

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

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

Кафедра комп'ютерних наук  
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка персонального мобільного застосунку для профілактики  
цукрового діабету та захворювань щитоподібної залози

Виконала: студентка IV курсу, групи СТс-41

спеціальності 126 Інформаційні системи та

технології

(шифр і назва спеціальності)

Вовнянка Г. Р.

(підпис)

(прізвище та ініціали)

Керівник

(підпис)

Матійчук Л. П.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Марценко С. В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І. О.

(прізвище та ініціали)

Рецензент

(підпис)

Станько А.А.

(прізвище та ініціали)

Тернопіль  
2024

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

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

Кафедра комп'ютерних наук  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.  
(підпис) (прізвище та ініціали)

« 23 » червня 2024 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр  
(назва освітнього ступеня)

за спеціальністю 126 Інформаційні системи та технології  
(шифр і назва спеціальності)

Студентці Вовнянка Галині Романівні  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози.

Керівник роботи Матійчук Любомир Павлович, доктор економічних наук, проф. кафедри КН  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 29 » квітня 2024 року № 4/7-471

2. Термін подання студентом завершеної роботи 24 червня 2024 р.

3. Вихідні дані до роботи Перелік літературних та інтернет джерел

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

Вступ. 1 Мобільні застосунки, основні поняття та процес розробки. 1.1 Предметна область розробки мобільних застосунків. 1.2 Класифікація мобільних застосунків. 1.3 ОС Android.

2 Засоби для розробки мобільних застосунків. 2.1 Android Studio. 2.2 Мова програмування Kotlin. 2.3 Особливості мови розмітки XML. 3 Практична реалізація персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози.

3.1 Створення реєстраційного меню персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози. 3.2 Створення основної сторінки персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози. 3.3 Створення вікна з інформацією про показники цукрового діабету та захворювань щитоподібної залози. 3.4 Створення сторінки з додаванням даних щодо цукрового діабету та захворювань щитоподібної залози. 3.5 Створення сторінки з корисною інформацією щодо профілактики цукрового діабету та захворювань щитоподібної залози.

4. Безпека життєдіяльності, основи охорони праці. Висновки. Перелік джерел.

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

1. Титульна сторінка. 2. Мета кваліфікаційної роботи. 3. Мобільні застосунки. 4. Технології, які були використані. 5. Перше вікно реєстрації. 6. Друге вікно інформаційної сторінки.

7. Третє вікно інформаційної сторінки. 8. Створення бази даних SQLite. 9. Створення кнопки та функції для додавання нових аналізів. 10. Картка з корисною інформацією. 11. Висновки.

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Сенчишин В. С., доцент кафедри МТ		

7. Дата видачі завдання 29 січня 2024 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	30.01.2024	
2.	Підбір джерел про мобільні застосунки для медичного застосунку	31.01.2024-03.02.2024	
3.	Опрацювання джерел по темі кваліфікаційної роботи про мобільні застосунки для медичного застосунку	04.02.2024-06.02.2024	
4.	Виконання дослідження щодо мобільних застосунків. Розроблення персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози	07.02.2024-11.02.2024	
5.	Оформлення розділу «Мобільні застосунки, основні поняття та процес розробки»	03.06.2024-05.06.2024	
6.	Оформлення розділу «Засоби для розробки мобільних застосунків»	06.06.2024-08.06.2024	
6.	Оформлення розділу «Практична реалізація персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози»	09.06.2024-11.06.2024	
7.	Виконання завдання до підрозділу «Безпека життєдіяльності»	12.06.2024-13.06.2024	
8.	Виконання завдання до підрозділу «Основи охорони праці»	14.06.2024-15.06.2024	
9.	Оформлення кваліфікаційної роботи	16.06.2024-17.06.2024	
10.	Нормоконтроль	18.06.2024-19.06.2024	
11.	Перевірка на плагіат	20.06.2024	
12.	Попередній захист кваліфікаційної роботи	21.06.2024	
13.	Захист кваліфікаційної роботи	24.06.2024	

Студентка

\_\_\_\_\_  
(підпис)

Вовнянка Г. Р.

\_\_\_\_\_  
(прізвище та ініціали)

Керівник роботи

\_\_\_\_\_  
(підпис)

Матійчук Л. П.

\_\_\_\_\_  
(прізвище та ініціали)

## АНОТАЦІЯ

Розробка персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози // Кваліфікаційна робота освітнього рівня «Бакалавр» // Вовнянка Галина Романівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СТс-41 // Тернопіль, 2024 // С. 85, рис. – 11, табл. – 1, кресл. – 12, додат. – 17, бібліогр. – 38.

**Ключові слова:** розробка, застосунок, програмування, база даних, профілактика, захворювання, моніторинг.

Кваліфікаційна робота присвячена дослідженню розробки персонального мобільного застосунку з метою профілактики цукрового діабету та захворювань щитоподібної залози.

У першому розділі кваліфікаційної роботи описано основні поняття мобільного застосунку, різновиди таких застосунків та операційну систему Android.

В другому розділі розглянуті засоби для розробки мобільних застосунків, зокрема Android Studio, мова програмування Kotlin, мова розмітки XML та реляційна система керування базами даних SQLite.

Третій розділ присвячено практичній реалізації персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози, де описано створення реєстраційного меню, основної сторінки, вікна з інформацією про показники, сторінки для додавання даних та сторінки з корисною інформацією.

