

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЄКТ

на тему:

«Програмний додаток контролю споживання спортивного харчування»

Студента 4 курсу, 6 групи,
спеціальності 121 «Інженерія
програмного забезпечення»
освітньої програми «Інженерія
програмного забезпечення»

Ауера
Данііла Сергійовича

підпис студента

Науковий керівник
старший викладач кафедри
інженерії програмного
забезпечення та кібербезпеки

Гнатченко Дмитро
Дмитрович

підпис керівника

Гарант освітньої програми
кандидат технічних наук,
доцент кафедри інженерії
програмного забезпечення та
кібербезпеки

Рзаєва Світлана
Леонідівна

підпис гаранта

Державний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії програмного
забезпечення та кібербезпеки

Криворучко О. В.

«14» листопада 2022 р.

Завдання на випускний кваліфікаційний проєкт студентів

Ауеру Даніілу Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема випускного кваліфікаційного проєкту «Програмний додаток
контролю споживання спортивного харчування»

Затверджена наказом ректора від «6» грудня 2022 р. № 3288

2. Строк здачі студентом закінченого проєкту 5 червня 2023

3. Цільова установка та вихідні дані до проєкту

Мета проєкту розробка програмного додатку на платформі Android для
контролю споживання спортивного харчування.

Об'єкт дослідження програмний додаток контролю споживання спортивного
харчування.

Предмет дослідження процес розробки програмного додатку контролю
споживання спортивного харчування.

4. Консультанти проєкту із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускного кваліфікаційного проєкту (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ ДОДАТКУ

1.1. Огляд спортивного харчування

1.2. Застосування програмного додатку для контролю споживання спортивного харчування

1.3. Технічне завдання

1.4. Висновок до розділу 1

РОЗДІЛ 2. ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ПРОЄКТУВАННЯ ДОДАТКУ

2.1. Вибір платформи розробки

2.2. Архітектура програмного додатку

2.3. Вибір мови програмування

2.4. Висновок до розділу 2

РОЗДІЛ 3. РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ

3.1. Налаштування середовища для розробки

3.2. Створення проєкту

3.3. Налаштування та інтеграція бібліотеки SQLite

3.4. Висновок до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання проєкту

№ пор.	Назва етапів випускного кваліфікаційного проєкту	Строк виконання етапів проєкту	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускного кваліфікаційного проєкту</i>	21.09.2022	21.09.2022
2.	<i>Розробка та затвердження завдання на проєкт</i>	14.11.2022	14.11.2022
3.	<i>Вступ та перелік літературних джерел</i>	23.12.2022	23.12.2022
4.	<i>Розділ 1. Аналіз предметної області застосування додатку</i>	27.01.2023	27.01.2023
5.	<i>Розділ 2. Вибір програмних засобів та проєктування додатку</i>	03.03.2023	03.03.2023
6.	<i>Розділ 3. Реалізація додатку</i>	14.04.2023	14.04.2023
7.	<i>Висновки</i>	28.04.2023	28.04.2023
8.	<i>Здача випускного кваліфікаційного проєкту на кафедрі (перша перевірка)</i>	17.05.2023	17.05.2023
9.	<i>Підготовка автореферату та презентації доповіді</i>	26.05.2023	26.05.2023
10.	<i>Попередній захист випускного кваліфікаційного проєкту</i>	29.05.2023 – 02.06.2023	
11.	<i>Зовнішнє рецензування випускного кваліфікаційного проєкту</i>	05.06.2023	05.06.2023
12.	<i>Здача прошого випускного кваліфікаційного проєкту на кафедрі</i>	05.06.2023	05.06.2023
13.	<i>Публічний захист випускного кваліфікаційного проєкту</i>		

7. Дата видачі завдання «14» листопада 2022 р.

8. Науковий керівник випускного кваліфікаційного проєкту _____

Гнатченко Д.Д.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми _____

Рзаєва С.Л.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент _____

Ауер Д.С.

(прізвище, ініціали, підпис)

АНОТАЦІЯ

Відповідно до мети дослідження робота присвячена розробці програмного додатку контролю споживання спортивного харчування. Метою додатку є надання користувачам зручного та ефективного інструменту для ведення обліку та контролю за споживанням спортивних продуктів.

У роботі проведений аналіз предметної області, визначені вимоги до додатку та обрано відповідні програмні засоби. Далі розглянуто процес проєктування додатку, включаючи структуру даних, дизайн інтерфейсу користувача та вибір мови програмування.

Реалізація додатку охоплює створення бази даних з використанням SQLite, розробку логіки додатку та реалізацію графічного інтерфейсу. Під час розробки були враховані особливості роботи з базою даних та забезпечено обробку помилок та оптимізацію продуктивності.

Основні переваги додатку полягають у зручному інтерфейсі, можливості ведення детального обліку продуктів, створення та керування раціонами, а також у наданні статистичних даних про споживання спортивного харчування.

Готовий програмний додаток було успішно протестовано емулятором.

Ключові слова: додаток, технологія, залежність, клас, інтеграція SQLite, База даних, Java, Android.

ABSTRACT

According to the purpose the work is devoted to the development of a software application for monitoring sports nutrition consumption. The purpose of the application is to provide users with a convenient and effective tool for keeping records and monitoring the consumption of sports products.

The paper analyzes the subject area, defines the requirements for the application, and selects appropriate software tools. Next, the application design process is considered, including the data structure, user interface design, and choice of programming language.

The application implementation includes creating a database using SQLite, developing application logic, and implementing a graphical interface. The development took into account the peculiarities of working with the database and ensured error handling and performance optimization.

The main advantages of the application are a user-friendly interface, the ability to keep detailed records of products, create and manage diets, and provide statistics on sports nutrition consumption.

The finished software application was successfully tested by an emulator.

Keywords: application, technology, dependency, class, SQLite integration, Database, Java, Android.

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ ДОДАТКУ	6
1.1. Огляд спортивного харчування	6
1.2. Застосування програмного додатку для контролю споживання спортивного харчування.....	7
1.3. Технічне завдання	8
1.4. Висновки до розділу 1	9
РОЗДІЛ 2 ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ПРОЄКТУВАННЯ ДОДАТКУ	11
2.1. Вибір платформи розробки	11
2.2. Архітектура програмного додатку.....	13
2.3. Вибір мови програмування	17
2.4. Висновок до розділу 2	22
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ДОДАТКУ	23
3.1. Налаштування платформи для розробки.....	23
3.2. Створення проекту	23
3.3. Налаштування та інтеграція бібліотеки SQL Lite	25
3.4. Висновок до розділу 3	29
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	30
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....
ДОДАТКИ

					<i>ДТЕУ 121 06-02.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Програмний додаток контролю споживання спортивного харчування	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		23.12.22		3	2	32
Керівник		Гнатченко Д.Д.		23.12.22		Факультет інформаційних технологій 4 курс, 6 група		
Гарант		Рзасва С.Л.		23.12.22				
Розробив		Ауер Д.С.		23.12.22				
					<i>Зміст</i>			

ВСТУП

Спортивне харчування відіграє важливу роль у досягненні оптимальної фізичної форми, покращенні результатів тренувань та відновленні організму після фізичних навантажень. Для досягнення цих цілей, спортсмени мають контролювати своє харчування, враховуючи специфіку тренувань та власні потреби в поживних речовинах.

