

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, назва факультету (відділення))

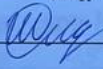
Кафедра комп'ютерних наук
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія організації та проведення тренувального процесу. Частина 2. Надання рекомендацій»

Виконав: студент 2-го курсу, групи 2КН-22м
спеціальності 122 – «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)


Шелестюк М. І.
(прізвище та ініціали)

Керівник: д.т.н., професор кафедри КН


Яровий А. А.
(прізвище та ініціали)

« 07 » 12 2023 р.

Опонент: д.т.н. в.о. зав. кафедри КСУ

Юхимчук М. С.
(прізвище та ініціали)

« 07 » 12 2023 р.

Допущено до захисту
Завідувач кафедри КН

д.т.н., проф. Яровий А.А.
« 08 » 12 2023 р.

Вінниця ВНТУ – 2023 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти П-й (магістерський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «Системи штучного інтелекту»

ЗАТВЕРДЖУЮ
Завідувач кафедри КН
Д.т.н., проф. Яровий А.А.

29.08. 2023 року

ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Шелестюку Максиму Ігоровичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Інформаційна технологія організації та проведення тренувального процесу. Частина 2. Надання рекомендацій.

керівник роботи: д.т.н., професор кафедри КН Яровий А.А.

затверджені наказом ВНТУ від «18» 09 2023 року № 247

2. Строк подання студентом роботи «13» 11 2023 року

3. Вихідні дані до роботи: середня швидкість відповіді серверу до 2000мс, максимальний розмір бази знань менше 40МБ, потужність бази знань не менше 30.



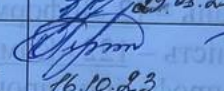
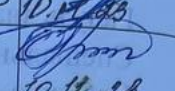
4. Зміст текстової частини:

Вступ, аналіз сучасного стану розвитку технологій організації та проведення тренувального процесу, проектування експертної системи для інформаційної технології організації та проведення тренувального процесу, програмна реалізація експертної системи для інформаційної технології організації та проведення тренувального процесу, програмна реалізація серверної частини інформаційної технології організації та проведення тренувального процесу, економічна частина, висновки, перелік використаних джерел, додатки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень):

Структура вхідних факторів, структура вихідних факторів, діаграма компонентів, діаграма розгортання, діаграма інфологічної моделі бази даних, діаграма даталогічної моделі бази даних.

6. Консультанти розділів роботи

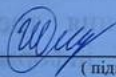
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1-4	Яровий Андрій Анатолійович, д.т.н., професор кафедри КН.	 09.09.23	 10.11.23
5	Ратушняк Ольга Георгіївна, к.т.н., доцент кафедри ЕПВМ	 16.10.23	 10.11.23

7. Дата видачі завдання «29» 08 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітки
		початок	закінчення	
1	Аналіз сучасного стану розвитку технологій організації та проведення тренувального процесу	01.09.23	07.09.23	
2	Проектування експертної системи для організації тренувального процесу	08.09.23	24.09.23	
3	Програмна реалізація експертної системи для організації тренувального процесу	25.09.23	15.10.23	
4	Розробка серверної частини інформаційної технології організації та проведення тренувального процесу	16.10.23	29.10.23	
5	Підготовка економічної частини	30.10.23	05.11.23	
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	06.11.23	10.11.23	

Студент


(підпис)

Шелестюк М.І.
(прізвище та ініціали)

Керівник роботи


(підпис)

Яровий А. А.
(прізвище та ініціали)

АНОТАЦІЯ

УДК 004.8

Шелестюк М. І. Інформаційна технологія організації та проведення тренувального процесу. Частина 2. Надання рекомендацій. Магістерська кваліфікаційна робота зі спеціальності 122 «Комп'ютерні науки», освітня програма «Системи штучного інтелекту». Вінниця: ВНТУ, 2023. 155 с.

Укр. мовою. Бібліогр.: 24 назв; рис.: 29; табл. 22.

Дана магістерська кваліфікаційна робота присвячена розробці інформаційної технології організації та проведення тренувального процесу, а саме експертної системи для надання рекомендацій щодо програми тренувань та серверної частини інформаційної технології. Було проведено аналіз сучасного стану розвитку технологій організації та проведення тренувального процесу. Аналіз програм-аналогів дозволив виявити їх переваги та недоліки. Здійснено проектування експертної системи для створення програми тренувань, в ході якого було обґрунтовано вибір продукційної моделі подання знань та проведено структурування знань. В ході програмної реалізації спроектованої експертної системи було розроблено базу знань в середовищі CLIPS. Виконано програмну реалізацію серверної частини з використанням технологій .NET Core та MS SQL Server. Тестування показало підвищення якості надання рекомендацій щодо визначення програми тренувань на 9.6% в порівнянні з програмами аналогами.

Графічна частина складається з 6 плакатів.

У економічному розділі визначено, що згідно узагальненого коефіцієнту конкурентоспроможності, науково-технічна розробка переважає існуючі аналоги приблизно в 2,26 рази. Термін окупності становить 2,86 р., що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Ключові слова: тренувальний процес, організація тренувального процесу, експертна система, програма тренувань, база знань.

ABSTRACT

Shelestiuk M.I. Information Technology for Organizing and Conducting Training Process. Part 2. Providing recommendations. Master's thesis in the specialty 122 «Computer sciences», educational program «Artificial intelligence systems». Vinnytsia: VNTU, 2023. 155 p.

In Ukrainian language. Bibliogr.: 24 titles; fig. 29; table 22.

This master's thesis is dedicated to the development of information technology for organizing and conducting training process, namely, an expert system for providing training program recommendations and the server part of information technology. An analysis of the current state of technology development of the organization and training process was conducted. The analysis of similar programs revealed their advantages and disadvantages. The design of an expert system for the training program creation was carried out, during which production model of knowledge presentation was approved and the structuring of knowledge was carried out. During the software implementation of the designed expert system, a knowledge base was developed in the CLIPS environment. The server part was implemented using .NET Core and MS SQL Server technologies. Testing showed a 9.6% improvement in the quality of providing recommendations for training program creation compared to similar programs.

The graphical part consists of 6 posters.

In the economic section, it is determined that according to the generalized coefficient of competitiveness, scientific and technical development exceeds existing analogues by approximately 2.26 times. The payback period is 2.86 years, which is less than 3 years, which indicates the commercial attractiveness of the scientific and technical development and may encourage a potential investor to finance the implementation of this development and its introduction to the market

Keywords: training process, organization of the training process, expert system, training program, knowledge base.

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ СУЧАСНОГО СТАНУ РОЗВИТКУ ТЕХНОЛОГІЙ ОРГАНІЗАЦІЇ ТА ПРОВЕДЕННЯ ТРЕНУВАЛЬНОГО ПРОЦЕСУ	7
1.1 Аналіз предметної області технологій організації та проведення тренувального процесу	7
1.2 Аналіз відомих програмних рішень та програм-аналогів	9
1.3 Аналіз об'єкту проектування технології організації та проведення тренувального процесу	13
1.3.1 Визначення ключових показників при складанні програми тренувань .	14
1.3.2 Опис типових програм тренувань	18
1.4 Висновок до розділу 1	30
2 ПРОЕКТУВАННЯ ЕКСПЕРТНОЇ СИСТЕМИ ДЛЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОРГАНІЗАЦІЇ ТА ПРОВЕДЕННЯ ТРЕНУВАЛЬНОГО ПРОЦЕСУ.....	31
2.1 Класифікація експертних систем.....	31
2.2 Критерії розробки експертних систем.....	34
2.3 Обґрунтування вибору моделі подання знань	39
2.4 Структуризація знань для створення програми тренувань	49
2.5 Висновок до розділу 2	53
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ЕКСПЕРТНОЇ СИСТЕМИ ДЛЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОРГАНІЗАЦІЇ ТА ПРОВЕДЕННЯ ТРЕНУВАЛЬНОГО ПРОЦЕСУ	55
3.1 Обґрунтування вибору середовища для розробки експертних систем	55
3.2 Розробка бази знань експертної системи	61
3.3 Тестування розробленої експертної системи	69

3.4 Висновок до розділу 3	77
4 ПРОГРАМНА РЕАЛІЗАЦІЯ СЕРВЕРНОЇ ЧАСТИНИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОРГАНІЗАЦІЇ ТА ПРОВЕДЕННЯ ТРЕНУВАЛЬНОГО ПРОЦЕСУ	78
4.1 Обґрунтування вибору засобів розробки серверної частини	78
4.2 Опис структури серверної частини	81
4.3 Тестування серверної частини інформаційної технології та аналіз отриманих результатів	85
4.4 Висновок до розділу 4	91
5 ЕКОНОМІЧНА ЧАСТИНА	92
5.1 Аналіз ринку фітнес-застосунків	92
5.2 Комерційний та технологічний аудит науково-технічної розробки.....	97
5.3 Розрахунок витрат на проведення науково-дослідної роботи	111
5.4 Розрахунок економічної ефективності науково-технічної розробки та її можливої комерціалізації потенційним інвестором	121
5.5 Розрахунок ефективності вкладених інвестицій та періоду їх окупності	125
5.6 Висновок до розділу 5	127
ВИСНОВКИ	129
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	131
Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	135
Додаток Б (обов'язковий) Лістинг програми.....	136
Додаток В (обов'язковий) ІЛЮСТРАТИВНА ЧАСТИНА.....	149

ВСТУП

Актуальність теми дослідження. З моменту інтенсивного впровадження інформаційних технологій та комп'ютерів в життя людей, актуальність необхідності регулярних фізичних навантажень та тренувань почала стрімко зростати у суспільстві [1]. Це обумовлено різноманітними чинниками.

Процес впровадження інформаційних технологій та цифровізація на всіх рівнях життєдіяльності викликав значні зміни звичного способу життя більшості населення планети. Організм людини не здатний швидко адаптуватись до таких змін, що призводить до розвитку численних захворювань та проблем, які раніше зустрічались зрідка. Внаслідок досліджень цих проблем вчені знайшли кореляцію між регулярними фізичними навантаженнями та позитивним впливом на якість життя та здоров'я людини [2]. Таким чином, спостерігається стрімке зростання інтересу до проблеми регулярних фізичних навантажень та тренувань.

Актуальність розробки інформаційної технології організації та проведення тренувального процесу очевидна, оскільки сучасне суспільство відчуває потребу в забезпеченні фізичного здоров'я через прості та доступні рішення в умовах сучасного способу життя. Такими рішеннями є веб-застосунки та мобільні застосунки для організації та проведення тренувань, які надають індивідуальні програми тренувань та раціонів з можливістю відслідковувати власні показники та результати. Користувачі отримують зручність, самостійність та організованість під час своїх тренувань без дороговартісних персональних консультацій чи тренерів.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Розробка прикладних інтелектуальних інформаційних технологій та систем» та плану наукової і навчально-методичної роботи кафедри.

Мета та завдання дослідження. Метою дослідження є розширення функціональних можливостей інформаційної технології організації та проведення тренувального процесу, а також підвищення якості надання рекомендацій щодо визначення програми тренувань.

Для досягнення даної мети, необхідно сформулювати та вирішити наступні завдання:

- здійснити аналіз сучасного стану розвитку технологій організації та проведення тренувального процесу;
- здійснити аналіз моделей подання знань та обґрунтування їх вибору;
- провести структурування знань для створення програми тренувань;
- виконати програмну реалізацію експертної системи для організації тренувального процесу;
- здійснити проектування серверної частини інформаційної технології організації та проведення тренувального процесу;
- виконати програмну реалізацію серверної частини інформаційної технології організації та проведення тренувального процесу.

Об’єкт дослідження – процес аналізу даних та надання рекомендацій при організації та проведення тренувань.

Предмет дослідження – програмні засоби аналізу даних та надання рекомендацій при організації та проведенні тренувань.

Методи дослідження. У роботі використані такі методи наукових досліджень: аналіз аналогічних технологій та програмних рішень, що дозволяє визначити більшість проблем та труднощів, які необхідно вирішити в даній роботі. Методи та моделі подання знань, теорія побудови експертних систем, методи інженерії знань, методи та підходи до розробки веб-орієнтованих програмних застосунків, методи об’єктно-орієнтованого програмування.

Наукова новизна одержаних результатів. Запропоновано інформаційну технологію організації та проведення тренувального процесу, що відрізняється від існуючих застосувань удосконаленої інформаційної моделі із

комбінованим застосуванням нечіткої логіки та експертних систем, що забезпечило розширення функціональних можливостей.

Практичне значення одержаних результатів полягає в тому, що визначено основні етапи роботи інформаційної технології організації та проведення тренувального процесу, здійснено проектування інформаційної технології організації та проведення тренувального процесу, а також на основі проведених досліджень здійснено програмну реалізацію серверної частини зазначеної інформаційної технології.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням математичних методів під час доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими та збіжністю результатів математичного моделювання з результатами, що отримані під час впровадження розроблених програмних засобів.

Особистий внесок магістранта. Усі результати, наведені у магістерській кваліфікаційній роботі, отримані самостійно. У роботах, опублікованих у співавторстві, автору належать такі результати: [3] – інтеграція розробленої експертної системи до веб-застосунку за допомогою бібліотеки CLIPSCLRWrapper.dll; [4] – порівняльний аналіз ефективності роботи нечіткої інтелектуальної системи з використанням узагальненої структури та із застосуванням дерева нечіткого логічного виведення.

Апробація результатів роботи. Результати досліджень апробовані на XI-й Міжнародній науково-практичній конференції EUROPEAN SCIENTIFIC CONGRESS, 27-29 листопада (Мадрид, 2023) [3] та на Міжнародній науково-практичній конференції "Молодь в науці: дослідження, проблеми, перспективи (МН-2024)", Вінниця 2023 [4].

Публікації. За результатами досліджень опубліковано дві статті [3, 4] у збірниках матеріалів міжнародних науково-практичних конференцій.

1 АНАЛІЗ СУЧАСНОГО СТАНУ РОЗВИТКУ ТЕХНОЛОГІЙ ОРГАНІЗАЦІЇ ТА ПРОВЕДЕННЯ ТРЕНУВАЛЬНОГО ПРОЦЕСУ

1.1 Аналіз предметної області технологій організації та проведення тренувального процесу

Аналізуючи сучасний стан розвитку технологій організації та проведення тренувального процесу, можна спостерігати значний прогрес і інновації у цій галузі. Прогнозується стрімке зростання ринку застосунків для фітнесу, який досягне колосальної суми в 14,7 мільярдів доларів до 2026 року [5]. Такі дані свідчать не тільки про економічну привабливість цієї сфери, а й про переосмислення звичок ведення здорового способу життя, перетворюючи концепцію персонального тренера на зручну технологію з індивідуальним підходом до користувача, яку можна використовувати за потреби. Можна констатувати факт успішного розвитку технологій у сфері тренувань за рахунок прагнення людей до ведення здорового способу життя у сучасний спосіб.

У 2020 році технології вбудовані в одяг або у вигляді аксесуару посіли перше місце у світових трендах фітнесу [5]. Застосування датчиків та пристроїв для моніторингу фізичної активності та здоров'я є адаптацією суспільства до використання подібних технологій у сфері фітнесу. Вбудовані датчики в фітнес-трекерах, смарт-годинниках та спортивних пристроях дозволяють вимірювати серцевий ритм, кількість кроків, споживану калорійність та інші параметри. Ці дані допомагають тренерам та спортсменам аналізувати фізичну активність, встановлювати цілі та відстежувати прогрес, що допомагає поліпшити ефективність тренувань, відстежити прогрес та надавати індивідуальну підтримку спортсменам.

Одним з головних трендів у розвитку технологій організації тренувального процесу є використання мобільних застосунків та платформ для тренувань. Такі застосунки використовують сучасні технології фітнесу та напрацювання тренажерних залів, щоб надавати користувачам індивідуальні плани тренувань

та дієт, а також відслідковувати показники та результати при виконанні запропонованих планів. Відповідно до звіту Всесвітнього економічного форуму, опублікованого у вересні 2020 року, кількість щоденних активних користувачів, які використовують фітнес-застосунки, зросла на 24% з 1 по 2 квартал 2020 року. У 2020 році близько 84 мільйонів людей скористалися застосунками та платформами для тренувань в США [5]. Ці застосунки дозволяють створювати та відстежувати тренувальні програми, записувати прогрес, аналізувати дані та надавати індивідуальні рекомендації. Вони стають універсальним інструментом для тренувань, доступним на різних пристроях, і дозволяють спортсменам бути більш самостійними та організованими у своїх тренуваннях.

Крім того, використання віртуальної реальності (VR) та інтерактивних тренажерів також стає все більш популярним. VR-технології дозволяють створювати іммерсивне середовище для тренувань, де спортсмен може взаємодіяти з віртуальними сценаріями та об'єктами, що покращує мотивацію та залучення до процесу. Інтерактивні тренажери, такі як бігові доріжки з відеоекранами або симулятори різних видів спорту, допомагають створити реалістичні умови тренувань та покращити техніку спортсменів. Наприклад, для боротьби з масовою проблемою нудьги в тренажерному залі компанії поєднують властивості відеоігор, що викликають звикання та занурення, з перевагами фізичних вправ для здоров'я та фізичної форми. Віртуальна реальність використовує технології тренажерного залу, щоб утримувати мотивацію та зацікавленість користувачів завдяки своєму інтерактивному підходу. Програмне забезпечення Oculus Quest Fitness – це популярна фітнес-програма віртуальної реальності, яка використовує ігри та унікальні ефекти, щоб мотивувати вас виконувати різні тренування.

Отже, застосування технологій у організації та проведенні тренувального процесу відкриває нові можливості для спортсменів та тренерів. Вони поліпшують ефективність тренувань, надають індивідуальну підтримку та сприяють досягненню кращих результатів. Мобільні застосунки, віртуальна реальність та датчики фізичної активності є лише кількома прикладами

інновацій, що розширюють можливості тренувань. Враховуючи швидкий розвиток технологій, цей напрямок продовжуватиме зростати та еволюціонувати, сприяючи поліпшенню тренувальних процесів та досягненню високих результатів у спорті та фітнесі.

Крім того, слід зазначити, що зростання популярності технологій організації та проведення тренувального процесу відбувається не тільки в професійному спорті, але й серед звичайних людей. Більшість людей сьогодні мають доступ до смартфонів, планшетів та інших пристроїв, що відкриває їм можливість використовувати ці технології для покращення свого фізичного здоров'я та збільшення активності. Це сприяє популяризації здорового способу життя, спорту та фітнесу в суспільстві загалом.

1.2 Аналіз відомих програмних рішень та програм-аналогів

На ринку фітнесу існує велика кількість застосунків, які дозволяють користувачу скласти програму тренувань. Основною ціллю таких застосунків зазвичай є спрощення процесу тренування або навіть заміна порад реального тренера, покладаючись на роботу застосунку.

Застосунок FitnessAI [6] набув значної популярності та заявляє про використання штучного інтелекту для створення більш персоналізованих та ефективних програм тренувань для кожного користувача. Ця програма спочатку була призначена лише для тренажерного залу, але згодом було проведено ряд покращень, щоб задовольнити зростаючий попит на заняття спортом в домашніх умовах. FitnessAI оптимізує підходи побудови ланцюгів вправ, корегує кількість повторень та вагу додаткового навантаження для кожної вправи на основі інформації зібраної користувачами. За словами розробника, алгоритм цього програмного забезпечення був навчений на 5,9 мільйонах проведених тренувань. Користувачу надається зручний інтерфейс, який дозволяє контролювати процес виконання вправ, їх послідовність та усю потрібну для цього статистичну інформацію, як показано на рисунку 1.1.

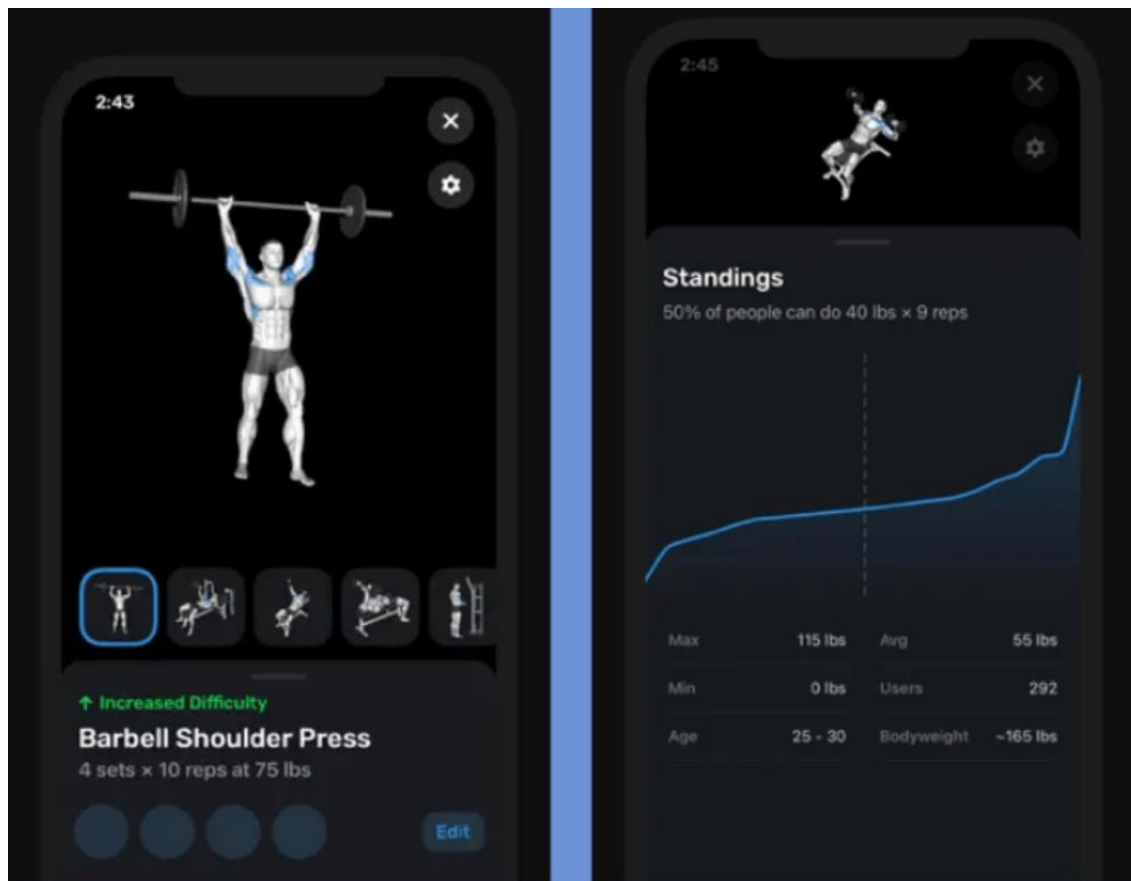


Рисунок 1.1 – Інтерфейс застосунку Fitness AI

До недоліків можна віднести підтримку лише однієї операційної системи iOS та недосконалу систему оцінювання програми тренувань. Штучний інтелект корегує комбінацію вправ для кожного тренування базуючись тільки на одній загальній оцінці всього тренування загалом. Такий спосіб не дає адекватних результатів, оскільки оцінювання підібраних для користувача навантажень та комбінації вправ потрібно розділити на декілька показників, які відображають окремі складові процесу тренувань. Таким чином, складена програма тренувань часто містить неправильний порядок виконання вправ. Наприклад, декілька вправ на одну групу м'язів підряд, тоді як правильніше складати порядок де чергуються групи м'язів.

Застосунок Fitbod [7] створює індивідуальні програми тренувань для своїх клієнтів, які змінюються з часом залежно від їхніх цілей та результатів. Він максимізує результати завдяки використанню кількох алгоритмів для зміни обсягу навантажень на основі попередніх виконаних вправ. Застосунок дозволяє

користувачу отримати індивідуальну програму тренувань, яка базується на його досвіді, типі фігури, цілях виконання фізичних вправ та наявності спортивного обладнання. Fitbod відображає набір вправ, а також підходи, повторення та додаткову вагу, як показано на рисунку 1.2. Завдяки цьому, користувачу не доведеться думати про те, що робити, коли він хоче провести тренування.

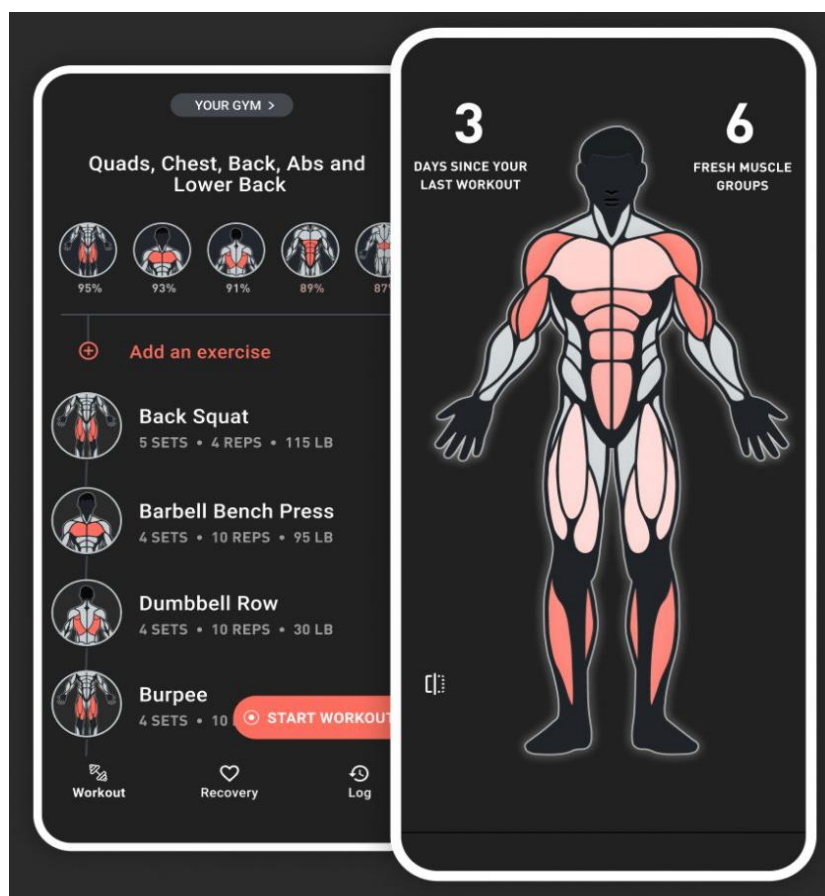



Рисунок 1.2 – Інтерфейс застосунку Fitbod

Основним недоліком такого застосунку є те, що система дозволяє користувачу створювати власні програми тренувань або змінювати уже створену програму будь-яким чином. Через це, застосунок не контролює ці зміни, тому користувач може зробити неефективні або безглузді програми тренувань, а система не повідомляє йому про це.


Hyperficient [8] – це унікальний веб-застосунок на основі штучного інтелекту, який допомагає максимізувати потенціал користувача під час


тренувань завдяки поєднанню штучного інтелекту та досвіду персонального тренера.


Розробники заявляють, що алгоритми машинного навчання побудовані на основі сучасних наукових досліджень та впливають на процес розробки персоналізованих програм тренувань, як показано на рисунку 1.3. Таким чином, користувач зможе досягати поставлених цілей більш ефективно. Hyperficient створює програму тренувань, адаптовану до різних факторів, таких як доступний час користувача, рівень попередньої підготовки та бажані цільові групи м'язів, щоб переконатися, що користувач не втомився від повторюваних одноманітних програм тренувань.

 **Tips**

- The 🟡 represents exercise optimality (rating out of 5)
- For each exercise do two warm-up sets, increasing in weight: first 8-12 reps, second 5-8 reps. If the muscle has already been worked, one warm-up set is fine
- Do the exercises in order — your programme is structured for optimal biomechanics

 Click the exercise name to open the page, and then the arrows ► next to each exercise for form tips and videos

 Programme

 **Your Programme**











 Exercise	 Day	 Sets	 Reps
 Shoulder Press	Day 1	2	5-8
 Incline Press	Day 1	2	5-8
 Fly	Day 1	1	6-10
 Preacher Curl	Day 1	2	8-10
 Lat Raises	Day 1	2	6-10
 Mid-position Curl	Day 1	1	6-10

Рисунок 1.3 – Персоналізована програма тренувань в застосунку Hyperficient

Від інших аналогічних рішень Hyperficient відрізняє те, що він має чіткий фокус на ефективності. Ідея полягає в тому, щоб забути про безцільні повторення та зосередитися на правильному виконанні вправ та програми тренувань. Веб-

застосунок оптимізує вибір і порядок вправ, зосереджуючись на таких факторах, як профіль опору, завантаженість та стабільність м'язових груп, одночасно дотримуючись принципів біомеханіки.

Hyperficient також пропонує інформаційну науково обґрунтованих тренувальних ідей, які допомагають користувачу зрозуміти ефективність вправ. Він також отримує поради щодо оптимізації способу життя, правильного харчування, кардіо-тренувань, сну та харчових добавок, що надзвичайно зручно для новачків у цій сфері. Для допомоги в техніці виконання вправ існують підказки щодо форми та навчальні відео про досконале виконання.

Існує ряд недоліків, які вказують на те, що розробники застосунку перебільшили з заявленими можливостями свого продукту. Основною проблемою є надмірна адаптація програми тренувань до вказаного користувачем доступного часу і попередньої підготовки. На рисунку 1.3 видно, що якщо вказати недостатньо доступного часу для тренування, або низький рівень фізичної підготовки – застосунок створить програму тренувань в якій буде 1 або 2 підходи на кожну вправу, що не відповідає сучасним науково доведеним дослідженням та рекомендаціям від професійних тренерів. Крім того, даний веб-застосунок не розраховано на розміри екранів мобільних пристроїв, тому ним незручно користуватись під час тренувань.

1.3 Аналіз об'єкту проектування технології організації та проведення тренувального процесу

Основною метою розробки інформаційної технології для організації та проведення тренувального процесу є надання кваліфікованих рекомендацій і готових рішень для складання індивідуальних програм тренування, які відповідають цілям та потребам користувача. Така технологія створює альтернативу дороговартісним персональним консультаціям у профільних спеціалістів, в галузі організації тренувального процесу.

Вимоги до інформаційної технології для організації та проведення тренувального процесу:

- формалізація антропометричних показників користувача;
- визначення індексу маси тіла;
- визначення базального рівня метаболізму;
- визначення відповідності параметрів фізичного стану здоров'я користувача до обраної цілі тренувань;
- створення програми тренувань з врахуванням параметрів фізичного стану здоров'я користувача та обраної цілі тренувань;
- база даних MS SQL Server для збереження інформації про основні сутності інформаційної технології;
- побудова Web API серверної частини за принципами REST для забезпечення продуктивності та масштабованості при взаємодії компонентів інформаційної технології.

1.3.1 Визначення ключових показників при складанні програми тренувань

Зазначимо найпоширеніші та всесвітньо прийняті, серед більшості організацій (в тому числі ВООЗ) і наукових спільнот, показники які використовуються при організації та проведенні тренувального процесу. Такими показниками є індекс маси тіла (ІМТ) та базальний рівень метаболізму (БРМ).

Індекс маси тіла (ІМТ, англ. BMI – body mass index) – величина, що дозволяє оцінити рівень відповідності маси людини та її зросту, дозволяючи тим самим приблизно визначити, чи є вага недостатньою, нормальною, надлишковою, або ж людина має ожиріння. Відомий бельгійський вчений Адольф Кетле запровадив формулу для розрахунку ІМТ, яку досі використовують повсюди, в тому числі й ВООЗ [9] у своїх рекомендаціях:

$$BMI = \frac{m}{h^2}, \quad (1.1)$$

де m – маса тіла (кг);

h – зріст (м).

Величина ІМТ відображає запаси жиру в організмі, і може вчасно сигналізувати про його надлишок, ризик розвитку ожиріння і пов'язаних із цим захворювань та ускладнень. Якщо індекс маси тіла є вищим за 25, то це вказує про наявність надмірної ваги, а його перевищення понад 35 – свідчить про ожиріння. В таблиці 1.1 представлено класифікацію ожиріння, за величиною ІМТ, згідно з рекомендаціями ВООЗ.

Таблиця 1.1 – Класифікація ожиріння, за величиною ІМТ, згідно рекомендацій ВООЗ.

Класифікація	Індекс маси тіла, кг/м ² (ІМТ за Кетле)	Ризик супутніх захворювань
Недостатня маса тіла	менше 18,5	Низький (підвищений ризик інших захворювань)
Нормальна маса тіла	18,5 – 24,9	Звичайний
Надлишкова маса тіла	25,0 – 29,9	Підвищений
Ожиріння I ступеня	30,0 – 34,9	Високий
Ожиріння II ступеня	35,0 – 39,9	Дуже високий
Ожиріння III ступеня	40,0 і більше	Надзвичайно високий

Звісно, ІМТ має свої недоліки та неточності, до яких можливо віднести:

- ІМТ не враховує індивідуальних антропометричних особливостей (таких як пропорції фігури, кількість м'язової маси), тому не може бути точним показником стану тіла;

- ІМТ не враховує різницю між жировою та м'язовою масою, а це означає, що величина ІМТ спортсменів та професійних атлетів може потрапляти у категорію зайвої ваги або ожиріння.

Але, не зважаючи на свої слабкі місця, ІМТ дозволяє відносно точно та швидко визначити людей у яких маса тіла знаходиться поза рамками нормального діапазону. Експерти, при створенні програм тренувань, у більшості випадків, враховують величину ІМТ. Даний показник дозволяє швидко оцінити і порекомендувати необхідну програму тренувань та скорегувати калорійність раціону людини. Наприклад, якщо ІМТ людини складає 28, і вона обрала своєю ціллю збільшення м'язової маси, то кваліфікований тренер порекомендує спочатку знизити масу тіла до нормальних значень, а вже потім переходити до програми збільшення м'язової маси.

Іншим поширеним показником є базальний рівень метаболізму [10] (БРМ). Базальний рівень метаболізму (БРМ, англ. BMR – Basal Metabolic Rate), або енергія основного обміну – це енергія, яка необхідна для життєдіяльності (метаболізму) організму, без фізичних навантажень. Величина БРМ залежить від багатьох факторів, зокрема статі, ваги, зросту та віку людини. Чим вища людина, і чим більша її вага – тим більше енергії необхідно для підтримання життєдіяльності, і навпаки. Існує багато методів для визначення величини БРМ, однак найчастіше експерти в галузі організації тренувального процесу використовують метод Харріса-Бенедикта. При обчисленні БРМ є дві формули, які обираються в залежності від статі людини.

Формула базального рівня метаболізму для жінок:

$$BMR_{\text{ж}} = (10 * m) + (6,25 * h(\text{cm})) - (5 * a) - 161, \quad (1.2)$$

де a – вік людини.

Формула базального рівня метаболізму для чоловік:

$$BMR_{\text{м}} = (10 * m) + (6,25 * h(\text{cm})) - (5 * a) + 5 \quad (1.3)$$

Наприклад, жінка вік якої складає 30 років, вага – 62 кг і зростом – 168 см, потребує для основного обміну речовин – 1659 кКал. Далі, в залежності від цілі людини та додавши орієнтовні енергетичні затрати на тренування, або іншу фізичну активність, можливо скоригувати добову калорійність раціону – зменшити, якщо необхідно позбавитись зайвої ваги, або збільшити, якщо необхідно збільшити м'язову масу, або тренуватись більш інтенсивно.

Експерт в галузі організації тренувального процесу, визначивши БРМ людини, зможе більш точно побудувати програму тренувань, зокрема:

- визначити необхідну інтенсивність тренування;
- визначити оптимальний набір вправ, обрахувавши енергетичні затрати на тренування;
- скоригувати ціль та структуру програми тренувань;
- скоригувати необхідний для тренувальної програми раціон.

Не зважаючи на те, що кількість досліджень і наукових публікацій на тему необхідності регулярних фізичних навантажень та тренувань значно збільшилась за останні роки – чітко визначеної кількості тренувань, їх тривалості, складу вправ, часу відпочинку між вправами та ще багатьох складових, необхідних для створення програми тренувань не визначено. Існує значна кількість рекомендацій, виявлених взаємозалежностей, статистичних даних, тобто накопичився значний обсяг емпіричних знань в даній галузі, але чітко сформованої та обґрунтованої наукової теорії – поки ще не існує. Зазначимо основні причини відсутності чітко визначених значень та складових для побудови універсальної, оптимальної програми тренувань, яку можливо було б застосувати для переважної більшості людей:

- варіабельність індивідуальних факторів. Кожна людина має унікальні антропометричні дані та характеристики, такі як вік, стан здоров'я, вага, зріст, рівень фізичної підготовки та інші фізіологічні та психологічні аспекти. Індивідуальні відмінності вимагають персоналізованого підходу до створення програми тренувань. Тому важко встановити універсальні стандарти, які підходять для всіх;

- різноманітність типів фізичної активності. Фізична активність може включати широкий спектр вправ і видів тренувань, таких як аеробні вправи, силові тренування, вправи на гнучкість та інші. Кожен тип фізичної активності може мати різні впливи на здоров'я та якість життя, і встановлення точних значень для кожного типу є складним завданням;
- різноманітність цілей. Люди мають різні цілі тренувань, такі як підтримка загального здоров'я, збільшення м'язової маси, поліпшення витривалості або зниження ваги. Кожна ціль вимагає інших підходів і може варіюватися за часом навантажень, кількістю тренувань та вправ;
- недостатній науковий прогрес. Попри значний обсяг досліджень, наукові дослідження у галузі тренувань та тренувального процесу продовжується. Нові відкриття, технології та підходи вносять зміни у розуміння оптимальних програм тренувань, але вони потребують подальшого дослідження, узагальнення та підтвердження;
- складність вимірювання та проведення досліджень. Визначення точних значень часу навантажень, кількості тренувань та вправ вимагає об'єктивних методів вимірювання. Існує багато методів оцінки ефективності фізичної активності, але вони можуть мати свої обмеження та недоліки;
- індивідуальна адаптація. Оптимальна програма тренувань повинна бути адаптована до реакції організму конкретної людини на навантаження. Реакція організму на тренування може бути індивідуальною, і вона може змінюватися з часом. Тому програма тренувань повинна бути гнучкою і враховувати індивідуальні потреби та зміни в тренувальному процесі.

1.3.2 Опис типових програм тренувань

Перед сучасними дослідниками, які працюють над питаннями визначення оптимальних програм тренувань та організації тренувального процесу існує досить складна задача по врахуванню усіх вищевказаних причин, при плануванні

та проведенні досліджень у цій галузі. Саме тому, сьогодні, при побудові оптимальних та ефективних програм тренувань, орієнтуються, як на ті наукові знання, які вже вдалось здобути, так і на залучення вагової кількості експертів. Саме експерти, які володіють значною кількістю емпіричних, перевірених на практиці знань – дозволяють частково компенсувати дефіцит наукових даних та інформації, при створенні програм тренувань. Поєднання дослідницьких, наукових знань та знань експертів дозволяє створити оптимальні програми тренувань для більшості людей, на даний момент. Ось чому найкращим рішенням є створення програми фізичних навантажень за допомогою експертної системи.

Якщо людина прийняла рішення, що їй необхідно розпочати регулярно давати своєму організму фізичні навантаження, то перше з чого розпочне переважна більшість, окрім безпосередньо пошуку місця для тренувань та пошуку всього необхідного для цього – це створення та формування програми тренувань. Це може бути як простий список вправ, які необхідно виконати за тренування так і детально розроблена програма тренувань з індивідуально підібраним раціоном харчування та погодинним режимом дня, графіком сну і т.д.