Об'єктом дослідження є розроблений персональний мобільний застосунок, а предметом – його функціональні можливості для профілактики зазначених захворювань.

## ANNOTATION

Development of a personal mobile application for the prevention of diabetes and thyroid diseases // Qualification work of the educational level "Bachelor" // Vovnianka Halyna Romanivna // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group STs-41 // Ternopil, 2024 // P. 85, fig. – 11, tabl. – 1, chair. – 12, annexes – 17, references – 38.

**Keywords:** development, application, programming, database, prevention, disease, monitoring.

The qualification paper is devoted to the research of developing a personal mobile application aimed at the prevention of diabetes and thyroid diseases.

The first chapter describes the basic concepts of mobile applications, types of such applications, and the Android operating system.

The second chapter covers the tools for developing mobile applications, including Android Studio, the Kotlin programming language, XML markup language, and the SQLite relational database management system.

The third chapter is dedicated to the practical implementation of the project, describing the creation of a registration menu, main page, window with information about indicators, pages for adding data, and pages with useful information.

The subject of the research is the developed personal mobile application, and the object is its functional capabilities for the prevention of the mentioned diseases.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

3D-ігри – тривимірні ігри: ігри, які мають тривимірний графічний движок, що дозволяє зробити ігровий процес більш реалістичним і захоплюючим.

Android – мобільна операційна система Google: використовується на різноманітних смартфонах та планшетах різних виробників.

Google Play – інтернет-магазин для завантаження та придбання застосунків, ігор, фільмів, музики та книг для пристроїв Android.

Id – ім'я об'єкта програми, що дозволяє звернутись до об'єкта; ознака, яка цілком визначає сутність.

IDE – інтегроване середовище розробки: програмне забезпечення, яке надає програмістам комплексні засоби для розробки програмного забезпечення.

iOS – мобільна операційна система від Apple: використовується на пристроях iPhone, iPad та iPod Touch.

Linux – безкоштовна операційна система з відкритим вихідним кодом, заснована на операційній системі Unix, широко використовується на серверах і як платформа для вбудованих систем.

OHA (Open Handset Alliance) – консорціум виробників мобільного обладнання, програмного забезпечення та телекомунікаційних компаній, які працюють разом для просування відкритих стандартів для мобільних пристроїв.

PWA – Progressive Web Apps: вебзастосунки, які можуть бути встановлені на пристрої користувача і надають користувачеві нативний досвід роботи з ними.

Swift – мова програмування, розроблена компанією Apple для створення застосунків для macOS, iOS, watchOS і tvOS.

UI – User Interface – інтерфейс користувача: візуальна частина комп'ютерної програми або операційної системи, з якою взаємодіє користувач.

WebView – компонент в Android та iOS, який дозволяє розробникам відображати вебвміст у застосунку.

## ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1 МОБІЛЬНІ ЗАСТОСУНКИ, ОСНОВНІ ПОНЯТТЯ ТА ПРОЦЕС РОЗРОБКИ.....	11
1.1 Предметна область розробки мобільних застосунків.....	11
1.2 Класифікація мобільних застосунків .....	14
1.3 ОС Android .....	16
1.4 Висновок до першого розділу .....	19
РОЗДІЛ 2 ЗАСОБИ ДЛЯ РОЗРОБКИ МОБІЛЬНИХ ЗАСТОСУНКІВ .....	20
2.1 Android Studio .....	20
2.2 Мова програмування Kotlin.....	22
2.3 Особливості мови розмітки XML .....	25
2.4 Реляційна система керування базами даних SQLite.....	27
2.5 Висновок до другого розділу .....	30
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПЕРСОНАЛЬНОГО МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ПРОФІЛАКТИКИ ЦУКРОВОГО ДІАБЕТУ ТА ЗАХВОРЮВАНЬ ЩИТОПОДІБНОЇ ЗАЛОЗИ .....	31
3.1 Створення реєстраційного меню персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози.....	31
3.2 Створення основної сторінки персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози.....	45
3.3 Створення вікна з інформацією про показники цукрового діабету та захворювань щитоподібної залози .....	48
3.4 Створення сторінки з додаванням даних щодо цукрового діабету та захворювань щитоподібної залози .....	51
3.5 Створення сторінки з корисною інформацією щодо профілактики цукрового діабету та захворювань щитоподібної залози.....	54
3.6 Висновок до третього розділу.....	56
РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ .....	57

4.1 Шляхи підвищення якості життя та профілактики захворювань.....	57
4.2 Загальні вимоги безпеки з охорони праці для користувачів мобільних застосунків .....	58
4.3 Оцінка ефективності мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози .....	59
4.4 Висновок до четвертого розділу .....	60
ВИСНОВКИ.....	61
ПЕРЕЛІК ДЖЕРЕЛ .....	62
ДОДАТКИ	



## ВСТУП

**Актуальність теми.** Сьогодні мобільні застосунки є невід'ємною складовою сучасного світу, який стає все більш залежним від технологій. Мобільні застосунки є зручним та простим інструментом, який дозволяє користувачам отримувати швидкий та простий доступ до різних сервісів та інформації [36].

Важливість мобільних застосунків у сучасному світі неможливо переоцінити. Такі програми дозволяють виконувати різні завдання в будь-який час та в будь-якому місці, що зробило їх популярними у всьому світі. Мобільні застосунки також дозволяють підвищувати ефективність бізнес-процесів та рівень задоволеності клієнтів. Але й звичайні користувачі смартфонів отримують максимум можливостей та функцій у своєму телефоні, завдяки завантаженим та встановленим застосункам. Програми для навчання мов, наукових дисциплін та навчальних курсів надають доступ до цінної інформації та навчальних ресурсів.