В контексті розростаючого використання мобільних технологій і популярності смартфонів, розробка програмного додатку на платформі Android, що допомагає спортсменам контролювати споживання спортивного харчування, є актуальною та перспективною задачею. Такий додаток може спростити процес контролю та аналізу споживання харчових продуктів, допомагаючи спортсменам складати індивідуальні раціони та отримувати інформацію про вміст поживних речовин.

Метою роботи є розробка програмного додатку на платформі Android для контролю споживання спортивного харчування. Додаток повинен надати спортсменам зручний інструмент для введення та обліку харчових продуктів, складання і збереження індивідуальних раціонів, аналізу даних про споживання та відображення статистики.

Об'єктом дослідження є сам додаток, який розробляється з метою надання користувачам зручного та ефективного інструменту для контролю та ведення обліку споживання спортивних продуктів. Об'єкт включає в себе всі аспекти розробки додатку, включаючи аналіз предметної області, вибір програмних засобів, проектування структури даних, реалізацію логіки додатку та створення графічного інтерфейсу.

					<i>ДТЕУ 121 06-02.БР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата	Програмний додаток контролю споживання спортивного харчування	Стадія	Аркуш	Аркушів
Зав. каф.	Криворучко О.В.			23.12.22		В	3	32
Керівник	Гнатченко Д.Д.			23.12.22		Факультет інформаційних технологій 4 курс, 6 група		
Гарант	Рзаєва С.Л.			23.12.22				
Розробив	Ауер Д.С.			23.12.22	<i>Вступ</i>			

Предметом дослідження є сам процес розробки додатку, включаючи всі етапи від постановки вимог до впровадження та тестування. Дослідження включає аналіз предметної області, вибір відповідних програмних засобів, проєктування структури даних, розробку функціональності, реалізацію графічного інтерфейсу та вирішення проблем, пов'язаних з використанням бази даних. Предметом дослідження також є аналіз та оцінка ефективності та користування додатком.

У дослідженні, пов'язаному з розробкою програмного додатку контролю споживання спортивного харчування, можуть бути застосовані наступні методи:

1. Аналіз предметної області: Здійснюється детальний аналіз та вивчення предметної області спортивного харчування, включаючи його основні принципи, рекомендації, типові продукти та їх складові. Цей метод дозволяє зрозуміти потреби та вимоги користувачів додатку.
2. Вибір програмних засобів: Проводиться аналіз і порівняння різних програмних засобів та технологій, які можуть бути використані для розробки додатку. Розглядаються їх переваги, недоліки та відповідність вимогам проєкту.
3. Проєктування програмного додатку: Застосовується метод проєктування, щоб визначити структуру додатку, включаючи моделі даних, архітектуру, логіку роботи та інтерфейс користувача. Використовуються діаграми класів, діаграми взаємодії та інші інструменти проєктування.
4. Реалізація програмного додатку: Здійснюється розробка програмного коду, який відповідає вимогам та проектним рішенням. Використовуються мови програмування, фреймворки та інші інструменти для створення функціональності додатку.

						Аркуш
					ДТЕУ 121 06-02.БР	4
Зм.	Аркуш	№ докум	Підпис	Дата		

5. Тестування та валідація: Проводяться різноманітні тести, щоб переконатися в коректності роботи додатку, виявити та усунути можливі помилки та недоліки. Використовуються автоматизовані тести, модульні тести, інтеграційні тести та інші методи тестування.
6. Оцінка ефективності та користувацької зручності: Проводяться оцінка продуктивності та користувацької зручності додатку. Використовуються методи анкетування, спостереження користувачів, збір статистичних даних та інші методи оцінки.
7. Впровадження та розгортання: Останній етап дослідження передбачає впровадження розробленого додатку та його розгортання на відповідному середовищі. Використовуються методи розгортання, встановлення та налаштування додатку на платформі.

Результатом даної дипломної роботи буде функціональний мобільний додаток, який може бути ефективним інструментом для спортсменів у контролі та оптимізації споживання спортивного харчування. Цей додаток може сприяти поліпшенню результатів тренувань, досягненню поставлених спортивних цілей та загальному покращенню фізичного стану спортсменів.

					<i>ДТЕУ 121 06-02.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		5

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ ДОДАТКУ

1.1. Огляд спортивного харчування

Спортивне харчування відіграє важливу роль у досягненні спортивних цілей, таких як покращення фізичної форми, збільшення м'язової маси або підвищення витривалості.

Спортсмени потребують належної кількості макро- і мікроелементів, вітамінів, мінералів та інших поживних речовин, щоб забезпечити оптимальну роботу м'язів, відновлення та підтримку енергетичного балансу.

Основні компоненти спортивного харчування включають:

- **Білки** - білки є будівельним матеріалом для м'язової тканини та необхідними для її ремонту та регенерації після тренувань. Спортсмени повинні забезпечити достатній прийом білків, включаючи м'ясо, рибу, яйця, молочні продукти, сою та рослинні джерела білка.
- **Вуглеводи** - вуглеводи є основним джерелом енергії для м'язів під час тренувань. Вони допомагають підтримувати глікогенові запаси в м'язах та забезпечують необхідну енергію для виконання фізичних навантажень. Збагачені вуглеводами продукти включають цільні злаки, фрукти, овочі та цукрівці.
- **Жири** - жири є джерелом концентрованої енергії та необхідні для нормального функціонування організму. Вони допомагають усвоювати жиророзчинні вітаміни, забезпечують захист внутрішніх органів та забезпечують надлишковий джерело

					<i>ДТЕУ 121 06-02.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Програмний додаток контролю споживання спортивного харчування	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		27.01.23		<i>РІ</i>	6	32
Керівник		Гнатченко Д.Д.		27.01.23		Факультет інформаційних технологій 4 курс, 6 група		
Гарант		Рзаєва С.Л.		27.01.23				
Розробив		Ауер Д.С.		27.01.23	<i>Аналіз предметної області застосування додатку</i>			

енергії під час тривалих тренувань. Жири можна отримати з риби, оливкової олії, горіхів, насіння та авокадо.

- **Вітаміни та мінерали** - вітаміни та мінерали є необхідними для нормального функціонування організму та підтримки фізичної активності. Деякі вітаміни, наприклад, вітамін С та Е, мають антиоксидантну дію і допомагають в боротьбі зі стресом, який виникає під час тренувань. Мінерали, такі як кальцій, залізо та магній, важливі для підтримки здорових кісток, кровоутворення та нормальної роботи м'язів.

