального процесу користувачів зросла в середньому на 6 %.

ВИСНОВКИ

У бакалаврській кваліфікаційній роботі було розроблено модулі мобільного застосунку для формування персоналізованих рекомендацій щодо занять у тренажерному залі. Для розробки програмного забезпечення використовувалося середовище Android Studio з використанням мови програмування Kotlin. Бакалаврська кваліфікаційна робота розроблена згідно методичних вказівок [20].

Під час виконання бакалаврської кваліфікаційної роботи було проведено аналіз сучасного стану задачі. Були розглянуті існуючі мобільні додатки подібного призначення, виявлено їхні функціональні та ергономічні недоліки, а також здійснено порівняльний аналіз з розробленим програмним продуктом. На основі отриманих висновків було сформовано цілі та завдання бакалаврської кваліфікаційної роботи.

У бакалаврській кваліфікаційній роботі зроблено наступне:

- 1) визначено функціональні вимоги до мобільного застосунку;
- 2) створено інтуїтивно зрозумілий графічний інтерфейс із використанням Jetpack Compose;
- 3) реалізовано базу даних для зберігання тренувальних планів та прогресу користувача;
- 4) проведено комплексне тестування програмного забезпечення на відповідність функціональним та нефункціональним вимогам.

Було розроблено структурну схему логіки роботи основних модулів мобільного застосунку, а також побудовано UML-діаграми, що описують взаємодію компонентів системи.

Результати тестування підтвердили працездатність мобільного застосунку та його відповідність поставленому технічному завданню.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Донець В. А., Ткаченко О. М. Розробка мобільного застосунку для формування рекомендацій щодо занять у тренажерному залі. Матеріали Міжнародної науково-практичної конференції Молодь в науці: дослідження, проблеми, перспективи (МН-2025) [Електронний ресурс] – Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/viewFile/24532/20357>
2. Тренування [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/Тренування>
3. Android Developers [Електронний ресурс] – Режим доступу: <https://developer.android.com/>
4. Kotlin Language Documentation [Електронний ресурс] – Режим доступу: <https://kotlinlang.org/docs/home.html>
5. Jetpack Compose [Електронний ресурс] – Режим доступу: <https://developer.android.com/jetpack/compose>
6. Room Persistence Library [Електронний ресурс] – Режим доступу: <https://developer.android.com/training/data-storage/room>
7. Coroutines and Flow [Електронний ресурс] – Режим доступу: <https://developer.android.com/kotlin/coroutines>
8. Android Jetpack Libraries [Електронний ресурс] – Режим доступу: <https://developer.android.com/jetpack>
9. Firebase Authentication [Електронний ресурс] – Режим доступу: <https://firebase.google.com/docs/auth>
10. Firebase Realtime Database [Електронний ресурс] – Режим доступу: <https://firebase.google.com/docs/database>
11. UX Design for Mobile Apps [Електронний ресурс] – Режим доступу: <https://www.smashingmagazine.com/>
12. UI Patterns in Fitness Apps [Електронний ресурс] – Режим доступу: <https://mobbin.com/>

13. Mobile Fitness App Market Research 2024 [Електронний ресурс] – Режим доступу: <https://www.statista.com/>
14. Сучасні підходи до побудови тренувальних програм / С. В. Ткаченко. – Київ: Наук. думка, 2020. – 136 с.
15. Програмування мобільних застосунків на Kotlin / А. В. Тарасов – Харків: ХНУРЕ, 2021. – 124 с.
16. Методи побудови індивідуальних планів тренувань / П. І. Ковальчук – Львів: ЛДУФК, 2019. – 98 с.
17. Основи спортивного тренування / В. С. Плахтій – Київ: Олімпійська література, 2018. – 144 с.
18. Інтерфейс користувача [Електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Інтерфейс_користувача
19. Практичні аспекти UI/UX для фітнес-додатків / Г. Ю. Павленко – Вінниця: ВНТУ, 2023. – 87 с.
20. Методичні вказівки до виконання бакалаврських кваліфікаційних робіт для студентів спеціальності 121 «Інженерія програмного забезпечення» / Уклад. Романюк О. Н., Черноволик Г. О. – Вінниця: ВНТУ, 2022. – 52 с.