Програма тренувань (фізичних навантажень) являє собою індивідуально складений план виконання певної послідовності вправ, який спрямований на досягнення попередньо визначених цілей. Такими цілями можуть бути: зниження відсотка жиру в організмі, збільшення м'язової маси, підвищення фізичної витривалості організму та багато інших. Програми тренувань досить різноманітні та не мають єдиного стандарту, чи шаблону. Вони можуть варіюватись в залежності від виду фізичних навантажень, деталізованості, цілей, яких хоче добитись людина завдяки тренуванням та ще безлічі параметрів.

Опишемо як виглядає типова, найбільш розповсюджена та впізнавана, на сьогодні, програма тренувань. Спочатку визначається ціль програми тренувань. Наприклад, чоловік віком у 34 роки, що має вагу тіла 80 кг, прагне збільшити м'язову масу і зальну вагу тіла до 90 кг. Така програма подається у вигляді послідовності тижневих планів тренувань, з обов'язковими полями для

контролю та відстежування результатів тренувань, як наведено у таблиці 1.2. Це водночас необхідно для корегування кожного наступного тижневого плану тренувань і так повторюється до моменту виконання визначених людиною цілей.

Таблиця 1.2 – Типова програма тренування в щотижневому представленні (Тижневий план тренування «3-денний спліт»).

Тренувальний тиждень	Кількість Тренувань на тиждень					
	Тренування №1		Тренування №2		Тренування №3	
Тренувальний Тиждень №1	Грудні м'язи та Біцепс		Спинні м'язи та Тріцепс		Ноги та Плечі	
Виконана робота (загальна вага за тренування)	Загальна вага, кг.	Зміна Δ , %	Загальна вага, кг.	Зміна Δ , %	Загальна вага, кг.	Зміна Δ , %
	4924	=	5364	=	5420	=
Вага тіла:	80 кг			Зміна Δ , %		=
Тренувальний Тиждень №2	Грудні м'язи та Біцепс		Спинні м'язи та Тріцепс		Ноги та Плечі	
Виконана робота (загальна вага за тренування)	Загальна вага, кг.	Зміна Δ , %	Загальна вага, кг.	Зміна Δ , %	Загальна вага, кг.	Зміна Δ , %
	5004	+ 1,6	5484	+ 2,2	5560	+2,5
Вага тіла:	80,8 кг			Зміна Δ , кг		+ 0,8
...
Тренувальний Тиждень №24	Грудні м'язи та Біцепс		Спинні м'язи та Тріцепс		Ноги та Плечі	
Виконана робота (загальна вага за тренування)	Загальна вага, кг.	Зміна Δ , %	Загальна вага, кг.	Зміна Δ , %	Загальна вага, кг.	Зміна Δ , %
	6038	+ 0,2	6108	+ 0,5	6880	+0,4
Вага тіла:	90 кг			Зміна Δ , кг		+ 0,4

За основу створення тижневого плану тренувань, береться календарний тиждень, який розбивається на сім днів, розпочинається з понеділка та завершується неділею. Далі обираються, зазвичай 2-3 конкретні дні для тренувань, з проміжком один день для відпочинку, як показано в таблиці 1.3.

Таблиця 1.3 – Типовий тижневий план тренувань («3-денний спліт»).

Понеділок	Вівторок	Середа	Четвер	П'ятниця	Субота	Неділя
Тренування №1	День Відпочинку	Тренування №2	День Відпочинку	Тренування №3	День Відпочинку	День Відпочинку
Грудні м'язи та Біцепс		Спинні м'язи та Трицепс		Ноги та Плечі		
19:00		19:00		19:00		

Для кожного з тренувальних днів визначаються цільові м'язові групи та складається послідовність виконання вправ, з вказанням назви вправи, кількості повторень виконання кожної вправи, кількості повторень при виконанні кожної вправи та за необхідності додаткове навантаження (масу) на спортивних снарядах, що наведено в таблиці 1.4.

Зазвичай, план тренувань на наступний тиждень не змінюється і залишається за набором вправ та розпорядку тренувальних днів на тиждень таким самим. Єдині змінні, що корегуються – це кількість повторень виконання кожної вправи, кількість повторень при виконанні кожної вправи та додаткове навантаження (вага) на спортивному знарядді. Таким чином, кожний наступний тиждень відбуваються коригування тижневого плану за цими змінними, до моменту досягнення цілей програми тренування.

Таблиця 1.4 – Типовий план тренування на день (Тижневий план тренування «3-денний спліт», Тренування №1, Цільові м'язи: грудні м'язи та біцепс).

План тренування: Понеділок, 12.12.2023, Грудні м'язи та Біцепс, 19:00				
Порядок виконання вправ	Назва вправи	Кількість повторень виконання вправи (підходи)	Кількість повторень при виконанні вправи (повтори)	Додаткове навантаження (вага) на спортивному снаряді, кг.
1.	Жим штанги лежачи на лавці	4	6	80
2.	Жим гантелей на похилій лавці (кут нахилу 20°)	4	8	26
3.	Підйом штанги на біцепс стоячи	4	6	30
4.	Тяга нижнього блоку перед собою стоячи (з акцентом на внутрішню та верхню частину грудних м'язів)	3	8	24
5.	Концентрований підйом гантелі на біцепс, використовуючи лавку Скотта	3	8	18
6.	Розведення гантелей лежачи на лавці	3	8	16

Зазначимо основні сучасні підходи до організації тренувального процесу.

Самостійно скласти вищеописані програми тренувань, особливо, коли людина не має досвіду, або знань у сфері організації тренувального процесу – досить складно. Крім того існує значний ризик скласти неефективну, або ж навіть шкідливу програму. Тобто таку, що не дозволить досягти визначеної цілі тренувань, а можливо й значно збільшить ризик отримання травми, в процесі її виконання.

Станом на сьогодні, у людей які тренуються, або тільки планують розпочати регулярно тренуватись, існує чотири найпоширеніших підходи до організації власного тренувального процесу:

- інтуїтивний підхід. Базується на підборі вправ та розкладу тренувань емпіричним шляхом. Зазвичай не має системності та чітко визначеної цілі. Людина виконує вправи, які їй подобаються, тоді коли їй подобається;

- шаблонний підхід. Базується на виборі готових, вже кимось складених тренувальних програм. Вибір шаблону виконується за принципом найбільшого рівня кореляції готової програми з цілями та можливостями людини;
- підхід з використанням програмних продуктів та застосунків для створення програм тренувань. Якісно відрізняється від двох попередніх підходів, оскільки при побудові подібних програм тренувань використовуються особисті дані та показники людей, для яких ця програма буде створюватись, що своєю чергою робить її більш індивідуально налаштованою;
- консультативний підхід. Базується на делегуванні роботи по складанню, контролю та корегуванню програми тренувань – профільним спеціалістам у галузі організації тренувального процесу, тобто тренерам, лікарям-реабілітологам і т.д.

Потрібно зауважити, що часто люди комбінують використання одразу декількох підходів одночасно, з метою мінімізувати їх недоліки та максимізувати переваги. Запропонована експертна рекомендаційна система також комбінує два найефективніших підходи – підхід з використанням програмних продуктів та застосунків для створення програм тренувань та консультативний підхід. Порівняємо та визначимо переваги та недоліки кожного із підходів.

Виділимо основні переваги та недоліки інтуїтивного підходу. Можна виділити наступні переваги інтуїтивного підходу:

- простий та зрозумілий у використанні;
- не вимагає володіти знаннями у галузі організації тренувального процесу;
- не потребує додаткових фінансових витрат;
- на початковому етапі викликає більш позитивні враження та відданість тренуванням у людини, оскільки базується на підборі вправ, що подобаються та проведенні тренування тоді, коли подобається.

До недоліків інтуїтивного підходу можна віднести:

- низька ефективність подібних програм, оскільки через безсистемний та не ґрунтовний підхід – досить складно досягнути реалізації визначених цілей програми;
- може дати результат тільки людині без попереднього досвіду тренувань на початковому етапі (до 4-6 місяців) і залежить від генетичного потенціалу та індивідуальних здібностей;
- може призвести до виникнення проблем зі здоров'ям та надмірного ризику отримати травму. Наприклад, через не дотримання базових тренувальних принципів, підбору не коректних вправ, не збалансованого навантаження та не достатнього часу для відновлення організму – людина може перейти в стан «перетренованості». Подібний стан значно знижує якість рівня життя, оскільки викликає проблеми зі сном, відчуття апатії, депресивний стан, біль у суглобах і т.д.;
- за час використання подібного підходу до організації тренувального процесу, людина може сформувати негативні, шкідливі звички (наприклад кожне тренування намагатись підняти максимально можливу вагу) та напрацювати деструктивні навички, які буде досить складно змінити, або скорегувати у подальшому.

Визначимо переваги та недоліки шаблонного підходу. Перевагами шаблонного підходу є:

- зрозумілий у використанні, оскільки потребує тільки безпосередньо виконання вже складеної програми тренувань;
- доступність програм тренувань. За багато років накопичилась значна база програм, практично під більшість потреб, видів спорту, тощо. До того ж значна частина програм тренувань є у вільному доступі в інтернеті, профільних журналах, виданнях і т.д. А та не значна частина програм, що надається на платній основі, зазвичай має низьку ціну;
- значний об'єм накопиченої інформації про результати використання різних готових програм тренувань. Це можуть бути і відгуки про

ефективність даної програми, від людей, які вже виконали її, рекомендації та відповіді на питання по програмі тренувань;

- оскільки шаблонний підхід є найбільш популярним та масовим – то значна частина тренажерних залів, спортивних майданчиків та спортивних комплексів орієнтуються на покриття вимог більшості популярних готових програм тренувань при комплектації свого простору тренажерами та спортивним інвентарем. А це у свою чергу полегшує пошук необхідного для тренувань місця, або закладу при використанні шаблонного підходу;
- готові програми тренувань, які створені компетентними та авторитетними організаціями, дослідниками та експертами у галузі організації тренувального процесу є ефективними у більшості випадків, та задовольняють базові запити значної кількості людей при складанні першої програми тренувань.

Недоліками шаблонного підходу є:

- необхідність зважено та обдумано підходити до вибору готового плану тренувань. Досить часто, базуючись на власній суб'єктивній оцінці, людина може обрати програму тренувань яка їй не підходить на даний момент. Тобто обрати занадто складну та інтенсивну програму, переоцінивши власні можливості, або програму, яка буде давати не достатнє навантаження, щоб стимулювати ріст м'язів;
- необхідність перевіряти актуальність програми тренувань по відношенню до сучасного стану речей в галузі організації тренувального процесу. Існують випадки, коли програми тренувань, які складені ще у 90-х роках, рекомендують та публікують у сучасних засобах масової інформації. Тобто, помилки та не ефективні рішення, які приймалися в ті часи, по причині браку знань та належної інформації, продовжують рекомендувати, не зважаючи на сучасні результати досліджень та знання;
- необхідність перевіряти авторство готових програм тренування. У нинішній інформаційній ері люди перенасичені доступною інформацією і нажаль не завжди вона є достовірною. Тому трапляються випадки коли

люди використовують програми складені не експертами та фахівцями у галузі організації тренувального процесу, а некомпетентними людьми та організаціями. Це може призвести до не досягнення цілей програми тренувань, або проблем зі здоров'ям, набутих в процесі її виконання;

- готові програми тренувань складно підлаштувати та адаптувати до індивідуальних потреб людини. Наприклад, якщо програма передбачає тренування у понеділок, середу та п'ятницю, а у людини для цього є тільки вівторок, серeda та неділя. Або перелік вправ у програмі не можливо виконати через брак відповідного спорядження, тощо;
- відсутність індивідуального підходу до особливостей та антропометричних параметрів людини. Готові програми тренувань не можуть в повній мірі враховувати вагу, зріст, вік, стать конкретної людини та скоригувати програму під неї задля підвищення ефективності тренувань.

Виділимо основні переваги та недоліки підходу з використанням програмних продуктів та застосунків для створення програм тренувань. Можливо визначити наступні переваги даного підходу:

- можливість створювати індивідуально орієнтовані тренувальні програми, підлаштувати та корегувати їх відповідно до введених антропометричних параметрів людини;
- у переважній більшості випадків, програмні продукти та застосунки періодично оновлюються, що дозволяє приводити у відповідність та впроваджувати актуальні дослідження, методології та наукові відкриття в галузі організації тренувального процесу;
- програмні продукти та застосунки дозволяють вести записи та збирати дані тренувань для подальшого аналізу та коригування програми тренувань;
- програмні продукти та застосунки дають змогу інтегрувати гаджети спортивного та фітнес напрямлення для збору більш точної та деталізованої інформації за результатами тренування та іншої необхідної для уточнення інформації;

- більша доступність та надійність збереження інформації щодо програм тренувань;
- кращий рівень взаємодії з людиною, можливість оповіщення та інформування користувача програмного продукту, або застосунку;
- більшість програмних продуктів та застосунків дають змогу вносити зміни та адаптувати програму тренувань в процесі її виконання. Тобто є більш гнучким інструментом при створенні програм тренувань.

Визначимо основні недоліки підходу з використанням програмних продуктів та застосунків для створення програм тренувань:

- необхідність володіти відповідними пристроями та засобами для використання відповідних програмних продуктів та застосунків;
- деякі якісні програмні продукти та застосунки, у сфері організації тренувального процесу, реалізуються на платній, комерційній основі, або ж частина функцій застосунку є платною. Це значно знижує доступність даного підходу для певної групи людей;
- дизайн та реалізація деяких програмних продуктів та застосунків є не зовсім вдалою, з точки зору інтуїтивності, зрозумілості та зручності використання для користувача;
- залежність від коректності, введених користувачем, необхідних даних, для роботи програмних продуктів та застосунків;
- може не задовольнити вимоги та потреби професійних і висококваліфікованих спортсменів.

Виокремимо переваги консультативного підходу при створенні програм тренувань:

- висока точність та деталізованість створених програм тренування;
- індивідуальний підхід, врахування особливостей фізіології та антропометричних даних людини;
- можливість корегування та підлаштування програми тренувань в процесі її виконання;
- вищий рівень мотивації та заохочення до виконання програми тренувань;

- якісний та швидкий процес аналізу і створення рекомендацій по оптимізації тренувального процесу;
- спрощення і делегування частини обов'язків та відповідальності за виконання програми тренувань;
- зазвичай використовуються актуальні та найбільш ефективні сучасні методи організації тренувального процесу;
- наявність значної практичної бази знань за результатами виконання різних програм тренувань, що спрощує пошук та надання необхідних рекомендацій.

Зазначимо недоліки консультативного підходу при створенні програм тренувань:

- висока вартість та витрати. Даний підхід є найдорожчим з усіх зазначених, через необхідність залучення профільних спеціалістів;
- необхідність підлаштовувати власний графік з графіком роботи обраного експерта;
- необхідність пошуку профільного експерта;
- залежність від географічного розташування експерта та людини, оскільки не у кожному населеному пункті може знайтись відповідний профільний експерт та належне місце для проведення відповідного тренування;
- залежність від суб'єктивного погляду експерта. Експерт може мати сформовані упередження стосовно певних методів організації та проведення тренувального процесу, підбору вправ, навантаження тощо.

Наведемо приклад та опишемо детальніше вигоду від поєднання підходу з використанням програмних продуктів і застосунків для створення програм тренувань та консультативного підходу. Переважна більшість програмних продуктів і застосунків для створення програм тренувань – генерують програму тренувань, використовуючи певний готовий шаблон і наповнюють його в залежності від введених користувачем даних. Досить часто подібні згенеровані програми є дійсно ефективними та оптимальними для розв'язання задач користувача. Але спільною проблемою подібних програмних рішень є те, що,

вони можуть побудувати програму тренувань, базуючись на побажаннях людини, не враховуючи її доцільність та корисність, тобто побудують програму тренувань, яку хоче людина, а не програму тренувань, яка дійсно потрібна людині. Наприклад, людина, яка ніколи раніше не тренувалась, але сповнена мотивації та переоцінюючи власні сили, обирає в застосунку, що бажає інтенсивно тренуватись 6 днів на тиждень, щоб максимально швидко збільшити м'язову масу тіла.

Переважає більшість застосунків побудує користувачу програму тренувань, з розрахунком на 6 днів на тиждень і ціллю – збільшити м'язову масу. Якщо ж людина б звернулась до професійного тренера з проханням скласти подібну програму – то тренер би пояснив, що подібні інтенсивні тренування та великий тренувальний об'єм не тільки де будуть сприяти збільшенню м'язової маси, а ще й призведуть до виникнення проблем зі здоров'ям внаслідок надмірного фізичного навантаження та порекомендував би почати хоча б з 2-3 не інтенсивних тренувань на тиждень.

Експертна система дозволить поєднати доступність та зручність програмних рішень з експертними знаннями профільних спеціалістів в галузі організації тренувального процесу. Повертаючись до наведеного вище прикладу, якби при застосуванні експертної системи, користувач який ніколи раніше не тренувався, обрав, що бажає проводити інтенсивні тренування 6 днів на тиждень, щоб максимально швидко збільшити м'язову масу тіла – то система замість виконання запиту користувача і побудови тренувальної програми на 6 днів – виведе детальне пояснення, про шкідливість подібної програми та порекомендує програму тренувань на 3 дні для початківців. Тобто побудує програму, яка дійсно потрібна та дозволить досягнути цілі користувача.

1.4 Висновок до розділу 1

Отже, в першому розділі було проведено аналіз предметної області технологій організації та проведення тренувального процесу. Можна підкреслити бурхливий розвиток та економічну привабливість сфери фітнесу та ведення здорового способу життя. Особливої актуальності в цій сфері набули веб-застосунки та мобільні застосунки для організації та проведення тренувань. Визначено основну причину актуальності таких застосунків – надання індивідуальних планів тренувань та раціонів харчування з можливістю відслідковувати власні показники та результати. Таким чином, користувачі отримують зручність, самостійність та організованість під час своїх тренувань без дороговартісних персональних консультацій чи тренерів.

Проведений аналіз аналогів системи дозволив виявити переваги та недоліки існуючих рішень, які слід врахувати при розробці. На основі попередніх кроків було проведено аналіз об'єкту проектування технології організації та проведення тренувального процесу. В ході аналізу було визначено ключові показники при складанні програми тренувань, а саме індекс маси тіла та базальний рівень метаболізму. Крім того, проведено опис типових програм тренувань та охарактеризовано переваги та недоліки найпоширеніших підходів до організації тренувального процесу.

2 ПРОЕКТУВАННЯ ЕКСПЕРТНОЇ СИСТЕМИ ДЛЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОРГАНІЗАЦІЇ ТА ПРОВЕДЕННЯ ТРЕНУВАЛЬНОГО ПРОЦЕСУ

2.1 Класифікація експертних систем

Експертні системи – це інтелектуальні програмні системи, які моделюють та використовують експертні знання та процедури виведення для вирішення конкретних завдань чи проблем у певній предметній області [11]. Вирішенням проблеми у контексті експертних систем є рекомендації, які механізм логічного виведення генерує у відповідь на отримані факти.

Експертні системи класифікуються за кількома основними критеріями. Якщо розглянути структуру ЕС, то виділяють наступні:

- експертні системи на основі правил (Rule-Based Expert Systems). Вони використовують базу знань у вигляді правил «якщо-то», де «якщо» – це умова, а «то» – це дія або рекомендація. Система обчислює результати на основі цих правил;
- мережеві експертні системи (Neural Expert Systems), що поєднують елементи нейромережі з експертними знаннями для прийняття рішень. Мережеві експертні системи можуть навчатися на даних та покращувати свої рішення з часом.

З точки зору основного методу роботи ЕС вони поділяються на:

- системи на основі фреймів (Frame-Based Systems), які організовані навколо понять-фреймів. Фрейми містять інформацію про об'єкти або ситуації в предметній області;
- системи на основі правил, які використовують правила виду «якщо-то» для прийняття рішень;
- системи на основі басових мереж довіри (Bayesian Networks), які використовують ймовірнісні моделі та статистику для оцінки ймовірностей різних подій та прийняття рішень;

- експертні системи на основі машинного навчання (Machine Learning-Based Expert Systems). Ці системи використовують алгоритми машинного навчання для модифікації та вдосконалення експертної системи на основі даних.

Крім того, експертні системи поділяють за предметною областю їх застосування, адже кожна сфера людської діяльності сильно ускладнюється в міру технічного прогресу та вимагає конкретних специфічних знань. Зазначимо основні сфери застосування:

- медичні експертні системи, що використовуються для діагностики та лікування різноманітних медичних станів;
- фінансові експертні системи, що допомагають приймати фінансові рішення, такі як інвестиції та управління інвестиційним портфелем;
- експертні системи в інженерії, що використовуються для проектування, діагностики та управління складними технічними системами та їх конкретними аспектами;
- експертні системи управління, що використовуються для автоматизації та оптимізації управління процесами в підприємствах та організаціях;
- експертні системи в освіті, що допомагають у навчанні та навчальному процесі, створюючи індивідуальні навчальні програми.

Одним з основних критеріїв класифікації ЕС є тип задачі, що вирішується. Загалом системи, що оперують знаннями можуть вирішувати задачі аналізу та синтезу. При розв'язанні задачі синтезу множина рішень будується поетапно на основі рішень компонентів або підпроблем, наприклад при проектуванні чи плануванні. При розв'язанні задачі аналізу множина рішень перераховується та включається в систему, наприклад при інтерпритації даних чи діагностиці. Також існують комбіновані задачі, такі як навчання, моніторинг та прогнозування [12]. Наведемо класифікацію експертних систем за типом задачі, що вирішується:

- моніторинг, основною задачею якого є зосередженість на постійній інтерпретації даних і сигналізації про те, чи знаходяться певні параметри в

прийнятних межах. Двома основними проблемами є пропуск тривожної ситуації та обернений зв'язок із помилковим спрацьовуванням. Труднощі у виявленні цих проблем пов'язані з розмитістю ідентифікації тривожних сигналів;

- інтерпретація даних, що виконує важливу операцію по визначенню значення даних, оскільки вимагає послідовних і правильних результатів. Зазвичай вважається, що при цьому виконується багатофакторний аналіз даних. Це звичне завдання для ЕС;
- діагностика. Процес діагностики передбачає виявлення будь-яких проблем із системою. Проблемою вважається відхилення від заданої норми. Ця інтерпретація дозволяє аналізувати несправності обладнання в технічних системах, природні аномалії та хвороби живих організмів з єдиної теоретичної точки зору. Ідентифікація функціональної структури діагностичної системи є особливо важливою в такій задачі;
- проектування, що передбачає запис процедур побудови для створення об'єктів з наперед визначеними властивостями. Ця специфікація охоплює всі відповідні документи, включаючи пояснювальні записки, креслення і т.д. Ключовими проблемами в цьому випадку є отримання точного структурного опису знань про об'єкт. Ефективна організація проектування вимагає розгляду як прийняття рішень, так і обґрунтування рішень. Таким чином, в задачах проектування відбуваються два тісно пов'язані процеси – прийняття рішень і процес пояснення;
- прогнозування. Системи прогнозування логічно обчислюють ймовірні наслідки за конкретних обставин. Як правило, система прогнозування використовує динамічну параметричну модель середовища, де параметри налаштовуються відповідно до вимог. Результати роботи цієї моделі служать основою для ймовірнісних прогнозів;
- планування, що передбачає виявлення планів дій для об'єктів, які можуть виконувати певні функції. Моделі поведінки реальних об'єктів

використовуються в таких ЕС для логічного виведення наслідків запланованої діяльності;

- навчання. Використання навчальних систем може виявити помилки в при вивченні будь-якої дисципліни та запропонувати відповідні рішення. Накопичуючи інформацію про гіпотетичного учня та його характерні недоліки, вони можуть визначити недоліки в знаннях своїх учнів і застосувати відповідні методи для їх усунення. Для передачі знань ці системи планують спілкування з учнем відповідно до його прогресу.

Отже, експертну систему для організації та проведення тренувального процесу можна класифікувати за типом задачі, що вирішується до задачі планування; за предметною областю їх застосування – експертна система фітнесу в поєднанні з медичними експертними системами; за основним методом роботи – система на основі правил.

2.2 Критерії розробки експертних систем

Перед інформаційною технологією організації та проведення тренувального процесу стоїть задача створення програми тренувань з врахуванням параметрів фізичного стану здоров'я користувача та обраної цілі тренувань. Фізичний стан здоров'я оцінюється на основі антропометричних показників користувача та за допомогою визначення індексу маси тіла і базального рівня метаболізму.

В теорії експертних систем визначають найбільш придатні для застосування ЕС задачі [12]:

- задачі прогнозу і класифікації, для яких притаманна велика кількість допустимих рішень;
- задачі спрямовані на обробку недостовірної інформації;
- задачі, що вимагають серйозний професійний досвід;
- задачі з великою кількістю параметрів, що необхідно враховувати та для яких складно або неможливо визначити аналітичні залежності.

З огляду на вищевказаний перелік, поставлена задача має велику кількість допустимих рішень, потребує відносно тривалу професійну підготовку та вимагає врахування великої кількості параметрів. Перед початком створення ЕС, необхідно перевірити інші важливі критерії розробки, такі як доцільність, виправданість та можливість.

Критерії доцільності розробки ЕС, що наведені на рисунку 2.1, описують на що повинне опиратись вирішення поставленої задачі щоб розробка ЕС виявилася доречною.



Рисунок 2.1 – Критерії доцільності розробки ЕС

Таким чином, вирішення поставленої задачі не повинне обмежуватись чистими розрахунками та обчисленнями по формулам, а має використовувати операції з символами, включаючи логічний аналіз та перебір варіантів. Крім того, задача не повинна мати чіткого алгоритмічного представлення та базуватись на евристичних методах, які призводять до отримання результату. Евристики дозволяють відійти від методу перебору всіх можливих варіантів вирішення, економлячи при цьому час, однак вони не гарантують виведення оптимального рішення.

Інші критерії доцільності визначають, що задача повинна мати кінцеву і прийнятну розмірність, щоб бути не надто складною для вирішення за допомогою комп'ютера, а також бути не тривіальною та зацікавлювати своїм способом знаходження рішення, яке неможливо або дуже складно отримати класичними методами.

Критерії виправданості розробки ЕС, що наведені на рисунку 2.2, не обов'язково всі повинні виконуватись – достатньо хоча б одного.

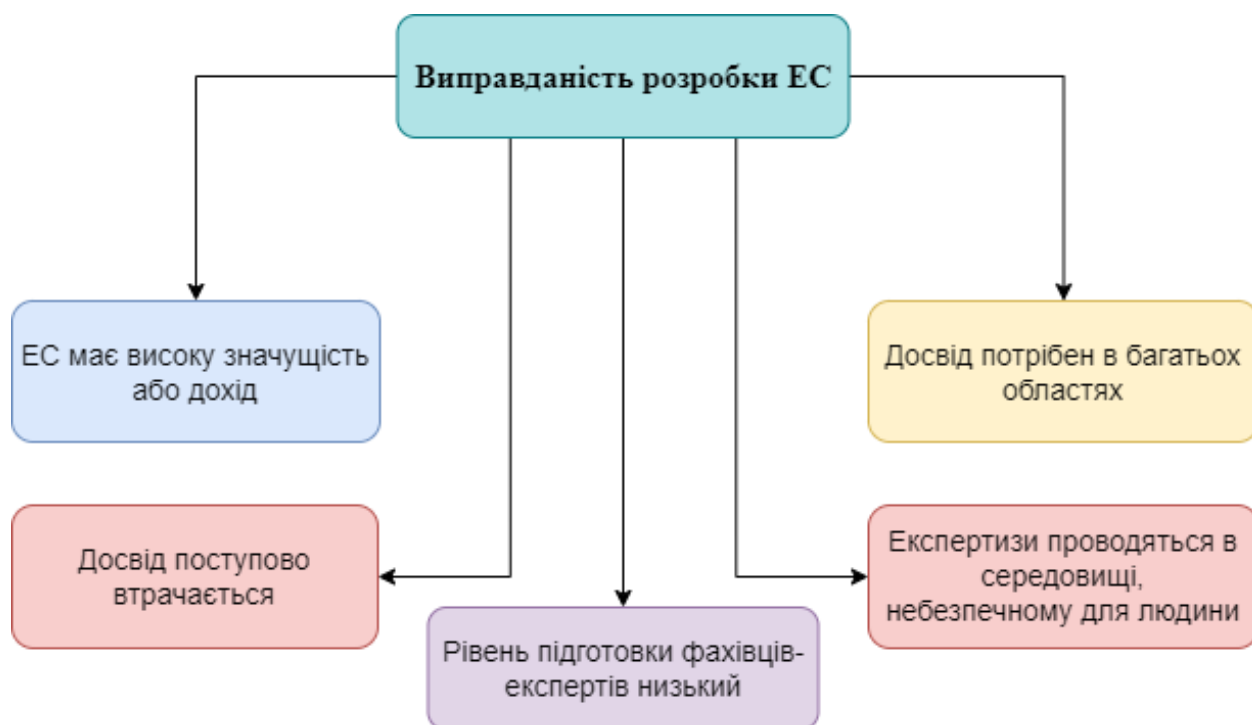


Рисунок 2.2 – Критерії виправданості розробки ЕС

Вважається, що експертну систему виправдано розробляти коли очікується високий рівень значущості чи доходу від вирішення поставленої задачі, коли вміння розв'язувати задачі з даної предметної області поступово втрачається і необхідно зберегти накопичений експертами досвід. Крім того, виправдано розробляти ЕС коли фахівці в певній галузі не мають належної підготовки, накопичений досвід необхідний для багатьох сфер людської діяльності або у випадку коли робота людини проводиться в небезпечному середовищі.

Важливими критеріями є можливість розробки ЕС. На рисунку 2.3 наведено критерії, від яких залежить можливість розробки ЕС, причому необхідне виконання усіх наведених умов.

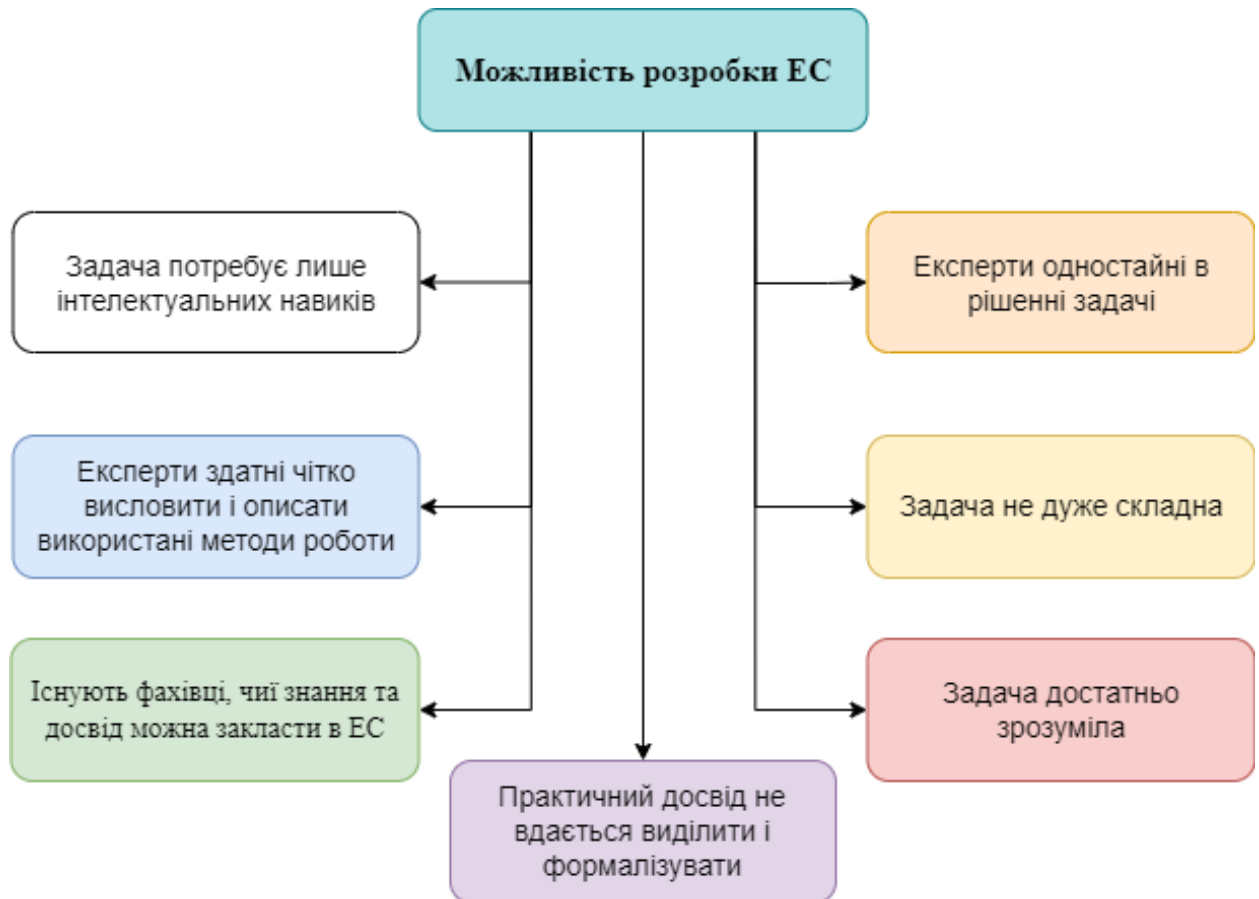


Рисунок 2.3 – Критерії можливості розробки ЕС

Для того щоб розробка ЕС вважалась можливою, необхідно мати експертів, які володіють потрібними знаннями та досвідом, щоб закласти їх в ЕС. Крім того, експерти повинні бути спроможні чітко охарактеризувати та пояснити методи, які використовувались для вирішення поставленої задачі, причому у рішеннях експертів не повинно бути розбіжностей.

Для виконання поставленої задачі необхідні лише інтелектуальні навички, причому задача не повинна бути надто важкою, адже коли для вирішення доводиться розробляти спеціальні методи чи унікальні підходи – ЕС в таких випадках не застосовують. Крім того, якщо вирішення задачі потребує знань,

здорового глузду та практичного досвіду, який з певних причин важко або неможливо виділити і формалізувати – то розробка ЕС вважається можливою.

Дослідивши основні критерії розробки ЕС, можна оцінити доцільність застосування експертної системи в інформаційній технології організації та проведення тренувального процесу.

Традиційні програми зазвичай мають визначену розробником послідовність кроків з використанням алгоритмічних методів для знаходження оптимального рішення. Тоді як, найбільш притаманними особливостями інтелектуальних програм є символна обробка даних, використання евристичних методів, неявне визначення кроків розв'язання та близькі до оптимальних шукані рішення [11].

Створення програми тренувань для кожної конкретної людини відбувається на основі її антропометричних показників, віку, статі, доступної кількості часу, та вподобань щодо цілі тренувань. Завдяки такому різноманіттю факторів, які впливають на результуючу програму тренувань, неможливо розробити універсальний науково доведений алгоритм. А точно ідентифікувати найкращу програму фізичних навантажень серед величезної кількості альтернативних варіантів неможливо, оскільки наразі це найчастіше досягається емпіричним шляхом та експертною думкою профільного спеціаліста.

Враховуючи такі особливості та значну кількість критеріїв, яку необхідно врахувати при формуванні програми фізичних навантажень, використання евристичних методів дозволить набагато простіше і практичніше знайти рішення за прийнятний час. Тому інтелектуальні програми більше підходять для рішення поставленої задачі.

Інтелектуальна програма для створення індивідуального плану тренувань оперує знаннями, що сконцентровані на конкретну предметну область – організацію та проведення тренувального процесу. Причому ці знання мають певну організацію, а окремі відомості пов'язані одне з одним, що дозволяє вивести рішення проблеми. Виходячи з цього, така інтелектуальна програма

відноситься до класу експертних та її доцільно розробляти для інформаційної технології організації та проведення тренувального процесу

2.3 Обґрунтування вибору моделі подання знань

Модель подання знань відіграє вирішальну роль в галузі експертних систем. Це пов'язано з «мисленням» експертних систем та сприяє їх розумній поведінці. В експертних системах подання знань – це систематичний спосіб опису на машинному рівні того, що знає експерт у певній предметній галузі. Після отримання знань їх слід зберігати в базі знань для подальшого використання при логічному виведенні. База знань зазвичай складається з правдивих фактів за визначенням. Структура фактів може варіюватися від простого стверджувального речення до складної онтології. Спосіб, яким організовано збереження фактів в базі знань, істотно впливає на механізми виведення, які можна застосувати до збережених знань. Тому необхідно ретельно підійти до вибору моделі подання знань, щоб полегшити процес створення бази знань та вибору відповідного механізму виведення.

Моделі подання знань зазвичай діляться на чотири класи [13]:

- моделі на основі логіки (logic-based);
- семантичні мережі;
- моделі на основі фреймів (frame-based);
- продукційні моделі.

Логічні моделі є складовою формальних систем, які забезпечують формальні мови та дедуктивні системи для опису знань і виведення відповідно. Семантичні мережі використовують більш природний підхід на основі графів до представлення знань. В даному випадку вузли графів відповідають поняттям, а дуги графів означають відношення між поняттями. Найбільш поширеними видами семантичних мереж є: доказові, імплікаційні, виконувані, мережі потоків даних, мережі, що навчаються, мережі для опису понять та гібридні мережі [14]. Фреймові моделі використовують об'єктно-орієнтований підхід, де знання

структуровані навколо поняття «об'єкт». Наявність об'єкта дозволяє групувати пов'язані властивості і процедури, тому спрощує опис і керування знаннями. Аркуш паперу також характеризує особливий вид фреймів – скрипти, які використовуються для опису стереотипної послідовності події в певному контексті [15]. Продукційні моделі представляють знання як набір правил «якщо-то» і забезпечені методами автоматизованого або напівавтоматичного виведення. Експертні системи є найвідомішою реалізацією цього підходу.

Останні дослідження проводяться в області онтологічної інженерії та теорії типів. Онтологія, як техніка побудови баз знань, поєднує в собі вищезгадані моделі представлення знань і має багато активних реалізацій, включаючи Protege, Cys, Dublin Core тощо.

Розглянемо особливості моделей на основі логіки, а саме логіку предикатів (логіку першого порядку). Логіка першого порядку (FOL) — це набір формальних систем, які складаються з формальної мови для опису тверджень та з дедуктивної системи виведення [16]. Існує багато варіацій логіки першого порядку, таких як інтуїтивна, багатосортна, модальна та класична, яку і буде розглянуто. Логіка першого порядку – це стандарт для формалізації математики в аксіоми, а також вона відіграє вирішальну роль у представленні знання майже будь-якого роду. Вона з'явилась, як спадкоємець пропозиційної логіки, в якій немає достатньої виразності, щоб відобразити знання реального світу. Основна відмінність між логікою першого порядку та пропозиційною логікою полягає в основному в онтологічній прихильності. Логіка першого порядку, як і природна мова, передбачає, що світ містить об'єкти, відношення між ними та функції, тоді як пропозиційна логіка оперує лише фактами. Логіка першого порядку корисна для відображення логічних зв'язків та їх значень. Крім того, вона дозволяє розділити логічний вислів на слова. Наприклад, якщо розглянути два логічних твердження «Будинок є математиком» та «Гаус є математиком». У пропозиційній логіці ці твердження розглядаються як непов'язані, тоді як у логіці предикатів, дозволено ввести предикат «Математик (x)», який явно показує про відношення між цими висловлюваннями.