Мобільні застосунки також розширюють можливості для звичайних користувачів і спрощують їх щоденні дії. Програми для здоров'я і фітнесу дозволяють відстежувати фізичну активність, контролювати харчування і отримувати поради щодо здорового способу життя.

**Мета і задачі дослідження.** Метою даної кваліфікаційної роботи освітнього рівня "Бакалавр" є розробка персонального мобільного застосунку, спрямованого на профілактику цукрового діабету та захворювань щитоподібної залози. Для досягнення цієї мети необхідно вирішити такі завдання:

- Проаналізувати стан досліджень у галузі профілактики та контролю цукрового діабету та захворювань щитоподібної залози.
- Розробити та реалізувати персональний мобільний застосунок, який забезпечить користувачам зручний та ефективний інструмент для моніторингу їх стану здоров'я та профілактики виникнення цих захворювань.

- Оцінити практичне значення отриманих результатів у плані їхнього впливу на підвищення якості медичного обслуговування та поліпшення здоров'я користувачів мобільного застосунку.

Розробка персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози є важливим кроком у покращенні якості життя та збереженні здоров'я сучасного суспільства.

## РОЗДІЛ 1 МОБІЛЬНІ ЗАСТОСУНКИ, ОСНОВНІ ПОНЯТТЯ ТА ПРОЦЕС РОЗРОБКИ

### 1.1 Предметна область розробки мобільних застосунків

Мобільні застосунки – це програми, призначені для використання на мобільних пристроях, таких як телефони, планшети або годинники [7]. Це відрізняється від десктопних програм, які працюють на стаціонарних комп'ютерах, та веб-застосунків, які запускаються у мобільних браузерах, а не на самому пристрої. На рисунку 1.1 наведено приклад мобільних застосунків.



Рисунок 1.1 – Мобільні застосунки

Спочатку такі застосунки створювалися для поліпшення продуктивності, наприклад, для управління електронною поштою, календарем і контактами, але з часом їх почали використовувати в різних галузях, таких як ігри, автоматизація виробництва, GPS-служби, відстеження замовлень і покупок. Тепер існує величезна кількість різноманітних застосунків, багато з яких потребують підключення до Інтернету [10].

Мобільні програми почали використовувати на початку 1990-х років, коли на телефонах Nokia з'явилися такі базові мобільні ігри, як «Змійка». Однак справжня еволюція мобільних застосунків почалася з випуском iPhone у 2007 році та подальшим відкриттям App Store у 2008 році. Це дозволило стороннім розробникам створювати та поширювати програми, що призвело до вибуху різноманітності та кількості доступних програм. Операційна система Android, розроблена Google, наслідувала цей приклад із власним ринком застосунків Google Play Store. На рисунку 1.2 наведено інтерфейс Google Play Store.

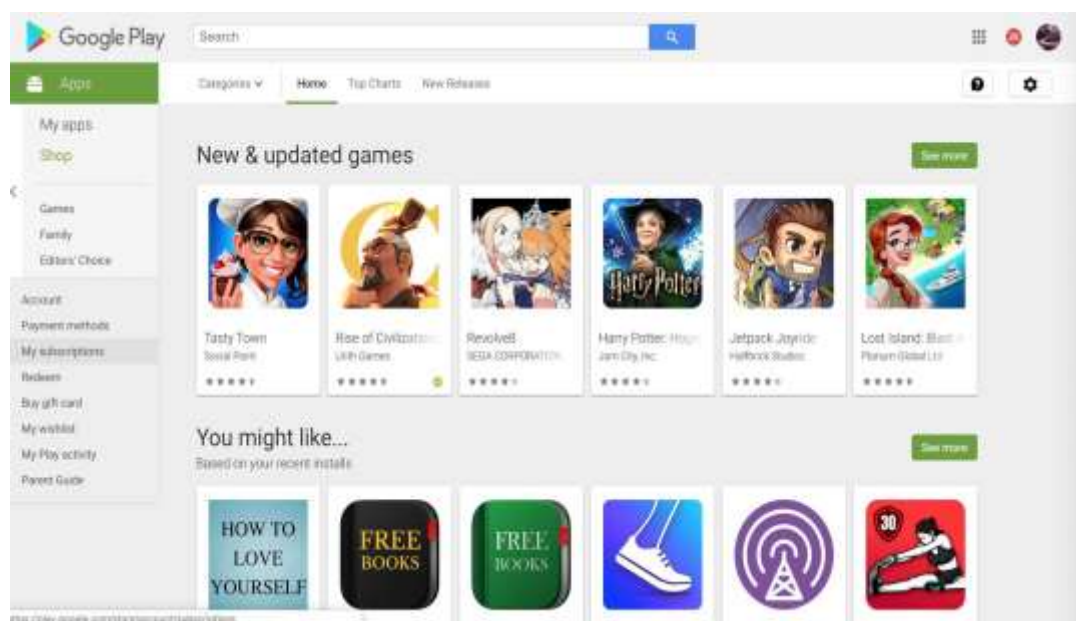


Рисунок 1.2 – Google Play Store

Впродовж тривалого періоду часу мобільні програми еволюціонували від простих утиліт та ігор до складних програм, які можуть виконувати завдання від соціальних мереж до банківських операцій, моніторингу здоров'я тощо [18].