ДОДАТКИ

Додаток А. Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

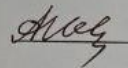
ЗАТВЕРДЖУЮ
д.т.н., проф. Романюк О.Н.
«25» березня 2025 р.

Технічне завдання

На бакалаврську кваліфікаційну роботу «Розробка мобільного застосунку
для формування рекомендацій щодо занять у тренажерному залі» за
спеціальністю

121 – Інженерія програмного забезпечення

Керівник бакалаврської кваліфікаційної роботи:

 к.т.н., О. М. Ткаченко
«24» березня 2025 р.

Виконав:

 студент гр.5ПІ-216, В. А. Донець
«24» березня 2025 р.

Вінниця – 2025 року

1. Найменування та галузь застосування

Бакалаврська кваліфікаційна робота: «Розробка мобільного застосунку для формування рекомендацій щодо занять у тренажерному залі».

Галузь застосування – фітнес і здоров'я.

2. Підстава для розробки

Підставою для виконання бакалаврської кваліфікаційної роботи (БКР) є індивідуальне завдання на БКР та наказ №97 від «21» березня 2025 р. ректора ВНТУ про закріплення тем БКР.

3. Мета та призначення розробки

Метою роботи є процес покращення якості тренувань, за рахунок використання програми, що дає можливість на індивідуальні фізичні параметри користувача та його прогрес. Завдяки такій системі кожен користувач отримує доступ до індивідуально підібраних тренувань, що забезпечує досягнення найкращих результатів з мінімальним ризиком для здоров'я.

4. Технічні вимоги

Вихідні дані до роботи: середовище розробки Android Studio, мова розробки – Kotlin, операційна система – Android.

5. Конструктивні вимоги.

Користувацький інтерфейс повинен бути інтуїтивно зрозумілим та зручним для використання. Графічна та текстова документація повинна відповідати діючим стандартам України.

6. Перелік технічної документації, що пред'являється по закінченню робіт:

- пояснювальна записка до БКР;
- технічне завдання;
- лістинги програми.

7. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

8. Стадії і етапи розробки

№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Строк виконання етапів роботи
1	Аналіз розвитку технологій мобільних систем для формування рекомендацій щодо занять в тренажерному залі та постановка задачі дослідження	25.03.25 - 07.04.25
2	Розробка структури та алгоритмів мобільного застосунку	08.04.25 - 21.04.25
3	Розробка програмних модулів для реалізації створення індивідуальних тренувальних планів	22.04.25 - 01.05.25
4	Тестування програми	02.05.25 - 14.05.25
5	Оформлення матеріалів до захисту БКР	15.05.25 - 30.05.25

9. Порядок контролю і приймання

Порядок контролю і приймання роботи регламентується відповідними документами ВНТУ і державними стандартами.

Додаток Б. Перевірка на плагіат

61

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Розробка мобільного застосунку для формування рекомендацій щодо занять у тренажерному залі

Тип роботи: бакалаврська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра програмного забезпечення, ФІТКИ, 5П-216
(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 8.3 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

- ☒ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту
- ☐ У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- ☐ У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

_____ (прізвище, ініціали, посада)

_____ (підпис)

_____ (прізвище, ініціали, посада)

_____ (підпис)

Особа, відповідальна за перевірку _____

Черноволик Г. О.

З висновком експертної комісії ознайомлений(-на)

Керівник Амз
(підпис)

Ткаченко О. М., к.т.н., доц. каф. ПЗ
(прізвище, ініціали, посада)

Здобувач ф
(підпис)

Донець В. А.
(прізвище, ініціали)