Формальна мова FOI складається з синтаксису та семантики. Синтаксис визначає, які колекції символи з алфавіту є дозволеними виразами, тоді як семантика визначає значення цих виразів. Алфавіт - це набір символів, з яких можуть бути сформовані речення формальної мови. У логіці першого порядку символи алфавіту поділяються на дві групи: логічні та нелогічні. Відмінність полягає в тому, що логічні символи завжди мають одне і те ж значення, тоді як значення нелогічних символів залежить від тлумачення. Логічні символи зазвичай представлені кванторами (\forall , \exists), логічними сполучниками (\wedge , \vee , \rightarrow , \leftrightarrow , \neg), розділовими знаками (дужками), змінними (нескінченна множина), рівністю ($=$). Нелогічні символи складаються з предикатів та функціями з валентністю (кількістю аргументів) більшою або рівною нулю. Предикат валентності 0 відповідає пропозиційній змінній, а функція валентності 0 називається константою. Правила формування визначають два типи дозволених синтаксичних виразів у логіці першого порядку, якими виступають терми та формули. Терми представлені змінними та функціями, тоді як формули складаються з предикатів (приймаючи терми як аргументи), символів рівності (працює з термами), логічних сполучників та кванторів. Змінна у формулі може бути зв'язаною або вільною (або обома одночасно), залежно від того, чи знаходиться вона у межах квантора чи ні. Формула без вільних змінних називається реченням першого порядку. Речення першого порядку – це формули, які матимуть чітко визначені значення істинності під час інтерпретації, тому їх використовують для подання знань; решта просто підтримує синтаксичний механізм. Семантика логіки першого порядку забезпечується інтерпретацією, яка визначає область дискурсу та позначення нелогічних символів [16].

Виведення в логіці першого порядку виконується в рамках дедуктивної системи, яка використовується, щоб показати що одна формула є логічним наслідком іншої формули. Більшість дедуктивних систем складаються з логічних аксіом і правил виведення, які можна використовувати для виведення теорем системи. Ці системи мають важливу особливість – вони є чисто синтаксичними, тобто похідні можна перевірити незважаючи на тлумачення. У результаті кожна

похідна є синтаксичним наслідком усіх виразів що їй передують. Існує багато дедуктивних систем для логіки першого порядку, включаючи систему Гільберта, натуральну дедукцію, секвенційне числення, метод таблиць і т. д. Дедуктивна система називається обґрунтованою, якщо вона створює лише логічно дійсні формули, тобто які гарантовано є логічним наслідком. Дедуктивна система називається повною, якщо вона гарантовано створює всі логічно достовірні формули. Обґрунтованість і повнота відносяться до семантики придатної для мови системи дедукції. На жаль, немає автоматизованого процесу виведення для логіки першого порядку, що може бути як обґрунтованим, так і повним, оскільки існує невирішувана проблема визначення чи є одне речення логічним наслідком інших речень [17].

Логіка першого порядку є хорошою відправною точкою для подання знань та виведення, оскільки вона має багато металогічних властивостей, яких не мають сильніші логіки. Тим не менш, така логіка також страждає від значних обмежень. Першим обмеженням є парадокс Сколема, який передбачає, що нескінченні структури (тобто дійсна лінія, натуральні числа) не можуть бути категорично аксіоматизовані в логіці першого порядку, ця проблема вирішується в сильніших логіках, таких як логіка другого порядку. По-друге, логіка першого порядку недостатня для вираження деяких типових особливостей природної мови, наприклад квантифікація над властивостями, що вимагає наявності квантифікатора над предикатами. Нарешті, обчислювальна складність цієї логіки є одним з основних факторів, які спонукають до розгляду різних інших моделей подання знань та виведення [17].

Розглянемо модель подання знань через семантичну мережу. Це техніка представлення знань, яка використовує графову нотацію для опису структури знань. Згідно з цими концепціями, екземпляри та властивості відповідають вузлам графа, тоді як зв'язки між ними (тобто таксономія) визначаються через спрямовані дуги. Формально семантична мережа визначається як кортеж $\langle I, R_1, R_2, \dots, R_n, M \rangle$, де I означає інформаційні сутності, R_1, R_2, \dots, R_n позначає типи зв'язків між інформаційними об'єктами, а M відповідає карті, яка встановлює

зв'язки між суб'єктами інформації [13]. Різноманітність природи елементів кортежу та доступними операціями над ними дозволяє виділити наступні найбільш поширені види семантичних мереж: доказові, імплікаційні, виконувані, мережі потоків даних, мережі, що навчаються, мережі для опису понять та гібридні мережі [14]. Незважаючи на значні відмінності в інформаційних об'єктах і мережевих застосунках, усі вони використовують декларативний графічний підхід до подання знань.

Існують різні види семантичних мереж, тому розглянемо їх більш детально. Мережі для опису понять використовуються для опису понять, екземплярів і відносин між ними. Вони активно використовують відношення типу «є чимось», «має щось», «екземпляр чогось» для формування таксономії знань, яка є важливою для механізму зовнішнього виведення. Приклад такої мережі наведено на рисунку 2.4. Доказова мережа графічно представляє логічні пропозиції, тому її елементи виконують додаткові семантичні ролі, які відповідають логічним операторам, таким як екзистенційний квантор, кон'юнкція, диз'юнкція та інші. Імплікаційні мережі є окремим випадком логічної семантичної мережі, які використовують імплікацію як основне відношення для з'єднання вузлів; інші відношення можуть бути вкладені в пропозиційні вузли, але вони ігноруються процедурами виведення. Виконувані мережі містять механізми, які можуть викликати деякі зміни в самій мережі. Найбільший загальними виконуваними механізмами є передача повідомлень, приєднані процедури та перетворення графів. Мережі що навчаються коригують свої внутрішні представлення у відповідь на нову інформацію, щоб ефективно працювати в своєму середовищі. Таку мережу можна налаштувати трьома способами: механічна пам'ять, зміна ваги та реструктуризація. Гібридні мережі – це мережі, які одночасно включають кілька типів семантичних мереж, щоб повторно використовувати переваги та усунути недоліки вищезазначених мереж.

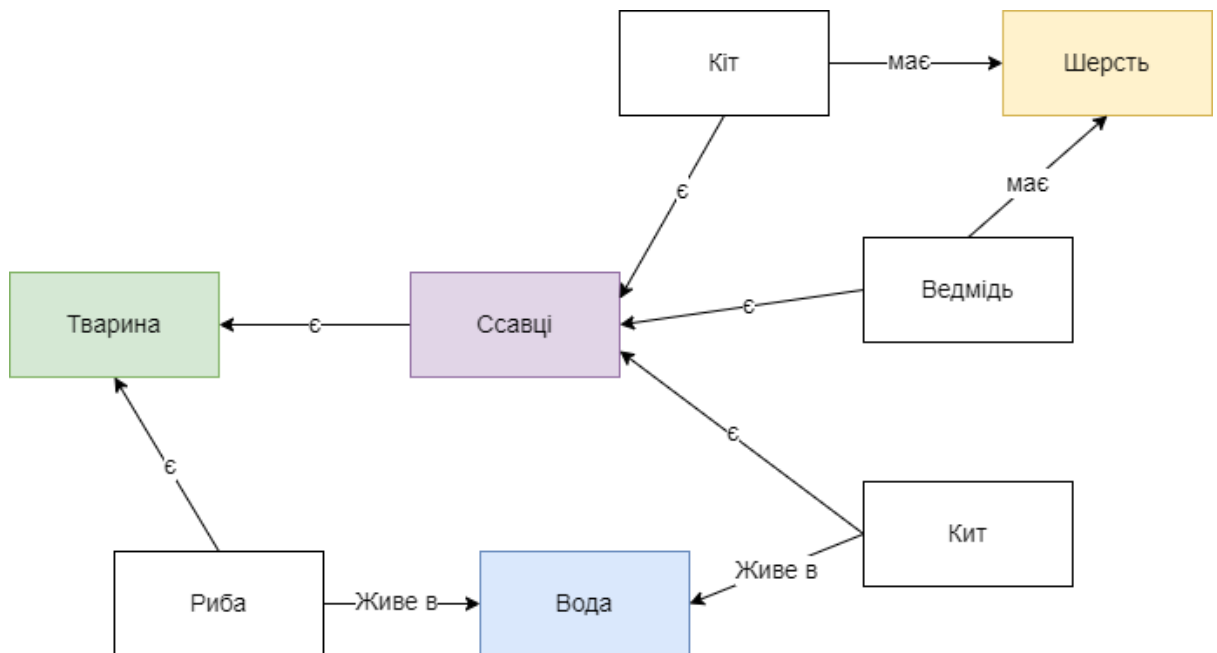


Рисунок 2.4 – Приклад семантичної мережі для опису понять

Серед основних переваг семантичних мереж – легкість сприйняття концепції знань людиною, що надається графічним відображенням. Спочатку графічна нотація показує прямі зв'язки між пов'язаними поняттями, тоді як лінійна нотація повинна спиратися на повторювані появи змінних або назви, щоб показати однакові зв'язки. По-друге, графічні нотації часто мають евристичну цінність у допомозі людині дослідити чи виявити шаблони, які було б важко або неможливо побачити в лінійній формі. Якщо говорити про недоліки семантичних мережах важливо підкреслити відсутність стандартизації значень вузлів та дуг, що створює складність для їх подальшої підтримки. Крім цього, деякі типи семантичних мереж страждають від конкретних проблеми, таких як множинні наслідування помилок, зв'язки «один до багатьох», відсутність формальної семантики, змішування різних рівнів абстракції. Тим не менш, природний підхід до подання знань робить семантичні мережі широко розповсюдженою технікою в інженерії систем знань.

Розглянемо таку модель подання знань як фрейми. Фрейм – це об'єктно-орієнтована форма подання знання, яка опирається на об'єкти, властивості, значення та відношення між ними [17]. В наслідок цього його часто називають представленням «об'єкт-властивість-значення». Об'єкти є природним способом

організації знань про фізичні об'єкти і ситуації. Важливо те, що об'єкт дозволяє групувати процедури для визначення його власних властивостей, а також властивостей інших об'єктів. У виразах логіки першого порядку опис світу стає розкиданим, тому важко контролювати послідовність та зрозумілість описаних знань. Об'єктно-орієнтоване представлення може бути досягнуто шляхом злиття багатьох властивостей об'єкта одного типу в одну структуру.

У системах на основі фрейму об'єкт, який містить властивості, заповнений значеннями, що відповідають «фрейму» який містить «слоти», заповнені даними. Тому фрейм можна визначити як структурне та реляційне представлення об'єкта. Кожен фрейм має назву та складається з набору слотів із відповідними даними. Дані можуть приймати значення примітивних типів (число, рядок, дата тощо), складних типів (набір, список) або служити посиланням на інший фрейм. Залежно від природи об'єкта розрізняють два типи фреймів – індивідуальні та узагальнені. Індивідуальний фрейм представляє єдиний об'єкт, який є екземпляром деякої категорії, тоді як узагальнений фрейм використовується для опису абстрактної категорії. Отже, окремі фрейми за замовчуванням мають спеціальний слот під назвою «Екземпляр чогось», де даними виступають ім'я або вказівник на інший відповідний узагальнений фрейм. У той же час, узагальнені фрейми можуть мати спеціальний слот під назвою «Є чимось» з даними, які вказують на більш узагальнений фрейм. Наявність цих спеціальних слотів має вирішальне значення, оскільки вони дозволяють реалізувати механізм успадкування [17].

Процес виведення зазвичай контролюється за допомогою процедур, які пов'язані зі слотами в узагальнених фреймах. Існує два типи процедур – «ЯКЩО ПОТРІБНО» та «ЯКЩО ДОДАНО». Перший виконується, коли дані слоту відсутні, а значення потрібне; другий виконується, коли є дані слотів які можуть вплинути на інші фрейми. Найпоширенішими випадками використання цих процедур є: надання значення за замовчуванням для порожніх слотів, вираження обмежень між слотами, підтримка узгодженості. Процедури та дані більш узагальненого фрейму можуть застосовуватись до більш конкретного фрейму

через успадкування. Виведення в системах на основі фреймів зазнають значного впливу від цих процедур і зазвичай мають наступний порядок:

- користувач створює новий фрейм, щоб оголосити існування якогось об'єкта чи ситуації;
- дані слотів успадковуються за можливістю;
- успадковані процедури «ЯКЩО ДОДАНО» виконуються, спонукаючи ініціалізацію та заповнення даних у інших фреймів;
- якщо користувач або будь-яка процедура вимагає даних для слотів, де дані відсутні, то виконується успадкована процедура «ЯКЩО ПОТРІБНО», що потенційно може викликати додаткові дії.

Звичайно, подання знань через фрейми має свої переваги та недоліки. Основні переваги походять від структурних особливостей. Об'єктно-орієнтований підхід дозволяє групувати властивості та процедури пов'язані з однотипними об'єктами, що спрощує процеси виведення та підтримки узгодженості. Таким чином, дані слотів, які вказують на інші фрейми, роблять їх неявно пов'язаними один з одним, зберігаючи кластеризацію пов'язаних знань. Нові властивості та відношення можна легко додати через нові слоти, тоді як пов'язані процедури можуть поширювати дані за замовчуванням для введених слотів до існуючих фреймів. Крім того, процедури роблять похідні (обчислені) властивості такими, що виглядають явно, що дозволяє потенційно уникнути непотрібних обчислень. Фрейми дозволяють як декларативний, так і процедурний контроль процесу виведення, таким чином зменшуючи загальні зусилля, необхідні для створення системи. Елегантна реалізація механізму успадкування підкреслює виразність фреймового підходу. Серед основних недоліків можна виділити: відсутність формального механізму виведення, відсутність стандартів для даних у слотах, залежність від сприйняття користувача – фрейм, що представляє один і той же об'єкт, може значно відрізнятися для двох різних користувачів.

Остання модель, яку буде розглянуто – продукційна модель подання знань. Правила продукції, також відомі як правила «ЯКЩО-ТОДІ», є одними з

найбільш широко розповсюджених форм подання знань, особливо в експертних системах та автоматизованих системах підтримки прийняття рішень. Вони характеризуються високою модульністю, що забезпечує гнучкість процесу ініціалізації та підтримки бази знань [11]. Кожне правило визначає відносно невелику та незалежну частину знань. Крім того, їх можна легко додавати та видаляти незалежно від інших правил. Простота синтаксису та природний спосіб вираження знань забезпечує просту інтерпретацію та легкість розуміння. Типове продукційне правило складається з антецеденту та консеквента. Антецедент відповідає посилю (умова, IF-clause), а консеквент означає дію (висновок, THEN-clause). Щоб увімкнути механізм виведення, правила зазвичай групуються в так звану продукційну систему, яка має власні структурно-функціональні особливості.

Продукційна система складається з бази знань і механізму логічного виведення, звідси слідує два типи правил – декларативні та процедурні. У декларативних правилах зазначаються всі факти та відношення щодо проблеми, тоді як процедурні правила виражають судження щодо того, як вирішити проблему, враховуючи що певні факти відомі. Відрізняються вони і місцем збереження: декларативні зберігаються в базі знань, а процедурні – включені до механізму виведення. База знань продукційної системи має свою специфіку і поділяється на робочу пам'ять та базу правил. Робоча пам'ять складається з набору фактів, які в основному є твердженнями про предметну область, але також можуть представляти різні структури даних [12]. Синтаксис і подання фактів не є строго формалізованим, тому вони можуть відрізнятися в різних системах. База правил – це сформований набір правил, які належать до спеціального типу правил «ЯКЩО-ТОДІ», де антецедент представляється кон'юнкцією умов, а консеквент відображає послідовність дій. Структура випадкового запису в базі правил визначається за допомогою формули $p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow a_1, a_2, \dots, a_k$. Члени антецеденту можуть мати форму заперечних або не заперечних предикатів та умов на певний об'єкт або кілька об'єктів. Консеквент зазвичай складається з наступних дій: «ДОДАТИ» факт до робочої

пам'яті, «ВИДАЛИТИ» факт з робочої пам'яті, «ЗМІНИТИ» поле атрибута, «ЗАПИТ» користувача на введення.

Механізм виведення у продукційних системах зазвичай реалізується за допомогою методу прямого ланцюжка. Він починається з фактів, наявних у робочій пам'яті, і використовує набір «активних» правил з бази знань, щоб отримати більше фактів, доки не буде досягнуто мети. Правило вважається активним, якщо його антецедент був задоволений, це необхідна умова для виконання консеквенту. Поширена ситуація, коли одночасно діють кілька правил, тому необхідно вказувати порядок їх виконання. Насправді, порядок виконання важливий, оскільки дії «ДОДАТИ» та «ВИДАЛИТИ» можуть змінити набір активних правил, щоб наступне активне правило в черзі могло бути дезактивовано попереднім. Стратегія вибору правила виконання з числа можливих кандидатів називається вирішенням конфлікту.

Інша проблема, яка виникає при виведенні в продукційних системах, пов'язана з процесом об'єднання. Деякі правила можуть вимагати великої кількості об'єднань, особливо якщо вони містять змінні для тестування всіх випадків, коли правила активні. Крім того, умови багатьох правил можуть збігатися, що призводить до багаторазових повторів одних і тих самих об'єднань. Щоб вирішити проблему неефективного об'єднання та проблему конфліктів був введений алгоритм Rete. Основні принципи його функціонування наступні: правила компілюються в мережу, яка об'єднує умови кількох правил разом, таким чином уникаючи дублювання; поширюються тільки правильні об'єднання; лише змінені умови оцінюються повторно.

Отже, правила продукції забезпечують велику модульність і гнучкість подання знань. Вони виразно та повноцінно описують знання, прості для розуміння та підтримуються потужним механізмом логічного виведення. Крім того, їх корисність вже доведена успіхом великих комерційних експертних систем. Говорячи про недоліки, важливо сказати, що не всі знання можуть бути виражені у формі правил. Правила продукції слабо відображають структурований опис знань, і важко дотримуватися ієрархії у порівнянні з

семантичними мережами. Великі системи зазвичай складаються з тисяч правил, що ускладнює закріплення знань і значно знижує продуктивність механізму виведення. Однак, перелічені недоліки можна вважати несуттєвими в межах інформаційної технології організації та проведення тренувального процесу, тому вирішено використовувати продукційну модель подання знань для розробки експертної системи.

2.4 Структуризація знань для створення програми тренувань

Важливим етапом проектування ЕС є структуризація знань. Зібравши великий обсяг знань про створення програми фізичних навантажень, необхідно визначити рівень деталізації системи. Структура отриманих знань визначається переліком основних об'єктів чи понять, що поєднані певними відношеннями, а також задається структура вхідної та вихідної інформації. Об'єкти предметної області мають набір характерних ознак, а ці ознаки мають свою область значень.

В результаті створюється ієрархічна система знань експерта, яка містить точну, повну та несуперечну множину об'єктів предметної області. На рисунку 2.5 наведено структуру вхідних факторів, від яких залежить результат логічного виведення ЕС. Такі фактори як «Вікова група», «Зріст» та «Маса» формують проміжні поняття «БРМ» - базальний рівень метаболізму та «ІМТ» - індекс маси тіла. Ці проміжні фактори впливають на те, які саме цілі тренувань підходять для користувача, враховуючи його фізичні показники. А це, в свою чергу, впливає на програму тренувань, яка задовольняє потреби та особливості користувача. Крім того, значною мірою впливає обрана користувачем ціль тренувань та доступна для нього кількість тренувань на тиждень.

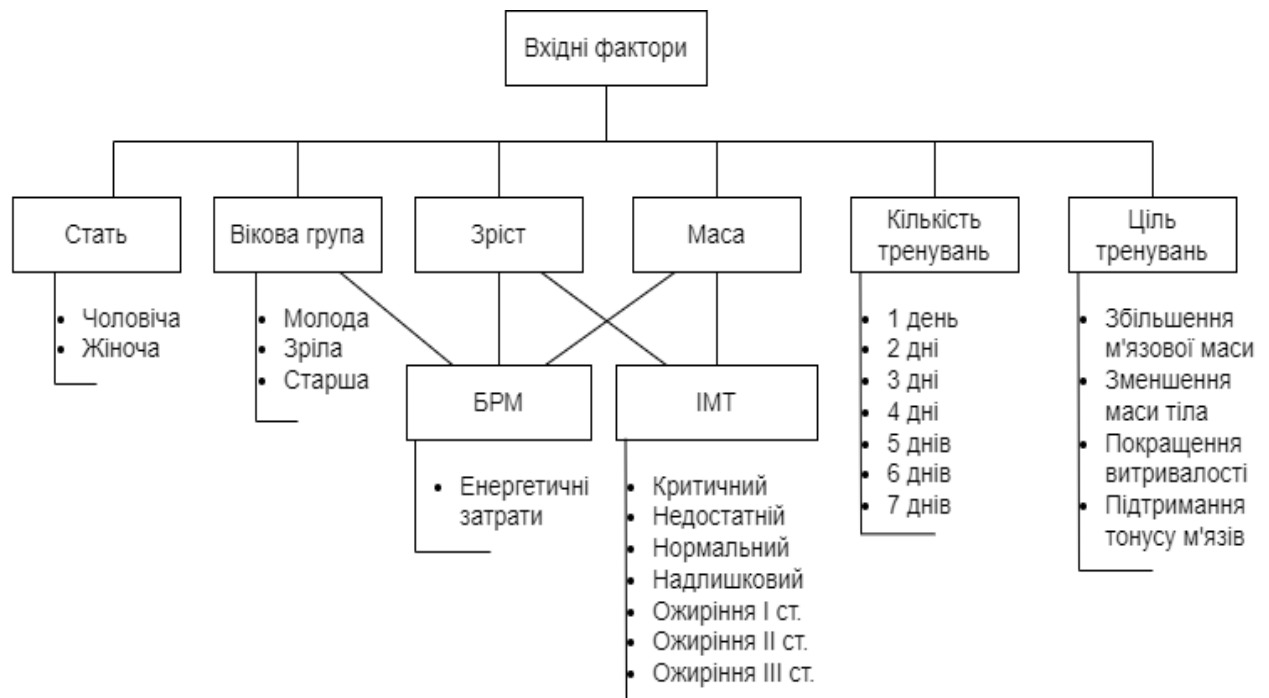


Рисунок 2.5 – Структура вхідних факторів

Структура вихідних факторів, що зображена на рисунку 2.6, демонструє залежність виведення ЕС від ІМТ, цілі тренувань, кількості тренувань на тиждень та вікової групи користувача. Виведенням є програма фізичних навантажень, що розрахована на тиждень. На рисунку 2.6 результуючі програми записані у вигляді «Програма - <ціль тренувань> - <вікова група> - <кількість днів>» для скорочення форми запису. Тобто, «Програма-ЗММ-М-1д» означає програму тренувань для збільшення м'язової маси для молодшої вікової групи розраховану на один день на тиждень. Можна помітити, що для будь-якої цілі тренувань максимальна кількість днів в програмі тренувань складає 6 днів для молодшої вікової групи та зрілої вікової групи. Для старшої вікової групи встановлено обмеження в 4 дні, оскільки з віком відновлювальні можливості людини помітно знижуються і немає сенсу тренуватись більше, адже це може призвести до стану перетренованості, що приносить шкоду для здоров'я людини, особливо старшої вікової групи.

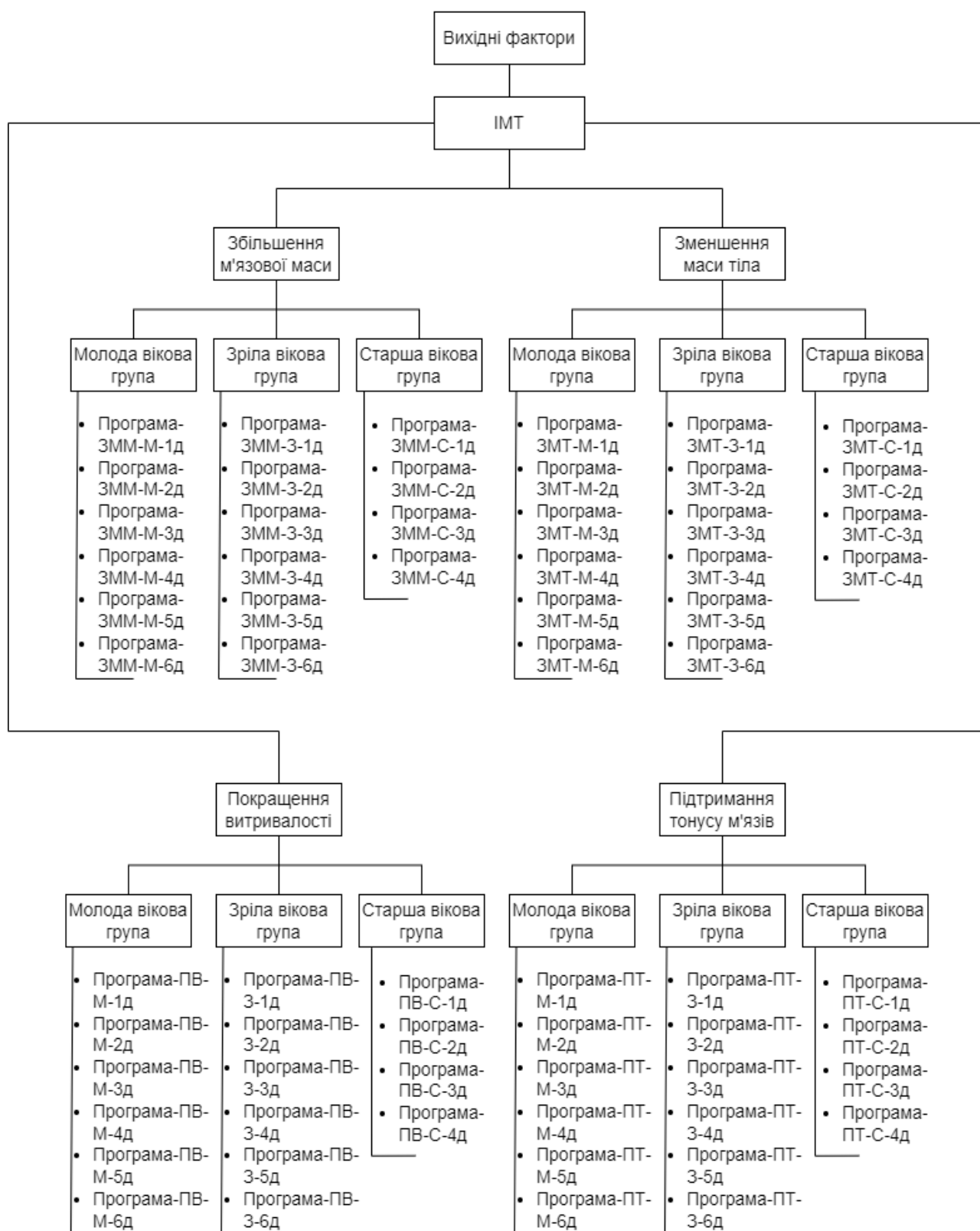


Рисунок 2.6 – Структура вихідних факторів

Якщо ввести структуру предметної області з таблиці в базу знань більшості продукційних оболонок, то можна отримати експертну систему, яка, отримуючи інформацію з відповідей користувача на питання про масу тіла, ціль програми

тренувань, вік, інтенсивність тренувань, калорійність раціону та кількість доступних для тренування днів (на тиждень) – порекомендує програму тренувань з вказанням кількості тренувань на тиждень. Функціональна структура демонструє модель прийняття рішення та міркувань, що використовує експерт при вирішенні поставлених перед ним задач. Згодом дана структура може бути формалізована за допомогою коротких правил «ЯКЩО-ТО», або у вигляді семантичних мереж. Зобразимо функціональну структуру предметної області у формі таблиці 2.1.

Таблиця 2.1 – Функціональна структура створення програми фізичних навантажень.

Маса тіла (ІМТ)	Ціль програми тренувань	Вікова група	Інтенсивність тренування	Калорійність раціону	Кількість тренувань (на тиждень)
Критично низька маса тіла	Збільшення м'язової маси тіла	Молода	Висока	Збільшення калорійності раціону	2,3,4
			Помірна		5,6
Зріла		Висока	2,3		
		Помірна	4,5		
		Низька	6		
		Старша	Помірна		1,2
Низька			3,4		
Нормальна маса тіла		Покращення витривалості, або Підтримання тону м'язів	Молода		Висока
	Помірна			4,5	
	Зріла		Висока	1,2	
			Помірна	3,4	
			Низька	5	
	Старша		Помірна	1	
			Низька	2,3	
	Надлишкова маса тіла		Зменшення ваги та зниження кількості жиру в організмі	Молода	Висока
Помірна	4,5,6				
Зріла	Висока	2			
	Помірна	3,4			
	Низька	5,6			
Ожиріння II ступеня	Старша	Помірна		1	
Ожиріння III ступеня		Низька		2,3,4	

Побудуємо декілька правил згідно даних функціональної структури:

ЯКЩО ІМТ = Недостатня маса тіла

І ціль програми тренувань = Збільшення м'язової маси тіла

І вікова група = Молода

І інтенсивність тренування = Помірна

І калорійність раціону = Збільшення калорійності раціону

ТО кількість тренувань на тиждень 5 АБО 6.

ЯКЩО ІМТ = Надлишкова маса тіла

І ціль програми тренувань = Зменшення ваги та кількості жиру в організмі

І вікова група = Зріла

І інтенсивність тренування = Висока

І калорійність раціону = Зниження калорійності раціону

ТО кількість тренувань на тиждень 2.

2.5 Висновок до розділу 2

В даному розділі наведено класифікацію експертних систем кількома основними критеріями. В результаті, експертну систему організації та проведення тренувального процесу було класифіковано за типом задачі, що вирішується до ЕС з задачею планування. За предметною областю застосування, дану ЕС визначено як система фітнесу в поєднанні з медичними експертними системами. Також проведено класифікацію за основним методом роботи та визначено, що дана експертна система відноситься до систем на основі правил.

Було оцінено доцільність розробки експертної системи за основними критеріями. Поставлену задачу охарактеризовано як таку, що має велику кількість допустимих рішень, потребує відносно тривалу професійну підготовку та вимагає врахування великої кількості параметрів. Відповідно до цієї задачі та критеріїв розробки дана ЕС виявилася доречною, виправданою та можливою для застосування в інформаційній технології організації та проведення тренувального процесу.

На етапі обґрунтування вибору моделі подання знань було досліджено основні чотири класи подання знань, а саме: моделі на основі логіки, семантичні мережі, моделі на основі фреймів та продукційні моделі. Розглянуто принципи

побудови знань для перелічених моделей та особливості роботи і застосування механізмів логічного виведення. Враховуючи усі переваги та недоліки кожної моделі, вирішено використовувати продукційну модель подання знань для розробки експертної системи для інформаційної технології організації та проведення тренувального процесу.

Крім того, було проведено процес структуризації знань, який дозволив створити ієрархічну систему знань експерта, що складається з повної та несуперечної сукупності об'єктів предметної області. В результаті, було розроблено структуру вхідних та вихідних факторів, де отримані знання визначаються переліком основних об'єктів чи понять, що поєднані певними відношеннями. Причому ці об'єкти предметної області володіють набором атрибутів, що мають свою область значень. А розроблена функціональна структура передала способи прийняття рішень, які використовує експерт при розв'язанні поставлених перед ним задач.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ЕКСПЕРТНОЇ СИСТЕМИ ДЛЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОРГАНІЗАЦІЇ ТА ПРОВЕДЕННЯ ТРЕНУВАЛЬНОГО ПРОЦЕСУ

3.1 Обґрунтування вибору середовища для розробки експертних систем

Розглянемо оболонку «Decision Support Expert-Shell». Вона дозволяє створити експертну систему продукційного типу, яка містить основні складові: робочу пам'ять, базу знань та механізм виведення.

База знань містить правила-продукції, факти, дозволені значення та питання. Правила-продукції подаються у вигляді: «ЯКЩО <умова> ТО <дія>».

В межах створення програми фізичних навантажень було задано такі дозволені значення:

дозволено(протипоказання)=Так, Ні
 дозволено(стать)=чоловік, жінка
 дозволено(вік)=18-25, 25-35, 35-45, 45-60, 60+
 дозволено(вага)=40-50, 50-60, 60-70, 70-80, 80-90, 90-100, 100-110, 110+
 дозволено(зріст)=155-165, 165-175, 175-185, 185-195, 195+
 дозволено(ціль_трен)=Позбавитись_зайвої_ваги, Бути_в_тонусі, Покращити_витривалість, Збільшити_силу_та_м'язи
 дозволено(кількість_трен)=1, 2, 3, 4, 5, 6, 7

Розглянемо фрагмент бази знань, який містить питання:

питання(протипоказання)=чи є у вас медичні протипоказання фізичним вправам?
 питання(стать)=якої ви статі?
 питання(вік)=вказіть віковий діапазон під який ви підпадаєте:
 питання(вага)=вказіть у якому діапазоні знаходиться ваша вага(кг)?
 питання(зріст)=вказіть у якому діапазоні знаходиться ваш зріст(см)?
 питання(ціль_трен)=яка основна ціль ваших тренувань?
 питання(кількість_трен)=скільки разів на тиждень ви можете тренуватись?

Для того щоб врахувати усі зазначені дозволені значення було створено 1161 правило-продукцію. Серед них 41 правило охоплює обчислення індексу

маси тіла користувача (ІМТ), що проходить консультацію. Розглянемо одне з правил для визначення ІМТ:

```

правило9: якщо
протипоказання=ні ІІ
вага=50-60 ІІ
зріст=175-185
то
ІМТ=недостатня_вага.

```

На основі ІМТ, цілі тренувань, віку, статі та бажаної кількості тренувань на тиждень генерується рекомендація, в якій описано план тренувань для досягнення цілі. Розглянемо декілька правил для визначення рекомендації:

```

правило50: якщо
протипоказання=ні ІІ
стать=Чоловік ІІ
вік=18-25 ІІ
ІМТ=недостатня_вага ІІ
ціль_трен=Бути_в_тонусі
то
рекомендація=При_недостатня_вага_рекомендовано_обрати_ціль_Збільшити_силу_та_м'язи.

```

```

правило437: якщо
протипоказання=ні ІІ
стать=Чоловік ІІ
вік=45-60 ІІ
ІМТ=надлишкова_вага ІІ
ціль_трен=Позбавитись_зайвої_ваги ІІ
кількість_трен=4
то
рекомендація=для_позбавитись_зайвої_ваги_проводити_4_тренування_за_тиждень,_з_них_2*Кардіотренування+2*Силових_тренування_та_зменшити_добову_калорійність.

```

```

правило955: якщо
протипоказання=ні ІІ
стать=Жінка ІІ
вік=45-60 ІІ
ІМТ=недостатня_вага ІІ
ціль_трен=Покращити_витривалість
то

```

рекомендація=ціль_Покращити_витривалість_не_рекомендовано_для_вашої_вікової_групи_Спробуйте_Бути_в_тонусі .

При проходженні консультації користувач вводить ціль «рекомендація» або «ІМТ». Далі користувачу необхідно відповісти на питання експертної системи за допомогою дозволених значень. В результаті система логічного виведення демонструє рекомендацію. На рисунку 3.1 наведено вікно фактів, де можна побачити результат консультації.

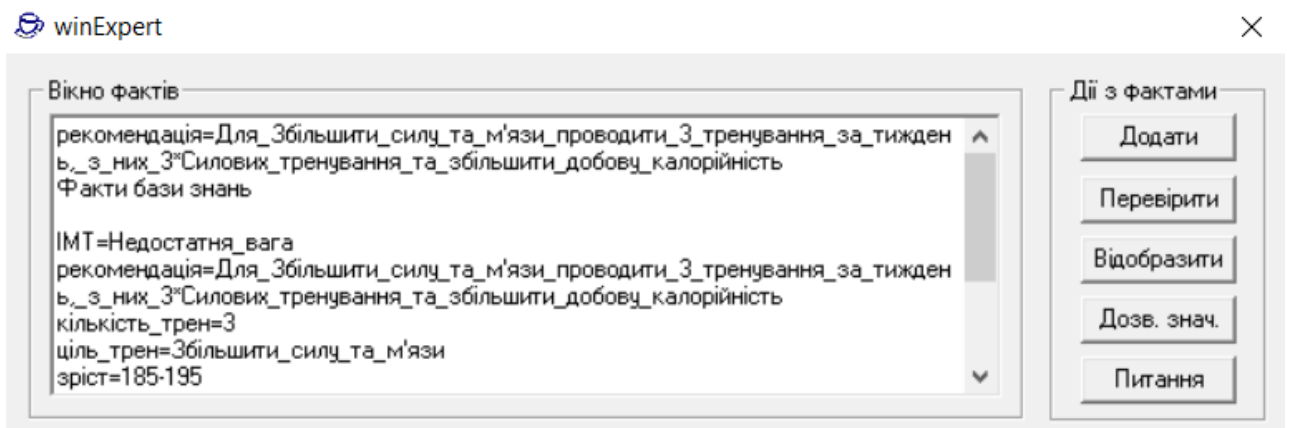


Рисунок 3.1 – Результат консультації у вікні фактів

Можна зробити висновок, що продукційна експертна система розроблена в оболонці «Decision Support Expert-Shell» не дозволяє повноцінно врахувати усі фактори при створенні програми фізичних навантажень. Збільшення кількості факторів тягне за собою збільшення кількості дозволених значень, що в свою чергу сильно збільшує кількість необхідних правил-продукцій для досягнення цілі. При великій кількості правил дуже важко перевіряти несуперечність нових правил та їх недетермінованість.

Розглянемо особливості розробки експертної системи в оболонці «Мала ЕС». Дана оболонка дозволяє створювати експертні системи, використовуючи логічне виведення на основі баєсового підходу. Користувачу пропонується пройти консультацію для отримання програми фізичних навантажень. Всього база знань містить 17 запитань та 28 різних рекомендацій по вибору програми тренувань.

Важливою особливістю є те, що інформація з якою працює експертна система не є точною, тобто задана з певною ймовірністю. Користувач може бути невпевнений в абсолютній істинності чи хибності свого свідчення при відповіді на запитання, тому можна давати відповідь з деяким ступенем впевненості. Після завершення консультації система формує виведення у виді імовірностей настання результатів і використовує для цього оцінку правдоподібності деяких передумов, одержуваних від користувача.

На рисунку 3.2 наведено перелік фактів отриманих від користувача під час проходження консультації. Також можна побачити обрані ним ступені впевненості до кожного факту.

1	КУ (-5)	Вам більше 46 років?
2	КУ (-2)	Чи хотіли б Ви і чи є у Вас можливість тренуватись 5 разів на тиждень?
3	КУ (-2)	Оцініть наскільки Ви худі і вам не вистачає ваги?
4	КУ (3)	Оцініть чи є у Вас зайва вага?
5	КУ (-2)	Оцініть чи Ви у хорошій фізичній формі?
6	КУ (-4)	У Вас є протипоказання від інтенсивних фізичних навантажень?
7	КУ (0)	Чи є у Вас необхідність збільшити силові показники та об'єми м'язів?
8	КУ (-5)	Чи хотіли б Ви і чи є у Вас можливість тренуватись 6 разів на тиждень?
9	КУ (-2)	Наскільки Вам подобаються кардіотренування?
10	КУ (5)	Чи є у Вас необхідність позбавитись зайвої ваги?
11	КУ (4)	Чи хотіли б Ви і чи є у Вас можливість тренуватись 4 рази на тиждень?
12	КУ (2)	Наскільки Вам подобаються силові тренування?
13	КУ (4)	Чи хотіли б Ви і чи є у Вас можливість тренуватись 3 рази на тиждень?
14	КУ (-2)	Чи хотіли б Ви і чи є у Вас можливість тренуватись 2 рази на тиждень?
15	КУ (-2)	Наскільки Ви фізично витривалий(ла)?
16	КУ (-5)	Чи хотіли б Ви і чи є у Вас можливість тренуватись 1 раз на тиждень?

Рисунок 3.2 – Перелік фактів отриманих від користувача

Ймовірності настання результатів визначаються за допомогою правила Баєса для обчислення апостеріорної умовної ймовірності $p(H|E)$ гіпотези H за умови, що відбулася подія свідчення E .

$$p(H|E) = \frac{p(E|H)}{p(E)}, \quad (3.1)$$

де $p(E)$ – безумовна (апріорна ймовірність свідчення E);

$p(E|H)$ – спільна ймовірність свідчення E і гіпотези H .