Загальна кількість та співвідношення цих компонентів спортивного харчування можуть різнитися в залежності від виду спорту, тривалості тренувань та індивідуальних потреб спортсмена. Оптимальний харчовий раціон може бути складним для розрахунку та контролю, особливо при великій кількості продуктів та інформації про їхній склад.

1.2. Застосування програмного додатку для контролю споживання спортивного харчування

Завдяки швидкому розвитку технологій і поширенню мобільних пристроїв, програмні додатки стали незамінними інструментами для контролю та управління різними аспектами нашого життя, включаючи харчування. В сфері спортивного харчування програмний додаток може стати надзвичайно корисним інструментом для спортсменів, тренерів та дієтологів.

Програмний додаток для контролю споживання спортивного харчування дозволяє спортсменам вести детальний облік свого харчування, включаючи введення їжі та напоїв, їх кількості, складу та калорійності. За допомогою вбудованих баз даних продуктів, спортсмени можуть швидко знайти необхідну інформацію про поживну цінність конкретного продукту.

						Аркуш
					<i>ДТЕУ 121 06-02.БР</i>	7
Зм.	Аркуш	№ докум	Підпис	Дата		

Також можливість додавання власних продуктів забезпечує індивідуалізацію додатку під потреби кожного користувача.

Одним із основних функціональних елементів такого додатку є ведення щоденника споживання, де користувач може фіксувати всі прийоми їжі та напоїв протягом дня. Додаток може відображати загальну кількість спожитих калорій, а також розподіл білків, вуглеводів і жирів у раціоні спортсмена. Це дає можливість контролювати збалансованість харчування та дотримання необхідного співвідношення поживних речовин.

Крім того, програмний додаток може надавати рекомендації щодо оптимального харчового раціону для досягнення конкретних спортивних цілей, таких як збільшення м'язової маси, зниження ваги або поліпшення енергетичного балансу. Він може розраховувати індивідуальні потреби спортсмена в поживних речовинах на основі його біологічних показників, тренувань та особистих цілей.

Програмний додаток також може мати функцію моніторингу забезпечення спортсмена необхідною кількістю води та вітамінів. Це дозволяє забезпечити оптимальну гідратацію та надати необхідні поживні речовини для забезпечення нормального функціонування організму під час фізичних навантажень.

Застосування програмного додатку для контролю споживання спортивного харчування може сприяти покращенню результатів тренувань та досягненню спортивних цілей.

1.3. Технічне завдання

Основні вимоги до функціональності додатку включають:

- Можливість реєстрації та авторизації користувачів.

						Аркуш
					ДТЕУ 121 06-02.БР	8
Зм.	Аркуш	№ докум	Підпис	Дата		

- Можливість додавання продуктів до бази даних, включаючи інформацію про їх калорійність, поживні речовини та інші властивості.
- Можливість створення та ведення раціону харчування користувача, включаючи додавання продуктів до раціону та відстеження споживання.
- Розрахунок калорійності та поживної цінності раціону на основі доданих продуктів.
- Надання рекомендацій щодо оптимального споживання спортивних продуктів, враховуючи потреби користувача.

Окрім функціональних вимог, технічне завдання також визначає інші важливі аспекти:

- Платформа та мова програмування: додаток буде розроблено для платформи Android з використанням мови програмування Java або Kotlin.
- База даних: для зберігання даних про продукти та раціон користувача буде використана локальна база даних SQLite.
- Інтерфейс користувача: додаток повинен мати зрозумілий та привабливий інтерфейс, що забезпечує зручну взаємодію з користувачем.

1.4. Висновки до розділу 1

Отже, було проведено аналіз предметної області застосування програмного додатку для контролю споживання спортивного харчування. Було розглянуто основні аспекти спортивного харчування, включаючи білки, вуглеводи, жири, вітаміни та мінерали, які є важливими компонентами збалансованого раціону спортсмена.

						Аркуш
						9
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 06-02.БР	

На основі отриманої інформації та аналізу можна зробити висновок, що розробка нового програмного додатку для контролю споживання спортивного харчування має великий потенціал для задоволення потреб спортсменів та покращення їхніх спортивних результатів.



						Аркуш
					ДТЕУ 121 06-02.БР	10
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 2

ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ПРОЄКТУВАННЯ ДОДАТКУ

2.1. Вибір платформи розробки

При розробці мобільного додатку для Android контролю споживання спортивного харчування необхідно вирішити питання вибору платформи розробки. Вибір правильної платформи є важливим етапом, який може вплинути на функціональність, продуктивність та доступність додатку.

Для розробки Android-додатків доступні різні мови та інструменти. Одним з найпоширеніших інструментів є Android Studio, який є офіційним інтегрованим середовищем розробки (IDE) для платформи Android. Android Studio надає розширені можливості для розробки, тестування та налагодження додатків, а також забезпечує доступ до багатьох бібліотек та інструментів, що спрощують процес розробки.

Для розробку додатку буде використано середовище розробки IntelliJ IDEA - комерційне інтегроване середовище розробки для різних мов програмування (Java, Python, Scala, PHP та ін.) від компанії JetBrains. В цьому середовищі є інтеграція Android Studio зі спрощеним процесом розробки додатків на платформі Android. До складу IntelliJ IDEA включені напрацювання, створені в результаті спільної роботи з компанією Google, яка використовувала IntelliJ IDEA як базис для свого нового відкритого середовища розробки Android Studio. Завдяки співпраці істотно розширені штатні можливості IntelliJ IDEA з розробки застосунків для платформи Android.

Зм.	Аркуш	№ докум.	Підпис	Дата			
Зав. каф.		Криворучко О.В.		03.03.23	Програмний додаток контролю споживання спортивного харчування		
Керівник		Гнатченко Д.Д.		03.03.23			
Гарант		Котенко Н.О.		03.03.23	Факультет інформаційних технологій 4 курс, 6 група		
Розробив		Ауер Д.С.		03.03.23			
					ДТЕУ 121 06-02.БР		
					Вибір програмних засобів та проєктування додатку		
					Стадія	Аркуш	Аркушів
					P2	11	32

Для розробки бази даних у додатку буде використано одну з вбудованих бібліотек SQLite.

SQLite - це легка, вбудовувана реляційна база даних, яка забезпечує збереження, організацію та доступ до даних в мобільних та вбудованих системах. Вона є одним з найпоширеніших способів збереження даних на платформі Android, оскільки входить до складу Android SDK і має вбудовану підтримку. SQLite втілена у вигляді бібліотеки, де реалізовано багато зі стандарту SQL-92. Початковий код SQLite поширюється як суспільне надбання (англ. public domain), тобто може використовуватися без обмежень та безоплатно з будь-якою метою.