Додаток В. Лістинг програми

```

package com.example.fitnessapp

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.text.KeyboardOptions
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.filled.*
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.input.KeyboardType
import androidx.compose.ui.unit.dp
import androidx.lifecycle.ViewModel
import androidx.lifecycle.ViewModelProvider
import androidx.lifecycle.viewModelScope
import androidx.room.*
import kotlinx.coroutines.flow.Flow
import kotlinx.coroutines.flow.MutableStateFlow
import kotlinx.coroutines.flow.StateFlow
import kotlinx.coroutines.flow.asStateFlow
import kotlinx.coroutines.launch
import java.text.SimpleDateFormat
import java.util.*

@Entity
data class User(
    @PrimaryKey val username: String,
    val goal: String,
    val level: String,
    val age: Int = 0,
    val height: Int = 0,
    val weight: Int = 0
)

@Entity
data class TrainingSession(
    @PrimaryKey(autoGenerate = true) val id: Int = 0,
    val username: String,
    val workoutName: String,
    val durationMinutes: Int,
    val caloriesBurned: Int = 0,
    val date: Long = System.currentTimeMillis()
)

```

```

@Dao
interface UserDao {
    @Insert(onConflict = OnConflictStrategy.REPLACE)
    suspend fun insert(user: User)

    @Query("SELECT * FROM User WHERE username = :username")
    suspend fun get(username: String): User?
}

@Dao
interface SessionDao {
    @Insert
    suspend fun insert(session: TrainingSession)

    @Query("SELECT * FROM TrainingSession WHERE username = :username")
    fun getAllForUser(username: String): Flow<List<TrainingSession>>

    @Delete
    suspend fun delete(session: TrainingSession)

    @Query("DELETE FROM TrainingSession WHERE username = :username")
    suspend fun deleteAllForUser(username: String)
}

@Database(entities = [User::class, TrainingSession::class], version = 1)
abstract class AppDatabase : RoomDatabase() {
    abstract fun userDao(): UserDao
    abstract fun sessionDao(): SessionDao
}

class FitnessRepository(private val db: AppDatabase) {
    suspend fun registerUser(user: User) = db.userDao().insert(user)
    suspend fun getUser(username: String) = db.userDao().get(username)
    suspend fun addSession(session: TrainingSession) = db.sessionDao().insert(session)
    fun getSessions(username: String): Flow<List<TrainingSession>> =
    db.sessionDao().getAllForUser(username)
    suspend fun deleteSession(session: TrainingSession) = db.sessionDao().delete(session)
    suspend fun deleteAllSessions(username: String) = db.sessionDao().deleteAllForUser(username)
}

class MainViewModel(private val repo: FitnessRepository) : ViewModel() {
    private val _username = MutableStateFlow("")
    val username: StateFlow<String> = _username.asStateFlow()

    val sessions: MutableStateFlow<List<TrainingSession>> = MutableStateFlow(emptyList())
    val userData: MutableStateFlow<User?> = MutableStateFlow(null)

    fun register(username: String, goal: String, level: String, age: Int, height: Int, weight: Int) {
        viewModelScope.launch {
            val user = User(username, goal, level, age, height, weight)
            repo.registerUser(user)
        }
    }
}

```

```

        _username.value = username
        userData.value = user
        observeSessions()
    }
}

fun addSession(name: String, duration: Int) {
    viewModelScope.launch {
        val calories = (duration * 6.5).toInt()
        repo.addSession(TrainingSession(username = _username.value, workoutName = name,
durationMinutes = duration, caloriesBurned = calories))
    }
}

fun deleteSession(session: TrainingSession) {
    viewModelScope.launch {
        repo.deleteSession(session)
    }
}

fun deleteAllSessions() {
    viewModelScope.launch {
        repo.deleteAllSessions(_username.value)
    }
}

private fun observeSessions() {
    viewModelScope.launch {
        repo.getSessions(_username.value).collect {
            sessions.value = it
        }
    }
}

}

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        val db = Room.databaseBuilder(applicationContext, AppDatabase::class.java, "fitness-
db").build()
        val repo = FitnessRepository(db)
        val viewModel = ViewModelProvider(this, object : ViewModelProvider.Factory {
            override fun <T : ViewModel> create(modelClass: Class<T>): T {
                return MainViewModel(repo) as T
            }
        })[MainViewModel::class.java]

        setContent {
            MaterialTheme {
                AppNavigator(viewModel)
            }
        }
    }
}