За результатами пройденого опитування, логічне виведення експертної системи виділило найбільш ймовірну програму тренувань для користувача, як показано на рисунку 3.3.

Експертна система для підбору оптимальної програми тренувань на тиждень
Виконали: Шелестюк Богдан та Шелестюк Максим

(0.99812)	Програма схуднення: 4 дні на тиждень - 2 високointенсивних кардіотренування 40 хвилин + 2 високointенсивних силових тренувань 40 хвилин
(0.99029)	Програма схуднення: 3 дні на тиждень - 2 високointенсивних кардіотренування 40 хвилин + 1 високointенсивне силове тренування 40 хвилин
(0.98733)	Програма схуднення: 5 днів на тиждень - 3 інтенсивних кардіотренування 40 хвилин + 2 високointенсивних силових тренувань 40 хвилин
(0.95763)	Програма збільшення сили та м'язів: 4 дні на тиждень - 4 силових тренування 45 хвилин
(0.94424)	Програма схуднення: 2 дні на тиждень - 1 високointенсивне кардіотренування 40 хвилин + 1 високointенсивне силове тренування 40 хвилин
(0.94298)	Програма тонус: 4 дні на тиждень - 2 загальнорозвиваючих тренування 60 хвилин + 1 кардіотренування 40 хвилин + 1 силове тренування 30 хвилин
(0.92649)	Програма витривалість: 4 дні на тиждень - 3 інтенсивних кардіотренування 60 хвилин + 1 силове тренування 30 хвилин
(0.91912)	Програма схуднення: 6 днів на тиждень - 3 інтенсивних кардіотренування 40 хвилин + 3 інтенсивних силових тренувань 30 хвилин
(0.89186)	Програма схуднення: 7 днів на тиждень - Не рекомендується тренуватись кожен день - спробуйте 6 і менше днів на тиждень
(0.85556)	Програма витривалість: 3 дні на тиждень - 3 інтенсивних кардіотренування 60 хвилин
(0.82249)	Програма збільшення сили та м'язів: 3 дні на тиждень - 3 силових тренування 45 хвилин
(0.72965)	Програма збільшення сили та м'язів: 5 днів на тиждень - 4 силових тренування 45 хвилин + 1 кардіотренування 45 хвилин
(0.69370)	Програма тонус: 5 днів на тиждень - 2 загальнорозвиваючих тренування 60 хвилин + 2 кардіотренування 40 хвилин + 1 силове тренування 30 хвилин
(0.68849)	Програма витривалість: 5 днів на тиждень - 3 інтенсивних кардіотренування 60 хвилин + 2 силових тренування 30 хвилин
(0.54410)	Програма схуднення: 1 день на тиждень - 1 високointенсивне кардіотренування 40 хвилин
(0.36863)	Програма збільшення сили та м'язів: 2 дні на тиждень - 2 інтенсивних силових тренування 60 хвилин
(0.36765)	Програма витривалість: 6 днів на тиждень - 3 інтенсивних кардіотренування 60 хвилин + 2 силових тренування 30 хвилин + 1 загальнорозвиваюче тренування 60 хвилин
(0.32486)	Програма витривалість: 2 дні на тиждень - 2 інтенсивних кардіотренування 60 хвилин
(0.24523)	Програма збільшення сили та м'язів: 6 днів на тиждень - 4 силових тренування 45 хвилин + 1 кардіотренування 45 хвилин + 1 загальнорозвиваюче тренування 45 хвилин
(0.18937)	Програма тонус: 6 днів на тиждень - 2 загальнорозвиваючих тренування 60 хвилин + 2 кардіотренування 40 хвилин + 2 силових тренування 30 хвилин
(0.18722)	Програма тонус: 3 дні на тиждень - 1 загальнорозвиваюче тренування 60 хвилин + 1 кардіотренування 40 хвилин + 1 силове тренування 30 хвилин
(0.10523)	Програма витривалість: 7 днів на тиждень - Не рекомендується тренуватись кожен день - спробуйте 6 і менше днів на тиждень
(0.09596)	Програма збільшення сили та м'язів: 7 днів на тиждень - Не рекомендується тренуватись кожен день - спробуйте 6 і менше днів на тиждень
(0.02888)	Програма тонус: 7 днів на тиждень - Не рекомендується тренуватись кожен день - спробуйте 6 і менше днів на тиждень
(0.01058)	Програма збільшення сили та м'язів: 1 день на тиждень - 1 інтенсивне силове тренування 60 хвилин
(0.00570)	Програма тонус: 2 дні на тиждень - 1 загальнорозвиваюче тренування 60 хвилин + 1 кардіотренування 40 хвилин
(0.00224)	Програма витривалість: 1 день на тиждень - 1 інтенсивне кардіотренування 60 хвилин
(0.00008)	Програма тонус: 1 день на тиждень - 1 інтенсивне загальнорозвиваюче тренування 60 хвилин

Сортировка результатов
 по именам по вероятностям Ввод

Рисунок 3.3 – Логічне виведення експертної системи

Співставляючи надані відповіді на запитання та результат консультації потрібно відзначити, що створена експертна система, працює коректно, враховує відповіді респондента та закладену думку експерта, і підбирає відповідну програму тренувань в залежності від наданих відповідей.

Розробка експертної системи в «Мала ЕС» супроводжується значними труднощами та функціональними обмеженнями. При визначенні ймовірностей вихідних фактів доводиться визначати їх методом статистичних випробувань, що призводить до значних затрат часу. Враховуючи значну кількість питань для врахування фізичного стану, кількості днів та бажаної цілі тренувань, такий спосіб логічного виведення програми фізичних навантажень стає дуже проблематичним. Крім того, неможливо здійснити числові обчислення для визначення ІМТ чи БРМ, а також дуже складно вносити зміни в уже розроблену базу знань. Отже, вирішено не використовувати логічне виведення на основі баєсового підходу.

Розглянемо процес розробки експертної системи в середовищі CLIPS, який і було обрано для розробки бази знань експертної системи створення програми тренувань. CLIPS є продукційною системою, однак на відміну від оболонки «Decision Support Expert-Shell», широко застосовується завдяки оновленням, підтримці спільноти, швидкості роботи та функціональним можливостям.

Середовище розробки CLIPS – професійний інструмент для розробки експертних систем. CLIPS постійно вдосконалюється з 1986 року і по сьогоднішній день. Це середовище призначене для полегшення розробки програмного забезпечення для моделювання людських знань або досвіду.

Існує три засоби подання знань у CLIPS [18]:

- правила, які в першу чергу призначені для евристичних знань на основі досвіду;
- конструкти створення функцій (deffunctions) та узагальнені функції, які першочергово призначені для подання процедурних знань;
- об'єктно-орієнтоване програмування, також головним чином призначене для процедурних знань. Підтримуються п'ять загальноприйнятих особливостей об'єктно-орієнтованого програмування: класи, обробники повідомлень, абстракція, інкапсуляція, успадкування, поліморфізм. Правила можуть збігатися при пошуку за шаблоном з об'єктами та фактами.

Розробнику можна вільно використовувати перелічені засоби подання знань, тобто можна розробляти програмне забезпечення, використовуючи лише правила, лише об'єкти або суміш об'єктів та правил.

CLIPS також був розроблений для інтеграції з іншими мовами, такими як мови C та Java. В самій назві CLIPS закладена аббревіатура «C Language Integrated Production System», що говорить про продукційну систему з інтегруванням мови C. Правила та об'єкти утворюють інтегровану систему, оскільки правила можуть зіставляти факти та об'єкти за шаблоном. Окрім використання в якості самостійного програмного інструменту, CLIPS можна викликати з процедурної мови, виконувати свої функції, а потім повертати керування програмі, що

викликає. Подібним чином процедурний код можна визначити як зовнішні функції та викликати його з CLIPS. Коли зовнішній код завершує виконання, керування повертається до CLIPS.

Суттєвою перевагою такого середовища є невисокі вхідні вимоги для нових розробників, якщо вони вже знайомі з об'єктно-орієнтованим програмуванням чи іншими мовами, такими як C++, Smalltalk, Objective C або Java, оскільки їм уже відомі переваги застосування об'єктів у розробці програмного забезпечення. Звичайно, середовище CLIPS містить усі необхідні можливості включно з вбудованим редактором та інструментом налагодження. Сама оболонка надає базові елементи експертних систем:

- список фактів та екземплярів, тобто глобальну пам'ять для збереження даних;
- базу знань, тобто містить всі правила – базу правил;
- механізм виведення.

ЕС написана в CLIPS може складатись з правил, фактів та об'єктів, поєднуючи в собі класичну продукційну модель подання знань з об'єктно-орієнтованим програмуванням. А механізм виведення уже вирішує які правила повинні виконатись та за яких обставин. Експертна система, що написана в CLIPS з використанням правил – це ЕС керована даними, де факти і об'єкти виступають даними, які запускають виконання завдяки механізму виведення.

Отже, середовище CLIPS задовольняє усі потреби при розробці експертної системи та не має значних функціональних обмежень.

3.2 Розробка бази знань експертної системи

Поставлена перед експертною системою задача складання програми тренувань має велику кількість допустимих рішень, вимагає врахування великої кількості параметрів та опирається на використання евристик. Тому для розробки бази знань в середовищі CLIPS було обрано правила, як основний засіб

подання знань, оскільки вони призначені для евристичних знань на основі досвіду.

Перед тим як створювати правила необхідно продумати, як буде представлена інформація в базі знань. Для цього в системі CLIPS існують факти, які є одними з основних форм представлення інформації високого рівня. Кожен факт представляє частину інформації, яка була розміщена в поточному списку фактів, який називається списком фактів. Факти є основною одиницею даних, яка використовується правилами.

Для середовища CLIPS передбачено використання впорядкованих та неупорядкованих фактів. Впорядковані факти складаються із символу, за яким слідує послідовність нуля або більше полів, розділених пробілами. Перше поле впорядкованого факту визначає «відношення», яке застосовується до решти полів упорядкованого факту. Наприклад, «батько Максима Ігор» стверджує, що Ігор є батьком Максима. Крім того, поля в неупорядкованому факті можуть належати до будь-якого з примітивних типів даних (за винятком першого поля, яке має бути символом), і жодних обмежень на впорядкування полів не накладено [18].

При розробці експертної системи для складання програми тренувань, впорядковані факти використовуються тільки 3 рази, а саме як проміжні факти в результаті активації трьох правил. Наведемо ці правила:

```
(defrule AgeGroupYoung
  (user_profile (age ?a&:(>= ?a 18)&:(<= ?a 23)))
  =>
  (assert (age_group Young))
)
(defrule AgeGroupModerate
  (user_profile (age ?a&:(>= ?a 24)&:(<= ?a 45)))
  =>
  (assert (age_group Moderate))
)
(defrule AgeGroupOld
  (user_profile (age ?a&:(>= ?a 46)))
  =>
  (assert (age_group Old))
)
```

Наведені правила визначають формальну вікову групу користувача на основі вказаного ним віку. В результаті, за допомогою ключового слова «assert» в базу знань додається впорядкований факт, наприклад «age_group Young», тобто вік користувача відноситься до формальної вікової групи «Молодий».

Було вирішено не використовувати впорядковані факти, як основну одиницю представлення інформації в даній базі знань, а тільки для деяких зручних для цього місцях. Причиною даного рішення стало те що, при подальшій інтеграції експертної системи в серверну частину інформаційної технології ускладнюється задача пошуку таких фактів. Тобто, якщо серверній частині необхідно знайти впорядкований факт – то він не має чітко визначеної назви та структури, що порушує цілісність даних та не дозволяє інтегрувати розроблену ЕС в застосунки зі строго типізованими мовами програмування, як С#.

Отже, вирішено використовувати невпорядковані факти для більшості інформації бази знань, які позбавлені таких незручностей. Впорядковані факти кодують інформацію позиційно. Щоб отримати доступ до цієї інформації, користувач повинен знати не тільки які дані зберігаються у факті, але й яке поле містить ці дані. Невпорядковані, або шаблонні факти, надають користувачеві можливість абстрагувати структуру факту, призначаючи імена кожному полю у факті. Конструкція «deftemplate» використовується для створення шаблону, який потім можна використовувати для доступу до полів факту за їх назвою. Конструкція «deftemplate» є аналогом запису або визначення структури даних в мовах програмування, таких як С.

За допомогою конструкції «deftemplate» визначається назва шаблону разом із нульовим або більше визначеннями іменованих полів або слотів. На відміну від упорядкованих фактів, слоти факту «deftemplate» можуть бути обмежені типом, значенням і числовим діапазоном. Крім того, для слота можна вказати значення за замовчуванням. Слот складається з відкриваючої дужки, за якою йде назва слота, нуля або більше полів і закриваючої дужки. Причому слоти не

можуть використовуватися в упорядкованому факті, а позиційні поля не можуть використовуватися у факті без шаблону [18].

З урахуванням вищевказаних особливостей було додано ряд шаблонів необхідних для розробки бази знань та збереження невпорядкованих фактів. Для подання інформації про вправи було додано шаблон «exercise», який містить слоти для збереження ідентифікатору, назви вправи, цільової групи м'язів, пріоритет вправи, залучення суглобів:

```
(deftemplate exercise
  (slot id)
  (slot name)
  (slot target_muscle_group)
  (slot priority)
  (slot jointness)
  (multislot support_muscle_groups)
  (multislot equipment))
```

Крім того, даний шаблон має мультислот для збереження декількох додаткових груп м'язів, що залучаються при виконанні вправи. Також є мультислот для збереження необхідного спортивного інвентаря для вправи.

Кожна вправа характеризується наявністю однієї цільової групи м'язів, яку вона найбільше включає в роботу при виконанні вправи. Таким чином, для кожної групи м'язів формується список вправ. Кожен такий список вправ за групою м'язів сортується по пріоритету, де найвищий пріоритет надається умовно найефективнішій вправі. Отже, слот пріоритету вправи відповідає саме за ефективність вправи щодо її цільової групи м'язів.

Слот залучення суглобів, відповідає за збереження інформації про кількість залучених суглобів для виконання вправи. Для даного слоту передбачено значення «односуглобний», або «багатосуглобний» (комплексний). Така інформація стає корисною, коли необхідно створити програму тренувань для користувача який має небагато доступних днів для цього, або якщо користувач початківець. В такому разі програма тренувань буде містити більше багатосуглобних вправ.

Для подання інформації про користувача було додано шаблон «user_profile», який містить слоти для збереження ідентифікатору, антропометричних показників, доступної кількості днів для тренувань та обраної цілі тренувань:

```
(deftemplate user_profile
  (slot id)
  (slot name)
  (slot gender)
  (slot weight)
  (slot height)
  (slot age)
  (slot restriction)
  (slot available_days)
  (slot goal))
```

Збереження знань про показники стану здоров'я користувача відбувається за допомогою наступних шаблонів:

```
(deftemplate health_indicator
  (slot parameterName)
  (slot parameterValue))
```

```
(deftemplate bmi_result
  (slot value)
  (slot formalName))
```

```
(deftemplate bmr_result
  (slot value))
```

```
(deftemplate bmi_formal
  (slot formal_name))
```

Шаблон «health_indicator» відповідає за узагальнене збереження ключових показників при складанні програми тренувань, а саме ІМТ та БРМ. Шаблони «bmi_result», «bmi_formal» та «bmr_result» створені для збереження числового значення показника та його формального значення, наприклад для ІМТ на рівні 20 одиниць формальне значення – нормальна маса тіла.

Одними з найголовніших шаблонів для створення програми тренувань є «daily_plan» та «weekly_plan»:

```
(deftemplate daily_plan
  (slot id_d)
  (slot age_group)
  (slot goal)
  (slot days)
  (slot day_n)
  (slot exercise_organization_type)
  (multislot exercises_id))
```

```
(deftemplate weekly_plan
  (slot id_w)
  (slot days)
  (slot rest_days)
  (slot week_plan_type)
  (multislot daily_plans_id))
```

Шаблон «daily_plan» відповідає за план тренування в межах одного дня. Він містить мультислот для збереження ідентифікаторів вправ, які використовуються в цьому плані. Тоді як шаблон «weekly_plan» відповідає за план тренування в межах одного тижня. Тому, він містить мультислот «daily_plans_id» для збереження ідентифікаторів добових планів тренувань.

Слід зауважити, що усі шаблони створені таким чином щоб розроблену ЕС можна було інтегрувати до серверної частини інформаційної технології. Саме тому використовуються слоти ідентифікатори, адже серверна частина зберігає дані в реляційній базі даних. Крім того, факти вправ, профілю користувача та плани тренування на добу будуть додаватись в базу знань з серверної частини, а саме з її бази даних. Це зроблено для гнучкості та зручності внесення змін чи додавання нових вправ. Можливість швидко скорегувати інформацію про вправи чи добові плани тренувань є досить важливою, адже з часом наукові дослідження можуть довести більшу ефективність однієї вправи над іншою і в такому випадку можна просто змінити значення в базі даних, не залучаючи при цьому базу знань.

Після створення усіх необхідних шаблонів для подання невпорядкованих фактів, необхідно створити правила, які будуть оперувати цими фактами. Знання предметної області задаються правилами, структура яких дозволяє використати весь спектр функціональних можливостей середовища CLIPS. Наведемо

декілька правил для визначення ІМТ та БРМ на основі інформації від користувача:

```
(defrule CalculateMaleBMR
  (user_profile (weight ?w) (height ?h) (age ?a) (gender male))
  =>
  (assert (health_indicator(parameterName BMR) (parameterValue =(- (+ (+ 66.5 (*
13.75 ?w)) (* 5.003 ?h)) (* 6.75 ?a))))))

(defrule CalculateFemaleBMR
  (user_profile (weight ?w) (height ?h) (age ?a) (gender female))
  =>
  (assert (health_indicator(parameterName BMR) (parameterValue =(- (+ (+ 655.1 (*
9.563 ?w)) (* 1.85 ?h)) (* 4.676 ?a))))))

(defrule CalculateBMI
  (user_profile (weight ?w) (height ?h))
  =>
  (assert (health_indicator(parameterName BMI) (parameterValue =(/ ?w (** (/ ?h
100) 2))))))
```

Для визначення формального значення ІМТ розроблено правила, які активуються якщо обраховане значення ІМТ потрапляє в певний проміжок, що відповідає одному з п'яти формальних проміжків. Наведемо правило для проміжку «Надлишкова маса»:

```
(defrule BMIOverWeight
  (health_indicator(parameterName BMI)(parameterValue ?bmi&:(>= ?bmi 25)&:(<=
?bmi 29.9)))
  =>
  (assert (bmi_formal (formal_name OverWeight))))
```

Також, розроблено правила для перетворення факту з одного шаблону в інший для зручності розробки:

```
(defrule BMIResult
  (health_indicator (parameterName BMI)(parameterValue ?value))
  (bmi_formal (formal_name ?formal))
  =>
  (assert (bmi_result (value ?value)(formalName ?formal))))

(defrule BMRResult
  (health_indicator (parameterName BMR)(parameterValue ?value))
```



```
=>
(assert (bmr_result (value ?value))))
```

Для визначення рекомендованого плану тренувань на тиждень, було розроблено правила, які мають наступний вигляд:

```
(defrule Week_Plan_Muscle_gain_2days
  (user_profile(available_days ?days&:(= ?days 2))(goal Muscle_gain))
  (age_group ?ag)
  (daily_plan(days 2)(goal Muscle_gain)(age_group ?ag)(exersice_organization_type
  Circullar)(day_n 1)(id_d ?id1))
  (daily_plan(days 2)(goal Muscle_gain)(age_group ?ag)(exersice_organization_type
  Circullar)(day_n 2)(id_d ?id2))
  =>
  (assert (weekly_plan(id_w ?*id*)(days 2)(rest_days d1r23d2)(week_plan_type
  Full_Body)(daily_plans_id ?id1 ?id2)))
  (bind ?*id*=(+ ?*id* 1))
  )
```

```
(defrule Week_Plan_Muscle_gain_3days
  (user_profile(available_days ?days&:(= ?days 3))(goal Muscle_gain))
  (age_group ?ag)
  (daily_plan(days 3)(goal Muscle_gain)(age_group ?ag)(exersice_organization_type
  Circullar)(day_n 1)(id_d ?id1))
  (daily_plan(days 3)(goal Muscle_gain)(age_group ?ag)(exersice_organization_type
  Circullar)(day_n 2)(id_d ?id2))
  (daily_plan(days 3)(goal Muscle_gain)(age_group ?ag)(exersice_organization_type
  Circullar)(day_n 3)(id_d ?id3))
  =>
  (assert (weekly_plan(id_w ?*id*)(days 3)(rest_days d1r1d2r1d3)(week_plan_type
  Full_Body)(daily_plans_id ?id1 ?id2 ?id3)))
  (bind ?*id*=(+ ?*id* 1))
  )
```

Наведені вище правила перевіряють наявність фактів з певною обраною користувачем кількістю днів тренувань, ціллю тренувань та віковою групою. Далі відбувається пошук фактів де плани тренувань на добу співпадають з попередньою перевіркою за кількістю днів тренувань, ціллю тренувань, віковою групою, а також мають круговий тип організації заняття, певний порядковий номер добового плану. Якщо такі добові плани знайшлися, то їх ідентифікатори

зберігаються в змінні і після цього створюється новий неупорядкований факт рекомендованого плану тренувань на тиждень для користувача. Такий факт містить усю необхідну інформацію та ідентифікатори добових планів тренувань, за допомогою яких можна знайти ці плани тренувань на кожну конкретну добу та дізнатись записану там інформацію.

Також було розроблено правила, які покривають випадки, коли обрана користувачем ціль тренувань не відповідає його показникам здоров'я. Наведемо приклад такого правила:

```
(defrule UnderWeight_Fat_Loss
  (bmi_formal(formal_name undereight))
  (user_profile(goal fat_loss))
  =>
  (assert (recomendation (bad_practice "Considering your BMI (UnderWeight) we
  recommend to change your goal for Muscle gain and follow next training program:")
  (weekly_plan_id ?*id*)))
  (bind ?*id*=(+ ?*id* 1))
  )
```

В даному випадку, якщо користувач має недостатню вагу за визначеним ІМТ і обрав ціль тренувань «позбавитись зайвої ваги» - то ЕС запропонує змінити ціль тренувань на «збільшення м'язової маси».

3.3 Тестування розробленої експертної системи

Перевіримо правильність роботи розробленої експертної системи. Для початку визначимо тестового користувача. Необхідні дані про нього заповнимо за шаблоном «user_profile» як в розробленій базі знань. Результат створення тестового користувача, наведено в таблиці 3.1.

Таблиця 3.1 – Профіль користувача для тестування системи.

Поле	Значення
Ім'я користувача	test user - 1
Стать	Чоловіча

Продовження таблиці 3.1.

Вік	28
Зріст	186 см
Маса	58 кг
Ціль	Збільшення м'язової маси тіла
Бажана кількість днів тренувань на тиждень	4
Медичні та інші обмеження	Відсутні

Тестування проводилось з використанням серверної частини інформаційної технології організації та проведення тренувального процесу. Тестового користувача було додано до бази даних, після чого було викликано експертну систему для складання програми тренувань для користувача. Його профіль було додано до бази знань як невпорядкований факт за шаблоном «user_profile». Після запуску механізму виведення, експертна система створила невпорядкований факт за шаблоном «weekly_plan» і серверна частина отримала відповідь у вигляді цього факту. Оскільки сервер дає відповідь у форматі JSON [19], було проведено перетворення отриманого від ЕС факту в зазначений формат. На рисунку 3.4 наведено створену програму тренувань для тестового користувача у форматі JSON.

Отриманий результат врахував сучасні підходи, принципи, методи та методології створення програм тренувань, а також індивідуальні особливості та потреби користувача, такі як антропометричні показники, бажана кількість днів тренувань та обрана ціль тренувань. ЕС правильно визначила ІМТ та перевірила чи не заперечує отриманий показник обраній цілі тренувань. Було визначено вікову групу «Moderate» (зрілий) та перевірено, чи можна зрілому користувачу займатись 4 рази на тиждень. Вірно визначено хронологію тренувальних днів та днів відпочинку «trainingRestBalance», отримавши результат «d1d2r1d3d4». Він розшифровується як, перший день тренувань «d1» слідом за яким другий день тренувань «d2», потім один день відпочинку «r1», потім знову 2 дні тренувань

«d3» та «d4». Спосіб організації тренувальних днів у тижні визначено як «UpperLowerBody», як і було задумано при розробці.

```
[
  {
    "id": 23,
    "totalDays": 4,
    "trainingRestBalance": "d1d2r1d3d4",
    "weekPlanType": "UpperLowerBody",
    "dailyPlans": [
      {
        "id": 42,
        "ageGroup": "Moderate",
        "goal": "MuscleGain",
        "totalDays": 4,
        "dayNumber": 1,
        "exerciseOrganizationType": "Circular",
        "exercises": []
      },
      {
        "id": 13,
        "ageGroup": "Moderate",
        "goal": "MuscleGain",
        "totalDays": 4,
        "dayNumber": 2,
        "exerciseOrganizationType": "Circular",
        "exercises": []
      },
      {
        "id": 2,
        "ageGroup": "Moderate",
        "goal": "MuscleGain",
        "totalDays": 4,
        "dayNumber": 3,
        "exerciseOrganizationType": "Circular",
        "exercises": []
      },
      {
        "id": 21,
        "ageGroup": "Moderate",
        "goal": "MuscleGain",
        "totalDays": 4,
        "dayNumber": 4,
        "exerciseOrganizationType": "Circular",
        "exercises": []
      }
    ]
  }
]
```

Рисунок 3.4 – Програма тренувань тестового користувача

Крім того, ЕС чітко визначила список планів тренувань на кожний тренувальний день в тижні «dailyPlans». Оскільки користувач обрав 4 дні, було отримано 4 плани, де вказано спосіб організації порядку виконання вправ «exerciseOrganizationType», порядковий номер плану серед 4 інших (dayNumber) та список вправ.

Перевіримо правильність роботи експертної системи для ще одного створеного тестового користувача, інформація про якого наведена в таблиці 3.2.

Таблиця 3.2 – Профіль користувача №2 для тестування системи.

Поле	Значення
Ім'я користувача	test user – 2
Стать	Жіноча
Вік	46
Зріст	168 см
Маса	75 кг
Ціль	Зменшення маси тіла
Бажана кількість днів тренувань на тиждень	2
Медичні та інші обмеження	Відсутні

На рисунку 3.5 наведено створену програму тренувань для тестового користувача №2 у форматі JSON. З огляду на отриманий результат, ЕС створила програму тренувань на 2 дні виходячи з зазначених користувачем даних, причому спосіб організації тренувальних днів у тижні визначено як «FullBody».

```
[
  {
    "id": 24,
    "totalDays": 2,
    "trainingRestBalance": "d1r3d2",
    "weekPlanType": "FullBody",
    "dailyPlans": [
      {
        "id": 56,
        "ageGroup": "Old",
        "goal": "FatLoss",
        "totalDays": 2,
        "dayNumber": 1,
        "exerciseOrganizationType": "Circullar",
        "exercises": []
      },
      {
        "id": 49,
        "ageGroup": "Old",
        "goal": "FatLoss",
        "totalDays": 4,
        "dayNumber": 2,
        "exerciseOrganizationType": "Circullar",
        "exercises": []
      }
    ]
  }
]
```

Рисунок 3.5 – Програма тренувань тестового користувача №2

Дане рішення є абсолютно правильним, оскільки для людини, яка бажає позбутися зайвої ваги буде важко це зробити за 2 дні. Тому «FullBody» тренування дозволять збільшити тренувальний об'єм, задіявши максимальну кількість м'язових груп, і тим самим відіграти малу кількість тренувань на тиждень.

Проведемо тестування якості надання рекомендацій щодо визначення програми тренувань. Для проведення такого тестування було обрано експертів, які досліджували тренувальний процес, підходи до організації тренувального процесу, а особливо створення програми тренувань під цілі клієнта, його фізичні показники та доступну кількість днів для тренувань на тиждень.

Експертам необхідно було визначити за десятибальною шкалою наскільки коректною та якісною є програма тренувань створена експертною системою та наскільки вона враховує фізичні показники клієнта, доступну кількість тренувань на тиждень та поставлену ціль тренувань. Таким чином головні фактори впливу на результуючу програму тренувань були охоплені. Окрім розробленої експертної системи, для порівняння оцінювались програми-аналоги FitnessAI та Fitbod.

Експертами було запропоновано 5 стереотипних ситуацій, які являють собою контрольні приклади. Кожна ситуація описує необхідні вхідні дані клієнта, тобто вхідні фактори створення програми тренувань. За цими даними експертна система та програми аналогів створювали програму тренувань, яку в свою чергу, оцінювали експерти. Результати оцінювання експертів наведено в таблиці 3.3.

Таблиця 3.3 – Результати оцінювання експертами якості надання рекомендацій щодо визначення програми тренувань

Інформаційна технологія організації та проведення тренувального процесу							
Стереотипні ситуації	Оцінки експертів					Середня оцінка якості	
	№1	№2	№3	№4	№5	(бали)	(%)
1	9	8	10	9	7	8.6	86%
2	8	9	7	10	8	8.4	84%
3	10	10	9	7	8	8.8	88%
4	9	10	8	9	7	8.6	86%
5	10	9	8	9	8	8.8	88%
Загальна оцінка якості надання рекомендацій: 86.4%							
Програма-аналог FitnessAI							
Стереотипні ситуації	Оцінки експертів					Середня оцінка якості	
	№1	№2	№3	№4	№5	(бали)	(%)
1	8	7	8	7	6	7.2	72%
2	9	9	8	7	7	8	80%
3	9	8	8	8	7	8	80%
4	8	7	8	8	6	7.4	74%
5	8	8	8	8	7	7.8	78%
Загальна оцінка якості надання рекомендацій: 76%							
Програма-аналог Fitbod							
1	8	7	8	8	7	7.6	76%
2	8	9	7	8	7	7.8	78%
3	8	9	8	7	8	8	80%
4	7	8	8	8	7	7.6	76%
5	8	8	8	8	7	7.8	78%
Загальна оцінка якості надання рекомендацій: 77.6%							

Стереотипні ситуації, які було запропоновано експертами наведено в таблиці 3.4. Розглянемо чому експерти поставили саме такі оцінки як в таблиці 3.3. В першій ситуації програма-аналог Fitbod створила програму тренувань для 5 днів на тиждень, однак кожне тренування було спрямоване на силові вправи – важкі фізичні навантаження. Таким чином не було враховано час необхідний на відновлення організму, а також не взятий до уваги віковий фактор, оскільки в 52 роки відновлювальні процеси в організмі людини сповільнюються. Розроблена експертна система врахувала вік людини та її ІМТ, на основі цього було запропоновано програму тренувань на 3 дні (2 силових тренування і одне кардіо-тренування). Також важкі силові вправи зі штангою з великим додатковим навантаженням, були замінені на безпечніші вправи використовуючи тренажери.

Таблиця 3.4 – Опис стереотипних ситуацій

Поле	Стереотипні ситуації				
	№1	№2	№3	№4	№5
Стать	Чоловіча	Жіноча	Чоловіча	Чоловіча	Жіноча
Вік	52	19	20	41	46
Зріст	180 см	175 см	171 см	182 см	184 см
Маса	87 кг	56 кг	50 кг	108 кг	74 кг
Ціль	Зменшення маси тіла	Збільшення м'язової маси тіла	Збільшення м'язової маси тіла	Зменшення маси тіла	Підтримання тонусу м'язів
Бажана кількіс-ть тренувань на тиждень	5	3	2	3	2

Програма-аналог FitnessAI створила програму тренувань на 3 силових і 2 кардіо-тренування, причому набір вправ переважно складався з важких силових вправ (присідання зі штангою, станова тяга, жим штанги лежачи і т.д.). Крім того,

додаткова вага на спортивних знаряддях була розрахована згідно застарілих формул заснованих на масі тіла. Внаслідок цього загальний тоннаж (виконана робота) виявився надмірним для людини з такими віковими та ваговими показниками.

В другій ситуації програма-аналог Fitbod створила стандартну програму тренувань на 3 дні з важкими силовими вправами. Однак, вона не врахувала специфіку підбору вправ для жінок, оскільки важкі вправи зі штангою стимулюють ріст (гіпертрофію) великих м'язових груп (найширший м'яз спини, квадрицепси, грудні м'язи), що підходить більше для чоловіків. Розроблена експертна система врахувала вік людини та недостатній ІМТ і рекомендувала програму тренувань на 3 дні, причому початковий етап тренувань не містить важких силових вправ. Тобто було враховано різницю між відновлювальними можливостями жінок та чоловіків після фізичних навантажень. Комплекс вправ фокусується на розвитку м'язових груп пріоритетних для жінок (сідничні м'язи, найдовший м'яз спини, м'язи кору), що краще відповідає цілі тренувань. FitnessAI розрахувала програму на 3 дні та підбрала комплекс вправ який підходить для жінок, замінивши силові вправи зі штангою на вправи з гантелями, однак залишився фокус на розвиток великих м'язових груп.

В третій ситуації розроблена експертна система рекомендувала збільшити кількість тренувань до 3 днів, оскільки для молодого чоловічої статі з недостатнім індексом маси тіла необхідно збільшити рівень стресу для м'язових груп, щоб стимуляція росту м'язів була достатньою. Інші програми аналогії рекомендували 2 дні тренувань на тиждень з адекватним набором вправ, однак цієї кількості тренувань на тиждень потенційно недостатньо для достатньої стимуляції росту м'язів.

В четвертій ситуації програми аналогії створили набір вправ для кардіо-тренувань (біг, велотренажер, еліптичні тренажери, ...), не враховуючи, що людина має ІМТ – ожиріння першого ступеню. Тому певні запропоновані вправи можуть зашкодити суглобам цієї людини через надмірну вагу і підвищене навантаження на опорно-рухового апарату.

В п'ятій ситуації програми аналоги рекомендували комплекс вправ, який погано відповідає цілі тренувань – підтримання тону м'язів. Аналоги поєднали вправи для набору м'язової маси та зменшення маси тіла, зменшивши інтенсивність фізичних навантажень та загальний тоннаж для цих вправ. Таким чином, ціль тренувань задовольняється тільки частково.

За результатами з таблиці 3.3 було визначено загальну оцінку якості надання рекомендацій для програм-аналогів та розробленої інформаційної технології. Для програм-аналогів вона становить 76% та 77.6%, тобто середня якість надання рекомендацій програм-аналогів становить 76.8%. Отже, оцінки експертів показали, що розроблена інформаційна технологія організації та проведення тренувального процесу підвищила якість надання рекомендацій щодо визначення програми тренувань на 9.6%.

3.4 Висновок до розділу 3

В даному розділі було здійснено обґрунтування вибору середовища для розробки експертних систем. Проведено попередню розробку ЕС в різних оболонках та середовищах, на основі чого визначено основні переваги та недоліки кожного з них. Середовище CLIPS виявилось найбільш вдалим та задовольняє усі вимоги при розробці експертної системи, не маючи при цьому значних недоліків.

На етапі розробки бази знань експертної системи було створено усі необхідні шаблони для збереження неупорядкованих фактів. Причому шаблони розроблено у такий спосіб, щоб можна було успішно інтегрувати ЕС до серверної частини інформаційної технології. Також, розроблено правила бази знань, що реалізують логіку складання програми тренувань. Крім того, проведено тестування розробленої експертної системи. Результати підтвердили правильність роботи правил бази знань та адекватність всієї ЕС. А тестування якості надання рекомендацій щодо визначення програми тренувань показало підвищення якості на 9.6% в порівнянні з програмами аналогами.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ СЕРВЕРНОЇ ЧАСТИНИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОРГАНІЗАЦІЇ ТА ПРОВЕДЕННЯ ТРЕНУВАЛЬНОГО ПРОЦЕСУ

4.1 Обґрунтування вибору засобів розробки серверної частини

Зважаючи на вимоги до функціональності, поставлених завдань, акторів і потенційних запитів на отримання або відправлення інформації всередині системи, для впровадження були обрані різні технології та засоби розробки.

Було обрано мову програмування C#. Рішення було обумовлено в першу чергу об'єктно-орієнтованим підходом до створення будь-якої частини системи. Враховуючи масштаб розвитку на міжнародному рівні, мови програмування пропонують розробникам численні корисні можливості та підходи, зокрема в межах їх власного середовища розробки . NET. У C# можна повністю застосувати принципи написання чистого коду для покращення функціональності без модифікації існуючих кодів, зменшення витрат і зусиль під час масштабування та прискорення обслуговування системи за рахунок скорочення часу, необхідного для читання та написання коду. Чистий код в першу чергу досягається за рахунок слабкого зв'язку об'єктів один з одним під час запису. Синтаксис LINQ є важливим компонентом розробки, що забезпечує взаємодію з об'єктами бази даних, списками об'єктів і всіма типами довідкових об'єктів в одному шаблоні з усіма необхідними можливостями. [20] У розробці розподілених систем LINQ дозволяє реалізувати взаємодію через асинхронні методи.

Крім того, мова програмування може бути використана для реалізації структурованих інформаційних об'єктів у системі, що призводить до керованої поведінки програми. Таким чином, ви можете підвищити надійність і безпеку системи, реалізувавши автоматичне очищення оперативної пам'яті від непотрібних або доступних об'єктів. Це вигідно. Контрольований код дозволяє зменшити надмірність даних, не вимагаючи інтегральних обмежень або ручного

введення даних [21]. Усі додаткові зусилля, необхідні для роботи з керованим кодом, були зроблені для використання середовища виконання CLR.

Управління всіма компонентами системи в рамках платформи розробки .NET здійснюється за допомогою середовища виконання CLR, яке керує їхнім кодом усередині компанії. Середовище дозволяє автоматичний розподіл пам'яті під час роботи системи, а також її очищення та виконання операцій. Основною перевагою є процес компіляції написаного коду. Компілятор JIT перетворює код, написаний на мові високого рівня, у проміжну мову IL, яка не залежить від будь-якого впливу на апаратне забезпечення пристрою. Проміжний мовний код перетворюється на машинний код компілятором JIT під час виконання, при цьому операційна система виконує команди, які виконуються на її власному комп'ютері [22]. Таким чином, середовище виконання CLR підлаштовує проміжний код мови IL під ту апаратну конфігурацію пристрою, де це середовище працює.

Для розробки серверної частини було обрано середовище розробки Visual Studio, яка пропонує широкий спектр функцій і є зручною для користувача. Розробнику надається продуманий інтерфейс користувача, який можна легко налаштувати, призначити автоматичне розташування та вирівнювання коду на основі вибраних ключових слів, а також виділення синтаксичних та семантичних помилок. Таким чином, можна виконувати налагодження коду одразу під час програми. Visual Studio має важливу функцію IntelliSense, яка надає інтерактивні підказки під час введення певних об'єктів і відображає параметри для методів, параметрів, класів, атрибутів і перевантажених методів.

Для створення, управління та маніпулювання даними обрано Microsoft SQL Server та відповідно реляційний тип бази даних. Така СУБД добре взаємодіє з іншими компонентами системи, оскільки також від компанії Microsoft.

Під час розробки інформаційної технології використовується Entity Framework та Identity Entity Framework, які виступають посередниками для взаємодії системи на платформі .NET з обраною реляційною базою даних. Entity Framework виконує роль ORM, який дозволяє взаємодіяти з таблицями та

кортежами реляційної бази даних, як з об'єктами та класами при об'єктно-орієнтованому програмуванні [23].

На рисунку 4.1 представлено внутрішню структуру Entity Framework.