Основні особливості та переваги SQLite включають:

Легкість використання: SQLite має простий та зрозумілий API, що спрощує роботу з ним. Його можна легко інтегрувати в Android-додаток і працювати з ним за допомогою стандартних SQL-запитів.

Вбудованість: SQLite є вбудованою базою даних, що означає, що для використання його не потрібно встановлювати додаткові сервери або налаштовувати зовнішні сервіси. Це спрощує розгортання та розповсюдження додатків, особливо на мобільних пристроях.

Компактність: SQLite база даних зберігається у вигляді одного файлу на пристрої. Це дозволяє ефективно використовувати ресурси пристрою та забезпечує низький рівень футпринта бази даних.

Підтримка транзакцій: SQLite підтримує транзакції, що дозволяє забезпечувати консистентність та цілісність даних. Транзакції дозволяють групувати операції з базою даних і забезпечують атомарність, цілісність та ізоляцію даних.

Ефективність: SQLite має низький рівень накладних витрат і добре працює з невеликими та середніми обсягами даних. Він швидко виконує запити та забезпечує ефективне використання ресурсів пристрою.

					<i>ДТЕУ 121 06-02.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		12

Підтримка SQL: SQLite повністю підтримує мову структурованих запитів SQL, що дозволяє легко виконувати операції створення, читання, оновлення та видалення даних. Це дозволяє розробникам використовувати потужний і знайомий мовний інтерфейс для роботи з базою даних.

SQLite є чудовим вибором для зберігання локальних даних в мобільних додатках Android, зокрема для контролю споживання спортивного харчування. Він надає зручність, ефективність та надійність для збереження та управління даними, дозволяючи користувачам зберігати та отримувати необхідну інформацію щодо їх спортивного харчування.

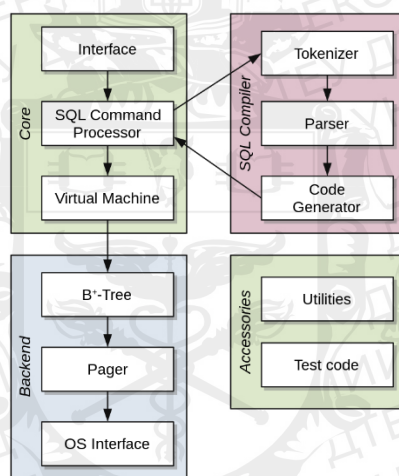


Рис. 2.1. Архітектура SQLite

2.2. Архітектура програмного додатку

Архітектура програмного додатку відіграє ключову роль у забезпеченні його розширюваності, підтримки та підтримки принципів доброї практики розробки програмного забезпечення. Для розробки програмного додатку на Android для контролю споживання спортивного харчування рекомендується використовувати розподілену архітектуру з низьким зв'язком між компонентами. Одним з популярних підходів до розробки таких архітектур є Model-View-ViewModel (MVVM).

						Аркуш
						13
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 06-02.БР	

MVVM використовується для відокремлення моделі та її відображення. Необхідністю цього є надання можливості змінювати їх незалежно одну від одної. Наприклад, розробник працює над логікою роботи з даними, а дизайнер — з користувацьким інтерфейсом.

MVVM була створена з метою поділу праці дизайнера і програміста, що є неможливим, коли Java-розробник намагається побудувати GUI в Swing або розробник на Visual C++ намагається створити користувацький інтерфейс в MFC. Розробники кмітливі і мають безліч навичок, але створення зручних і привабливих інтерфейсів вимагає абсолютно інших талантів, ніж ті, якими вони володіють. Ця робота більше підходить для дизайнерів інтерфейсів. Хороші дизайнери інтерфейсів краще знають, чого бажають користувачі, ніж експерти в області проектування і написання коду. Зрозуміло, буде краще, якщо дизайнер інтерфейсів створить інтерфейс, а розробник напише код, який реалізує логіку цього інтерфейсу, але технології типу Swing або MFC просто-напросто не дозволяють чинити таким чином.

Архітектура MVVM складається з наступних компонентів:

- **Модель (Model):** Модель відповідає за представлення даних додатку та логіку їх обробки. Вона може включати класи, які взаємодіють з базою даних, зовнішніми службами або мережевими джерелами. Модель не залежить від реалізації користувацького інтерфейсу та не взаємодіє напряму з користувачем.
- **Представлення (View):** Представлення відповідає за відображення даних користувачу та обробку користувацьких дій. Воно включає компоненти, такі як активності (Activities) або фрагменти (Fragments), які відповідають за відображення інтерфейсу користувача. Представлення повинно бути

						Аркуш
					ДТЕУ 121 06-02.БР	14
Зм.	Аркуш	№ докум	Підпис	Дата		

безпосередньо пов'язане з моделлю і отримувати дані через ViewModel.

- **ViewModel:** ViewModel виступає посередником між представленням та моделлю. Вона містить бізнес-логіку та стан додатку. ViewModel надає методи та дані, які можуть бути прив'язані до представлення для відображення та оновлення даних. Вона також реагує на користувацькі дії, обробляє їх та оновлює модель.

Ця архітектура дозволяє досягти розподіленості компонентів, що полегшує тестування та модифікацію окремих частин додатку. Крім того, вона сприяє повторному використанню коду та полегшує розробку масштабованих додатків. На наступному рисунку (Рисунок 2) представлена діаграма взаємодії між компонентами шаблону MVVM.