Мобільні застосунки працюють за допомогою апаратного та програмного забезпечення пристрою для ефективного виконання завдань. Операційні системи, такі як iOS від Apple або Android від Google, забезпечують стандартизоване середовище з такими службами, як керування пам'яттю та безпека. Розробники створюють програми, використовуючи певні мови програмування та інструменти, надані цими системами.

Розробка мобільних застосунків – це процес створення програмних застосунків для мобільних пристроїв. Він включає в себе кілька етапів, як зазначено вище.

Різні методології керують цим процесом. Гнучка розробка використовує ітераційні спринти для гнучкості та постійного вдосконалення. Метод водоспаду є лінійним, завершуючи кожен етап перед переходом до наступної. Ключові ролі включають розробників, які пишуть код, дизайнерів, які створюють макет програми, тестувальників, які гарантують відсутність помилок у програмі, і менеджерів проєктів, які контролюють процес.

Мобільні програми взаємодіють із апаратним забезпеченням пристрою (камера, GPS, датчики) і програмним забезпеченням (ОС, інші програми), щоб забезпечити функціональність. Наприклад, навігаційна програма використовує GPS для надання послуг визначення місцезнаходження, а програма соціальних мереж може використовувати камеру для фотографування.

ОС керує цими взаємодіями, гарантуючи, що застосунки мають необхідні дозволи та ресурси, зберігаючи продуктивність і безпеку пристрою [37]. Більшість мобільних пристроїв поставляються з попередньо встановленими застосунками, такими як браузер, поштовий клієнт, календар і картографічна програма. Інші можна завантажити з магазинів застосунків, таких як App Store або Google Play Store, або встановити вручну.

Деякі застосунки безкоштовні, інші мають ціну або пропонують мікротранзакції та рекламу. Прибуток зазвичай ділиться між розробником і магазином застосунків. Також варіюється ціна застосунка в залежності від мобільної платформи.

Спочатку мобільні застосунки були спрямовані на підвищення продуктивності та отримання інформації, такої як пошта, календар, контакти та погодні умови. Проте з часом вони почали використовуватися у багатьох інших галузях. Широкий вибір застосунків ускладнює їхнє виявлення, тому з'явилися різні джерела рекомендацій і оглядів [23].

## 1.2 Класифікація мобільних застосунків

З урахуванням технологічного розвитку, мобільні застосунки можна класифікувати за трьома основними категоріями: нативні, гібридні та кросплатформенні (див. табл. 1.1). При цьому до категорії мобільних застосунків можна віднести також інтернет-застосунки PWA [17].

Таблиця 1.1 – Основні характеристики нативних, гібридних та кросплатформних мобільних застосунків

Характеристика	Нативні застосунки	Гібридні застосунки	Кросплатформенні застосунки
Мова програмування	Мова, специфічна для платформи (наприклад, Kotlin або Java для Android)	HTML, CSS, JavaScript	Фреймворки (наприклад, React Native, Xamarin) або Web-технології
Доступ до пристроїв	Повний доступ до функцій пристрою, таких як камера, геолокація, датчики тощо	Обмежений доступ, зазвичай за допомогою плагінів	Обмежений доступ, але покращується з кожним оновленням
Розробка	Розробка для кожної платформи окремо	Єдиною кодовою базою для кількох платформ	Єдиною кодовою базою для кількох платформ, але потребує тестування на кожній платформі
Швидкість розробки	Потрібен окремих час для розробки для кожної платформи	Швидша розробка за рахунок єдиної кодової бази	Швидша розробка, але потребує додаткового тестування на кожній платформі
Вартість розробки	Дорожча, оскільки потрібно окремо оплачувати розробку для кожної платформи	Зазвичай дешевший, оскільки потрібен менший обсяг роботи	Зазвичай дешевший, оскільки використовує єдину кодову базу
Підтримка та оновлення	Потрібна окрема підтримка та оновлення для кожної платформи	Можлива єдина підтримка та оновлення для всіх платформ	Можлива єдина підтримка та оновлення для всіх платформ, але може бути неоднаковою

Переваги нативних застосунків включають швидкість та відмінну реакцію інтерфейсу, високу продуктивність у складних обчисленнях, можливість створення унікального користувацького досвіду та інтерфейсу, ефективну взаємодію з пристроями та надійність. Нативні застосунки забезпечують високу якість користувацького досвіду, що позитивно впливає на різні бізнес-метрики [31].

Недоліки нативної розробки включають високі витрати на розробку для обох платформ, потребу у висококваліфікованих розробників, постійну підтримку для відповідності вимогам нових версій операційних систем, обмежену доступність через пошукові системи та комісії магазинів застосунків.

Гібридні застосунки розробляються для обох платформ за допомогою веб-технологій [38], таких як HTML, CSS і JavaScript, і використовуються в якості застосунку за допомогою WebView. Їх переваги включають швидкість розробки та економію, а недоліки – можливість проблем з інтерфейсом, низьку інтеграцію з ОС та складності з підгонкою під гайдлайни магазинів застосунків [32].

Кросплатформенні застосунки, такі як React Native, Flutter та Xamarin, розробляються для обох платформ за допомогою повноцінних мов програмування і забезпечують близьку до нативних застосунків продуктивність.

Їх переваги полягають у збереженні часу та коштів завдяки використанню однієї кодової бази, проте вони можуть стикатися з тестуванням та іншими проблемами.