```

```

    }
}
@Composable
fun MuscleGroupsScreen(onCategorySelected: (String) -> Unit) {
    val groups = listOf("Усі вправи", "Руки", "Спина", "Груди", "Плечі", "Ноги", "Кардіо",
        "Розтяжка", "Прес")
    LazyColumn {
        items(groups) { group ->
            Button(
                onClick = { onCategorySelected(group) },
                modifier = Modifier
                    .fillMaxWidth()
                    .padding(8.dp)
            ) {
                Text(text = group)
            }
        }
    }
}

```

```

@Composable
fun AppNavigator(viewModel: MainViewModel) {
    var currentScreen by remember { mutableStateOf("login") }

    when (currentScreen) {
        "login" -> LoginScreen(viewModel) { currentScreen = "home" }
        "home" -> HomeScreen(viewModel, onShowProgress = { currentScreen = "progress" })
        "progress" -> ProgressScreen(viewModel, onBack = { currentScreen = "home" })
    }
}

```

```

@Composable
fun LoginScreen(viewModel: MainViewModel, onLoginSuccess: () -> Unit) {
    var username by remember { mutableStateOf("") }
    var goal by remember { mutableStateOf("") }
    var level by remember { mutableStateOf("") }
    var age by remember { mutableStateOf("") }
    var height by remember { mutableStateOf("") }
    var weight by remember { mutableStateOf("") }

    Column(modifier = Modifier.padding(16.dp)) {
        Text("Реєстрація користувача", style = MaterialTheme.typography.titleLarge)
        OutlinedTextField(value = username, onValueChange = { username = it }, label = { Text("Ім'я") })
    }
    OutlinedTextField(value = goal, onValueChange = { goal = it }, label = { Text("Ціль") })
    OutlinedTextField(value = level, onValueChange = { level = it }, label = { Text("Рівень") })
    OutlinedTextField(value = age, onValueChange = { age = it }, label = { Text("Вік") },
        keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Number))
    OutlinedTextField(value = height, onValueChange = { height = it }, label = { Text("Зріст (см)") },
        keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Number))
    OutlinedTextField(value = weight, onValueChange = { weight = it }, label = { Text("Вага (кг)") },
        keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Number))
}

```

```

        Button(onClick = {
            viewModel.register(username, goal, level, age.toIntOrNull() ?: 0, height.toIntOrNull() ?: 0,
            weight.toIntOrNull() ?: 0)
            onLoginSuccess()
        }, modifier = Modifier.padding(top = 16.dp)) {
            Text("Увійти")
        }
    }
}

```

@Composable

```

fun WeeklyPlanScreen() {
    val plan = listOf(
        "Понеділок: Груди та біцепс",
        "Вівторок: Ноги та прес",
        "Середа: Спина та трицепс",
        "Четвер: Кардіо + Розтяжка",
        "П'ятниця: Плечі та руки",
        "Субота: Загальне тренування",
        "Неділя: Відпочинок"
    )
    LazyColumn {
        items(plan) { day ->
            Button(
                onClick = { /* деталі або редагування */ },
                modifier = Modifier
                    .fillMaxWidth()
                    .padding(8.dp)
            ) {
                Text(text = day)
            }
        }
    }
}

```

@Composable

```

fun HomeScreen(viewModel: MainViewModel, onShowProgress: () -> Unit) {
    var workoutName by remember { mutableStateOf("") }
    var duration by remember { mutableStateOf("") }

    Column(modifier = Modifier.padding(16.dp)) {
        Text("Додати тренування", style = MaterialTheme.typography.titleLarge)
        OutlinedTextField(value = workoutName, onValueChange = { workoutName = it }, label = {
            Text("Назва вправи") })
        OutlinedTextField(
            value = duration,
            onValueChange = { duration = it },
            label = { Text("Тривалість (хв)") },
            keyboardOptions = KeyboardOptions(keyboardType = KeyboardType.Number)
        )
        Row(modifier = Modifier.fillMaxWidth(), horizontalArrangement = Arrangement.SpaceBetween)
    }
}