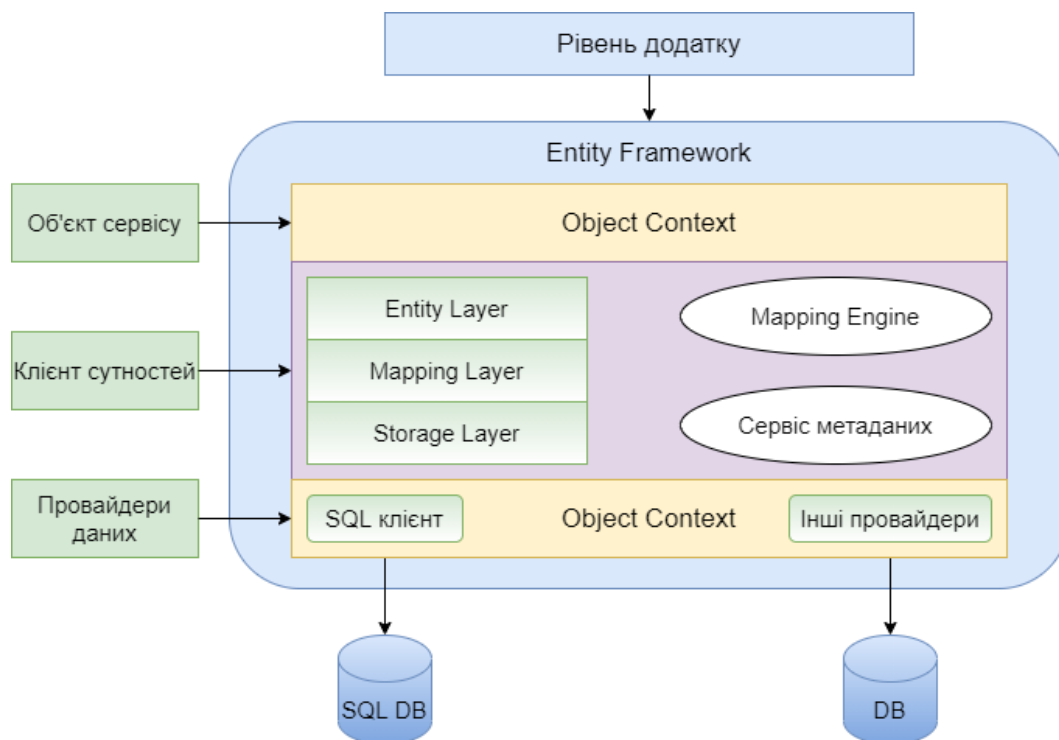


Рисунок 4.1 - Внутрішня структура Entity Framework.

Identity Entity Framework є доповненням до Entity Framework для впровадження системи авторизації та аутентифікації користувачів системи [23].

Фреймворк ASP.NET Core було обрано за основу для розробки інформаційної технології організації та проведення тренувального процесу. Фреймворк є у відкритому доступі та надає можливості створення сучасних кросплатформних веб-застосунків, мобільних застосунків, мікросервісів та Web Api. В комплекті до фреймворку є стандартний набір базових бібліотек класів, які є сумісними з усіма .NET Core застосунками. Оскільки система в межах даної роботи буде працювати на девайсах з різними операційними системами, то розроблене рішення має бути розраховане на таке різноманіття. ASP.NET Core якраз забезпечує таку потребу, а також має вбудований функціонал для впровадження залежностей (Dependency Injection), можливість інтеграції з

сучасними фреймворками розробки клієнтської частини застосунку, готовність до конфігурації на основі хмарного середовища, неперевантажений модульний канал обробки HTTP запитів. Крім того, для точного налаштування роботи Web API є можливість змінювати канал обробки запитів за допомогою власних, або вбудованих проміжних шарів застосунку. Отже, фреймворк ASP.NET Core максимально повно забезпечує потреби при розробці системи.

4.2 Опис структури серверної частини

Веб-браузер на стороні клієнта відображає інтерфейс користувача, тобто HTML та CSS код веб застосунку. Таким чином, користувач може взаємодіяти з системою через веб-браузер будь-якого девайса, що має доступ до мережі інтернет. Клієнтська частина системи посилає HTTP або HTTPS запити до сервера системи, чиї функції виконує Web API.

На рисунку 4.2 наведено діаграму розгортання інформаційної технології організації та проведення тренувального процесу.

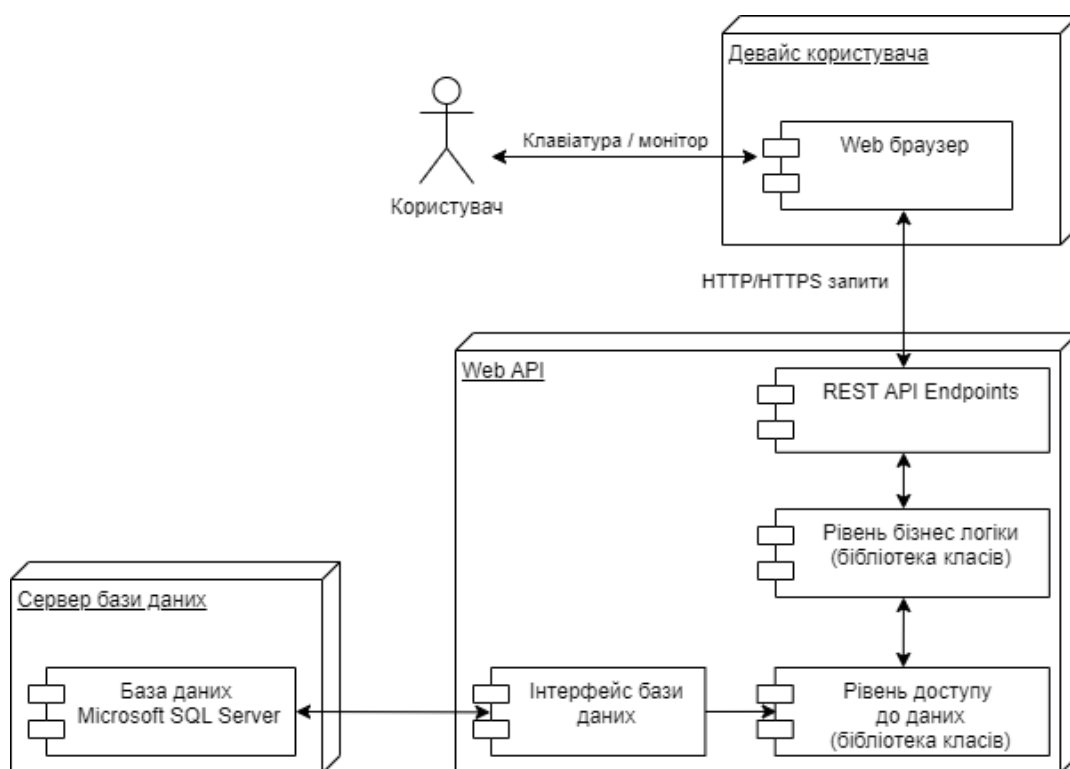


Рисунок 4.2 – Діаграма розгортання інформаційної технології

Сервер системи представлений розробленим Web API. Він побудований з певного набору компонентів, що взаємопов'язані між собою. На рисунку 4.3 наведено діаграму компонентів системи, де показано основні розроблені компоненти та взаємозв'язки між ними. HTTP запити від клієнтської частини системи надходять по каналу обробки запитів. Після цього запити обробляються ланцюгом проміжних шарів (middleware), де спочатку відпрацьовує проміжний шар обробки помилок та передає обробку проміжному шару обслуговування статичних файлів, таких як файли JavaScript або CSS. В кінці каналу обробки запитів ланцюг замикається і обробка передається до кінцевої точки певного контролера. Підсистема маршрутизації отримує інформацію про маршрут до кінцевої точки та пересилає їй подальшу обробку запиту. Кінцевими точками є методи контролерів. Вони побудовані за принципами REST, що дозволяє забезпечити масштабованість взаємодії компонентів всієї системи, загальність інтерфейсів, незалежне впровадження компонентів, проміжні компоненти, що знижують часову затримку або посилюють безпеку.

Контролери виконують ряд певних дій та взаємодіють з бібліотекою класів рівня бізнес-логіки. На цьому рівні описано сервіси та додаткову інфраструктуру для їхньої роботи, таку як логіка взаємодії цієї системи з об'єктами та функціями експертної системи CLIPS. Робота з базою знань експертної системи полягає в наступному:

- завантаження фактів про вправи для фізичних навантажень з бази даних MS SQL Server до бази знань;
- завантаження фактів про плани тренувань на добу з бази даних MS SQL Server до бази знань;
- завантаження факту профілю користувача до бази знань;
- обчислення індексу маси тіла та його формального значення;
- визначення базального рівня метаболізму;
- визначення вікової групи;
- створення плану фізичних навантажень на тиждень, враховуючи індивідуальні особливості та потреби користувача, таких як

антропометричні показники, бажана кількість днів тренувань та обрана ціль тренувань;

- перевірка відповідності фізичних показників користувача до обраної ним цілі тренувань та виведення відповідної рекомендації.

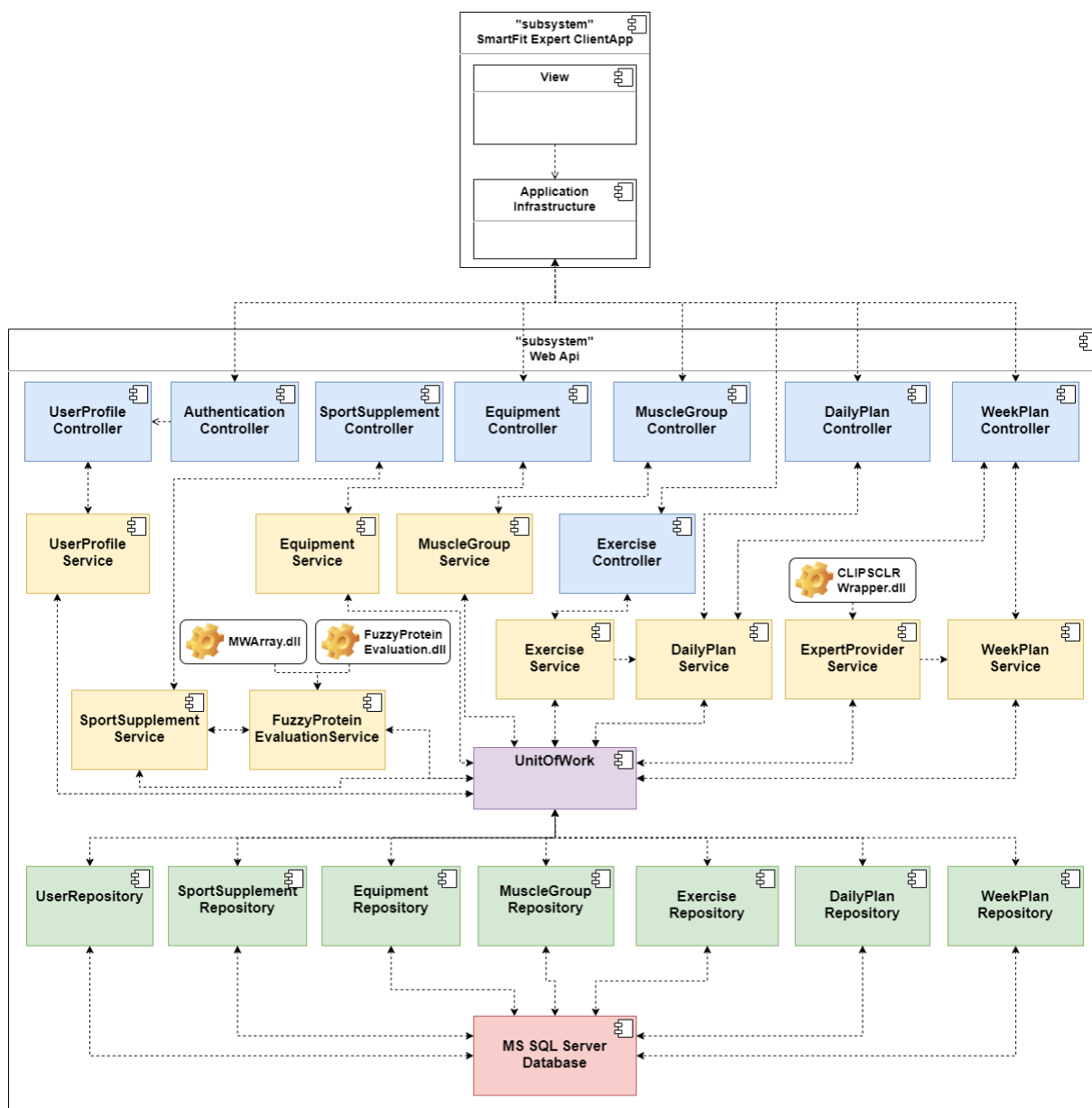


Рисунок 4.3 – Діаграма компонентів серверної частини

Така робота з базою знань відбувається на рівні бізнес-логіки в сервісі **ExpertProviderService**. Даний сервіс використовує можливості **CLIPS** за допомогою додаткового ресурсу динамічно зв'язаних бібліотек (DLL). Бібліотека класів **CLIPSCLRWrapper.dll** містить об'єкти, що забезпечують обмін

даними між середовищем .NET та експертною системою CLIPS, а також надає доступ до розробленої бази знань, що дозволяє інтегрувати її в веб-застосунок. Рівень бізнес-логіки слугує посередником між контролерами та рівнем доступу до даних. Сервіси взаємодіють з рівнем доступу до даних, коли потрібно взаємодіяти з інформацією бази даних.

Рівень доступу до даних отримує керування від сервісів та займається виключно задачами по роботі з даними та взаємодіє з базою даних. Для організації ефективної та правильної роботи на цьому рівні, вирішено застосувати шаблон проектування UnitOfWork та Repository. Шаблон UnitOfWork дозволяє взаємодіяти різним сервісам системи з єдиним об'єктом, що містить весь необхідний набір репозиторіїв з даними системи. Таким чином, реалізується підхід транзакцій, коли виконується одразу ряд дій з даними, а не постійне звернення до бази даних при кожній нагоді. Це забезпечує пришвидшення роботи з даними та дозволяє відмінити транзакцію, тобто весь ланцюг дій, в разі виникнення проблем. Шаблон проектування Repository використовувався попереднім шаблоном, та являє собою додатковий рівень абстракції доступу до даних. Це забезпечує інкапсуляцію деталей реалізації механізму збереження і отримання даних від джерела, що дозволяє розробникам мати велику гнучкість при зміні джерел інформації. Крім цього, код легше повторно використовувати, підтримувати та проводити тестування.

Розробка бази даних відбувалась в декілька етапів. Спочатку було розроблено інфологічну модель бази даних (Додаток В, рис В.5), що визначає предметну область інформаційної технології та має високий рівень абстракції. Для формалізації такої моделі було обрано нотацію Чена. На основі інфологічної моделі, було розроблено даталогічну модель бази даних (Додаток В, рис В.6), орієнтовану на систему управління базами даних MS SQL Server.

На усіх рівнях роботи Web API застосовується шаблон проектування Dependency Injection. Для полегшення розробки та подальшого масштабування системи необхідно забезпечити слабку зв'язаність компонентів між собою. Dependency Injection розв'язує таку задачу завдяки інверсії управління, тобто

створення залежного об'єкта відбувається не всередині цього об'єкта, в зовні. Утворений зовні залежний об'єкт надається у розпорядження головному декількома шляхами. В даній системі це відбувається методом впровадження залежності в конструкторі класу, де використовується залежний клас. Таким чином, контролери системи залежать від абстракції, а не від конкретного класу сервісу бізнес-логіки. Аналогічно сервіси не залежать напряду від рівня доступу до даних.

4.3 Тестування серверної частини інформаційної технології та аналіз отриманих результатів

Для проведення тестування серверної частини інформаційної технології було використано Swagger, як показано на рисунку 4.4. Це інструмент для документації та тестування API, який дозволяє автоматично створювати документацію розробленого API завдяки опису її структури в форматі YAML або JSON файлу. Для кожного з методів HTTP, Swagger дозволяє описати параметри запитів і формат відповідей. Swagger також дозволяє описувати параметри та формат відповідей для методів POST, PUT і DELETE.

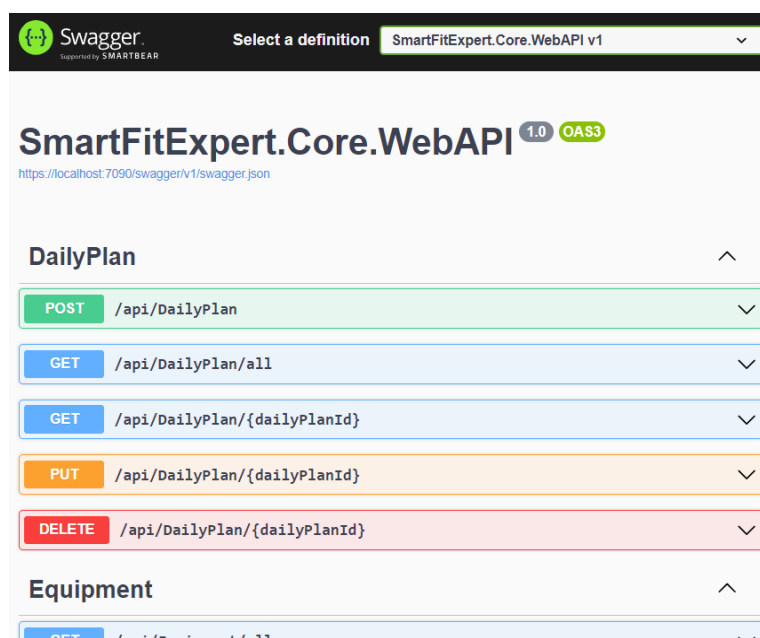


Рисунок 4.4 – Серверна частина інформаційної технології у Swagger

Проведемо тестування методу POST для сервісу роботи з профілями користувачів. Для цього заповнимо об'єкт профілю користувача у форматі JSON. Таким чином відбувається наповнення тіла HTTP запиту, як показано на рисунку 4.5. Натиснувши кнопку «Execute», до серверної частини інформаційної технології, а саме до контролеру Web API, відправляється сформований HTTP запит з методом POST. Контролер для обробки запитів з об'єктами профілів користувачів (UserProfileController), отримує відправлений HTTP запит та обробляє його використовуючи відповідний сервіс (UserProfileService). Даний сервіс використовує контекст даних Entity Framework для створення нового профілю користувача в базі даних. В результаті, порядок виконання повертається до контролеру та повертає відповідь на початковий запит, як показано на рисунку 4.6.

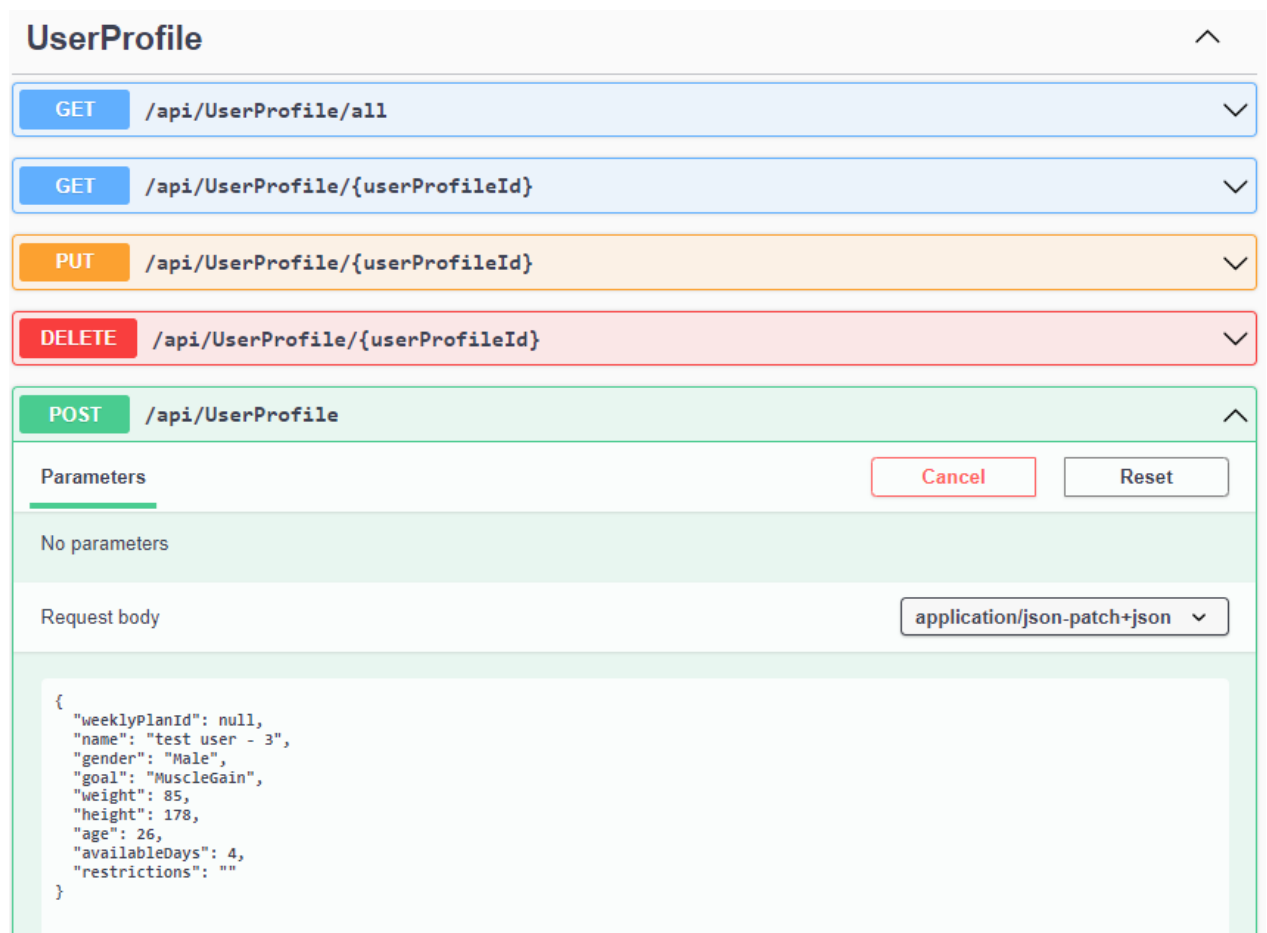


Рисунок 4.5 – Створення запиту з методом POST для профілю користувача

З рисунку 4.6 можна побачити, що код статусу відповіді серверу 200, тобто запит успішно виконався і в результаті сервер повернув новий профіль користувача. Перевіримо чи дійсно було створено новий об'єкт профілю користувача за допомогою середовища управління та розробки баз даних Microsoft SQL Server Management Studio 18. Для цього необхідно встановити підключення до серверу бази даних та виконати запит на отримання усіх профілів користувачів, як показано на рисунку 4.7. В результаті можна побачити новий кортеж з ідентифікатором 6, який відповідає новому профілю користувача.

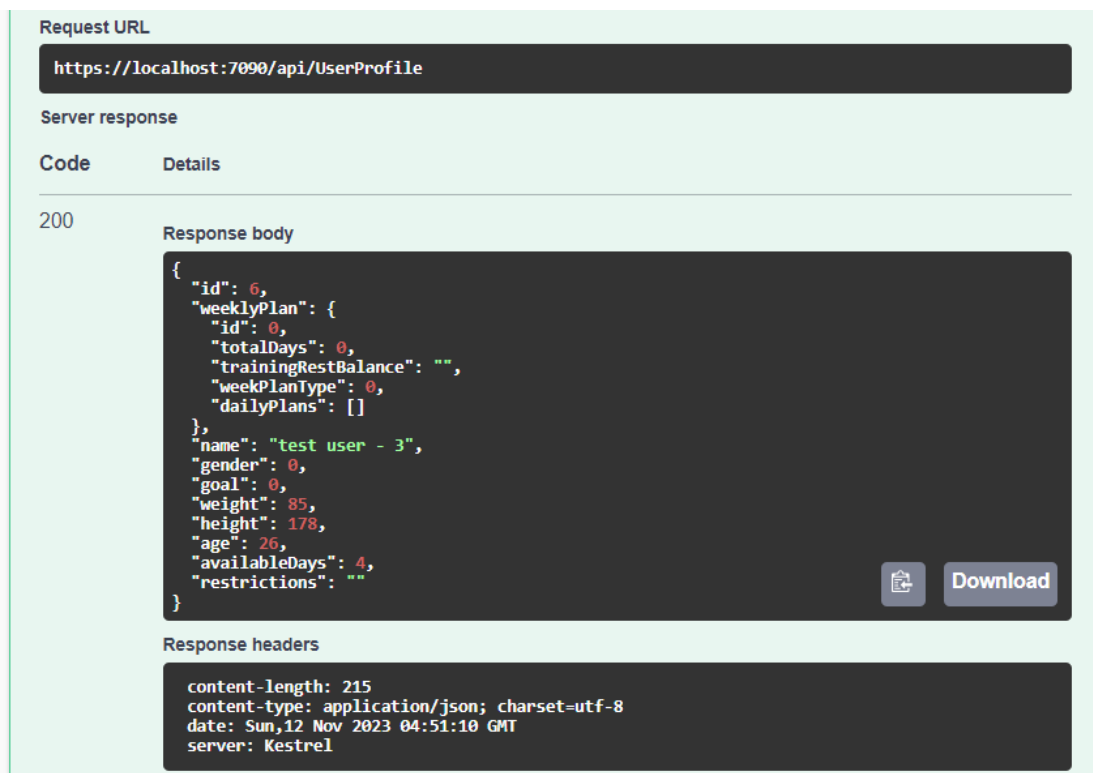


Рисунок 4.6 – Відповідь на отриманий запит з методом POST для профілю користувача

Проведемо тестування методу GET для сервісу роботи з профілями користувачів. А саме, виконаємо запит на отримання усіх користувачів. На рисунку 4.8 показано відповідь серверу на отриманий запит. Код статусу відповіді серверу дорівнює 200.

Id	WeeklyPlanId	Name	Gender	Goal	Weight	Height	Age	AvailableDays	Restrictions
1	1	Max	0	0	82	177	22	4	
2	4	test user - 1	0	0	58	186	28	4	
3	5	test user - 2	1	1	75	168	46	2	
4	6	test user - 3	0	0	85	178	26	4	

Рисунок 4.7 – Результат виконання запиту в базі даних

Це означає, що запит успішно виконався і було отримано колекцію об'єктів профілів користувачів у форматі JSON. Останнім в даній колекції є новостворений користувач при попередньому тестуванні.

```
curl -X 'GET' \
  'https://localhost:7090/api/UserProfile/all' \
  -H 'accept: text/plain'
```

Request URL: https://localhost:7090/api/UserProfile/all

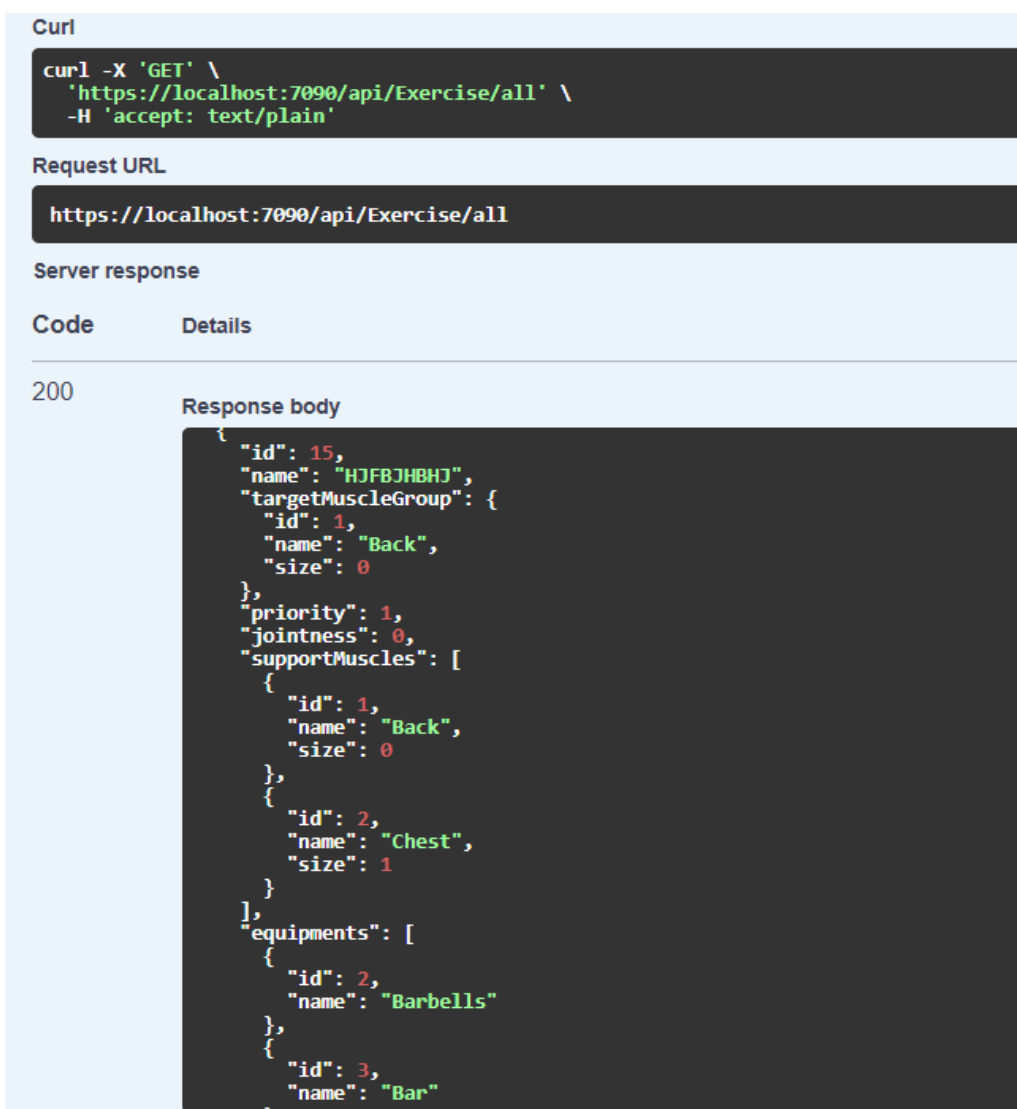
Server response: 200

```
{
  "name": "test user - 2",
  "gender": 1,
  "goal": 1,
  "weight": 75,
  "height": 168,
  "age": 46,
  "availableDays": 2,
  "restrictions": ""
},
{
  "id": 6,
  "weeklyPlan": {
    "id": 0,
    "totalDays": 0,
    "trainingRestBalance": "",
    "weekPlanType": 0,
    "dailyPlans": []
  },
  "name": "test user - 3",
  "gender": 0,
  "goal": 0,
  "weight": 85,
  "height": 178,
  "age": 26,
  "availableDays": 4,
  "restrictions": ""
}
]
```

Response headers: content-length: 871, content-type: application/json; charset=utf-8, date: Sun, 12 Nov 2023 04:56:20 GMT

Рисунок 4.8 – Відповідь на отриманий запит з методом GET для профілю користувача

Проведемо тестування методу GET для сервісу роботи з вправами. А саме, виконаємо запит на отримання усіх вправ. На рисунку 4.9 показано відповідь серверу на отриманий запит. Код статусу відповіді серверу 200, тобто запит було отримано та оброблено успішно. У відповідь сервер відправив колекцію об'єктів вправ, які містять вкладений об'єкт м'язової групи та вкладений об'єкт спортивного спорядження необхідного для виконання вправи. Однак, можна помітити що на рисунку 4.9 вправа має неадекватну назву, яку потрібно змінити. Для цього проведемо тестування методу PUT для сервісу роботи з вправами.



```
Curl
curl -X 'GET' \
'https://localhost:7090/api/Exercise/all' \
-H 'accept: text/plain'

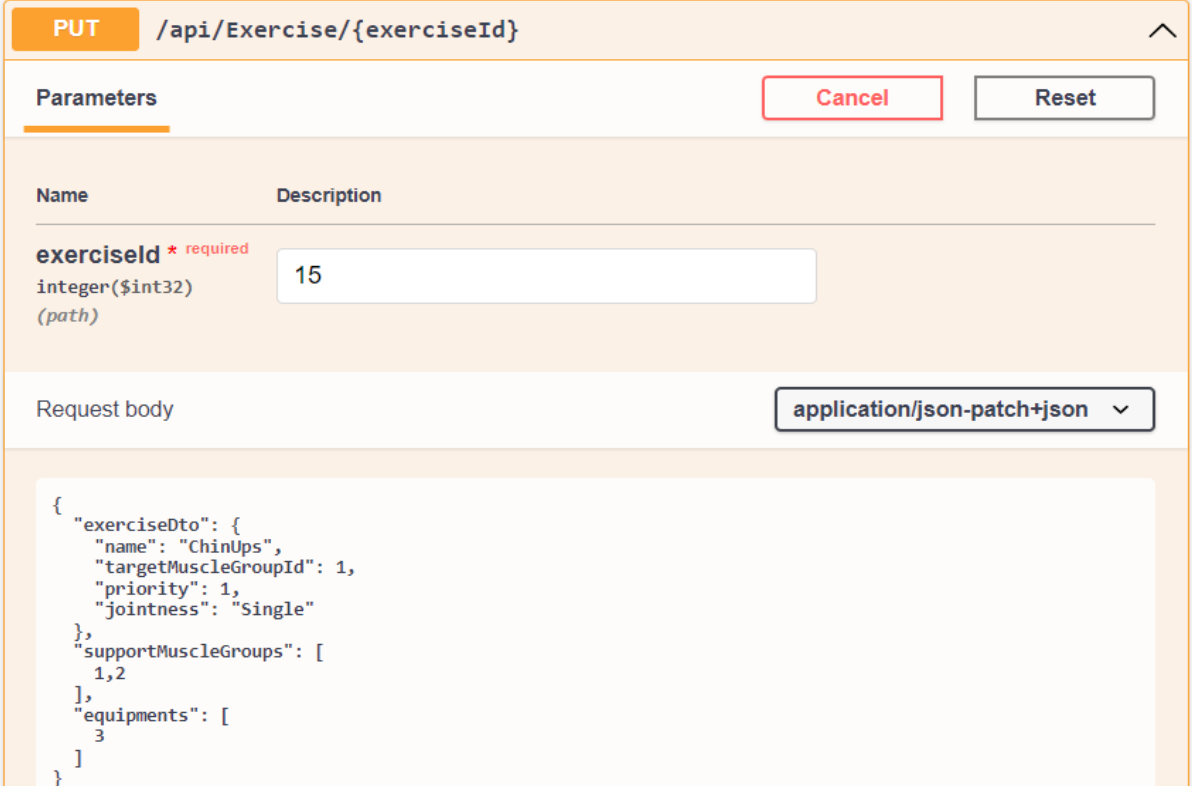
Request URL
https://localhost:7090/api/Exercise/all

Server response
Code    Details
200
Response body
{
  "id": 15,
  "name": "HJFBJHBHJ",
  "targetMuscleGroup": {
    "id": 1,
    "name": "Back",
    "size": 0
  },
  "priority": 1,
  "jointness": 0,
  "supportMuscles": [
    {
      "id": 1,
      "name": "Back",
      "size": 0
    },
    {
      "id": 2,
      "name": "Chest",
      "size": 1
    }
  ],
  "equipments": [
    {
      "id": 2,
      "name": "Barbells"
    },
    {
      "id": 3,
      "name": "Bar"
    }
  ]
}
```

Рисунок 4.9 – Відповідь на отриманий запит з методом GET для вправ

Даний метод призначений для внесення змін в уже існуючі об'єкти. Тому для відправки такого запиту на сервер, необхідно вказати унікальний

ідентифікатор об'єкта (вправи), щоб серверна частина могла здійснити пошук в базі даних. Крім того, необхідно створити тіло запиту за аналогією з методом POST. На рисунку 4.10 показано процес створення такого запиту, де було змінено назву вправи на більш адекватну.



The screenshot shows a REST client interface for a PUT request to the endpoint `/api/Exercise/{exerciseId}`. The interface is divided into several sections:

- Method and Path:** PUT `/api/Exercise/{exerciseId}`
- Parameters:** A table with two columns: Name and Description. The parameter `exerciseld` is marked as required and has a value of `15` entered in the input field. The type is `integer($int32)` and the location is `(path)`.
- Request body:** A dropdown menu is set to `application/json-patch+json`.
- Request body content:** A JSON object is displayed in a text area:

```
{  "exerciseDto": {    "name": "ChinUps",    "targetMuscleGroupId": 1,    "priority": 1,    "jointness": "Single"  },  "supportMuscleGroups": [    1,2  ],  "equipments": [    3  ]}
```

Рисунок 4.10 – Створення запиту з методом PUT для об'єкта вправи

З рисунку 4.11 можна побачити, що код статусу відповіді серверу 200, тобто запит успішно виконався і в результаті сервер повернув оновлений об'єкт вправи, де вказана правильна назва. Аналогічним чином, було перевірено усі сервіси серверної частини, усі перевірки пройдено успішно.

Server response

Code	Details
200	<p>Response body</p> <pre>{ "id": 15, "name": "ChinUps", "targetMuscleGroup": { "id": 1, "name": "Back", "size": 0 }, "priority": 1, "jointness": 0, "supportMuscles": [{ "id": 1, "name": "Back", "size": 0 }, { "id": 2, "name": "Chest", "size": 1 }], "equipments": [{ "id": 1 }] }</pre>

Рисунок 4.11 – Відповідь на отриманий запит з методом PUT для вправ

4.4 Висновок до розділу 4

В даному розділі було проведено опис технологій та засобів розробки, а також наведено обґрунтування оптимальності їх вибору, враховуючи вимоги та особливості предметної області. Розроблено діаграму розгортання інформаційної технології та діаграму компонентів. В результаті, основним компонентом системи є Web Арі побудований за принципами REST. Web Арі має рівень бізнес логіки та рівень доступу до даних. Застосування різноманітних шаблонів проектування дозволяє розробити систему компоненти якої слабо зв'язані між собою. Таким чином отримано ряд важливих переваг: використання підходу транзакцій при зверненні до бази даних, гнучкість при зміні сховища даних, чистота написаного коду, можливість повторного використання коду, використання об'єктів заміників при тестуванні, підвищення масштабованості системи за рахунок впровадження нового функціоналу без необхідності змінювати існуючий код.

Крім того, було виконано тестування серверної частини інформаційної технології. Здійснене тестування кожного методу HTTP для усіх сервісів серверної частини, показало тільки успішні результати.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Аналіз ринку фітнес-застосунків

У 2020 році світовий ринок фітнес-застосунків оцінювався в 13,78 мільярда доларів США, а до 2030 року, за прогнозами, досягне 120,37 мільярда доларів США, зростаючи на 24,3% у середньому з 2021 по 2030 рік. [24] Індустрія фітнес-застосунків включає невеликі спеціалізовані програми, розроблені для підтримки мотивації користувачів. під час бігу, ходьби, їзди на велосипеді або будь-якої іншої фізичної активності. Крім того, більшість фітнес-застосунків використовуються для підрахунку калорій, інші ведуть статистичні записи про тренування або збирають дані про прогулянки, пробіжки та їзду на велосипеді. Крім того, фітнес-застосунки підключають користувача до особистого тренера або дієтолога, щоб допомогти вирішити проблеми під час використання певної вправи або просто під час тренувань. Крім того, застосунки також забезпечують узгоджену серію пісень, кожна з яких має однаковий ритм під час виконання тренувань, наприклад бігу та занять фітнесом.