Прогресивні веб-застосунки (Progressive Web Apps, PWA) представляють новий підхід до створення мобільних програм, який об'єднує переваги веб-сайтів та можливостей звичайних мобільних застосунків. PWA функціонують у середовищі веб-браузера, але можуть бути встановлені на домашній екран смартфона. Основна мета PWA полягає в тому, щоб забезпечити користувачам веб-застосунки, що наближені до звичайних мобільних застосунків. Вони використовують передові веб-технології, такі як Service Workers, Web App Manifest та Cache API, для забезпечення офлайн-режиму, швидкого

завантаження сторінок, пуш-сповіщень, доступу до апаратних можливостей пристрою і багато іншого.

Однією з ключових переваг PWA є можливість використання без встановлення з магазину застосунків, що зручно для користувачів та економить їхній час. Крім того, PWA є платформонезалежними, тобто вони можуть працювати на будь-якому пристрої з підтримкою веб-браузера, незалежно від операційної системи. Ще однією важливою перевагою PWA є можливість автоматичного оновлення без необхідності встановлення оновлень з магазину застосунків. Це дозволяє розробникам швидко виправляти помилки та впроваджувати нові функції, а користувачам – завжди отримувати останню версію застосунка.

Загалом, мобільні вебзастосунки PWA набувають все більшої популярності завдяки зручності, швидкості, доступності та низькій залежності від певної платформи [33].

### 1.3 ОС Android

ОС Android є відкритою операційною системою на основі ядра Linux, розробленою спеціально для мобільних пристроїв. Логотип операційної системи зображений на рисунку 1.3.



Рисунок 1.3 – ОС Android



Впродовж останнього періоду часу Android переживала величезне зростання, стрімко ставши найпопулярнішою мобільною операційною системою у світі. За даними Statcounter, станом на січень 2022 року Android займає понад 72% світової частки ринку мобільних операційних систем, значно перевершуючи своїх конкурентів.

Деякі особи відзначають, що Android виявляється більш привабливим порівняно зі своїм конкурентом, Apple iOS, у певних аспектах, таких як перегляд вебсторінок та інтеграція з сервісами Google та іншими. Додатково, Android, на відміну від iOS, є відкритою платформою, що дає змогу реалізувати більше функцій [2].

Хоча спочатку існувала заборона на встановлення програм з "неперевіраних джерел", це обмеження можна легко вимкнути через стандартні налаштування пристрою, що дозволяє встановлювати програми без доступу до Інтернету. Android також дозволяє будь-кому безкоштовно розробляти та тестувати програми на своїх пристроях.

Платформа Android доступна для різних апаратних платформ, таких як ARM, MIPS, x86. Окрім офіційного Google Play, існують альтернативні магазини застосунків, такі як Amazon Appstore, Opera Mobile Store, GetUpps! та F-Droid. У версії 4.3 з'явилася підтримка багатокористувацького режиму [21].

Android, будучи універсальною операційною системою, знаходить застосування на різних пристроях:

1. Смарт-телевізори: Android підтримує смарт-телевізори, дозволяючи користувачам транслювати вміст із різних джерел, отримувати доступ до застосунків, переглядати вебсторінки та керувати переглядом телепередач за допомогою голосових команд або пультів дистанційного керування.

2. Побутова техніка: Android інтегрується в різні побутові прилади, такі як холодильники, пральні машини та духовки. Ця інтеграція дозволяє користувачам отримувати доступ до розумних функцій, дистанційно керувати налаштуваннями та отримувати сповіщення про стан своїх приладів.

3. Пристрої для охорони здоров'я: Android використовується в різних пристроях для охорони здоров'я, зокрема в портативних фітнес-трекерах, розумних годинниках і пристроях для медичного моніторингу. Ці пристрої можуть відстежувати показники здоров'я користувачів, надавати зворотній зв'язок у реальному часі та синхронізувати дані з мобільними застосунками для аналізу.

4. Пристрої IoT (Інтернет речей): Android служить платформою для пристроїв IoT, таких як розумні термостати, камери безпеки та підключені системи освітлення. Android дозволяє цим пристроям спілкуватися один з одним, а також смартфонам або планшетам, надаючи користувачам можливість дистанційного керування та моніторингу.

5. Автомобільні інформаційно-розважальні системи: Android Auto – це версія Android, призначена для використання в автомобілях. Він надає водіям доступ до навігації, музики, обміну повідомленнями та інших програм через інформаційно-розважальну систему автомобіля, мінімізуючи відволікання та сприяючи безпечній водійській практиці.

6. Цифрові вивіски та кіоски: Android використовується в рішеннях для цифрових вивісок та інтерактивних кіосках, розгорнутих у роздрібних магазинах, аеропортах, ресторанах та інших громадських місцях. Це забезпечує легке керування вмістом, дистанційне оновлення та інтерактивний досвід користувача.

7. Промислові та корпоративні програми: Android все частіше використовується в промислових і корпоративних середовищах для таких завдань, як керування запасами, відстеження матеріально-технічного забезпечення та обслуговування на місцях. Індивідуальні рішення для Android розроблено відповідно до конкретних бізнес-потреб і вимог.

Існує інформація, що Google може видаляти застосунки з телефонів користувачів у разі порушення умов використання. Конкуренти Android критикують платформу за її фрагментацію, проте Google запевняє, що подібні проблеми не існують [28].