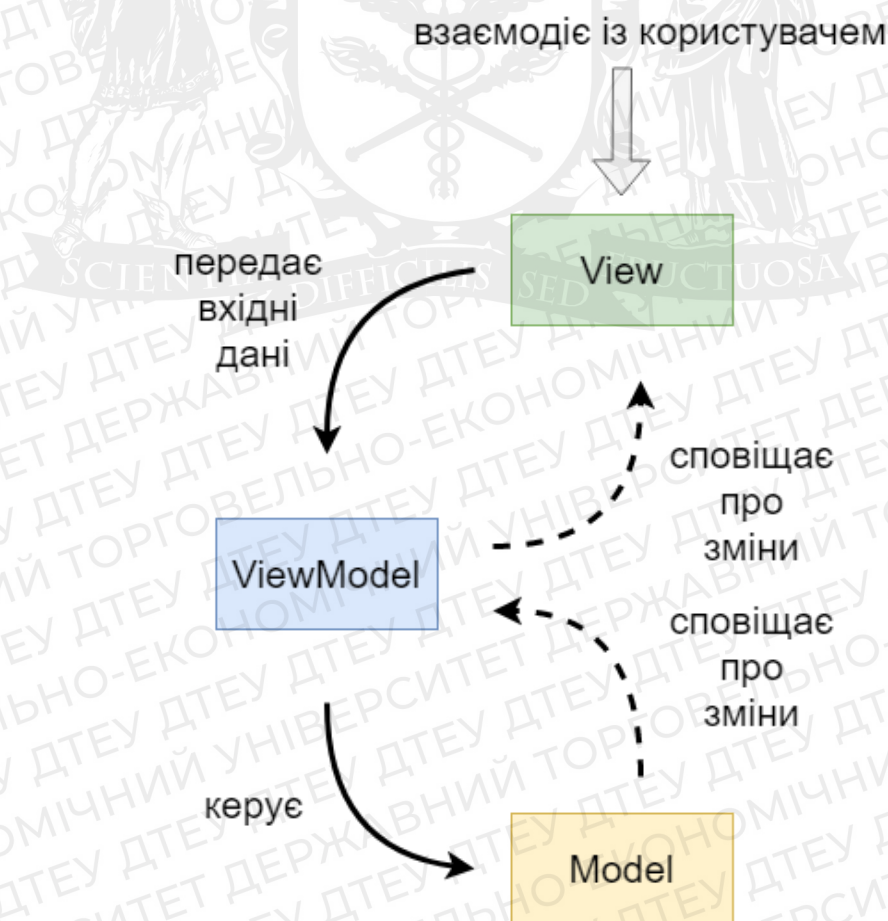


Рис. 2.2. Діаграма взаємодії MVVM

					ДТЕУ 121 06-02.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		15

У контексті програмного додатку для контролю споживання спортивного харчування, модель може включати класи для зберігання і керування даними про продукти харчування, їх характеристики та калорійні значення. Представлення може включати активності або фрагменти для відображення списків продуктів, детальну інформацію про продукти та введення даних користувача. ViewModel відповідатиме за забезпечення потрібних даних для представлення, обробку введення користувача та взаємодію з моделлю для збереження та отримання даних.

MVVM зручно використовувати замість класичного MVC та йому подібних у тих випадках, коли на платформі, де ведеться розробка, присутне «зв'язування даних».

В MVC/MVP зміни у користувацькому інтерфейсі не впливають безпосередньо на модель, а йдуть через Контролер/Presenter. У таких технологіях, як WPF та Silverlight, присутня концепція «зв'язування даних», що дозволяє зв'язувати дані із візуальними елементами в обидві сторони.

Архітектура MVVM вирішує цю проблему ясним поділом відповідальності:

- Розробка користувацького інтерфейсу здійснюється дизайнером інтерфейсів за допомогою технології, більш-менш природної для такої роботи (XML)
- Логіка користувацького інтерфейсу реалізується розробником як компонент ViewModel
- Функціональні зв'язки між користувацьким інтерфейсом та ViewModel реалізуються через біндинги (bindings), які, по суті, є правилами типу «якщо кнопка А була натиснута, повинен бути викликаний метод `onButtonAClick()` з ViewModel». Біндинги можуть бути написані в коді або визначені декларативним шляхом (Android використовує обидва типи).

						ДТЕУ 121 06-02.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата			16

Архітектура MVVM використовується в тому чи іншому вигляді усіма сучасними технологіями, наприклад Microsoft WPF і Silverlight, Oracle JavaFX, Adobe Flex, AJAX.

Застосування архітектури MVVM допоможе створити добре організований та легко розширюваний програмний додаток для контролю споживання спортивного харчування на платформі Android.

2.3. Вибір мови програмування

Вибір мови програмування є важливим етапом розробки, оскільки від цього вибору залежать продуктивність, ефективність та зручність розробки додатку. Платформа Android підтримує дві мови для розробки додатків – Java та Kotlin.

Мова програмування Java є об'єктно-орієнтованою мовою з високим рівнем абстракції. Вона була розроблена компанією Sun Microsystems (пізніше придбана компанією Oracle) і вперше випущена в 1995 році. Java була створена з метою створення платформонезалежного мовного середовища, що дозволяє розробляти програми, які можуть працювати на різних операційних системах та пристроях.

Однією з найважливіших особливостей мови Java є її платформонезалежність. Програми, написані на Java, компілюються в байт-код, який може бути виконаний на будь-якій віртуальній машині Java (JVM). Це дозволяє розробникам писати код один раз і запускати його на різних платформах без необхідності в перекомпіляції. Віртуальна машина Java забезпечує керування пам'яттю, збирання сміття та інші операції, що спрощують роботу з пам'яттю та полегшують розробку.

Java має потужну стандартну бібліотеку, яка надає різноманітні класи та інтерфейси для роботи зі строками, колекціями, мережами, базами даних,

						Аркуш
					ДТЕУ 121 06-02.БР	17
Зм.	Аркуш	№ докум	Підпис	Дата		

графікою та багатьма іншими аспектами програмування. Це спрощує розробку програм та забезпечує готові рішення для багатьох типових задач.

Ще однією з сильних сторінок Java є безпека. Вона має механізми для керування виключеннями, що допомагають уникати помилок та вирішувати проблеми під час виконання програми. Java також має можливості для контролю доступу, що дозволяє обмежувати доступ до ресурсів та забезпечувати безпеку даних.

Java підтримує розробку великих та складних додатків завдяки своїй об'єктно-орієнтованій природі. Розробники можуть створювати класи, об'єкти та використовувати концепції успадкування, поліморфізму та інкапсуляції для створення модульних та масштабованих додатків.

Java також має велику та активну спільноту розробників, що сприяє обміну досвідом, підтримці та розробці різноманітних інструментів та фреймворків.

У підсумку, мова програмування Java є потужним інструментом для розробки різноманітних програмних додатків. Вона пропонує платформнонезалежність, багату бібліотеку, безпеку та широкі можливості розробки завдяки своїй об'єктно-орієнтованій природі. З цими перевагами Java є популярним вибором для розробки додатків у різних сферах, включаючи підприємницькі, мобільні та веб-додатки.

Мова програмування Kotlin є статично типізованою мовою, що працює на віртуальній машині Java (JVM). Вона була розроблена компанією JetBrains і представлена в 2011 році. Kotlin поєднує в собі вигоди об'єктно-орієнтованого та функціонального програмування, що робить її привабливим вибором для розробки різних типів додатків.

Однією з основних переваг Kotlin є його виразність та читабельність коду. Вона надає простий та зрозумілий синтаксис, що дозволяє зменшити кількість написаного коду порівняно з мовою Java. Kotlin підтримує такі

						Аркуш
					ДТЕУ 121 06-02.БР	18
Зм.	Аркуш	№ докум	Підпис	Дата		

конструкції, як null-безпечність, розширення функціональності інтерфейсів, лямбда-функції та інші, що сприяють більш ефективному та зрозумілому програмуванню.

Мова Kotlin інтегрується з Java, що дозволяє розробникам поступово переходити від коду на Java до Kotlin без втрати сумісності з існуючими Java-бібліотеками та інфраструктурою. Це дозволяє використовувати як Kotlin, так і Java у одному проекті та забезпечує гнучкість вибору мови залежно від потреб розробки.