Збільшення уваги до підтримки здорового способу життя та збільшення використання смартфонів, планшетів і переносних пристроїв позитивно впливає на зростання розміру ринку застосунків для фітнесу. Крім того, підвищення обізнаності про захворювання, пов'язані з дієтою, сприяє зростанню частки ринку фітнес-застосунків у всьому світі. Однак такі фактори, як технічні проблеми і висока вартість покупок у застосунках, а також занепокоєння щодо безпеки даних і суворі правила обмежують зростання ринку фітнесу. Навпаки, очікується, що технологічний прогрес у сфері штучного інтелекту та машинного навчання, а також зростання поширеності захворювань, таких як гіпертонія, ожиріння та серцеві проблеми, запропонують вигідні можливості для розширення ринку застосунків для фітнесу протягом прогнозованого періоду.

За типом, очікується, що сегмент фізичних вправ і схуднення отримає значну частку в індустрії застосунків для фітнесу протягом прогнозованого

періоду, через збільшення випадків ожиріння. Крім того, клінічні дослідники дають поради щодо корисності мобільних застосунків, і багато людей уже інтегрували їх у свої спроби контролювати вагу. Проте очікується, що сегмент відстеження активності зростатиме найвищими темпами протягом прогнозованого періоду завдяки зростанню попиту на системи відстеження здоров'я вбудовані в одяг та завдяки запуску нових пристроїв відстеження активності.

Якщо говорити про регіони, то у 2020 році на ринку застосунків для фітнесу домінувала Північна Америка, і очікується, що він спостерігатиме найвищі темпи зростання протягом прогнозованого періоду через збільшення попиту на модернізацію фітнес-систем у всьому регіоні. Крім того, очікується, що сплеск попиту на проведення безпроблемної та безризикової трансформації фітнесу серед закладів охорони здоров'я сприятиме зростанню ринку. Проте очікується, що індустрія застосунків для фітнесу в Азіатсько-Тихоокеанському регіоні спостерігатиме значне зростання протягом прогнозованого періоду через сплеск потреби відстежувати зростаючу кількість проблем із охороною здоров'я та фітнесом, як-от ожиріння [24].

Глобальний ринок фітнес-застосунків сегментовано за типом пристрою, операційною системою, типом і регіоном. Залежно від типу пристрою ринок поділяється на смартфони, планшети та носимі пристрої. Залежно від операційної системи він поділяється на iOS і Android. Залежно від типу він поділяється на тренування та схуднення, дієту та харчування, відстеження активності та інші. Що стосується регіонів, ринок вивчається в Північній Америці, Європі, Азіатсько-Тихоокеанському регіоні та Латинській Америці, Близькому Сході і Африці.

На рисунку 5.1 наведено розподіл ринку фітнес-застосунків за типом пристрою у порівнянні показників 2020 року з прогнозом на 2030р.

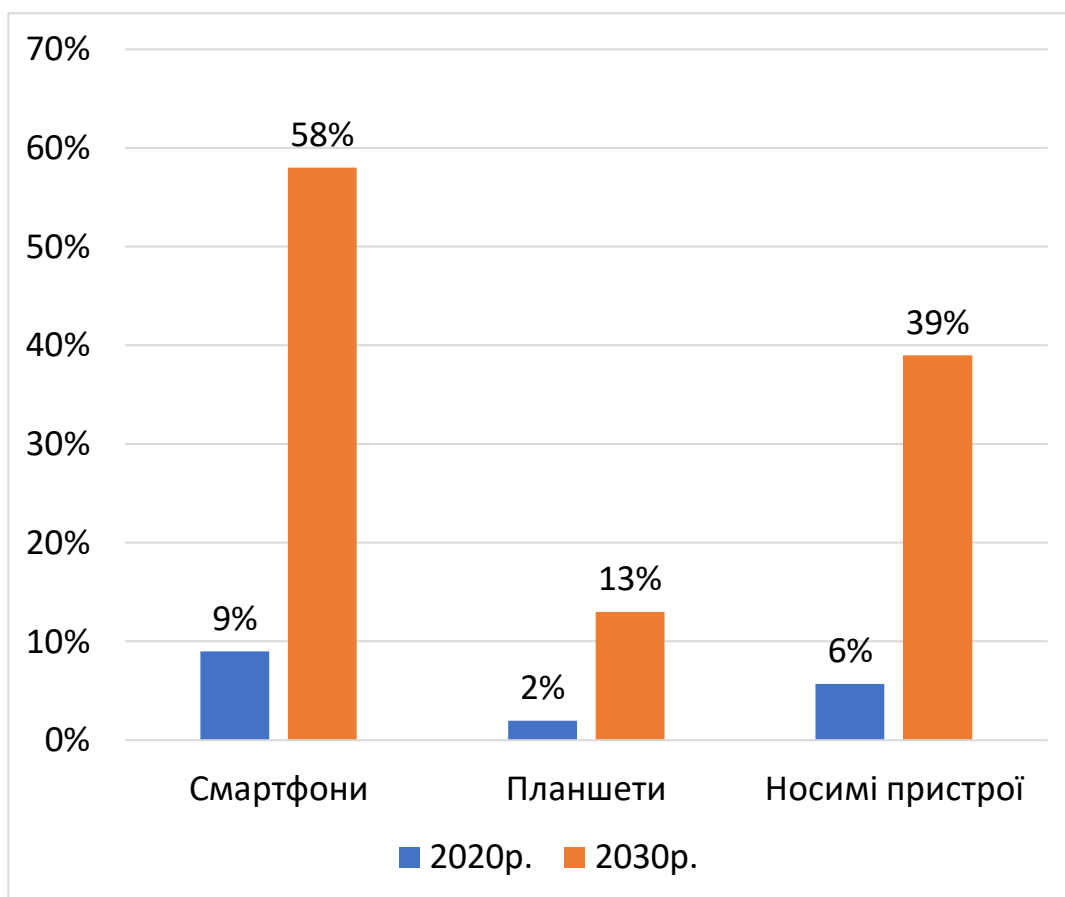


Рисунок 5.1 – Розподіл ринку фітнес-застосунків за типом пристрою

Ключовими гравцями під час аналізу ринку застосунків для фітнесу є: Aaptiv Inc, Addias, Applico Inc., Azumio, Inc., Fitbit LLC., FitnessKeeper, MyfitnessPal, Inc., Nike, Inc., Noom Inc. і Under Armour, Inc. Ці гравці прийняли різні стратегії, щоб збільшити своє проникнення на ринок і зміцнити свої позиції в індустрії застосунків для фітнесу.

Спалах COVID-19 суттєво вплинув на зростання ринку фітнес-застосунків, головним чином через запровадження карантину урядами більшості країн і впровадження роботи з дому та онлайн-навчання по всьому світу. Крім того, щоб утриматися на конкурентному ринку, компанії перейшли до цифровізації, щоб надавати онлайн-послуги, допомагаючи клієнтам порівнювати ціни з іншими конкурентами, вибирати правильні варіанти та пропозиції для здійснення процесів планування фітнесу онлайн. Стало зрозуміло, що COVID-19 матиме довгостроковий вплив на життя та економіку, тому відомі програми для тренувань, такі як Nike Training Club, Map My Run і Strava, стали свідками

збільшення кількості нових користувачів. Крім того, Peloton у відповідь зробив свій додаток безкоштовним протягом 90 днів. Очікується, що ці безкоштовні користувачі перетворяться на платних підписників застосунку або придбають підключене обладнання. Таким чином, очікується, що кількість таких розробок у всьому світі створить вигідні умови для розширення ринку.

На рисунку 5.2 наведено розподіл ринку фітнес-застосунків за типом операційної системи пристрою у порівнянні показників 2020 року з прогнозом на 2030р.

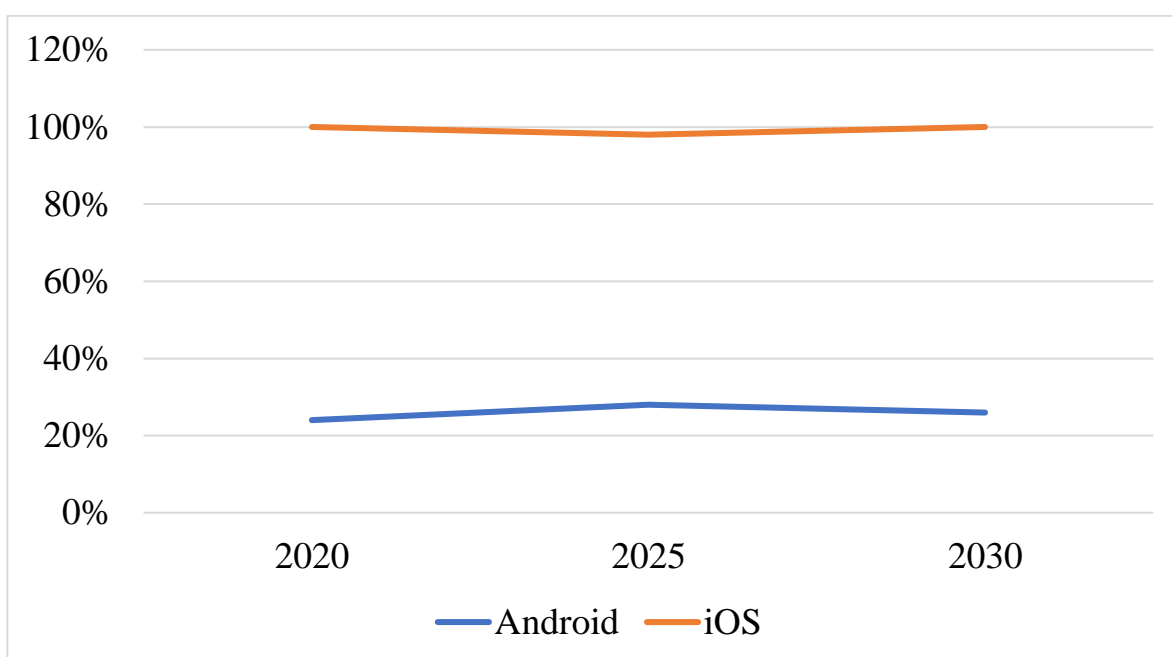


Рисунок 5.2 – розподіл ринку фітнес-застосунків за типом операційної системи пристрою

Операційна система iOS прогнозується як один із найприбутковіших сегментів, що логічно впливає з того, що найбільшим ринком на даний момент є США, де найрозповсюдженішою операційною системою є iOS.

У 2022 році на Північну Америку припадала найбільша частка доходу понад 37,1%, як показано на рисунку 5.3. Різноманітні фактори, такі як швидке зростання використання смартфонів, зростання мережі покриття, зростання поширеності хронічних захворювань, збільшення геріатричної популяції та триваюча пандемія COVID-19, спонукають до впровадження фітнес-застосунків

у Північній Америці. США є провідним ринком фітнес-застосунків у всьому світі. Одним із головних факторів, що сприяє зростанню ринку, є високе впровадження mHealth у Північній Америці. Згідно з опитуванням, проведеним Freeletics, під час пандемії COVID-19 використання фітнес-застосунків перевершило всі очікування, оскільки під час карантину, пов'язаного з COVID-19, 74,0% американців були свідками використання принаймні одного фітнес-застосунку. Крім того, 60,0% цих споживачів планували скасувати абонемент у спортзал.

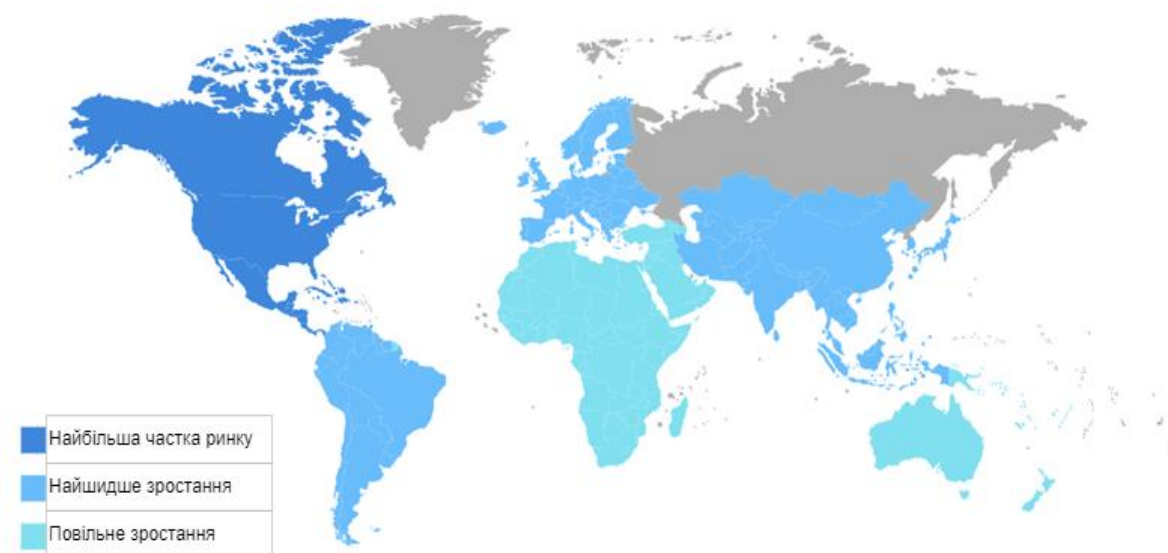


Рисунок 5.3 – Стан розподілу доходів ринку фітнес-застосунків за регіонами світу

Очікується, що в Азіатсько-Тихоокеанському регіоні буде найшвидший темп зростання в 18,9% протягом прогнозованого періоду завдяки зростаючому проникненню смартфонів і розумних пристроїв, що носяться, і зростаючому використанню послуг mHealth. Згідно зі звітом GSMA The Mobile Economy 2017, підключення смартфонів в Азіатсько-Тихоокеанському регіоні досягло 47% на кінець 2016 року та, за оцінками, досягне 62% до 2020 року. Зростання доступності призводить до збільшення використання смартфонів для доступу різні програми для фітнесу. Такі фактори, як зростання витрат на охорону

здоров'я, збільшення випадків ожиріння та зростання кількості спортсменів спонукають уряди та приватні організації розробляти нові моделі режимів фітнесу. Згідно зі статтею, опублікованою Всесвітнім економічним форумом, кількість завантажень програм для здоров'я та фітнесу в Індії зросла на 157%, що дорівнює 58 мільйонам нових користувачів застосунків.

5.2 Комерційний та технологічний аудит науково-технічної розробки

Метою проведення комерційно-технологічного аудиту є оцінка науково-технічного рівня, а також рівня комерційного потенціалу інформаційної технології для організації та проведення тренувального процесу, яку було розроблено в результаті науково-технічної діяльності, зокрема під час виконання магістерської кваліфікаційної роботи. Інформаційна технологія для організації та проведення тренувального процесу отримала назву «SmartFit Expert», та пройшла комерційно-технологічний аудит із залученням п'яти незалежних експертів, інформація про яких надана в таблиці 5.1.

Таблиця 5.1 – Інформація про експертів, що приймали участь в комерційному та технологічному аудиті інформаційної технології для організації та проведення тренувального процесу

№	Експерт	Посада	Місце роботи
1	Ben Milton	Team Lead	Exercise.com
2	Коваленко Ольга Олександрівна	Business Analyst	ScienceSoft
3	Andrzej Jaworski	Business Analyst	Mighty Pro
4	Todor Hristov	Team Lead	Trainerize
5	Назаренко Вероніка Сергіївна	Data Analyst	Innowise Group

У таблиці 5.2 наведено 12 критеріїв, за якими оцінюється науково-технічний рівень розробки, а також її комерційна доцільність за 5-бальною системою.

Таблиця 5.2 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка.

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	1	2	3	4	5
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає

Продовження таблиці 5.2.

Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Експертні оцінки науково-технічного рівня і комерційного потенціалу науково-технічної розробки інформаційної технології для організації та проведення тренувального процесу наведено в таблиці 5.3.

Таблиця 5.3 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки.

Критерії	Експерт (ПІБ, посада)				
	1	2	3	4	5
	Бали:				
1. Технічна здійсненність концепції	4	3	4	4	4
2. Ринкові переваги (наявність аналогів)	3	3	4	2	3
3. Ринкові переваги (ціна продукту)	4	5	4	4	4
4. Ринкові переваги (технічні властивості)	3	3	3	2	3
5. Ринкові переваги (експлуатаційні витрати)	3	3	2	3	3
6. Ринкові перспективи (розмір ринку)	3	2	3	3	3
7. Ринкові перспективи (конкуренція)	3	3	2	3	4
8. Практична здійсненність (наявність фахівців)	4	4	4	2	3
9. Практична здійсненність (наявність фінансів)	1	1	1	2	3
10. Практична здійсненність (необхідність нових матеріалів)	3	4	3	3	4
11. Практична здійсненність (термін реалізації)	2	2	2	3	3
12. Практична здійсненність (розробка документів)	3	4	3	3	4
Сума балів	36	37	35	34	41
Середньоарифметична сума балів $СБ_c$	36,6				

Середньоарифметична сума балів $СБ_c$ (описана в таблиці 5.3) може бути використана для прийняття обґрунтованого рішення з приводу науково-технічного рівня розробки, а також її комерційної доцільності. Щоб досягти цього, використаємо рекомендації, наведені в таблиці 5.4.

Таблиця 5.4 – Науково-технічні рівні та комерційні потенціали розробки.

Середньоарифметична сума балів CB_c , розрахована на основі висновків експертів	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вищий середнього
21...30	Середній
11...20	Нижчий середнього
0...10	Низький

Середньоарифметична сума балів CB_c становить 36,6 та ґрунтується на висновках п'яти експертів і означає науково-технічний або комерційний потенціал розробки вище середнього. Поєднання евристичних методів, таких як нечітка логіка та експертні системи, було використане для досягнення більшого рівня точності та відображення більш реального стану речей в ситуаціях, коли недостатньо достовірних знань і досліджень. Крім того, розроблена інформаційна технологія для організації та проведення тренувального процесу забезпечує простоту тестування, зручність розробки та внесення змін в систему, завдяки використанню зрозумілих і добре організованих баз знань. Варто відзначити, що вартість використання розробки значно нижча, ніж у аналогічних програм. На основі проведених досліджень за отриманими результатами можна відзначити перспективність і привабливість для інвестицій в розробку.

Оцінимо рівень конкурентоспроможності розробки інформаційної технології для організації та проведення тренувального процесу. Найближчими конкурентами на ринку є такі програми-аналоги: FitnessAI, Fitbod, Fitbit Coach, Fitness Buddy та 8fit. Порівнюючи власну розробку з конкурентами, можна оцінити конкурентоспроможність базуючись на системі економічних та якісних показників. Економічні показники конкурентоспроможності базуються на загальній сумі витрат споживачів, необхідних для задоволення їх потреб за допомогою розробки (застосунку). Якісні показники відображають властивості,

які дозволяють задовольнити конкретні вимоги споживачів. Оцінка конкурентоспроможності інформаційних технологій проходитиме в три етапи.

Спочатку необхідно знайти одиничні параметричні індекси. Якісні одиничні параметричні індекси визначаються за їх технічними показниками (показниками якості) та розраховуються за відповідними формулами.

Одиничний параметричний індекс розраховується за формулою 5.1, якщо збільшення величини параметра свідчить про підвищення якості нової розробки:

$$q_i = \frac{P_i}{P_{\text{баз } i}}, \quad (5.1)$$

де q_i – одиничний параметричний індекс, розрахований за i -м параметром;

P_i – значення i -го параметра розробки;

$P_{\text{баз } i}$ – аналогічний параметр базової розробки-аналога, з якою проводиться порівняння.

Одиничний параметричний індекс обчислюється за оберненою формулою 5.2, якщо зменшення величини параметра свідчить про підвищення якості нової розробки:

$$q_i = \frac{P_{\text{баз } i}}{P_i}. \quad (5.2)$$

Для розрахунку одиничного параметричного індексу необхідно визначити основні параметри розробки, які впливають на рівень конкурентоспроможності застосунку. Основними якісними параметрами є:

- врахування антропометричних, вікових та статевих особливостей користувача при створенні програми тренувань;
- врахування необхідного часу для відновлення при формуванні тижневого плану тренувань, з метою уникнення стану «перетренованості»;
- відповідність складеної програми тренувань цілям користувача;
- підбір вправ у програму тренувань відповідно до можливостей та оснащення тренувального місця користувача;

- надання рекомендацій щодо раціону харчування в залежності від створеної програми тренувань;
- можливість зберігання даних користувача у застосунку (програми тренувань, результати, історія користування);
- наявність безкоштовної версії застосунку, безкоштовного функціонала, або пробної версії;
- зручність застосунку та дизайн (UX/UI);
- інтеграція до застосунку даних з фітнес-гаджетів.

До економічних параметрів відносять – ціну застосунку.

Якісні параметри, що впливають на рівень конкурентоспроможності розробки «SmartFit Expert» на ринку фітнес-застосунків були оцінені експертами (див. табл. 5.1) за 10-бальною шкалою. Середні оцінки 5-ти експертів були внесені у таблицю 5.5

Таблиця 5.5 – Експертна оцінка якісних параметрів фітнес-застосунків

№	Параметри:	Застосунки:					
		SmartFit Expert	FitnessAI	Fitbod	Fitbit Coach	Fitness Buddy	8fit
1	Відповідність складеної програми тренувань цілям користувача	9	8	7	8	5	6
2	Підбір вправ до програми тренувань відповідно до можливостей та оснащення тренувального місця користувача	7	5	8	3	2	3
3	Врахування вікових особливостей при створенні програми тренувань	8	6	7	6	3	4
4	Врахування статевих особливостей при створенні програми тренувань	7	5	7	6	5	5
5	Врахування антропометричних особливостей користувача при створенні програми тренувань	6	3	4	2	1	2
6	Надання рекомендацій щодо раціону харчування в залежності від створеної програми тренувань	6	2	3	4	1	9
7	Врахування необхідного часу для відновлення при формуванні тижневого плану тренувань, з метою уникнення стану «перетренованості»	8	7	8	7	6	5

Продовження таблиці 5.5

8	Можливість зберігання даних користувача у застосунку (програми тренувань, результати, історія користування)	4	6	8	8	6	7
9	Інтеграція до застосунку даних з фітнес-гаджетів	2	4	5	9	2	6
10	Наявність безкоштовної версії застосунку, безкоштовного функціонала, або пробної версії	7	2	6	6	3	5
11	Зручність застосунку та дизайн (UX/UI)	5	6	6	8	5	7

Оскільки, збільшення величини усіх параметрів буде свідчити про підвищення якості нової розробки, то одиничний параметричний індекс розраховується за формулою 5.1. Результати проведеного оцінювання наведено у таблиці 5.6.

Таблиця 5.6 – Результати розрахунку одиничних якісних параметричних індексів для SmartFit Expert відносно кожного застосунку-аналога

№	Параметри:	Застосунки:				
		FitnessAI	Fitbod	Fitbit Coach	Fitness Buddy	8fit
		q_i	q_i	q_i	q_i	q_i
1	Відповідність складеної програми тренувань цілям користувача	1,13	1,29	1,13	1,8	1,5
2	Підбір вправ до програми тренувань відповідно до можливостей та оснащення тренувального місця користувача	1,4	0,88	2,33	3,5	2,33
3	Врахування вікових особливостей при створенні програми тренувань	1,33	1,14	1,33	2,67	2
4	Врахування статевих особливостей при створенні програми тренувань	1,4	1	1,17	1,4	1,4
5	Врахування антропометричних особливостей користувача при створенні програми тренувань	2	1,50	3	6	3
6	Надання рекомендацій щодо раціону харчування в залежності від створеної програми тренувань	3	2	1,5	6	0,67
7	Врахування необхідного часу для відновлення при формуванні тижневого плану тренувань, з метою уникнення стану "перетренованості"	1,14	1	1,14	1,33	1,6

Продовження таблиці 5.6

8	Можливість зберігання даних користувача у застосунку (програми тренувань, результати, історія користування)	0,67	0,5	0,5	0,67	0,57
9	Інтеграція до застосунку даних з фітнес-гаджетів	0,5	0,4	0,22	1	0,33
10	Наявність безкоштовної версії застосунку, безкоштовного функціонала, або пробної версії	3,5	1,17	1,17	2,33	1,4
11	Зручність застосунку та дизайн (UX/UI)	0,83	0,83	0,63	1	0,71

Визначено, що на рівень конкурентоспроможності розробки «SmartFit Expert» на ринку фітнес-застосунків впливає лише один економічний параметр – ціна застосунку. Преміум-підписка SmartFit Expert коштує 89 гривень на місяць. Проведемо розрахунок одиничного економічного параметричного індексу для розробки відносно кожного застосунку-аналога:

- Fitbod, при ціні 175 грн. $q_i = 89/175 = 0,51$;
- Fitbit Coach, при ціні 265 грн. $q_i = 89/265 = 0,34$;
- FitnessAI, при ціні 310 грн. $q_i = 89/310 = 0,29$;
- Fitness Buddy, при ціні 91,7 грн. $q_i = 89/91,7 = 0,97$;
- 8fit, при ціні 540 грн. $q_i = 89/540 = 0,16$.

Під час другого етапу, були визначені групові параметричні індекси. Експертами було визначено вагомі коефіцієнти кожного якісного та економічного параметра, щоб з'ясувати важливість кожного параметра для користувача, при виборі фітнес-застосунку. У таблиці 5.6 показано зміни значення параметра, причому коли воно більше 1 – то параметр якості розробки вищий, ніж у конкурента, якщо ж менше 1 – то нижчий, якщо вони рівні – то жоден немає переваги. Для економічного параметра – все навпаки. Для полегшення подальших розрахунків, наведемо отримані значення в таблиці 5.7.

Таблиця 5.7 – Індекс зміни значення і коефіцієнти вагомості якісних та економічних параметрів.

№	Параметри	Застосунки:					Коефіцієнт вагомості
		FitnessAI	Fitbod	Fitbit Coach	Fitness Buddy	8fit	
	Якісні параметри:	Індекс зміни значення параметра					α_i
1	Відповідність складеної програми тренувань цілям користувача	1,13	1,29	1,13	1,8	1,5	0,21
2	Підбір вправ до програми тренувань відповідно до можливостей та оснащення тренувального місця користувача	1,4	0,88	2,33	3,5	2,33	0,06
3	Врахування вікових особливостей при створенні програми тренувань	1,33	1,14	1,33	2,67	2	0,11
4	Врахування статевих особливостей при створенні програми тренувань	1,4	1	1,17	1,4	1,4	0,09
5	Врахування антропометричних особливостей користувача при створенні програми тренувань	2	1,5	3	6	3	0,05
6	Надання рекомендацій щодо раціону харчування в залежності від створеної програми тренувань	3	2	1,5	6	0,67	0,07
7	Врахування необхідного часу для відновлення при формуванні тижневого плану тренувань, з метою уникнення стану "перетренованості"	1,14	1	1,14	1,33	1,6	0,04
8	Можливість зберігання даних користувача у застосунку (програми тренувань, результати, історія користування)	0,67	0,5	0,5	0,67	0,57	0,07
9	Інтеграція до застосунку даних з фітнес-гаджетів	0,5	0,4	0,22	1	0,33	0,09
10	Наявність безкоштовної версії застосунку, безкоштовного функціонала, або пробної версії	3,5	1,17	1,17	2,33	1,4	0,08
11	Зручність застосунку та дизайн (UX/UI)	0,83	0,83	0,63	1	0,71	0,13
	Економічні параметри:	Індекс зміни значення параметра					β_i
12	Ціна застосунку	0,29	0,51	0,34	0,97	0,16	1

Нормативні параметри оцінюються показником, який приймає два значення: 1 – товар відповідає нормам і стандартам; 0 – не відповідає. Однак, на ринку фітнес-застосунків відсутні чітко визначені нормативні параметри, тому беруться до уваги певні вимоги для нормальної роботи застосунків:

- безпека персональних даних користувачів;
- безпека фінансових операцій, передбачених функціональними можливостями самого застосунку;
- відповідність заявленим характеристикам;
- стабільна робота на усіх заявлених розробником пристроях;
- результат роботи застосунку повинен задовольняти очікування користувачів.

Добуток частинних показників за кожним нормативним параметром складають груповий показник конкурентоспроможності за нормативними параметрами, який обчислюється за формулою 5.3:

$$I_{\text{НП}} = \prod_{i=1}^n q_i, \quad (5.3)$$

де $I_{\text{НП}}$ – загальний показник конкурентоспроможності за нормативними параметрами;

q_i – одиничний (частинний) показник за i -м нормативним параметром;

n – кількість нормативних параметрів, які підлягають оцінюванню.

Під час розрахунку групового показника конкурентоспроможності за нормативними параметрами, було перевірено відповідність кожного параметра вимогам до нормальної роботи застосунків. Якщо параметр відповідає вимогам – його було оцінено, як 1, якщо не відповідає – 0. Отримані результати наведено в таблиці 5.8.

Таблиця 5.8 – Оцінка нормативних параметрів роботи фітнес-застосунків

№	Нормативні вимоги (параметри)	SmartFit Expert	FitnessAI	Fitbod	Fitbit Coach	Fitness Buddy	8fit
1	Відповідність заявленим характеристикам	1	1	1	1	1	1
2	Стабільна робота на усіх заявлених розробником пристроях	1	1	1	1	1	1
3	Безпека персональних даних користувачів	1	1	1	1	1	1
4	Безпека фінансових операцій та транзакцій	1	1	1	1	1	1
5	Результат роботи застосунку задовольняє очікування користувачів	1	1	1	1	1	1

За результатами з таблиці 5.8, можна підтвердити виконання усіх вищезазначених нормативних вимог до нормальної роботи застосунків. Отже, груповий показник конкурентоспроможності за нормативними параметрами, для кожного застосунку $I_{ГП} = 1$, тобто усі застосунки є конкурентоспроможними.

Визначимо груповий параметричний індекс за якісними параметрами, враховуючи вагомість кожного параметра. Для цього розрахунок використаємо формулу 5.4:

$$I_{ГП} = \sum_{i=1}^n (q_i \cdot \alpha_i), \quad (5.4)$$

де $I_{ГП}$ – груповий параметричний індекс за якісними показниками (порівняно з аналогом);

q_i – одиничний параметричний показник i -го параметра;

α_i – вагомість i -го параметричного показника, $\sum_{i=1}^n \alpha_i = 1$;

n – кількість якісних параметрів, за якими оцінюється конкурентоспроможність.

Розрахуємо груповий параметричний індекс за якісними параметрами для кожного застосунку-аналога відносно SmartFit Expert.

FitnessAI:

$$I_{\text{III}} = (1,13 \cdot 0,21) + (1,4 \cdot 0,06) + (1,33 \cdot 0,11) + (1,4 \cdot 0,09) + (2 \cdot 0,05) + (3 \cdot 0,07) + (1,14 \cdot 0,04) + (0,67 \cdot 0,07) + (0,5 \cdot 0,09) + (3,5 \cdot 0,08) + (0,83 \cdot 0,13) = 1,43;$$

Fitbod: $I_{\text{III}} = 1,15$;

Fitbit Coach: $I_{\text{III}} = 1,16$;

Fitness Buddy: $I_{\text{III}} = 2,23$;

8fit: $I_{\text{III}} = 2,23$.

Можна зробити висновок про якісну перевагу SmartFit Expert над своїми основними конкурентами, адже усі значення групового параметричного індексу за якісними параметрами виявились більшими за одиницю. Найближчими конкурентами за якістю є Fitbod та Fitbit Coach, які відстають на 15% і 16% відповідно.

Визначимо груповий параметричний індекс за економічними параметрами (за ціною застосунку). Для цього розрахунку використаємо формулу 5.5:

$$I_{\text{EП}} = \sum_{i=1}^m (q_i \cdot \beta_i), \quad (5.5)$$

де $I_{\text{EП}}$ – груповий параметричний індекс за економічними показниками (порівняно з аналогом);

q_i – економічний параметр i -го виду;

β_i – частка i -го економічного параметра, $\sum_{i=1}^m \beta_i = 1$;

m – кількість економічних параметрів, за якими здійснюється оцінювання.

Проведемо розрахунок групового параметричного індексу за економічними параметрами для кожної програми-аналога відносно SmartFit Expert.

FitnessAI: $I_{\text{EП}} = 0,29 \cdot 1 = 0,29$;

Fitbod: $I_{\text{EП}} = 0,51$;

Fitbit Coach: $I_{\text{EП}} = 0,34$;

Fitness Buddy: $I_{\text{EП}} = 0,97$;

8fit: $I_{\text{EП}} = 0,16$.

Можна зробити висновок, що SmartFit Expert вигідніше своїх основних конкурентів, адже усі отримані значення групового параметричного індексу за економічними параметрами менші за одиницю. Найближчим конкурентом за вартістю є Fitness Buddy, що поступається лише на 3%.

На останньому етапі, проведено розрахунок інтегрального показника конкурентоспроможності. Використовуючи отримані значення групових параметричних індексів за нормативними, якісними та економічними показниками, було визначено інтегральний показник за формулою 5.6:

$$K_{\text{ИИТ}} = I_{\text{ИП}} \cdot \frac{I_{\text{ТП}}}{I_{\text{ЕП}}} . \quad (5.6)$$

Проведемо розрахунки інтегрального показника конкурентоспроможності для кожного застосунку-аналога відносно SmartFit Expert.

FitnessAI: $K_{\text{ИИТ}} = 1 \cdot (1,43/0,29) = 4,98$;

Fitbod: $K_{\text{ИИТ}} = 2,26$;

Fitbit Coach: $K_{\text{ИИТ}} = 3,45$;

Fitness Buddy: $K_{\text{ИИТ}} = 2,3$;

8fit: $K_{\text{ИИТ}} = 8,11$.

Розрахунки показують, що усі інтегральні показники конкурентоспроможності більші за одиницю, що свідчить про те, що розробка перевершує своїх найближчих конкурентів. Слід підкреслити, що значення інтегрального показника в першу чергу обумовлено низькою вартістю програми SmartFit Expert. Висока якість і багата функціональність Fitbod роблять його сильним конкурентом, тоді як Fitness Buddy відносно недорогий. Підсумовуючи, представлені результати дослідження підтверджують високий рівень конкурентоспроможності та перспективності розробки, а також свідчать про значний потенціал для комерціалізації застосунку SmartFit Expert.

5.3 Розрахунок витрат на проведення науково-дослідної роботи

Витрати на проведення науково-дослідної, конструкторсько-технологічної, дослідно-конструкторської роботи, створення дослідного зразка і проведенням тестувань розробки, під час планування, обліку і калькулювання собівартості науково-дослідної роботи можуть бути класифіковані наступним чином:

- витрати на оплату праці;
- витрати на придбання обладнання;
- програмне забезпечення для наукових (експериментальних) робіт;
- енергія для науково-виробничих цілей;
- витрати на забезпечення діяльності офісу (місця роботи);
- відрахування на соціальні заходи;
- накладні (загальновиробничі) витрати.

Визначимо витрати на оплату праці. До цієї групи витрат включається виплата основної та додаткової заробітної плати працівникам, науковцям, розробникам, які безпосередньо залучені до виконання конкретних завдань і обов'язків, розрахованих із застосуванням посадових окладів, відрядних розцінок, тарифних ставок до встановленого окладу.

Проведемо розрахунок основної заробітної плати дослідників, розробників та працівників. Витрати на основну заробітну плату дослідників, розробників та працівників (Z_o) розраховують відповідно до посадових окладів, за формулою 5.7:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (5.7)$$

де k – кількість посад дослідників, залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – кількість днів роботи конкретного дослідника, дн.;

T_p – середня кількість робочих днів в місяці, $T_p = 20$ днів.

Внесемо проведені розрахунки основної заробітної плати до таблиці 5.9.

Таблиця 5.9 – Витрати на заробітну плату дослідників, розробників та працівників.

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Кількість днів роботи	Витрати на заробітну плату, грн
Керівник проекту	42 000	2 100	108	226 800
Керівник відділу розробки	40 000	2 000	102	204 000
Програміст (серверна частина)	36 000	1 800	71	127 800
Програміст (клієнтська частина)	32 000	1 600	64	102 400
Спеціаліст з продажу та роботі з клієнтами	24 000	1 200	49	58 800
UX/UI дизайнер	18 000	900	18	16 200
Бухгалтер-фінансист	21 000	1 050	5	5 250
Маркетолог	20 000	1 000	26	26 000
Юрист	22 000	1 100	3	3 300
Всього				770 550

$$\begin{aligned}
 Z_o = & \frac{42000 \cdot 108}{20} + \frac{40000 \cdot 102}{20} + \frac{36000 \cdot 71}{20} + \frac{32000 \cdot 64}{20} + \frac{24000 \cdot 49}{20} + \frac{18000 \cdot 18}{20} \\
 & + \frac{21000 \cdot 5}{20} + \frac{20000 \cdot 26}{20} + \frac{22000 \cdot 3}{20} = 770\,550 \text{ грн.}
 \end{aligned}$$

Загальний розмір витрат на заробітну плату науковців, розробників і працівників, які займаються дослідженням та розробкою інформаційної технології для організації та проведення тренувального процесу, становить 770 550 грн.

Проведемо розрахунок додаткової заробітної плати дослідників, розробників та працівників. Дослідникам, розробникам і робітникам розраховується додаткова заробітна плата в розмірі 12% від суми основної заробітної плати, за формулою 5.8:

$$З_{\text{дод}} = З_0 \cdot \frac{H_{\text{дод}}}{100\%}, \quad (5.8)$$

де $H_{\text{дод}}$ – норма нарахування додаткової заробітної плати.

$$З_{\text{дод}} = 770\,550 \cdot \frac{12\%}{100\%} = 92\,466 \text{ грн.}$$

Загальний розмір витрат на додаткову заробітну плату дослідників, розробників і працівників при проведенні дослідження та розробки інформаційної технології організації та проведення тренувального процесу складає 92 466 грн.

Проведемо розрахунок відрахувань на соціальні заходи. Ця група витрат включає відрахування внеску на загальнообов'язкове державне соціальне страхування та впровадження заходів ЄСВ на соціальний захист.

Відповідно до формули 5.9 нарахування на соціальні заходи для дослідників та робітників обчислюється як 22% від суми їх основної та додаткової заробітної плати:

$$З_{\text{н}} = (З_0 + З_{\text{дод}}) \cdot \frac{H_{\text{зп}}}{100\%}, \quad (5.9)$$

де $H_{\text{зп}}$ – норма нарахування на заробітну плату.

$$З_{\text{н}} = (770\,550 + 92\,466) \cdot \frac{22\%}{100\%} = 189\,863,52 \text{ грн.}$$

Загальна сума відрахувань на соціальні заходи дослідників, розробників і працівників складає 189 863,52 грн.

Проведемо розрахунок витрат на придбання обладнання. Дана група витрат охоплює витрати, понесені на закупівлю обладнання та іншого оснащення для персоналу, дослідників і розробників під час дослідження та розробки

інформаційної технології. Для визначення загальної суми витрат на придбання обладнання застосуємо формулу 5.10:

$$B_o = \sum P_o \cdot k_o , \quad (5.10)$$

де P_o – ціна за одиницю необхідного обладнання, або оснащення;

k – кількість необхідного обладнання, або оснащення.

Детальний розрахунок витрат на придбання необхідного обладнання, а також оснащення для дослідників, розробників та працівників наведено в таблиці 5.10.

Таблиця 5.10 – Витрати на придбання обладнання та оснащення.