Основні характеристики Android включають:

1. Відкритий код: Android є відкритою платформою, що дозволяє розробникам створювати та модифікувати програми за своїм бажанням.
2. Широкий асортимент застосунків: через Google Play Store користувачі мають доступ до мільйонів програм, які задовольняють різноманітні потреби, від розважальних ігор до корисних робочих інструментів.
3. Мультизадачність: Android дозволяє виконувати декілька програм одночасно.
4. Інтеграція з Google: Android тісно пов'язаний з різними сервісами Google, такими як пошук, Gmail, Google Maps, YouTube та інші.
5. Персоналізація: користувачі можуть налаштувати вигляд і функції своєї системи за допомогою тем, шпалер, віджетів та інших параметрів.
6. Підтримка різноманітного обладнання: Android працює на різних пристроях від різних виробників, таких як Samsung, LG, Sony, Huawei, Xiaomi та інші.

#### **1.4 Висновок до першого розділу**

В першому розділі кваліфікаційної роботи розглянуто основні поняття, пов'язані з розробкою мобільних застосунків, включаючи визначення основного поняття мобільного застосунку, типи таких застосунків та характеристики операційної системи Android. Цей аналіз є важливим етапом у розробці персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози, оскільки він надає базові знання та розуміння основних принципів роботи мобільних застосунків і їхніх характеристик.

Дана інформація буде використана в подальших розділах для конкретизації завдань розробки та функціональних можливостей запропонованого застосунку.

## РОЗДІЛ 2 ЗАСОБИ ДЛЯ РОЗРОБКИ МОБІЛЬНИХ ЗАСТОСУНКІВ

### 2.1 Android Studio

Android Studio є інтегрованим середовищем розробки для створення застосунків в операційній системі Android. На рисунку 2.1 продемонстровано програмне середовище Android Studio.

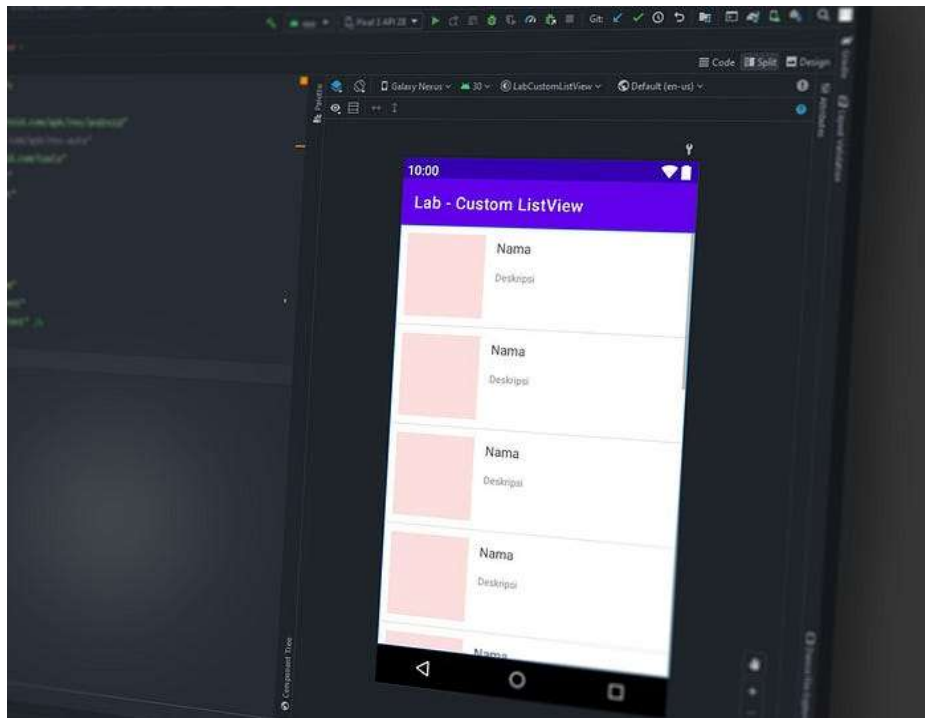


Рисунок 2.1 – Android Studio

Щоб налаштувати Android Studio, необхідно завантажити IDE з офіційної сторінки завантаження Android Studio. Спочатку треба вибрати версію, сумісну з вашою операційною системою, і завантажити програму встановлення. Android Studio доступний для Windows, macOS і Linux. Після завершення завантаження запустити програму встановлення та дотримуватись вказівок, щоб установити Android Studio на комп'ютер [3].

У процесі інсталяції Android Studio запропонує встановити необхідні компоненти. До них належать Android SDK, Android Virtual Device (AVD) Manager і Android Emulator. Android SDK – це набір бібліотек і інструментів, які

розробники використовують для створення програм Android. AVD Manager використовується для створення та керування віртуальними пристроями для тестування програм. Емулятор Android – це віртуальний пристрій, який дозволяє розробникам тестувати свої програми без використання фізичного пристрою.

Після встановлення Android Studio користувачу потрібно налаштувати програму, перш ніж почати використовувати. Спочатку треба вибрати «Стандартну» конфігурацію та натиснути «Далі». На наступному екрані можна вибрати тему IDE та знову натиснути «Далі». Також можна налаштувати параметри відповідно до своїх уподобань [5].

Після налаштування Android Studio можна почати створювати першу програму для Android. Щоб створити новий проект, треба натиснути «Почати новий проект Android Studio» на екрані привітання або вибрати «Новий проект» у меню «Іле». Користувачу буде запропоновано вибрати назву проекту, назву пакета та інші деталі проекту [6].