Одним з найважливіших аспектів Kotlin є безпека та надійність коду. Kotlin підтримує строгу статичну типізацію, що дозволяє виявляти помилки ще на етапі компіляції. Він також надає безпечну обробку виключень, забезпечує захист від пам'ятевих помилок та інших типових проблем програмування.

Іншою перевагою Kotlin є його підтримка великої спільноти розробників. Kotlin має активне та зростаюче співтовариство, що сприяє обміну знаннями, розробці інструментів та фреймворків. Багато популярних бібліотек та фреймворків, таких як Android, Spring та Ktor, активно підтримують Kotlin.

У підсумку, Kotlin є потужною та зрілою мовою програмування, яка має багато переваг порівняно з іншими мовами, зокрема з Java. Вона пропонує виразний синтаксис, підтримку функціонального програмування, безпеку та надійність коду, а також інтеграцію з Java. Завдяки цим перевагам, Kotlin став популярним вибором для розробки різних типів додатків, включаючи мобільні додатки, веб-програми, серверні додатки та інші.

Під час вибору мови програмування були розглянуті наступні аспекти:

- Платформа Android: Оскільки додаток розробляється для платформи Android, необхідно вибрати мову програмування, яка підтримується на цій платформі. Для розробки Android-додатків

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 06-02.БР	19

основними мовами програмування є Java та Kotlin. Обидві мови мають добру підтримку від Google та широкий спільноту розробників.

- **Продуктивність та ефективність:** Одним з критеріїв вибору мови програмування є продуктивність та ефективність розробки. У цьому контексті Kotlin може бути перевагою, оскільки вона надає багато функцій, які полегшують розробку, такі як інтероперабельність з Java, нульова безпека, розширення функціональності стандартної бібліотеки та інші.
- **Спільнота розробників:** Важливо також врахувати спільноту розробників та доступність ресурсів для вивчення та підтримки обраної мови програмування. Якщо мова має активну спільноту розробників і велику кількість ресурсів, це може полегшити розробку та забезпечити швидшу підтримку в разі виникнення проблем.
- **Майбутнє розвитку:** Одним з важливих аспектів є оцінка майбутнього розвитку обраної мови програмування. Важливо враховувати активність розробників мови, оновлення та плани розвитку мови. Kotlin, наприклад, має активне розвитку та підтримку від Google, що робить її перспективною мовою для розробки Android-додатків.
- **Платформна незалежність:** Мова Java є платформонезалежною, що означає, що програмний код, написаний на Java, може працювати на різних платформах без необхідності в перекомпіляції. Це важливо для додатків, які можуть бути розгорнуті на різних операційних системах або пристроях.
- **Велика спільнота розробників:** Java має велику та активну спільноту розробників, що забезпечує доступ до великої кількості

						Аркуш
						20
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 06-02.БР	

ресурсів, документації та підтримки. Це дозволяє розробникам швидко знаходити рішення для виникаючих проблем і отримувати оновлення щодо мови та платформи.

- Багата екосистема: Java має велику екосистему з великою кількістю бібліотек, фреймворків та інструментів, які сприяють розробці широкого спектру додатків. Це включає рішення для роботи з базами даних, мережевими операціями, веб-розробкою та багато іншого. Використання такої розширеної екосистеми спрощує розробку та розширення функціональності додатку.
- Надійність та безпека: Java відома своєю надійністю та безпекою. Мова має вбудовану систему керування пам'яттю, яка допомагає уникнути багатьох типових помилок програмування, таких як утечки пам'яті. Крім того, Java має багато інструментів для контролю безпеки, таких як механізм виключень, обмеження доступу та інші.
- Широке застосування: Java є однією з найпопулярніших мов програмування у світі. Вона використовується у багатьох індустріях та сферах, включаючи розробку підприємницьких додатків, фінансові системи, веб-розробку та багато іншого. Це означає, що знання Java має велику цінність на ринку праці та забезпечує широкі можливості для розробників

З урахуванням усіх цих факторів було прийняте рішення використовувати мову програмування Java для реалізації програмного додатку контролю споживання спортивного харчування. Java надає надійну та безпечну платформу для розробки, має велику спільноту розробників та широку екосистему, що сприяє швидкій та ефективній розробці додатку.

						Аркуш
					ДТЕУ 121 06-02.БР	21
Зм.	Аркуш	№ докум	Підпис	Дата		

2.4. Висновок до розділу 2

Отже, при виборі програмних засобів для розробки було визначено використання мови програмування Java, з урахуванням її популярності та широкої підтримки на платформі Android. Для реалізації локального зберігання даних було обрано SQLite, який забезпечує зручний та ефективний доступ до даних на мобільному пристрої.

У процесі проектування програмного додатку була використана архітектура MVVM, яка дозволяє розділити компоненти додатку та забезпечити їх взаємодію з мінімальним зв'язком. Ця архітектура сприяє модульності, тестуванню та підтримці додатку в майбутньому.

Результатом вибору програмних засобів та проектування став детальний план розробки мобільного додатку для контролю споживання спортивного харчування на платформі Android. Подальша реалізація та імплементація цього плану дозволить створити ефективний та зручний інструмент для користувачів, що допоможе їм досягти своїх спортивних та харчових цілей.

						Аркуш
					ДТЕУ 121 06-02.БР	22
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ДОДАТКУ

3.1. Налаштування платформи для розробки

Для початку розробки проекту, першим кроком необхідно налаштувати платформу для розробки. Для цього необхідно завантажити IntelliJ IDEA з офіційного сайту та інтегрувати Android Studio.

Для використання Android Studio в IntelliJ IDEA необхідно:

1. Відкрити IntelliJ IDEA
2. Перейти до розділу “Plugins” в налаштуваннях IntelliJ IDEA.
3. У полі пошуку ввести “Android”
4. Активувати плагін натиснувши на чек-бокс

3.2 Створення проекту

Для того щоб створити проект необхідно натиснути кнопку New Project.

Після натискання у спеціальному вікні необхідно обрати операційну систему Android, тип пристрою, у нашому випадку Phone and Tablet , та натиснути кнопку Next.

Далі потрібно обрати назву проекту, назву пакетів проекту, а також місце збереження, та натиснути кнопку Finish.

Після виконаних дій буде створено пустий проект в якому буде одна активність MainActivity, а також один XML файл для неї, в якому можна опрацьовувати візуальну частину проекту.

					<i>ДТЕУ 121 06-02.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Програмний додаток контролю споживання спортивного харчування <i>Реалізація додатку</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Зав. каф.		Криворучко О.В.		14.04.23		<i>РЗ</i>	23	32
Керівник		Гнатченко Д.Д.		14.04.23		<i>Факультет інформаційних технологій 4 курс, 6 група</i>		
Гарант		Рзаєва С.Л.		14.04.23				
Розробив		Ауер Д.С.		14.04.23				

Після створення проекту можна приступати до розробки додатку слідуючи етапам процесу розробки:

Постановка завдань: На цьому етапі визначаються основні вимоги до програмного додатку. Встановлюються функціональні та нефункціональні вимоги, які визначають, яким чином додаток повинен працювати та які функції він має надавати.