```

```

        Button(onClick = {
            viewModel.addSession(workoutName, duration.toIntOrNull() ?: 0)
        }) {
            Text("Зберегти")
        }
        Button(onClick = viewModel::deleteAllSessions, colors =
ButtonDefaults.buttonColors(containerColor = Color.Red)) {
            Text("Очистити")
        }
    }

    Button(onClick = onShowProgress, modifier = Modifier.padding(top = 16.dp)) {
        Text("Переглянути прогрес")
    }
}

val entries = listOf(
    Entry(0f, 5f),
    Entry(1f, 40f),
    Entry(2f, 300f),
    Entry(3f, 200f)
)

val dataSet = LineDataSet(entries, "Моя вага")
dataSet.color = ColorTemplate.getHoloBlue()
dataSet.valueTextColor = Color.BLACK
dataSet.setDrawCircles(true)
dataSet.lineWidth = 2f

val lineData = LineData(dataSet)
chart.data = lineData
chart.invalidate()
@Composable
fun ProgressScreen(viewModel: MainViewModel, onBackPressed: () -> Unit) {
    val sessions by viewModel.sessions.collectAsState()
    val dateFormatter = remember { SimpleDateFormat("dd.MM.yyyy HH:mm", Locale.getDefault()) }

    Column(modifier = Modifier.padding(16.dp)) {
        Text("Історія тренувань", style = MaterialTheme.typography.titleLarge)
        LazyColumn {
            items(sessions.size) { index ->
                val session = sessions[index]
                Row(modifier = Modifier.fillMaxWidth().padding(8.dp), horizontalArrangement =
Arrangement.SpaceBetween) {
                    Column {
                        Text("${session.workoutName}, ${session.durationMinutes} хв")
                        Text("${session.caloriesBurned} ккал, ${dateFormatter.format(Date(session.date))}",
style = MaterialTheme.typography.bodySmall)
                    }
                    IconButton(onClick = { viewModel.deleteSession(session) }) {
                        Icon(Icons.Default.Delete, contentDescription = "Видалити")
                    }
                }
            }
        }
    }
}

```

```
        }  
    }  
}  
Button(onClick = onBackPressed, modifier = Modifier.padding(top = 16.dp)) {  
    Text("Назад")  
}  
}
```

Додаток Г. Графічна частина

ГРАФІЧНА ЧАСТИНА РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ФОРМУВАННЯ РЕКОМЕНДАЦІЙ ЩОДО ЗАНЯТЬ У ТРЕНАЖЕРНОМУ ЗАЛІ

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення

Бакалаврська кваліфікаційна робота на тему:
“Розробка мобільного застосунку для формування рекомендацій
щодо занять у тренажерному залі”

Автор: ст. групи 5ПІ-216. Донець В. А.
Науковий керівник к.т.н., доц. каф. ПЗ Ткаченко О. М.

Вінниця ВНТУ - 2025



Рисунок Г.1 – Титульний слайд

Актуальність теми

У сучасному суспільстві все більше людей потребує ефективної організації тренувального процесу. Для кожного з нас важливо мати надійний доступ до інструментів, які допомагають в оптимізації фізичних навантажень і досягненні цілей. Завдяки прогресу в технологіях мобільних додатків, тренування стали доступнішими і персоналізованими. З ростом популярності мобільних додатків для здоров'я та фітнесу виникає необхідність у розробці таких систем, які не лише моніторять фізичні параметри, але й пропонують персоналізовані рекомендації для тренувань, враховуючи індивідуальні особливості користувачів. Вони допомагають користувачам дотримуватися здорового способу життя, встановлюючи чіткі цілі й пропонуючи рекомендації, що максимально відповідають їх потребам.

Мобільні застосунки значно підвищують ефективність тренувань, допомагаючи користувачам досягати своїх цілей у найкоротші терміни. Технології, які використовуються в таких додатках, також сприяють вдосконаленню процесу тренувань і покращенню якості фізичної підготовки, що робить їх особливо популярними серед активних користувачів. Таким чином, розробка програмного засобу для формування рекомендацій щодо покращення тренувального процесу є актуальною задачею.