Найменування обладнання та оснащення	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Ноутбук HP Pavilion 15-eh2038ua	5	27 999	139 995
Монітор 23.8" HP Omen 24	3	8 169	24 507
Комп'ютерна миша Logitech M500S Advanced	3	1 600	4 800
Килимок для комп'ютерної миші A4Tech X7-300MP Speed	3	169	507
Крісло офісне Ultra C-11	6	2 849	17 094
Стіл офісний Luxe Studio 2	6	2 842	17 052
Шафа офісна картотечна Emby CF-495-3	1	7 690	7 690
Шафа гардеробна БЮ 409	1	2 876	2 876
Кулер для води ABC D270F	1	1 799	1 799
БФП Canon i-Sensys MF272dw, Wi-Fi	1	10 999	10 999
Маршрутизатор TP-LINK Archer A64	1	1 399	1 399
Всього			228 718

$$B_o = (27999 \cdot 5) + (8169 \cdot 3) + (1600 \cdot 3) + (169 \cdot 3) + (2849 \cdot 6) + (2842 \cdot 3) + (7690 \cdot 1) + (2876 \cdot 1) + (1799 \cdot 1) + (10999 \cdot 1) + (1399 \cdot 1) = 228\,718 \text{ грн}$$

Загальний розмір витрат на придбання обладнання та оснащення, що необхідні для дослідників, розробників і працівників складає 228 718 грн.

Проведемо розрахунок витрат на забезпечення діяльності офісу (місяця роботи). Ця група витрат охоплює витрати, пов'язані з придбанням або орендою необхідних приміщень, а також із забезпеченням необхідних умов праці для дослідників чи розробників та співробітників. Розрахувати загальні витрати на забезпечення офісної діяльності можна за формулою 5.11:

$$B_{wp} = \left(\left(\sum P_{wp} \cdot k_{wp} \right) + P_{of} \right) \cdot n, \quad (5.11)$$

де P_{wp} – ціна за одиницю необхідного товару, чи послуги;

k_{wp} – необхідна кількість товару, чи послуги;

P_{of} – ціна на оренду необхідного приміщення;

n – кількість місяців необхідних для дослідження та розробки.

Наведемо детальні розрахунки витрат для забезпечення необхідних робочих умов для дослідників, розробників та співробітників в таблиці 5.11.

Таблиця 5.11 – Витрати на забезпечення діяльності офісу.

Найменування товару чи послуги	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Чорнила для Canon, PIXMA G1411 Canon	1	687	687
Папір офісний HP Home&Office A4	1	166	166
Доставка води	11	120	1 320
Оренда офісного приміщення, 45м ²	1	169	507
Кількість місяців необхідних для дослідження та розробки			6
Всього			53 538

$$B_{wp} = (((687 \cdot 1) + (166 \cdot 1) + (120 \cdot 11)) + 6750) \cdot 6 = 53\,538 \text{ грн}$$

Загальна сума витрат на забезпечення необхідних робочих умов для дослідників, розробників та працівників складає 53 538 грн.

Проведемо розрахунок витрат на програмне забезпечення для наукових (експериментальних) робіт. Ця група витрат включає придбання спеціалізованих програмних засобів і програмного забезпечення для дослідницького використання, а також витрати на проектування, розробку та встановлення таких засобів. Точні деталі придбання програмного забезпечення необхідно брати до уваги під час розрахунку витрат, оскільки деякі програмні продукти продаються за місячною передплатою, а інші купуються та використовуються без оплати.

Для розрахунку загальних витрат на програмне забезпечення для наукових досліджень та розробки використовується формула 5.12:

$$B_{\text{прг}} = \left(\left(\sum P_{\text{sub}} \cdot k_{\text{sub}} \right) \times n_{\text{sub}} \right) + \left(\sum P_{\text{sw}} \cdot k_{\text{sw}} \right), \quad (5.12)$$

де P_{sub} – ціна за щомісячне користування програмним продуктом;

k_{sub} – кількість щомісячних підписок на програмний продукт;

n_{sub} – кількість місяців підписки на програмний продукт;

P_{sw} – ціна за програмний продукт;

k_{sw} – кількість програмних продуктів.

Наведемо детальні розрахунки витрат на придбання спеціальних програмних засобів і програмного забезпечення для дослідників, розробників та працівників в таблиці 5.12.

Таблиця 5.12 – Витрати на забезпечення діяльності офісу.

Найменування програмного забезпечення	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Середовище розробки MATLAB	1	2 980	2 980
JetBrains All Products Pack	3	1 098	3 294
Azure Cloud	1	2 482	2 482
Microsoft 365	1	2 599	2 599
Кількість місяців необхідних для дослідження та розробки			6
Всього			68 130

$$V_{\text{прог}} = ((2980 \cdot 1) + (1098 \cdot 3) + (2482 \cdot 1) + (2599 \cdot 1)) \cdot 6 = 68\,130 \text{ грн.}$$

Загальна сума витрат на програмне забезпечення для дослідників, розробників та працівників складає 68 130 грн.

Проведемо розрахунок витрат на амортизацію обладнання, програмних засобів та приміщень. Ця категорія витрат включає амортизаційні відрахування на обладнання, прилади та програмне забезпечення, яке використовується в науково-дослідній роботі, якщо воно є в науково-дослідній організації чи підприємстві. За формулою 5.13 амортизаційні відрахування обладнання, приміщень, програмного забезпечення та інших об'єктів можна спростити за допомогою прямолінійного методу амортизації.

$$A_{\text{обл}} = \frac{Ц_{\text{б}}}{T_{\text{в}}} \cdot \frac{t_{\text{вик}}}{12}, \quad (5.13)$$

де $Ц_{\text{б}}$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{\text{вик}}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_{\text{в}}$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

Наведемо детальні розрахунки витрат на амортизацію обладнання, програмних засобів та приміщень в таблиці 5.13.

Таблиця 5.13 – Витрати на амортизаційні відрахування.

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Ноутбук HP Pavi-lion 15-eh2038ua	139 995	2	6	34 998,75
Монітор 23.8" HP Omen 24	24 507	2	6	6 126,75
Комп'ютерна миша Logitech M500S Advanced	4 800	2	6	1 200
Крісло офісне Ultra C-11	17 094	4	6	2 136,75
Стіл офісний Luxe Studio 2	17 052	4	6	2 131,5
Шафа офісна картотечна Emby CF-495-3	7 690	4	6	961,25
Шафа гардеробна БЮ 409	2 876	4	6	359,50
БФП Canon i-Sen-sys MF272dw, Wi Fi	10 999	4	6	1 374,88
Всього				49 289,38

$$A_{\text{обл}} = \left(\frac{139995}{2} \cdot \frac{6}{12} \right) + \left(\frac{24507}{2} \cdot \frac{6}{12} \right) + \left(\frac{4800}{2} \cdot \frac{6}{12} \right) + \left(\frac{17094}{4} \cdot \frac{6}{12} \right) + \left(\frac{17052}{4} \cdot \frac{6}{12} \right) + \left(\frac{7690}{4} \cdot \frac{6}{12} \right) + \left(\frac{2876}{4} \cdot \frac{6}{12} \right) + \left(\frac{10999}{4} \cdot \frac{6}{12} \right) = 49\,289,38 \text{ грн.}$$

Загальна сума витрат на амортизацію обладнання, програмних засобів та приміщень складає 49 289,38 грн.

Проведемо розрахунок витрат на енергію для науково-виробничих цілей. Ця група витрат складається з закупівлі електроенергії в неурядових організацій, фірм та установ, які здійснюють дослідження та розробки для технологічних цілей. Дані витрати формуються у разі виконання енергоємних наукових

досліджень, за методом прямого введення витрат і досягає значної питомої ваги у собівартості досліджень. Витрати на силову електроенергію (B_e) розраховують за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{впi}}{\eta_i}, \quad (5.14)$$

де W_{yi} – встановлена потужність обладнання на певному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн;

$K_{впi}$ – коефіцієнт, що враховує використання потужності, $K_{впi} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = 0,511 \cdot 720 \cdot 7,5 \cdot (0,95/0,97) = 2\,702,51 \text{ грн.}$$

Загальна сума витрат на енергію для науково-виробничих цілей складає 2 702,51 грн.

Проведемо розрахунок накладних (загальновиробничих) витрат. Група накладних (загальновиробничих) витрат розраховується у розмірі 110% основної заробітної плати наукових працівників, розробників і службовців за формулою 5.15 яка включає витрати на організаційне управління, раціоналізацію винахідництва, навчання персоналу, витрати на наймання робочої сили, банківське обслуговування. фінансування платежів, витрати на освоєння продукції.

$$B_{нзв} = Z_o \cdot \frac{H_{нзв}}{100\%}, \quad (5.15)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$B_{нзв} = 770\,550 \cdot 1,1 = 847\,605 \text{ грн.}$$

Загальна сума накладних (загальновиробничих) витрат складає 847 605 грн.

Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою 5.16:

$$B_{\text{заг}} = Z_o + Z_{\text{дод}} + Z_n + B_o + B_{\text{wr}} + B_{\text{прг}} + A_{\text{обл}} + B_e + B_{\text{нзв}}, \quad (5.16)$$

$$B_{\text{заг}} = 770550 + 92466 + 189863,52 + 228718 + 53538 + 68130 + 49289,38 + 2702,51 + 847605 = 2\,302\,862,4 \text{ грн.}$$

Загальна сума витрат на проведення науково-дослідної роботи складає 2 302 862,4 грн.

Розподіл витрат на проведення науково-дослідної роботи за групами витрат наведено на рисунку 5.4



Рисунок 5.4 - Розподіл за групами витрат

За формулою 5.17 можна розрахувати загальні витрати ZB на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів.

$$ZB = \frac{B_{\text{заг}}}{\eta}, \quad (5.17)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оскільки розробка, на даний момент, знаходиться на стадії дослідного зразка $\eta=0,5$.

$$ЗВ = 2\,302\,862,4 / 0,5 = 1\,151\,431,2 \text{ грн.}$$

Загальна сума витрат на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів оцінюється в 1 151 431,2 грн.

5.4 Розрахунок економічної ефективності науково-технічної розробки та її можливої комерціалізації потенційним інвестором

Загалом, збільшення чистого прибутку є сприятливим результатом, який може отримати потенційний інвестор, як результат впровадження науково-технічних розробок. Збільшення чистого прибутку може забезпечити додаткове фінансування потенційним інвесторам, підвищити фінансові результати та конкурентоспроможність на ринку. Це також може позитивно вплинути на рішення про комерціалізацію розробки.

Для визначення потенційної можливості збільшення чистого прибутку інвестора за рахунок науково-технічних розробок, вирішуються наступні задачі:

- зазначається час впровадження результатів науково-технічної розробки;
- вказується тривалість в роках час очікування основних позитивних результатів для потенційного інвестора після впровадження науково-технічної розробки, очікуються;
- оцінюється кількісно величина існуючого та майбутнього попиту на розроблену або аналогічну науково-технічну розробку, крім того визначаються основні суб'єкти цього попиту;
- встановлюється ціна реалізації на ринку фітнес-застосунків з аналогічними чи подібними функціями.

Здійснюючи оцінку ефективності інноваційних проектів здійснюється розрахунок важливих показників, таких як:

- абсолютний економічний ефект (чистий дисконтований дохід);
- внутрішня економічна дохідність (внутрішня норма дохідності);
- термін окупності.

Майбутній економічний ефект розробки визначатиметься наданими даними, які вказують на те, що характеристики продукту з часом покращуються, а кількість споживачів збільшується. До цих даних відносяться: ΔN – збільшення кількості споживачів продукту, в розглянуті та проаналізовані періоди часу, від покращення його певних характеристик; N – кількість споживачів, які використовували аналогічний продукт у році до того, як результати нової науково-технічної розробки були впроваджені; C_0 – вартість програмного продукту у році до впровадження результатів розробки; $\pm \Delta C_0$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в розглянуті та проаналізовані періоди часу. Для всіх випадків, наведених вище, формула 5.18 використовується для розрахунку того, скільки потенційно може заробити інвестор, обчислюючи свій чистий прибуток у роки, коли очікується, що науково-технічна розробка принесе позитивні результати.

$$\Delta\Pi_i = (\pm\Delta C_0 \cdot N + C_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (5.18)$$

де $\pm\Delta C_0$ – зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай, таким показником може бути зміна ціни реалізації одиниці нової розробки в аналізованому році (відносно року до впровадження цієї розробки); $\pm\Delta C_0$ може мати як додатне, так і від'ємне значення (від'ємне – при зниженні ціни відносно року до впровадження цієї розробки, додатне – при зростанні ціни);

N – основний кількісний показник, який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної розробки;

C_o – основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році, $C_o = C_b \pm \Delta C_o$;

C_b – основний якісний показник, який визначає ціну реалізації існуючої (базової) науково-технічної розробки у році до впровадження результатів;

ΔN – зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році. Зазвичай таким показником може бути зростання попиту на науково-технічну розробку в аналізованому році (відносно року до впровадження цієї розробки);

λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2023 році ставка податку на додану вартість становить 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту (послуги). $\rho = 0,4$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2023 році $\vartheta = 18\%$.

Якщо припустити, що очікувана ціна 89 грн. за одиницю передбачає 4-річний період збільшення прибутку, то після завершення його розробки та вдосконалення ціну можна підвищити на 40 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 46000 шт., протягом другого року – на 42500 шт., протягом третього року – на 39000 шт. і протягом четвертого року – на 31000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

Можливе збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = ((0 \cdot 40 + (89+40) \cdot 46000) \cdot 0,8333 \cdot 0,4) \cdot (1-0,18) = 1\,621\,895,12 \text{ грн.};$$

Можливе збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = ((0 \cdot 40 + (89+40) \cdot (46000+42500)) \cdot 0,8333 \cdot 0,4) \cdot (1-0,18) = 3\,120\,385,18 \text{ грн.};$$

Можливе збільшення чистого прибутку 3-го року:

$$\begin{aligned} \Delta\Pi_3 &= ((0 \cdot 40 + (89+40) \cdot (46000+42500+39000)) \cdot 0,8333 \cdot 0,4) \cdot (1-0,18) = \\ &= 4\,495\,470,17 \text{ грн.}; \end{aligned}$$

Можливе збільшення чистого прибутку 4-го року:

$$\begin{aligned} \Delta\Pi_4 &= ((0 \cdot 40 + (89+40) \cdot (46000+42500+39000+31000)) \cdot 0,8333 \cdot 0,4) \cdot (1-0,18) = \\ &= 5\,588\,486,45 \text{ грн.} \end{aligned}$$

Проведемо розрахунки приведеної вартості збільшення всіх чистих прибутків $\Pi\Pi$, які потенційний інвестор може отримати від можливого впровадження та комерціалізації науково-технічної розробки, за допомогою формули 5.19:

$$\Pi\Pi = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (5.19)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,15$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

Збільшення прибутку планується отримувати починаючи з першого року:

$$\begin{aligned} \Pi\Pi &= (1621895,12/(1+0,15)^1) + (3120385,18/(1+0,15)^2) + (4495470,17/(1+0,15)^3) + \\ &+ (5588486,45/(1+0,15)^4) = 9\,920\,882,96 \text{ грн.} \end{aligned}$$

5.5 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Проведемо розрахунок на початкові PV інвестиції необхідний для визначення, скільки потенційний інвестор має вкласти для впровадження та комерціалізації науково-технічної розробки. При таких розрахунках використовується формула 5.20:

$$PV = k_{инв} \cdot 3B, \quad (5.20)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; $k_{инв}=2,6$;

$3B$ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2,6 \cdot 1\,151\,431,2 = 2\,993\,721,12 \text{ грн.}$$

Таким чином, абсолютний економічний ефект $E_{абс}$ або чистий приведений дохід (NPV , *Net Present Value*) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП - PV, \quad (5.21)$$

де $ПП$ – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн;

PV – теперішня вартість початкових інвестицій, грн.

$$E_{абс} = 9\,920\,882,96 - 2\,993\,721,12 = 6\,927\,161,84 \text{ грн.}$$

З огляду на те, що абсолютний економічний ефект більше нуля – інвестування коштів на розробку та впровадження результатів даної науково-дослідної роботи ймовірно доцільне. Для остаточного прийняття рішення про доцільність, визначається внутрішньо економічну дохідність E_e або показник внутрішньої норми дохідності вкладених інвестицій. Її порівнюють з бар'єрною

ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно. Проведемо розрахунок відносної (щорічної) ефективності вкладених у науково-технічну розробку інвестицій E_e :

$$E_B = \sqrt[T_{ж}]{1 + \frac{E_{абс}}{PV}} - 1, \quad (5.22)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, грн;

PV – теперішня вартість початкових інвестицій, грн;

$T_{ж}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, роки.

$$E_B = \sqrt[4]{1 + \frac{6\,927\,161,84}{2\,993\,721,12}} - 1 = 0,35$$

Розрахована відносна (щорічна) ефективність вкладених у науково-технічну розробку інвестицій E_e становить 0,35.

Проведемо розрахунок бар'єрної ставки дисконтування $\tau_{мін}$. Вона визначає мінімальну внутрішню економічну дохідність інвестицій, нижче якої немає сенсу інвестувати кошти у впровадження науково-технічної розробки та її комерціалізацію. Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$ визначається за формулою:

$$\tau_{мін} = d + f, \quad (5.23)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках в 2020 році в Україні $d = 0,16$;

f – показник, що характеризує ризикованість вкладення інвестицій; $f = 0,1$.

Розрахована мінімальна внутрішня економічна дохідність вкладених інвестицій становить 0,26, тобто $E_e > \tau_{мін}$. Тому можна зробити висновок про те, що потенційний інвестор може бути зацікавлений інвестувати у впровадження

науково-технічної розробки та виведення цієї розробки на ринок, тобто в комерціалізації.

Проведено розрахунок періоду окупності інвестицій $T_{ок}$ (*DPP, Discounted Payback Period*), які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_B}, \quad (5.24)$$

$$T_{ок} = 1/0,3 = 2,86 \text{ р.}$$

Розрахований період окупності $T_{ок} < 3$ -х років. Таким чином, науково-технічна розробка вважається комерційно привабливою і може спонукати потенційного інвестора профінансувати впровадження цієї розробки та виведення її на ринок.

5.6 Висновок до розділу 5

В ході виконання економічної частини було проведено комерційний та технологічний аудит. За їх результатами було оцінено рівень комерційного потенціалу інформаційної технології для організації та проведення тренувального процесу та науково-технічний рівень - вище середнього, тобто потенційним інвесторам дана розробка буде досить привабливою та перспективною, особливо враховуючи швидке зростання ринку фітнес застосунків. Розрахунок витрат на проведення науково-дослідної роботи показав, що найбільшими групами витрат є основна заробітня плата та накладні (загальновиробничі) витрати, що разом складають 70,28% усіх витрат науково-дослідної роботи. Крім того, загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів складають 1 151 431,2 грн.

Проведене дослідження рівня конкурентоспроможності застосунку SmartFit Expert, у порівнянні з п'ятьма найближчими конкурентами, показало

високий рівень конкурентоспроможності та перспективності розробки, а також значний потенціал для комерціалізації застосунку.

Було підтверджено доцільність інвестування коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи завдяки визначенню чистого приведеного доходу розробки інформаційної технології для організації та проведення тренувального процесу.

Можливість залучення потенційних інвесторів висока, оскільки відносна ефективність інвестицій у науково-технічну розробку перевищує мінімальну внутрішньоекономічну віддачу інвестицій, а період окупності інвестицій у 2,86 роки доводить комерційну привабливість розробки.

ВИСНОВКИ

Магістерську кваліфікаційну роботу було присвячено інформаційній технології організації та проведення тренувального процесу, а саме проектуванню та розробці експертної системи для надання рекомендацій щодо програми тренувань та серверної частини інформаційної технології.

В першому розділі було проведено аналіз предметної області, який виявив досить високу актуальність такої інформаційної технології, оскільки відзначається швидке зростання та економічна привабливість сфери фітнесу та ведення здорового способу життя. Основною причиною стало надання персональних програм тренувань, що підкріплюється можливістю моніторингу індивідуальних показників і результатів. Таким чином, користувачі отримують зручність, самостійність та організованість під час проведення тренувань без потреби в дорогих персональних тренерах чи консультацій. Проведений аналіз аналогів системи дозволив виявити переваги та недоліки існуючих рішень, які слід врахувати при розробці. В ході аналізу об'єкту проектування визначено ключові показники при складанні програми тренувань та охарактеризовано переваги та недоліки найпоширеніших підходів до організації тренувального процесу. В результаті, це дозволило визначити задачі, які необхідно вирішити для досягнення мети.

В другому розділі було оцінено доцільність розробки експертної системи за основними критеріями. Відповідно до поставленої задачі та критеріїв розробки дана ЕС виявилася доречною, виправданою та можливою для застосування в інформаційній технології організації та проведення тренувального процесу. Обґрунтовано вибір продукційної моделі подання знань та проведено структурування знань, яка дозволила створити ієрархічну систему знань експерта та розробити структуру вхідних та вихідних факторів.

В третьому розділі було здійснено програмну реалізацію експертної системи, що є ключовим етапом для успішного надання рекомендацій щодо визначення програми тренувань. Обґрунтовано вибір середовища CLIPS, де і

було розроблено базу знань експертної системи. В результаті було створено усі необхідні шаблони для збереження невпорядкованих фактів та розроблено правила бази знань. Причому шаблони розроблено у такий спосіб, щоб можна було успішно інтегрувати ЕС до серверної частини інформаційної технології. Результати тестування розробленої експертної системи підтвердили правильність роботи бази знань, а якість надання рекомендацій щодо визначення програми тренувань збільшилась на 9.6% в порівнянні з програмами аналогами.

В четвертому розділі було здійснено програмну реалізацію серверної частини інформаційної технології. Обґрунтовано вибір засобів розробки, враховуючи вимоги та особливості предметної області. Розроблено інфологічну та даталогічну модель бази даних, діаграму розгортання інформаційної технології та діаграму компонентів. Основним компонентом є Web Api побудований за принципами REST, який поєднує рівень бізнес логіки та рівень доступу до даних. Застосування різноманітних шаблонів проектування дозволило реалізувати слабку зв'язаність компонентів інформаційної технології між собою. Тестування серверної частини інформаційної технології для кожного методу HTTP усіх розроблених сервісів показало тільки успішні результати.

В п'ятому розділі було визначено, що згідно узагальненого коефіцієнту конкурентоспроможності, науково-технічна розробка переважає існуючі аналоги приблизно в 2,26 рази. Термін окупності становить 2,86 роки, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Додавання цілей тренувань «Покращення витривалості» та «Підтримання тону м'язів», а також врахування ІМТ, віку та статі користувача при складанні програми тренувань дозволило розширити функціональні можливості розробленої інформаційної технології в порівнянні з програмами-аналогами.

В результаті, усі задачі, поставлені в магістерській кваліфікаційній роботі були виконані, а мету роботи було досягнуто.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Darejeh A. An Investigation on the Use of Expert Systems in Developing Web-Based Fitness Exercise Plan Generator [Електронний ресурс] / A. Darejeh, H. Haddadpajouh, A. Darejeh // International Review on Computers and Software (IRECOS) 9(8):1442-1448 – Режим доступу до ресурсу: https://www.researchgate.net/publication/267035338_An_Investigation_on_the_Use_of_Expert_Systems_in_Developing_Web-Based_Fitness_Exercise_Plan_Generator (дата звернення: 01.09.2023).
2. Kim U. A fuzzy expert system for designing customized workout programs [Електронний ресурс] / U. Kim, J. Kim // Conference: 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/document/7737993> (дата звернення: 01.09.2023).
3. Інтеграція експертної системи для створення програми тренувань в середовищі виконання CLR / Шелестюк М.І., Шелестюк Б.І., Яровий А.А., Козловський А.В. // European scientific congress. Proceedings of the 11th International scientific and practical conference. Barca Academy Publishing. Madrid, Spain. 2023. Pp. 213-219.
4. Використання дерева нечіткого логічного виведення для оптимізації бази знань нечіткої інтелектуальної системи вибору спортивного харчування / Шелестюк Б.І., Шелестюк М.І., Яровий А.А., Козловський А.В. : Збірник матеріалів Міжнародної науково-практичної конференції "Молодь в науці: дослідження, проблеми, перспективи (МН-2024)". – В.: ВНТУ, 2023. – С. 1-7. [Електронний ресурс] – Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2024/paper/viewFile/19549/16201> (дата звернення: 30.11.2023).
5. Essential Technology In Fitness Statistics In 2023 [Електронний ресурс] – Режим доступу до ресурсу: <https://zipdo.co/statistics/technology-in-fitness/#:~:text=Technology%20has%20significantly%20revolutionized%20the,time%20tracking%20of%20fitness%20progress> (дата звернення: 14.09.2023).

6. FitnessAI uses artificial intelligence to generate personalized workouts. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.fitnessai.com/> (дата звернення: 14.09.2023).
7. Fitbod is a fitness app that provides a personalized exercise plan [Електронний ресурс] – Режим доступу до ресурсу: <https://fitbod.me/> (дата звернення: 14.09.2023).
8. Science-backed gym AI. [Електронний ресурс] – Режим доступу до ресурсу: https://hyperficient.org/?linkId=lp_541423&sourceId=rareconnections&tenantId=hyperficient (дата звернення: 14.09.2023).
9. Body mass index (BMI) [Електронний ресурс] // WHO, topics – Режим доступу до ресурсу: <https://www.who.int/data/gho/data/themes/topics/topic-details/GHO/body-mass-index> (дата звернення: 17.09.2023).
10. Human energy requirements [Електронний ресурс] // Report of a Joint FAO/WHO/UNU Expert Consultation – Режим доступу до ресурсу: <https://www.fao.org/3/y5686e/y5686e.pdf> (дата звернення: 18.09.2023).
11. Експертні системи. Частина 1. Навчальний посібник. – Вінниця: ВНТУ, 2006.– 114 с.
12. Яровий А. А. Експертні системи. Частина 2: навчальний посібник / А. А. Яровий, І. Р. Арсенюк, В. І. Месюра. – Вінниця: ВНТУ, 2017. – 106 с.
13. Глибовець М. М. Штучний інтелект / М. М. Глибовець, О. В. Олецкий., 2002. – 366 с.
14. Sowa J. Semantic networks [Електронний ресурс] – Режим доступу до ресурсу: <http://www.jfsowa.com/pubs/semnet.htm> (дата звернення: 11.10.2023).
15. C. Schank R., Abelson R. Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures. / C. Schank R., Abelson R. – Lawrence Erlbaum Associates, 1977. – 256с.
16. Awati R. first-order logic [Електронний ресурс] / Rahul Awati. – 2023. – Режим доступу до ресурсу: [https://www.techtarget.com/whatis/definition/first-order-logic#:~:text=First%2Dorder%20logic%20\(FOL\),or%20first%2Dorder%20functional%20calculus..](https://www.techtarget.com/whatis/definition/first-order-logic#:~:text=First%2Dorder%20logic%20(FOL),or%20first%2Dorder%20functional%20calculus..) (дата звернення: 14.10.2023).

17. Brachman R., Levesque H. Knowledge Representation and Reasoning. / Brachman R., Levesque H. – Morgan Kaufmann Publishers, 2004. – 381с.
18. Riley G. Reference Manual – CLIPS [Электронный ресурс] / Gary Riley. – 2021. – Режим доступа до ресурсу: <https://clipsrules.sourceforge.io/documentation/v640/ig.pdf> (дата звернення: 14.10.2023).
19. Дуглас К. JSON [Электронный ресурс] / Крокфорд Дуглас. – 2021. – Режим доступа до ресурсу: <https://uk.wikipedia.org/wiki/JSON>. (дата звернення: 19.10.2023).
20. What is "managed code"? URL: <https://learn.microsoft.com/en-us/dotnet/standard/managed-code> (дата звернення: 19.10.2023).
21. A tour of the C# language. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/> (дата звернення: 20.10.2023).
22. Introduction to .NET. URL: <https://learn.microsoft.com/en-us/dotnet/core/introduction> (дата звернення: 20.10.2023).
23. Entity Framework – Short guideline. URL: <https://coderlessons.com/tutorials/microsoft-technologies/izuchite-entity-framework/entity-framework-kratkoe-rukovodstvo> (дата звернення: 20.10.2023).
24. Fitness App Market Size, Share & Trends Analysis Report By Type [Электронный ресурс]. – 2023. – Режим доступа до ресурсу: <https://www.grandviewresearch.com/industry-analysis/fitness-app-market> (дата звернення: 15.11.2023).

ДОДАТКИ

Додаток А (обов'язковий)
Протокол перевірки кваліфікаційної роботи на
наявність текстових запозичень

Назва роботи: Інформаційна технологія організації та проведення тренувального процесу. Частина 2. Надання рекомендацій.

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФІІТА
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 88,0% Схожість 12,0%

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи




Шелестюк М.І.

Керівник роботи

Яровий А.А.

Опис прийнятого рішення

Магістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку 

Озеранський В.С.

Додаток Б (обов'язковий)

Лістинг програми

Фрагмент серверної частини інформаційної технології на мові програмування C#

```

using AutoMapper;
using SmartFitExpert.Core.BLL.Interfaces;
using SmartFitExpert.Core.DAL.Entities.JoinEntities;
using SmartFitExpert.Core.DAL.Entities;
using SmartFitExpert.Core.DAL;
using SmartFitExpert.Core.Common.DTO.WeeklyPlan;
using Microsoft.EntityFrameworkCore;

namespace SmartFitExpert.Core.BLL.Services
{
    public class WeeklyPlanService : IWeeklyPlanService
    {
        private readonly SmartFitExpertCoreContext _context;
        private readonly IMapper _mapper;

        public WeeklyPlanService(SmartFitExpertCoreContext context, IMapper mapper)
        {
            _context = context;
            _mapper = mapper;
        }

        public async Task<ResponseWeeklyPlanDto>
        CreateWeeklyPlanAsync(CreateWeeklyPlanDto newWeeklyPlan)
        {
            ValidateEntity(newWeeklyPlan.WeeklyPlanDto);
            var createdWeeklyPlan = (await
            _context.WeeklyPlans.AddAsync(_mapper.Map<WeeklyPlan>(newWeeklyPlan.WeeklyPlanDto))).E
            ntity;

            if (newWeeklyPlan.DailyPlans.Any())
            {
                foreach (var wpId in newWeeklyPlan.DailyPlans)
                {
                    createdWeeklyPlan.WeekDailyPlans.Add(new WeekDailyPlans
                    {
                        DailyPlanId = wpId,
                    });
                }
            }

            await _context.SaveChangesAsync();
            return _mapper.Map<ResponseWeeklyPlanDto>(await
            GetByIdWithDetailsAsync(createdWeeklyPlan.Id));
        }

        public async Task<ICollection<ResponseWeeklyPlanDto>> GetAllWeeklyPlansAsync()
        {
            return _mapper.Map<List<ResponseWeeklyPlanDto>>(await _context.WeeklyPlans
            .Include(wp => wp.WeekDailyPlans).ThenInclude(wdp => wdp.DailyPlan)
            .ToListAsync());
        }

        public async Task<ResponseWeeklyPlanDto> GetWeeklyPlanAsync(int weeklyPlanId)
        {
            return _mapper.Map<ResponseWeeklyPlanDto>(await
            GetByIdWithDetailsAsync(weeklyPlanId));
        }
    }
}

```

```

public async Task DeleteWeeklyPlanAsync(int weeklyPlanId)
{
    var weeklyPlan = await _context.WeeklyPlans.FindAsync(weeklyPlanId);

    ValidateEntity(weeklyPlan);

    _context.WeeklyPlans.Remove(weeklyPlan!);
    await _context.SaveChangesAsync();
}

public async Task<ResponseWeeklyPlanDto> UpdateWeeklyPlanAsync(int
weeklyPlanId, CreateWeeklyPlanDto newWeeklyPlan)
{
    var existingWeeklyPlan = await GetByIdWithDetailsAsync(weeklyPlanId);

    _mapper.Map(newWeeklyPlan.WeeklyPlanDto, existingWeeklyPlan);

    await _context.SaveChangesAsync();

    return _mapper.Map<ResponseWeeklyPlanDto>(existingWeeklyPlan!);
}

private async Task<WeeklyPlan> GetByIdWithDetailsAsync(int id)
{
    var weeklyPlan = await _context.WeeklyPlans.Include(wp =>
wp.WeekDailyPlans).ThenInclude(wdp => wdp.DailyPlan)
    .FirstOrDefaultAsync(wp => wp.Id == id);
    ValidateEntity(weeklyPlan);
    return weeklyPlan!;
}

private void ValidateEntity<T>(T? entity)
{
    if (entity is null)
    {
        throw new Exception($"Entity '{nameof(T)}' not found");
    }
}
}
}

using Microsoft.AspNetCore.Mvc;
using SmartFitExpert.Core.BLL.Interfaces;
using SmartFitExpert.Core.Common.DTO.WeeklyPlan;

namespace SmartFitExpert.Core.WebAPI.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class WeeklyPlanController : ControllerBase
    {
        private readonly IWeeklyPlanService _weeklyPlanService;
        private readonly IGenerateInitialPlanService _generateInitialPlanService;

        public WeeklyPlanController(IWeeklyPlanService weeklyPlanService,
IGenerateInitialPlanService generateInitialPlanService)
        {
            _weeklyPlanService = weeklyPlanService;
            _generateInitialPlanService = generateInitialPlanService;
        }

        [HttpGet("generate/{userProfileId}")]
        public async Task<ActionResult<string>> GetInitialPlan(int userProfileId)
        {

```

```

        return Ok(await
_generateInitialPlanService.CreateInitialPlan(userProfileId));
    }

    [HttpPost]
    public async Task<ActionResult<ResponseWeeklyPlanDto>>
CreateWeeklyPlanAsync([FromBody] CreateWeeklyPlanDto newWeeklyPlan)
    {
        return Ok(await _weeklyPlanService.CreateWeeklyPlanAsync(newWeeklyPlan));
    }

    [HttpGet("all")]
    public async Task<ActionResult<ICollection<ResponseWeeklyPlanDto>>>
GetAllWeeklyPlansAsync()
    {
        return Ok(await _weeklyPlanService.GetAllWeeklyPlansAsync());
    }

    [HttpGet("{weeklyPlanId}")]
    public async Task<ActionResult<ResponseWeeklyPlanDto>> GetWeeklyPlanAsync(int
weeklyPlanId)
    {
        return Ok(await _weeklyPlanService.GetWeeklyPlanAsync(weeklyPlanId));
    }

    [HttpPut("{weeklyPlanId}")]
    public async Task<ActionResult<ResponseWeeklyPlanDto>>
UpdateWeeklyPlanAsync(int weeklyPlanId, [FromBody] CreateWeeklyPlanDto newWeeklyPlan)
    {
        return Ok(await _weeklyPlanService.UpdateWeeklyPlanAsync(weeklyPlanId,
newWeeklyPlan));
    }

    [HttpDelete("{weeklyPlanId}")]
    public async Task<IActionResult> DeleteWeeklyPlanAsync(int weeklyPlanId)
    {
        await _weeklyPlanService.DeleteWeeklyPlanAsync(weeklyPlanId);
        return NoContent();
    }
}

}

using SmartFitExpert.Core.DAL.Entities.Abstract;
using SmartFitExpert.Core.DAL.Entities.JoinEntities;
using SmartFitExpert.Core.DAL.Enums;

namespace SmartFitExpert.Core.DAL.Entities
{
    public sealed class WeeklyPlan: BaseEntity<int>
    {
        public int TotalDays { get; set; }
        public string TrainingRestBalance { get; set; } = string.Empty;
        public WeekPlanType WeekPlanType { get; set; }
        public ICollection<WeekDailyPlans> WeekDailyPlans { get; set; } = new
List<WeekDailyPlans>();
        public ICollection<UserProfile> UserProfiles { get; set; } = new
List<UserProfile>();
    }
}

using SmartFitExpert.Core.DAL.Entities.Abstract;

namespace SmartFitExpert.Core.DAL.Entities.JoinEntities

```



```

{
    public sealed class WeekDailyPlans: BaseEntity<int>
    {
        public int WeekPlanId { get; set; }
        public int DailyPlanId { get; set; }
        public WeeklyPlan WeeklyPlan { get; set; } = null!;
        public DailyPlan DailyPlan { get; set; } = null!;
    }
}

using SmartFitExpert.Core.DAL.Entities.Abstract;
using SmartFitExpert.Core.DAL.Enums;

namespace SmartFitExpert.Core.DAL.Entities
{
    public sealed class UserProfile : BaseEntity<int>
    {
        public int? WeeklyPlanId { get; set; }
        public WeeklyPlan? WeeklyPlan { get; set; } = null!;
        public string Name { get; set; } = string.Empty;
        public Gender Gender { get; set; }
        public Goal Goal { get; set; }
        public int Weight { get; set; }
        public int Height { get; set; }
        public int Age { get; set; }
        public int AvailableDays { get; set; }
        public string Restrictions { get; set; } = string.Empty;
    }
}

using Microsoft.EntityFrameworkCore;
using SmartFitExpert.Core.DAL.Entities;
using SmartFitExpert.Core.DAL.Entities.JoinEntities;
using SmartFitExpert.Core.DAL.Extensions;

namespace SmartFitExpert.Core.DAL
{
    public class SmartFitExpertCoreContext: DbContext
    {
        public DbSet<UserProfile> UserProfiles => Set<UserProfile>();
        public DbSet<DailyPlan> DailyPlans => Set<DailyPlan>();
        public DbSet<Exercise> Exercise => Set<Exercise>();
        public DbSet<Equipment> Equipment => Set<Equipment>();
        public DbSet<WeeklyPlan> WeeklyPlans => Set<WeeklyPlan>();
        public DbSet<MuscleGroup> MuscleGroups => Set<MuscleGroup>();
        public DbSet<ExerciseDailyPlans> ExerciseDailyPlans =>
Set<ExerciseDailyPlans>();
        public DbSet<ExerciseSupportMuscles> SupportMuscleExercises =>
Set<ExerciseSupportMuscles>();
        public DbSet<WeekDailyPlans> WeekDailyPlans => Set<WeekDailyPlans>();
        public DbSet<ExerciseEquipments> ExerciseEquipments =>
Set<ExerciseEquipments>();

        public SmartFitExpertCoreContext(DbContextOptions<SmartFitExpertCoreContext>
options) : base(options)
        {
        }
        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            modelBuilder.Configure();
        }
    }
}

```



```

using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Metadata.Builders;
using SmartFitExpert.Core.DAL.Entities.JoinEntities;

namespace SmartFitExpert.Core.DAL.EntityConfigurations
{
    public sealed class WeekDailyPlansConfig :
    IEntityConfiguration<WeekDailyPlans>
    {
        public void Configure(EntityTypeBuilder<WeekDailyPlans> builder)
        {
            builder.HasOne(x => x.DailyPlan)
                .WithMany(x => x.WeekDailyPlans)
                .HasForeignKey(x => x.DailyPlanId)
                .OnDelete(DeleteBehavior.Cascade);

            builder.HasOne(x => x.WeeklyPlan)
                .WithMany(x => x.WeekDailyPlans)
                .HasForeignKey(x => x.WeekPlanId)
                .OnDelete(DeleteBehavior.Cascade);
        }
    }
}

using SmartFitExpert.Core.Common.DTO.DailyPlan;
using SmartFitExpert.Core.Common.DTO.Exercise;
using SmartFitExpert.Core.DAL.Entities;
using System.Text;

namespace SmartFitExpert.Core.BLL.Extensions
{
    public static class ClipsMapperExtensions
    {
        public static string ToExerciseTemplate(this ResponseExerciseDto exercise)
        {
            var exerciseTemplate = new StringBuilder();
            exerciseTemplate.Append($"(assert (exercise (id {exercise.Id}))(name
{exercise.Name}))(target_muscle_group {exercise.TargetMuscleGroup.Name})" +
                $"(priority {exercise.Priority}))(jointness {exercise.Jointness})");

            exerciseTemplate.Append($"(support_muscle_groups");
            foreach (var supports in exercise.SupportMuscles)
            {
                exerciseTemplate.Append($" {supports.Name}");
            }

            exerciseTemplate.Append($"(equipment");
            foreach (var equip in exercise.Equipments)
            {
                exerciseTemplate.Append($" {equip.Name}");
            }
            exerciseTemplate.Append(")))");
            return exerciseTemplate.ToString();
        }

        public static string ToDailyPlanTemplate(this ResponseDailyPlanDto dailyPlan)
        {
            var dailyPlanTemplate = new StringBuilder();
            dailyPlanTemplate.Append($"(assert (daily_plan (id_d
{dailyPlan.Id}))(age_group {dailyPlan.AgeGroup}))(goal {dailyPlan.Goal})" +