Проектування: На етапі проектування визначаються структура та архітектура додатку. Розробляється модель даних, проектується екрани та інтерфейс користувача. Важливо врахувати зручність взаємодії користувача з додатком та його естетичний вигляд.

Реалізація: На цьому етапі розробляється програмний код додатку. Використовуються обрані мови програмування та технології для створення функціональності, включаючи роботу з базою даних, обробку даних та взаємодію з користувачем. Розробка може бути поділена на окремі модулі або компоненти для забезпечення кращої організації та підтримки коду.

Тестування: Після реалізації додатку проводиться тестування для виявлення та виправлення помилок та недоліків. Виконуються різні види тестів, включаючи модульні тести, функціональні тести, тести на взаємодію та тести на продуктивність. Метою тестування є забезпечення якості та надійності додатку перед його впровадженням.

Впровадження: Після успішного проходження тестування додаток готовий до впровадження. Це включає підготовку пакету додатку для розповсюдження та публікацію в магазині додатків. Важливо забезпечити встановлення та налагодження додатку на цільових пристроях користувачів.

Підтримка та вдосконалення: Після впровадження додатку необхідно забезпечувати підтримку користувачів та виконувати вдосконалення додатку. Це включає виправлення помилок, врахування отриманого відгуку користувачів та впровадження нових функцій та можливостей.

						Аркуш
						24
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 06-02.БР	

3.3 Налаштування та інтеграція бібліотеки SQL Lite

Впровадження бази даних SQLite у програмний додаток контролю споживання спортивного харчування вимагає деяких особливих розглядів та вирішення питань, що стосуються створення, зміни та використання бази даних. Основні особливості впровадження SQLite у додаток включають:

- Створення бази даних: Першим кроком є створення бази даних SQLite, включаючи таблиці та відповідні стовпці для зберігання необхідної інформації. Для цього можна скористатися SQL-скриптами або використати інструменти, які надає Android SDK.
- Модель даних: Необхідно визначити модель даних для додатку, тобто структуру таблиць та взаємозв'язки між ними. Це допоможе організувати збереження даних про продукти та раціон користувача з урахуванням їхніх взаємозв'язків та залежностей.
- Маніпулювання даними: Для взаємодії з базою даних SQLite потрібно реалізувати необхідні операції, такі як додавання, оновлення, видалення та отримання даних. Це включає в себе написання SQL-запитів та використання відповідних методів API для взаємодії з базою даних.
- Маніпулювання схемою бази даних: Іноді може виникнути потреба в зміні схеми бази даних, наприклад, додавання нових таблиць, стовпців або виконання міграцій. Для цього необхідно використовувати механізми, що надаються SQLite, та враховувати вплив таких змін на існуючі дані та логіку додатку.

						Аркуш
						25
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 06-02.БР	

- Обробка помилок та оптимізація: Важливо враховувати можливі помилки, які можуть виникнути при роботі з базою даних, і відповідно їх обробляти. Крім того, можна виконати деякі оптимізації, такі як використання індексів, транзакцій та інших методів для покращення продуктивності та ефективності роботи з базою даних.

Пропоную розпочати роботу з інтеграції необхідних бібліотек для нього. Для цього достатньо перейти в файл build.gradle, і додати залежність наступним чином:

```
dependencies {
    implementation 'androidx.room:room-runtime:2.4.0'
    annotationProcessor 'androidx.room:room-compiler:2.4.0'
}
```

Після цього потрібно створити клас-сутність (entity) для представлення таблиці бази даних. Нам потрібно створити таблицю "Products" з колонками "id", "name" та "price", клас-сутність буде мати наступний вигляд:

```
import androidx.room.Entity;
import androidx.room.PrimaryKey;

@Entity(tableName = "products")
public class Product {
    @PrimaryKey(autoGenerate = true)
    private int id;
    private String name;
    private double price;
}
```

						Аркуш
						26
Зм.	Аркуш	№ докум	Підпис	Дата	ДТЕУ 121 06-02.БР	

Далі створюємо клас, який буде представляти базу даних та міститиме абстрактний метод для отримання інтерфейсу доступу до таблиць:

```
import androidx.room.Dao;  
import androidx.room.Insert;  
import androidx.room.Query;
```

@Dao

```
public interface ProductDao {  
    @Insert  
    void insert(Product product);  
  
    @Query("SELECT * FROM products")  
    List<Product> getAllProducts();  
}
```

Ініціалізація бази даних у додатку буде мати наступний вигляд:

```
AppDatabase database = Room.databaseBuilder(getApplicationContext(),  
AppDatabase.class, "my-database")  
    .build();
```

```
ProductDao productDao = database.productDao();
```

Далі використовуємо ці залежності у Main Activity:

```
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;
```

```
public class MainActivity extends AppCompatActivity {
```

						Аркуш
					ДТЕУ 121 06-02.БР	27
Зм.	Аркуш	№ докум	Підпис	Дата		

```

private EditText productNameEditText;
private EditText caloriesEditText;
private Button addButton;

private DatabaseHelper databaseHelper;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    productNameEditText = findViewById(R.id.productNameEditText);
    caloriesEditText = findViewById(R.id.caloriesEditText);
    addButton = findViewById(R.id.addButton);

    databaseHelper = new DatabaseHelper(this);

    addButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String productName =
productNameEditText.getText().toString();
            int calories =
Integer.parseInt(caloriesEditText.getText().toString());

            Product product = new Product(productName, calories);
            databaseHelper.addProduct(product);
        }
    });
}

```

					<i>ДТЕУ 121 06-02.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		28

```

    }
  });
}
}
public class Product {
    private String name;
    private int calories;

    public Product(String name, int calories) {
        this.name = name;
        this.calories = calories;
    }
}
}

```

3.4. Висновок до розділу 3

Отже, завдяки розробленим функціям інтерфейсу користувача, інтеграції бази даних SQLite та зручним механізмам обробки даних розроблено мобільний додаток, який забезпечує зручний та ефективний контроль споживання спортивного харчування для користувачів.

					<i>ДТЕУ 121 06-02.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		29

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Отже, у дипломній роботі було розроблено програмний додаток контролю споживання спортивного харчування, який дозволяє користувачам вести облік свого раціону харчування, визначати калорійність продуктів, стежити за вмістом поживних речовин та забезпечувати свій раціон оптимальними спортивними продуктами.

У першому розділі було проведено аналіз предметної області застосування програмного додатку, досліджено проблеми, пов'язані зі споживанням спортивного харчування, та визначено потреби користувачів. Виявлені проблеми та потреби послужили основою для подальшої розробки додатку.