Після того, як проект створено, можна почати створювати програму за допомогою різних інструментів і функцій, які надає Android Studio. Можна використовувати редактор візуального макета, щоб створити інтерфейс користувача, написати код на Java або Kotlin і використовувати Android SDK для доступу до таких функцій пристрою, як камера, датчики та GPS. Користувач також може використовувати вбудовані засоби налагодження, щоб усунути проблеми та оптимізувати свою програму [20].

Тестування програми є важливим кроком у процесі розробки. Android Studio поставляється з емулятором, який дозволяє тестувати програму на різних віртуальних пристроях. Користувач також може підключити пристрій Android до комп'ютера та протестувати програму безпосередньо на пристрої. Щоб запустити та протестувати програму на емуляторі або пристрої, треба використовувати кнопку «Виконати» в Android Studio. Можна використовувати вбудований профайлер для аналізу продуктивності програми та виявлення будь-яких вузьких місць або проблем із продуктивністю. Налаштування Android Studio є вирішальним кроком у розробці програм для Android. Виконуючи ці

кроки, можна легко налаштувати Android Studio на комп'ютері та почати створювати високоякісні програми для Android [34].

## 2.2 Мова програмування Kotlin

Kotlin є мовою програмування високого рівня, кросплатформеною та статично типізованою, яка має виведення типів. Вона розроблена для тісної взаємодії з Java, залежність JVM-версії стандартної бібліотеки Kotlin від бібліотеки класів Java. Однак виведення типів дозволяє його синтаксису бути більш компактним [8].

Kotlin в основному спрямований на використання на JVM, але також може бути скомпільований в JavaScript (наприклад, для вебзастосунків з використанням React) або нативний код через LLVM, для наприклад, нативних iOS-застосунків, що обмінюються бізнес-логікою з Android-застосунками [11].

Котирування мови Kotlin покривається компанією JetBrains, і мова підтримується Kotlin Foundation. Компілятор Android Kotlin за замовчуванням генерує байт-код Java 8, але програміст може вибрати цільову версію Java 9 до 18 для оптимізації або отримання застосункових можливостей. На рисунку 2.2 зображено приклад мови програмування Kotlin.

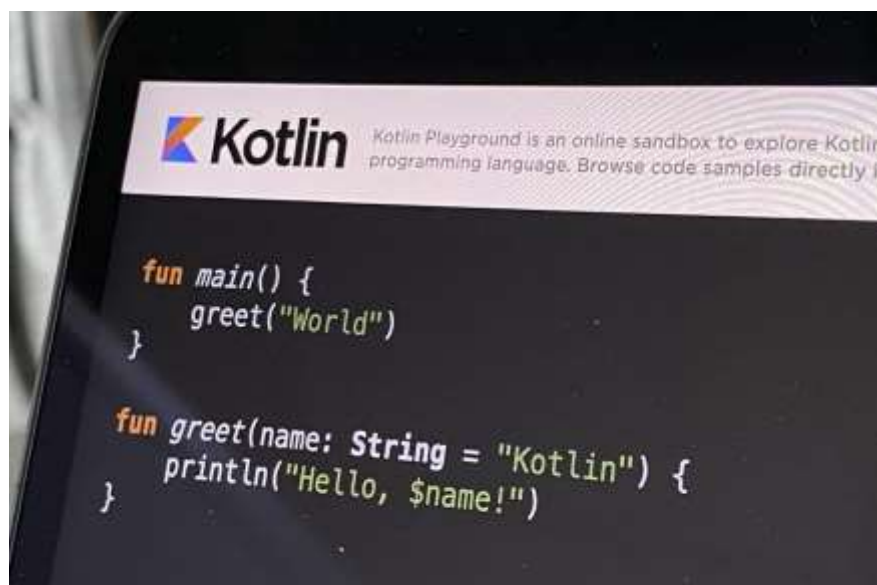


Рисунок 2.2 – Мова програмування Kotlin

Kotlin має підтримку мережі, оскільки може бути скомпільований в JavaScript (наприклад, Kotlin/JS з класичним бекендом, стабільний з версії 1.3), а новіша версія Kotlin/JS (на основі IR) перебуває в бета-версії з версії 1.5.30. Kotlin/Native (наприклад, для підтримки пристроїв Apple) також є бета-версією з версії 1.3 [12].

У липні 2011 року компанія JetBrains представила Project Kotlin – нову мову для JVM, над створенням якої працювали протягом року. За словами керівника JetBrains Дмитра Ємерова, більшість існуючих мов не відповідали їхнім вимогам, за винятком Scala. Проте, він вказав на повільний час компіляції Scala як недолік. Однією з основних цілей Kotlin було зробити компіляцію так само швидкою, як у Java. У лютому 2012 року JetBrains оприлюднила проєкт під відкритою ліцензією Apache 2 [13].

Нижче наведено основні особливості мови Kotlin:

- Стислий. Kotlin значно скорочує додатковий код, написаний на інших об'єктно-орієнтованих мовах програмування.
- Статично типізований. Статична типізація означає, що кожна змінна та вираз відомі під час компіляції.
- Безпечний. Ця мова програмування підтримує нульові значення як частину своєї системи, забезпечуючи захист від найбільш дратівливих `NullPointerException`.
- Класи даних. Kotlin має класи даних, які призводять до автоматичної генерації шаблонів, таких як `hashCode`, `equals`, `toString`, `getters/setter` тощо.
- Smart Cast. Kotlin приводить незмінні значення та автоматично додає значення до безпечного приведення.
- Взаємодія з Java. Kotlin і Java можуть взаємодіяти безпроблемно, оскільки Kotlin вбудований у Java і підтримує виклики Java-коду без необхідності використання будь-яких спеціальних обгортки або перекладачів.
- Час компіляції. Ця мова розробки Android має кращу продуктивність і швидший час компіляції.