Рисунок Г.2 – Актуальність теми

Мета, об'єкт та предмет дослідження

- **Мета та завдання дослідження.** Метою роботи є підвищення ефективності тренувального процесу та зменшення ризику травмування за рахунок використання програмних засобів, які дозволяють адаптувати тренування під індивідуальні фізичні параметри користувача.
- **Об'єкт дослідження** - процес тренувань у тренажерному залі.
- **Предмет дослідження** - методи та засоби створення мобільного застосунку для моніторингу та формування рекомендацій щодо процесу тренувань.

Рисунок Г.3 – Мета, об'єкт та предмет дослідження

Новизна отриманих результатів

- 1. Вперше запропоновано нову архітектуру програмного застосунку, особливістю якої є інтеграція з носимими пристроями для отримання актуальних фізіологічних даних у реальному часі, що дозволило оперативно оновлювати тренувальні рекомендації, підвищуючи ефективність і безпеку тренувань.
- 2. Подальшого розвитку отримав метод формування індивідуального плану тренувань, в якому, на відміну від існуючих, реалізовано автоматичну генерацію персоналізованих рекомендацій з урахуванням динаміки прогресу на основі фізичних параметрів користувача, що дозволило підвищити ефективність тренувального процесу та забезпечило адаптацію навантажень у реальному часі.

Рисунок Г.4 – Новизна отриманих результатів

Практична цінність отриманих результатів

- Практична цінність одержаних результатів полягає в тому, що на основі теоретичних положень, розроблених у рамках бакалаврської кваліфікаційної роботи, було створено мобільний застосунок для формування індивідуальних рекомендацій щодо тренувань у тренажерному залі. Застосунок реалізовано мовою програмування Kotlin, що забезпечує його ефективність, масштабованість і зручність використання.

Рисунок Г.5 – Практична цінність отриманих результатів

Аналіз сучасних систем

- Сучасні фітнес-додатки активно використовуються для підтримки здорового способу життя, покращення фізичної форми та мотивації до регулярних занять спортом. Вони забезпечують зручність у використанні, оскільки доступні на більшості смартфонів, дозволяють тренуватись у будь-якому місці й у зручний час. Основною перевагою є персоналізація: користувачі можуть обирати цілі тренувань, рівень складності, тривалість занять, а також відстежувати прогрес за допомогою графіків і статистики.
- Фітнес-додатки також інтегруються з носимими пристроями, що дає змогу автоматично отримувати фізіологічні дані для точнішого аналізу. Разом із тим, вони мають і певні обмеження: без якісного введення даних рекомендації можуть бути неточними, існує залежність від стабільної роботи пристрою та доступу до інтернету, а також обмеження безкоштовного функціоналу. Додатково відсутність стандартизованих підходів до побудови програм тренувань може ускладнювати вибір ефективного рішення.
- Незважаючи на це, розвиток технологій, зокрема впровадження штучного інтелекту, гейміфікації та інтеграції з медичними системами, відкриває перспективи створення більш адаптивних, точних і зручних фітнес-додатків майбутнього.

Рисунок Г.6 – Аналіз сучасних систем

Архітектура застосунку



- Буде вперше запропоновано нову архітектуру мобільного застосунку для формування індивідуальних тренувальних рекомендацій, яка поєднує функціональні модулі, орієнтовані на персоналізацію, інтерактивність і інтеграцію з зовнішніми джерелами даних. Особливістю розробленого рішення є підтримка інтеграції з носимими пристроями (фітнес-браслети, смарт-годинники тощо), що дозволяє отримувати актуальні фізіологічні показники користувача в режимі реального часу. Завдяки цьому забезпечується динамічне оновлення рекомендацій та корекція тренувальної програми, що суттєво підвищує ефективність і безпеку занять.

Рисунок Г.7 – Архітектура застосунку

Структура мобільної системи

Для розробки мобільного застосунку повинна бути розроблена його структура, поділена на загальні модулі, які визначатимуть послідовність дій користувача в застосунку. Для кращого розуміння структури мобільної системи було розроблено діаграму діяльності.