У другому розділі було проведено вибір програмних засобів та проектування програмного додатку. Було визначено вимоги до функціональності додатку, обгрунтовано вибір мови програмування Java та зазначено основні компоненти системи.

У третьому розділі було реалізовано програмний додаток згідно з вимогами, які були визначені у попередніх розділах. Були розроблені класи для керування базою даних, розрахунку калорійності продуктів та управління раціоном користувача.

Після реалізації додатку були проведені тести, що підтвердили його працездатність та відповідність вимогам, поставленим у початковому етапі розробки

В результаті дипломної роботи було успішно розроблено програмний додаток контролю споживання спортивного харчування, який може бути використаний користувачами для планування та аналізу свого раціону.

					<i>ДТЕУ 121 06-02.БР</i>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.		28.04.23	Програмний додаток контролю споживання спортивного харчування	Стадія	Аркуш	Аркушів
Керівник		Гнатченко Д.Д.		28.04.23		ВП	30	32
Гарант		Рзаєва С.Л.		28.04.23		Факультет інформаційних технологій 4 курс, 6 група		
Розробив		Ауєр Д.С.		28.04.23				
					<i>Висновки та пропозиції</i>			

Додаток дозволяє зручно вести облік продуктів, визначати їх харчову цінність та надає корисні рекомендації щодо оптимального споживання спортивних продуктів.

На основі результатів роботи можна зробити висновок, що програмний додаток контролю споживання спортивного харчування є ефективним та корисним інструментом для підтримки здорового способу життя та досягнення спортивних цілей. Розроблений додаток може бути використаний як спортсменами та фітнес-ентузіастами, так і будь-якими людьми, які цікавляться збалансованим харчуванням та покращенням фізичної форми.

Пропозиції для подальшого розвитку даного додатку включають розширення бази даних продуктів, додавання нових функцій, таких як ведення тренувальних планів та моніторинг фізичної активності, а також покращення інтерфейсу та забезпечення підтримки для інших мобільних платформ.

У цілому, розробка програмного додатку контролю споживання спортивного харчування була успішно завершена, і результати роботи відповідають поставленим вимогам та сприяють поліпшенню здоров'я та фізичної форми користувачів.

					ДТЕУ 121 06-02.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		31

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Платформа Android [Електронний ресурс]. – Режим доступу:
https://www.android.com/intl/ru_ru/
2. Інформація про SQLite [Електронний ресурс]. – Режим доступу:
<https://uk.wikipedia.org/wiki/SQLite>
3. Технологія SQLite [Електронний ресурс]. – Режим доступу:
<https://www.sqlite.org/docs.html>
4. Приклад реалізації SQLite [Електронний ресурс]. – Режим доступу:
<https://github.com/requery/sqlite-android>



					<i>ДТЕУ 121 06-02.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
Зав. каф.		Криворучко О.В.		23.12.22	Програмний додаток контролю споживання спортивного харчування	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
Керівник		Гнатченко Д.Д.		23.12.22		<i>СВД</i>	32	32
Гарант		Рзаєва С.Л.		23.12.22		Факультет інформаційних технологій 4 курс, 6 група		
Розробив		Ауер Д.С.		23.12.22				
					<i>Список використаних джерел</i>			

ДОДАТКИ

ДОДАТОК А

Код класу MainActivity

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
    private EditText productNameEditText;
    private EditText caloriesEditText;
    private Button addButton;
    private DatabaseHelper databaseHelper;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        productNameEditText = findViewById(R.id.productNameEditText);
        caloriesEditText = findViewById(R.id.caloriesEditText);
        addButton = findViewById(R.id.addButton);
        databaseHelper = new DatabaseHelper(this);

        addButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String productName = productNameEditText.getText().toString();
                int calories = Integer.parseInt(caloriesEditText.getText().toString());
                Product product = new Product(productName, calories);
                databaseHelper.addProduct(product);
            }
        });
    }
}
```

Код класу Product

```
public class Product {  
    private String name;  
    private int calories;  
  
    public Product(String name, int calories) {  
        this.name = name;  
        this.calories = calories;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public int getCalories() {  
        return calories;  
    }  
  
    public void setCalories(int calories) {  
        this.calories = calories;  
    }  
}
```


Код класу DatabaseHelper

```

DatabaseHelper
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DatabaseHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "nutrition_db";
    private static final int DATABASE_VERSION = 1;
    private static final String TABLE_PRODUCTS = "products";
    private static final String COLUMN_ID = "_id";
    private static final String COLUMN_NAME = "name";
    private static final String COLUMN_CALORIES = "calories";

    public DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String createTableQuery = "CREATE TABLE IF NOT EXISTS " + TABLE_PRODUCTS +
            "(" + COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
            COLUMN_NAME + " TEXT, " +
            COLUMN_CALORIES + " INTEGER)";
        db.execSQL(createTableQuery);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_PRODUCTS);
        onCreate(db);
    }

    public void addProduct(Product product) {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(COLUMN_NAME, product.getName());
    }

```

```
values.put(COLUMN_CALORIES, product.getCalories());  
db.insert(TABLE_PRODUCTS, null, values);  
db.close();  
}  
  
public Cursor getAllProducts() {  
    SQLiteDatabase db = this.getReadableDatabase();  
    return db.rawQuery("SELECT * FROM " + TABLE_PRODUCTS, null);  
}  
}
```



Код XML розмітки Activity_main

XML

Activity_main

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Список продуктів"
        android:textSize="18sp"
        android:textStyle="bold"
        android:layout_marginBottom="16dp"/>
    <ListView
        android:id="@+id/listViewProducts"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1" />
    <Button
        android:id="@+id/btnAddProduct"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Додати продукт"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="16dp"
        android:paddingStart="16dp"
        android:paddingEnd="16dp"/>
```

```
</LinearLayout>
build.gradle (module app)

apply plugin: 'com.android.application'

android {
    compileSdkVersion 30
    buildToolsVersion "30.0.3"

    defaultConfig {
        applicationId "com.example.myapplication"
        minSdkVersion 21
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }

    dependencies {
        implementation 'androidx.appcompat:appcompat:1.3.0'
        implementation 'androidx.core:core-ktx:1.6.0'
        implementation 'androidx.constraintlayout:constraintlayout:2.1.0'
        implementation 'androidx.sqlite:sqlite:2.1.0' // Залежність для SQLite
        testImplementation 'junit:junit:4.13.2'
        androidTestImplementation 'androidx.test.ext:junit:1.1.3'
        androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
    }
}

build.gradle (module project)
// Top-level build file where you can add configuration options common to all sub-projects/modules.
buildscript {
    repositories {
```

```
google()
jcenter()
}

dependencies {
    classpath 'com.android.tools.build:gradle:4.2.2'
}

allprojects {
    repositories {
        google()
        jcenter()
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