```

```

        $"(days {dailyPlan.TotalDays})(day_n
{dailyPlan.DayNumber})(exercise_organization_type
{dailyPlan.ExerciseOrganizationType})");

        dailyPlanTemplate.Append($"(exercises_id");
        foreach (var exercise in dailyPlan.Exercises)
        {
            dailyPlanTemplate.Append($" {exercise.Id}");
        }
        dailyPlanTemplate.Append(")))");
        return dailyPlanTemplate.ToString();
    }
    public static string ToUserProfileTemplate(this UserProfile user)
    {
        var userProfileTemplate = new StringBuilder();
        userProfileTemplate.Append($"(assert (user_profile (name
{user.Name})(gender {user.Gender})(weight {user.Weight})(height {user.Height})" +
            $"(age {user.Age})(restriction {user.Restrictions})(available_days
{user.AvailableDays})(goal {user.Goal})))");
        return userProfileTemplate.ToString();
    }
}

using CLIPSNET;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Xml.Linq;

namespace CLIPSExpertSystem
{
    public class ExpertProviderService : IExpertProviderService
    {
        private CLIPSNET.Environment _clips;
        private IUnitOfWork _db;
        public ExpertProviderService(IUnitOfWork db, UserProfile userProfile)
        {
            _db = db;
            _clips = new CLIPSNET.Environment();
            _clips.LoadFromResource("CLIPSExpertSystem",
"CLIPSExpertSystem.HealthIndicators.clp");
            _clips.LoadFromResource("CLIPSExpertSystem",
"CLIPSExpertSystem.Exercises.clp");
            _clips.LoadFromResource("CLIPSExpertSystem",
"CLIPSExpertSystem.Recomendations.clp");
            _clips.Reset();
            AssertExercises(_db.Exercises);
            AssertDailyPlans(_db.DailyPlans);
            AssertUserProfile(userProfile);
            _clips.Run();
        }

        public void AssertExercises(IEnumerable<Exercise> exercises)
        {
            foreach (var exercise in exercises)
            {
                _clips.AssertString($"(exercise(id {exercise.ExerciseId})(name
{exercise.Name}))" +
                    $"(target_muscle_group {exercise.TargetMuscleGroup})(priority
{exercise.Priority})" +
                    $"(jointness {exercise.Jointness})(support_muscle_groups
{exercise.SupportMuscleGroups})" +
                    $"(equipment {exercise.Equipment}))");
            }
        }
    }
}

```

```

    }
}
public void AssertUserProfile(UserProfile user)
{
    _clips.AssertString($"(user_profile (name {user.Name}))(gender
{user.Gender})(weight {user.Weight})" +
        $"(height {user.Height})(age {user.Age})(restriction
{user.Restrictions})" +
        $"(available_days {user.AvailableDays})(goal {user.Goal}))");
}
public void AssertDailyPlans(IEnumerable<DailyPlan> dailyPlans)
{
    foreach (var plan in dailyPlans)
    {
        _clips.AssertString($"(daily_plan(id_d {plan.DailyPlanId})(age_group
{plan.AgeGroup})" +
            $"(goal {plan.Goal})(days {plan.AvailableDays})(day_n
{plan.DayNumber})" +
            $"(exersice_organization_type
{plan.ExerciseOrganizationType})(exercises_id {plan.Exercises}))");
    }
}
public double GetBMIValue()
{
    var bmiFact = _clips.FindFact("?f", "health_indicator", "(eq
?f:parameterName BMI)");
    double bmiValue = (double)(((NumberValue)bmiFact["parameterValue"]));
    return bmiValue;
}
public double GetBMRValue()
{
    var bmrFact = _clips.FindFact("?f", "health_indicator", "(eq
?f:parameterName BMR)");
    double bmrValue = (double)(((NumberValue)bmrFact["parameterValue"]));
    return bmrValue;
}
public string GetBMIFormal()
{
    var bmiFormalFact = _clips.FindFact("bmi_formal");
    String bmiFormalName = ((LexemeValue)bmiFormalFact["formal_name"]).Value;
    return bmiFormalName;
}
public List<string> GetExercises()
{
    var exercises = _clips.FindAllFacts("exercise");
    List<string> names = new List<string>();
    foreach (var exercise in exercises)
    {
        string name = ((LexemeValue)exercise["name"]).Value;
        names.Add(name);
    }
    return names;
}
public Tuple<string, int> GetRecomendation()
{
    var recomendationFact = _clips.FindFact("recomendation");
    String badPractice =
((LexemeValue)recomendationFact["bad_practice"]).Value;
    int id = (int)(((NumberValue)recomendationFact["weekly_plan_id"])); ;
    return new Tuple<string, int>(badPractice, id);
}
public Tuple<int, string, List<string>> GetDailyPlan()
{
    var dailyPlanFact = _clips.FindFact("daily_plan");
    int day = (int)(NumberValue)dailyPlanFact["day_n"];
}

```

```

        String type =
        ((LexemeValue)dailyPlanFact["exercice_organization_type"]).Value;

        List<string> names = new List<string>();
        var exercises = ((MultifieldValue)dailyPlanFact["exercises"]).Value;
        foreach (var name in exercises)
        {
            names.Add(name.ToString());
        }
        return new Tuple<int, string, List<string>>(day, type, names);
    }
    public Tuple<int, int, string, string, List<int>> GetWeekPlan()
    {
        var weekPlanFact = _clips.FindFact("weekly_plan");
        int weekId = (int)(NumberValue)weekPlanFact["id_w"];
        int days = (int)(NumberValue)weekPlanFact["days"];
        string restDays = ((LexemeValue)weekPlanFact["rest_days"]).Value;
        string weekPlanType = ((LexemeValue)weekPlanFact["week_plan_type"]).Value;

        List<int> dayPlansIds = new List<int>();
        var dayPlans = ((MultifieldValue)weekPlanFact["daily_plans_id"]).Value;
        foreach (var id in dayPlans)
        {
            dayPlansIds.Add((int)(NumberValue)id);
        }
        return new Tuple<int, int, string, string, List<int>>(weekId, days,
restDays, weekPlanType, dayPlansIds);
    }
}
}
}

```

Фрагмент бази знань в середовищі розробки CLIPS

```

(deftemplate exercise
  (slot id)
  (slot name)
  (slot target_muscle_group)
  (slot priority)
  (slot jointness)
  (multislot support_muscle_groups)
  (multislot equipment))

(deftemplate user_profile
  (slot name)
  (slot gender)
  (slot weight)
  (slot height)
  (slot age)
  (slot restriction)
  (slot available_days)
  (slot goal))

(deftemplate health_indicator
  (slot parameterName)
  (slot parameterValue))

(deftemplate bmi_result
  (slot value)
  (slot formalName))

```

```

(deftemplate bmr_result
  (slot value))

(deftemplate bmi_formal
  (slot formal_name))

(deftemplate recommendation
  (slot bad_practice)
  (slot weekly_plan_id))

(deftemplate daily_plan
  (slot id_d)
  (slot age_group)
  (slot goal)
  (slot days)
  (slot day_n)
  (slot exercise_organization_type)
  (multislot exercises_id))

(deftemplate weekly_plan
  (slot id_w)
  (slot days)
  (slot rest_days)
  (slot week_plan_type)
  (multislot daily_plans_id))

(defrule AgeGroupYoung
  (user_profile (age ?a&:(>= ?a 18)&:(<= ?a 23)))
  =>
  (assert (age_group Young))
)

(defrule AgeGroupModerate
  (user_profile (age ?a&:(>= ?a 24)&:(<= ?a 45)))
  =>
  (assert (age_group Moderate))
)

(defrule AgeGroupOld
  (user_profile (age ?a&:(>= ?a 46)))
  =>
  (assert (age_group Old))
)

(defrule CalculateMaleBMR
  (user_profile (weight ?w) (height ?h) (age ?a) (gender male))
  =>
  (assert (health_indicator(parameterName BMR) (parameterValue =(- (+ (+ 66.5 (* 13.75 ?w)) (* 5.003 ?h)) (* 6.75 ?a))))))

(defrule CalculateFemaleBMR
  (user_profile (weight ?w) (height ?h) (age ?a) (gender female))
  =>
  (assert (health_indicator(parameterName BMR) (parameterValue =(- (+ (+ 655.1 (* 9.563 ?w)) (* 1.85 ?h)) (* 4.676 ?a))))))

```

```

(defrule CalculateBMI
  (user_profile (weight ?w) (height ?h))
  =>
  (assert (health_indicator(parameterName BMI) (parameterValue =(/ ?w (** (/ ?h 100) 2))))))

(defrule BMIOptimal
  (health_indicator(parameterName BMI)(parameterValue ?bmi&:(>= ?bmi 18.5)&:(<= ?bmi 24.9)))
  =>
  (assert (bmi_formal (formal_name Optimal))))

(defrule BMICritical
  (health_indicator(parameterName BMI)(parameterValue ?bmi&:(>= ?bmi 40)|:(<= ?bmi 13)))
  =>
  (assert (bmi_formal (formal_name Critical))))

(defrule BMIUnderWeight
  (health_indicator(parameterName BMI)(parameterValue ?bmi&:(< ?bmi 18.5)))
  =>
  (assert (bmi_formal (formal_name UnderWeight))))

(defrule BMIOverWeight
  (health_indicator(parameterName BMI)(parameterValue ?bmi&:(>= ?bmi 25)&:(<= ?bmi 29.9)))
  =>
  (assert (bmi_formal (formal_name OverWeight))))

(defrule BMIObese1
  (health_indicator(parameterName BMI)(parameterValue ?bmi&:(>= ?bmi 30)&:(<= ?bmi 34.9)))
  =>
  (assert (bmi_formal (formal_name Obese1))))

(defrule BMIObese2
  (health_indicator(parameterName BMI)(parameterValue ?bmi&:(>= ?bmi 35)&:(<= ?bmi 39.9)))
  =>
  (assert (bmi_formal (formal_name Obese2))))

(defrule BMIResult
  (health_indicator (parameterName BMI)(parameterValue ?value))
  (bmi_formal (formal_name ?formal))
  =>
  (assert (bmi_result (value ?value)(formalName ?formal))))

(defrule BMRResult
  (health_indicator (parameterName BMR)(parameterValue ?value))
  =>
  (assert (bmr_result (value ?value))))

(defrule UnderWeight_Fat_Loss
  (bmi_formal(formal_name undereight))
  (user_profile(goal fat_loss))
  =>
  (assert (recomendation (bad_practice "Considering your BMI (UnderWeight) we recommend to change
your goal for Muscle gain and follow next training program:")
  (weekly_plan_id ?*id*))
  (bind ?*id*=(+ ?*id* 1))
)

(defrule Young_LimitDays

```

```

(user_profile(available_days ?days&:(= ?days 1)|:(= ?days 7)))
(user_profile(age ?age&:(>= ?age 18)&:(<= ?age 23)))
=>
(assert (recomendation (bad_practice "Considering your Age we recommend to change number of training
days for 2-6 days")
(weekly_plan_id 0)))
)

(defrule Week_Plan_Muscle_gain_1days
  (user_profile(available_days ?days&:(= ?days 1))(goal Muscle_gain))
  (age_group ?ag)
  (daily_plan(days 1)(goal Muscle_gain)(age_group ?ag)(exercice_organization_type Circullar)(day_n
1)(id_d ?id1))
  =>
  (assert (weekly_plan(id_w ?*id*)(days 1)(rest_days d1)(week_plan_type Full_Body)(daily_plans_id
?id1)))
  (bind ?*id*=(+ ?*id* 1))
  )

(defrule Week_Plan_Muscle_gain_2days
  (user_profile(available_days ?days&:(= ?days 2))(goal Muscle_gain))
  (age_group ?ag)
  (daily_plan(days 2)(goal Muscle_gain)(age_group ?ag)(exercice_organization_type Circullar)(day_n
1)(id_d ?id1))
  (daily_plan(days 2)(goal Muscle_gain)(age_group ?ag)(exercice_organization_type Circullar)(day_n
2)(id_d ?id2))
  =>
  (assert (weekly_plan(id_w ?*id*)(days 2)(rest_days d1r23d2)(week_plan_type Full_Body)(daily_plans_id
?id1 ?id2)))
  (bind ?*id*=(+ ?*id* 1))
  )

(defrule Week_Plan_Muscle_gain_3days
  (user_profile(available_days ?days&:(= ?days 3))(goal Muscle_gain))
  (age_group ?ag)
  (daily_plan(days 3)(goal Muscle_gain)(age_group ?ag)(exercice_organization_type Circullar)(day_n
1)(id_d ?id1))
  (daily_plan(days 3)(goal Muscle_gain)(age_group ?ag)(exercice_organization_type Circullar)(day_n
2)(id_d ?id2))
  (daily_plan(days 3)(goal Muscle_gain)(age_group ?ag)(exercice_organization_type Circullar)(day_n
3)(id_d ?id3))
  =>
  (assert (weekly_plan(id_w ?*id*)(days 3)(rest_days d1r1d2r1d3)(week_plan_type
Full_Body)(daily_plans_id ?id1 ?id2 ?id3)))
  (bind ?*id*=(+ ?*id* 1))
  )

(defrule Week_Plan_Muscle_gain_4days
  (user_profile(available_days ?days&:(= ?days 4))(goal Muscle_gain))
  (age_group ?ag)
  (daily_plan(days 4)(goal Muscle_gain)(age_group ?ag)(exercice_organization_type Circullar)(day_n
1)(id_d ?id1))
  (daily_plan(days 4)(goal Muscle_gain)(age_group ?ag)(exercice_organization_type Circullar)(day_n
2)(id_d ?id2))
  (daily_plan(days 4)(goal Muscle_gain)(age_group ?ag)(exercice_organization_type Circullar)(day_n
3)(id_d ?id3))
  (daily_plan(days 4)(goal Muscle_gain)(age_group ?ag)(exercice_organization_type Circullar)(day_n
4)(id_d ?id4))

```

```

=>
(assert (weekly_plan(id_w ?*id*)(days 4)(rest_days d1r0d2r12d3r0d4)(week_plan_type
Upper_and_Lower_Body)(daily_plans_id ?id1 ?id2 ?id3 ?id4)))
(bind ?*id*=(+ ?*id* 1))
)
(defrule Week_Plan_Muscle_gain_5days
(user_profile(available_days ?days&:(= ?days 5))(goal Muscle_gain))
(age_group ?ag&:(neq ?ag "Old"))
(daily_plan(days 5)(goal Muscle_gain)(age_group ?ag)(exersice_organization_type Circullar)(day_n
1)(id_d ?id1))
(daily_plan(days 5)(goal Muscle_gain)(age_group ?ag)(exersice_organization_type Circullar)(day_n
2)(id_d ?id2))
(daily_plan(days 5)(goal Muscle_gain)(age_group ?ag)(exersice_organization_type Circullar)(day_n
3)(id_d ?id3))
(daily_plan(days 5)(goal Muscle_gain)(age_group ?ag)(exersice_organization_type Circullar)(day_n
4)(id_d ?id4))
(daily_plan(days 5)(goal Muscle_gain)(age_group ?ag)(exersice_organization_type Circullar)(day_n
5)(id_d ?id5))
=>
(assert (weekly_plan(id_w ?*id*)(days 5)(rest_days d1r0d2r0d3r1d4r0d5)(week_plan_type
Profi_Upper_and_Lower_Body)(daily_plans_id ?id1 ?id2 ?id3 ?id4 ?id5)))
(bind ?*id*=(+ ?*id* 1))
)
(defrule Week_Plan_Muscle_gain_6days
(user_profile(available_days ?days&:(= ?days 6))(goal Muscle_gain))
(age_group ?ag&:(neq ?ag "Old"))
(daily_plan(days 6)(goal Muscle_gain)(age_group ?ag)(exersice_organization_type Circullar)(day_n
1)(id_d ?id1))
(daily_plan(days 6)(goal Muscle_gain)(age_group ?ag)(exersice_organization_type Circullar)(day_n
2)(id_d ?id2))
(daily_plan(days 6)(goal Muscle_gain)(age_group ?ag)(exersice_organization_type Circullar)(day_n
3)(id_d ?id3))
(daily_plan(days 6)(goal Muscle_gain)(age_group ?ag)(exersice_organization_type Circullar)(day_n
4)(id_d ?id4))
(daily_plan(days 6)(goal Muscle_gain)(age_group ?ag)(exersice_organization_type Circullar)(day_n
5)(id_d ?id5))
(daily_plan(days 6)(goal Muscle_gain)(age_group ?ag)(exersice_organization_type Circullar)(day_n
6)(id_d ?id6))
=>
(assert (weekly_plan(id_w ?*id*)(days 6)(rest_days d1r0d2r0d3r1d4r0d5r0d6)(week_plan_type
Profi_Upper_and_Lower_Body)(daily_plans_id ?id1 ?id2 ?id3 ?id4 ?id5 ?id6)))
(bind ?*id*=(+ ?*id* 1)))

(defrule UnderWeight_Fat_Loss
(bmi_formal(formal_name undereight))
(user_profile(goal fat_loss))
=>
(assert (recomendation (bad_practice "Considering your BMI (UnderWeight) we recommend to change
your goal for Muscle gain and follow next training program:")
(weekly_plan_id ?*id*)))
(bind ?*id*=(+ ?*id* 1))
)

(defrule Young_LimitDays
(user_profile(available_days ?days&:(= ?days 1)|:(= ?days 7)))
(user_profile(age ?age&:(>= ?age 18)&:(<= ?age 23)))
=>

```



```
(assert (recomendation (bad_practice "Considering your Age we recommend to change number of training  
days for 2-6 days")  
(weekly_plan_id 0))))
```

Додаток В (обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ОРГАНІЗАЦІЇ ТА ПРОВЕДЕННЯ
ТРЕНУВАЛЬНОГО ПРОЦЕСУ. ЧАСТИНА 2. НАДАННЯ
РЕКОМЕНДАЦІЙ

Рис. 2.1 – Структура впливних факторів

Виконав: студент 2-го курсу,
групи 2КН-22м

спеціальності 122 «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)



Шелестюк М. І.

(прізвище та ініціали)

Керівник: д.т.н., професор кафедри КН

Яровий А. А.

(прізвище та ініціали)

« 04 » 12 2022 р.

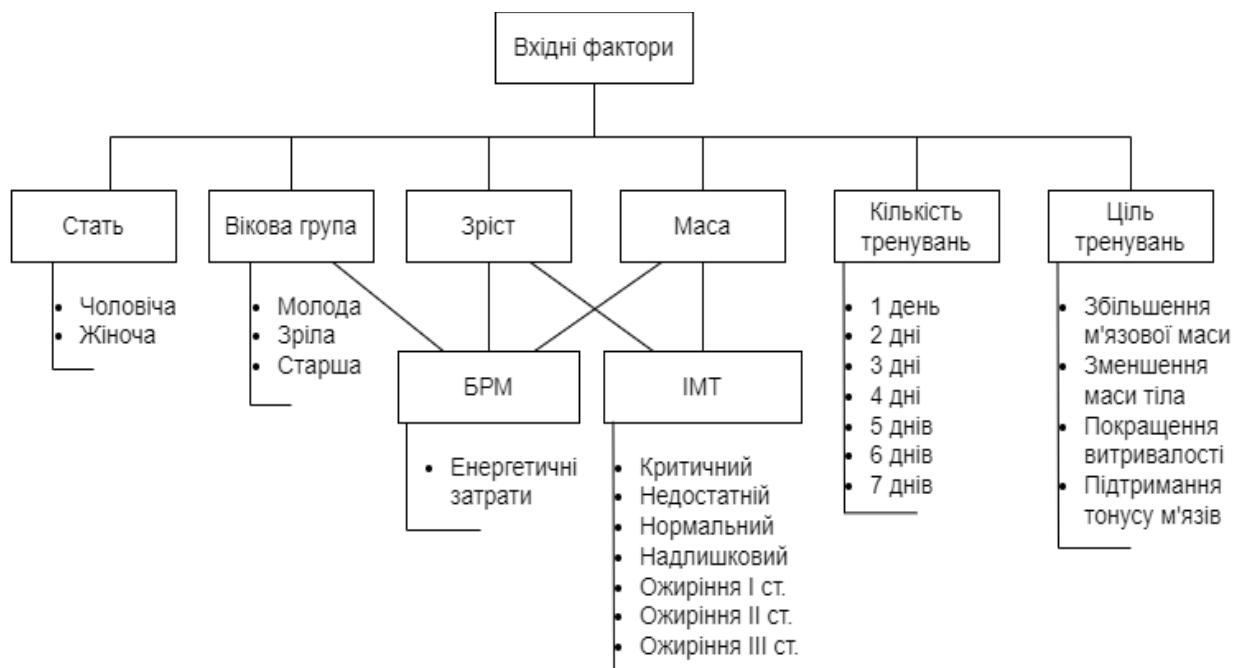


Рисунок В.1 – Структура вхідних факторів

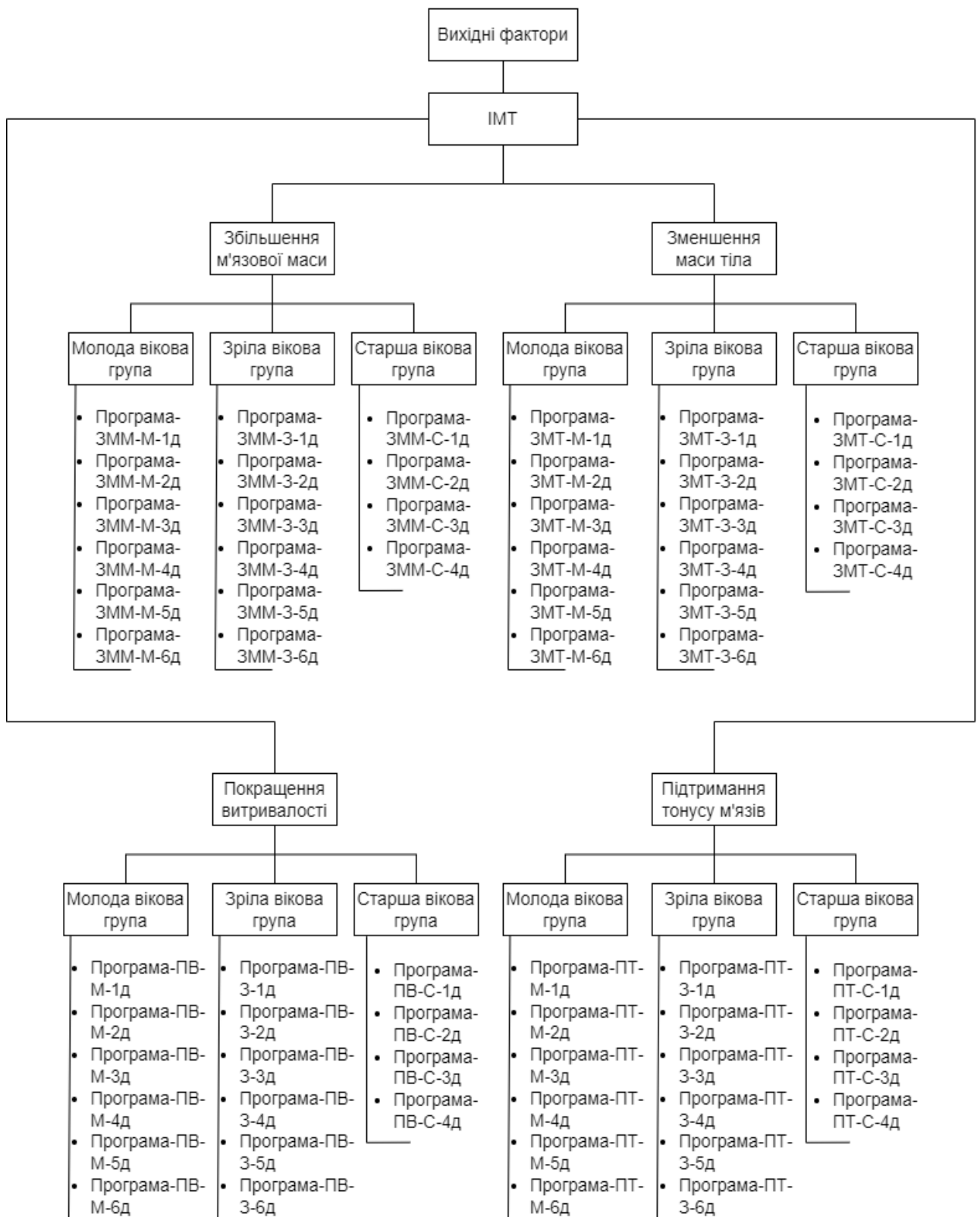


Рисунок В.2 – Структура вихідних факторів

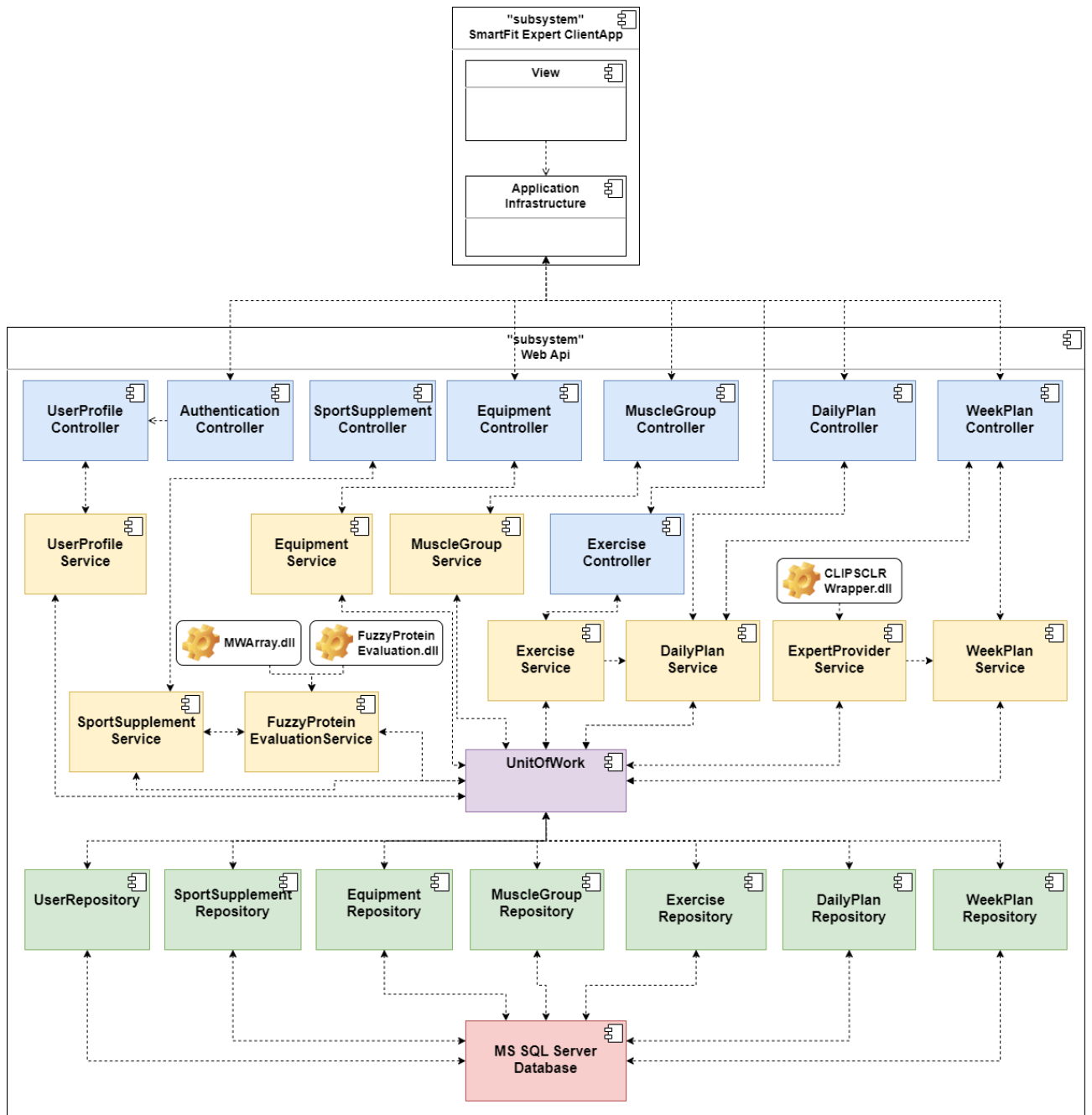


Рисунок В.3 – Діаграма компонентів серверної частини інформаційної технології

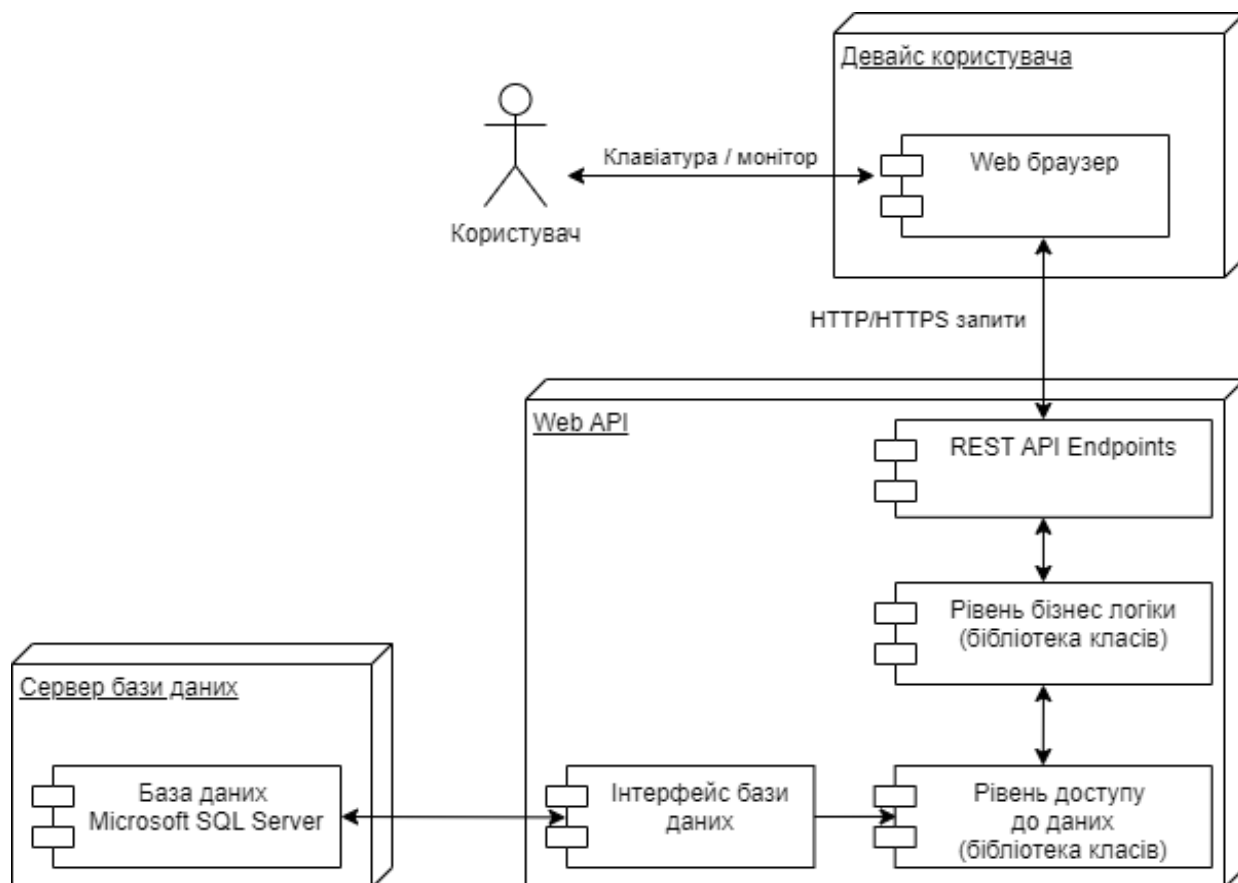


Рисунок В.4 – Діаграма розгортання інформаційної технології

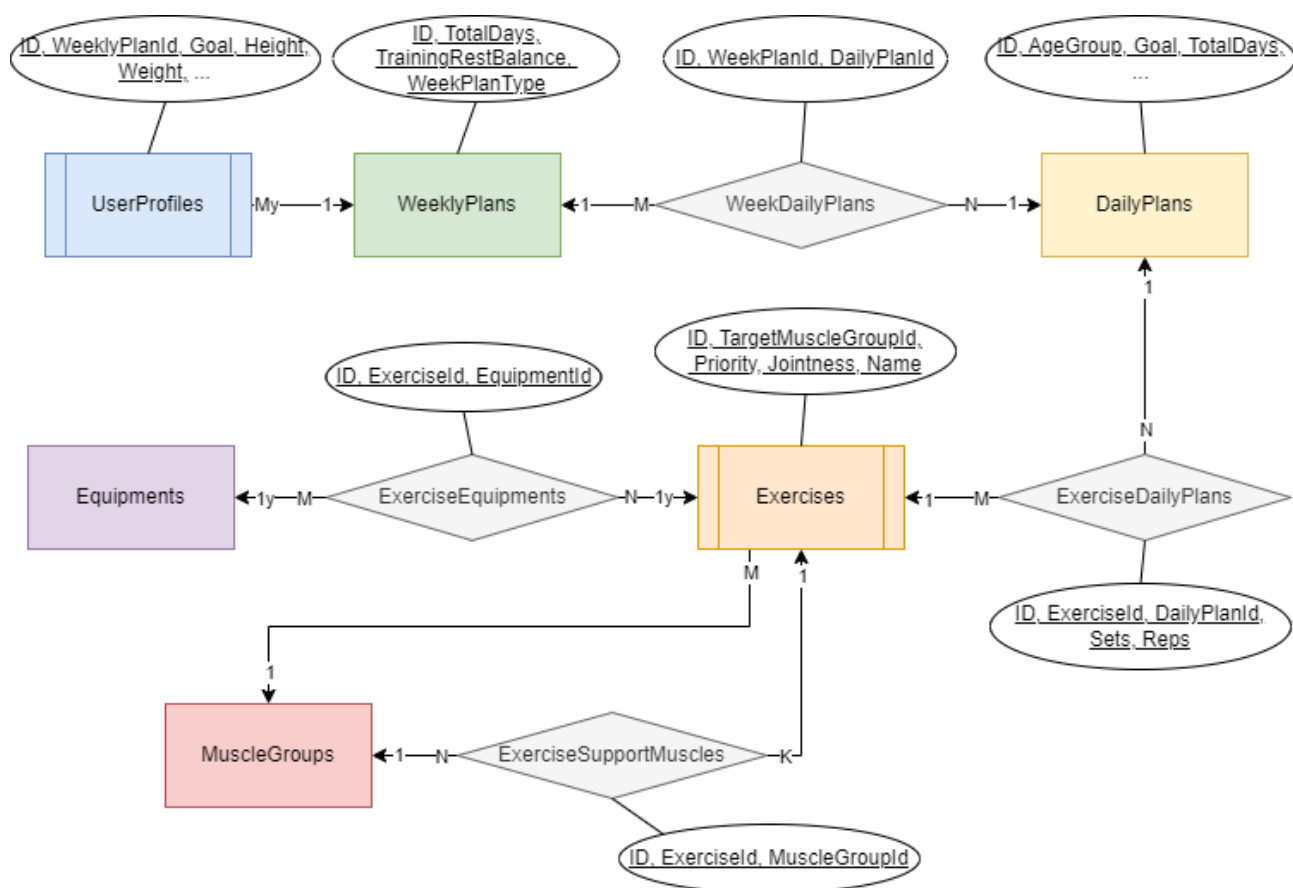


Рисунок В.5 – Інфологічна модель бази даних інформаційної технології

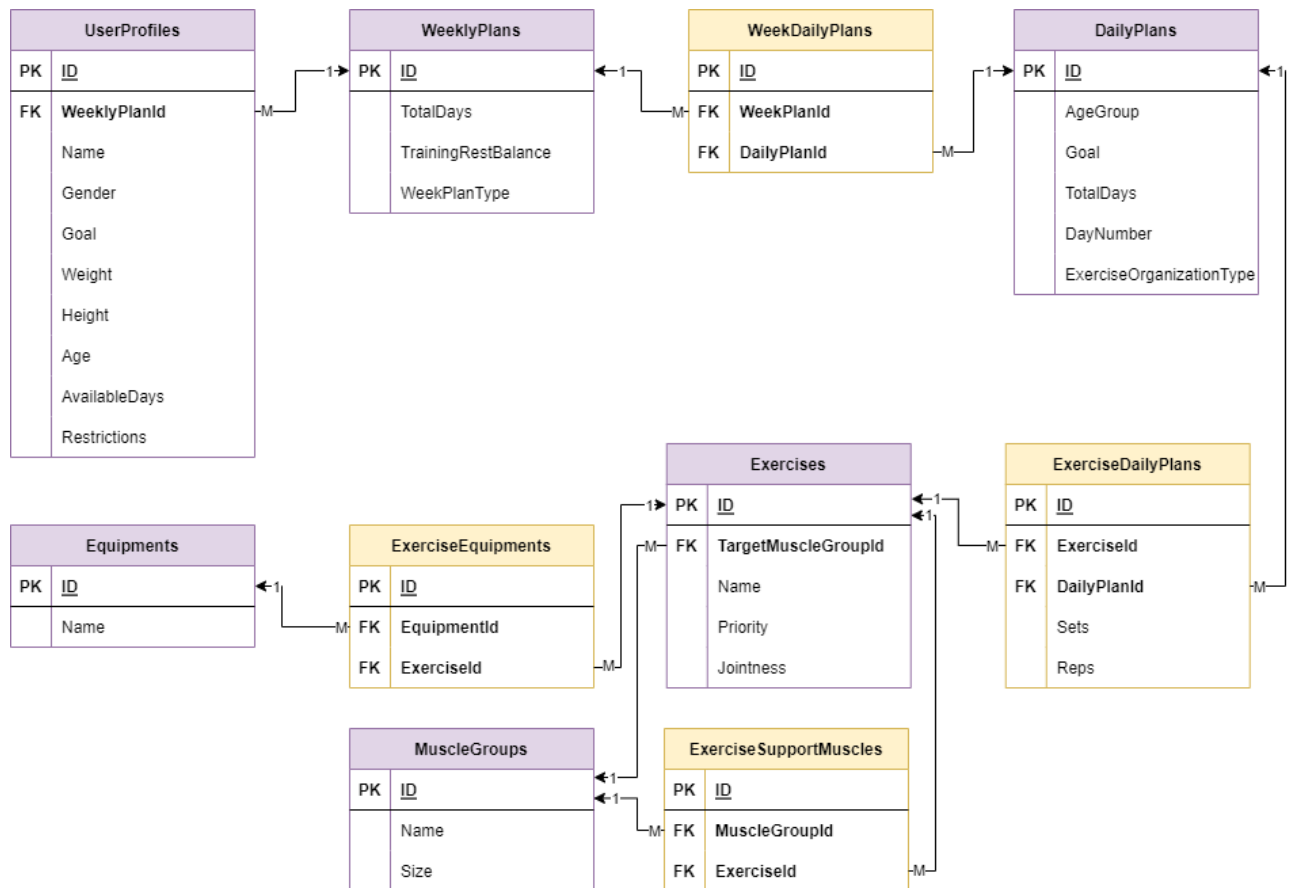


Рисунок В.6 – Даталогічна модель бази даних інформаційної технології