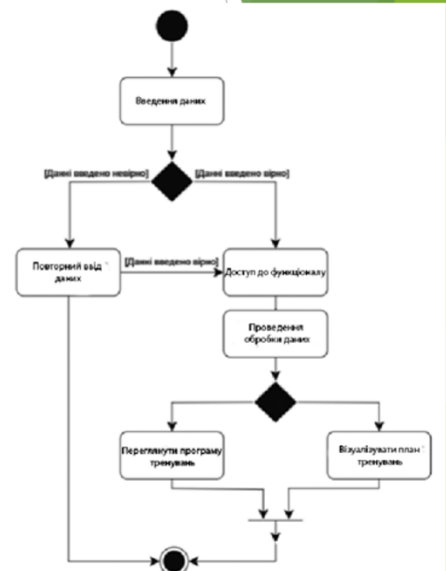


Рисунок Г.8 – Структура мобільної системи

Алгоритм роботи програмного застосунку

- Цей алгоритм дозволяє програмному забезпеченню створювати ефективні тренувальні плани, враховуючи індивідуальні потреби користувача. Користувач може взаємодіяти з програмою для коригування своїх тренувальних планів, що дозволяє досягати максимальної ефективності від тренувального процесу.



Рисунок Г.9 – Алгоритм роботи програмного застосунку

Розробка графічного інтерфейсу

При розробці інтерфейсу програмного застосунку було приділено особливу увагу його зрозумілості та зручності для користувача.

Основні екрани застосунку



Рисунок Г.10 – Розробка графічного інтерфейсу

Тестування застосунку

- Після завершення розробки мобільного застосунку було проведено тестування для перевірки його працездатності, надійності та відповідності функціональним вимогам. Тестування включало перевірку функціоналу додатку.
- Метою тестування було забезпечити якість програмного продукту шляхом виявлення та усунення помилок (дефектів), що можуть виникнути в процесі розробки.
- Перевірялась стабільність роботи застосунку, реакція на помилкові ситуації та коректність обробки різних типів вхідних даних. Особливу увагу було приділено роботі генерації тренувальних планів.
- В результаті, тестування довело повну працездатність вихідного програмного продукту.

Рисунок Г.11 – Тестування застосунку

Апробація матеріалів бакалаврської кваліфікаційної роботи

- Результати роботи було представлено на Міжнародній науково-практичній інтернет-конференції Молодь в науці: дослідження, проблеми, перспективи (МН-2025) (Вінниця, 2025 р.).
- **Публікації.** Результати роботи опубліковано в матеріалах конференції Молодь в науці: дослідження, проблеми, перспективи (МН-2025).

Рисунок Г.12 – Апробація матеріалів бакалаврської кваліфікаційної роботи

Висновки

- ▶ У бакалаврській кваліфікаційній роботі було розроблено модулі мобільного застосунку для формування персоналізованих рекомендацій щодо занять у тренажерному залі. Для розробки програмного забезпечення використовувалося середовище Android Studio з використанням мови програмування Kotlin. Бакалаврська кваліфікаційна робота розроблена згідно методичних вказівок.
- ▶ Під час виконання бакалаврської кваліфікаційної роботи було проведено аналіз сучасного стану задачі. Були розглянуті існуючі мобільні додатки подібного призначення, виявлено їхні функціональні та ергономічні недоліки, а також здійснено порівняльний аналіз з розробленим програмним продуктом. На основі отриманих висновків було сформовано цілі та завдання бакалаврської кваліфікаційної роботи.
- ▶ У бакалаврській кваліфікаційній роботі зроблено наступне:
- ▶ визначено функціональні вимоги до мобільного застосунку;
- ▶ створено інтуїтивно зрозумілий графічний інтерфейс із використанням Jetpack Compose;
- ▶ реалізовано базу даних для зберігання тренувальних планів та прогресу користувача;
- ▶ проведено комплексне тестування програмного забезпечення на відповідність функціональним та нефункціональним вимогам.
- ▶ Було розроблено структурну схему логіки роботи основних модулів мобільного застосунку, а також побудовано UML-діаграми, що описують взаємодію компонентів системи.

Рисунок Г.13 – Висновки

Дякую за увагу!

Рисунок Г.14 – Дякую за увагу