- Функціональні та об'єктно-орієнтовані можливості. Kotlin пропонує багатий набір корисних методів, включаючи перевантаження операторів, функції вищого порядку, відкладене обчислення, лямбда-вирази тощо. Функції вищого порядку стосуються функцій, які можуть приймати функцію як параметр, повертати функцію або робити те й інше.

- Функція розширення. Kotlin підтримує як функції розширення, так і властивості розширення.

Перший коміт в репозиторії Kotlin Git було здійснено 8 листопада 2010 року [14].

Kotlin 1.0 було випущено 15 лютого 2016 року. Це став першим офіційно стабільним релізом, і JetBrains зобов'язалася забезпечити довгострокову зворотно сумісність починаючи з цієї версії.

На конференції Google I/O 2017 компанія Google оголосила про першокласну підтримку Kotlin на Android [15].

Kotlin 1.2 було випущено 28 листопада 2017 року. У цій версії було внесено функцію спільного використання коду між платформами JVM та JavaScript (починаючи з версії 1.4 це була альфа-версія багатоплатформного програмування, яка була покращена з "експериментальної"). За допомогою нового плагіна Kotlin/JS Gradle було створено демонстраційну версію повного стеку [22].

Kotlin 1.3 було випущено 29 жовтня 2018 року, в ньому були додані підпрограми для асинхронного програмування.

Kotlin 1.4 було випущено в серпні 2020 року, в ньому були внесені деякі незначні зміни в підтримку платформи Apple, зокрема взаємодію з Objective-C/Swift.

Kotlin 1.5 вийшов у травні 2021 року. Kotlin 1.6 був випущений у листопаді 2021 року. Kotlin 1.7 вийшов у червні 2022 року, включаючи альфа-версію нового компілятора Kotlin K2 [35].



## 2.3 Особливості мови розмітки XML

Мова розмітки XML (розширювана мова розмітки) є стандартом, запропонованим консорціумом World Wide Web Consortium (W3C), для створення ієрархічно структурованих даних з метою обміну між різними застосунками, включаючи використання через Інтернет. Ця мова є спрощеною версією мови розмітки SGML і складається з текстових знаків, що є придатними для читання людиною [1].

Мова розмітки з правилами для визначення будь-яких даних називається Extensible Markup Language (XML). На відміну від інших мов програмування, XML не здатний самостійно виконувати обчислення. Натомість керування структурованими даними можна реалізувати за допомогою будь-якої комп'ютерної мови чи частини програмного забезпечення. На рисунку 2.3 зображено схему обробки XML-документа.

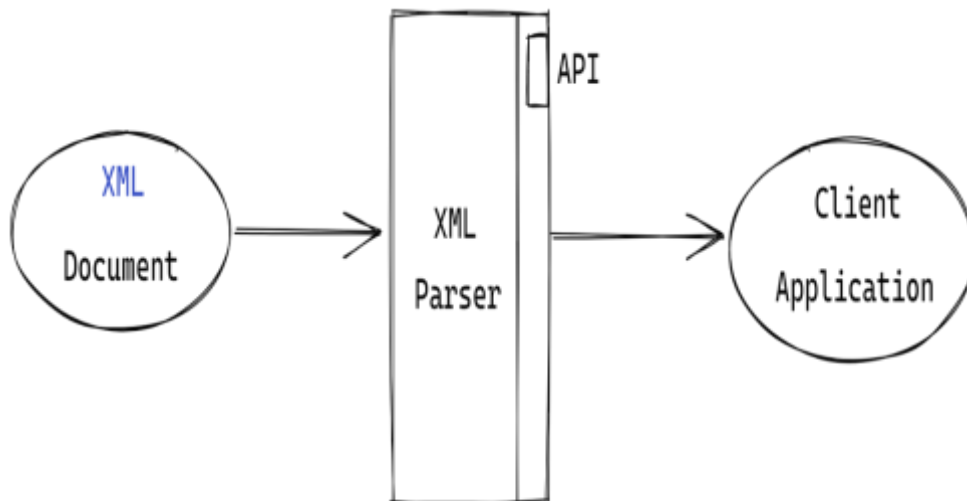


Рисунок 2.3 – Схема обробки XML-документа

Коректність XML-документа полягає в тому, що він відповідає всім синтаксичним правилам XML. Наприклад, коректний XML-документ має лише один елемент у корені, елементи мають бути розмічені початковим та кінцевим

тегами, атрибути не повинні дублюватися, теги можуть вкладатися один в одного і т.д [9].

Валідність документа означає, що він є коректним та повністю відповідає граматичним правилам, визначеним у схемі.

Синтаксичний аналізатор, або синтаксичний аналізатор, використовується для читання XML-документа, проведення синтаксичного аналізу та відтворення його структури. Наприклад, синтаксичний аналізатор може бути валідатором, який перевіряє документ на валідність [19].

Назви елементів у XML чутливі до регістру літер. Важливо вибрати змістовні назви для елементів, що підкреслюють значення даних у створеній мові розмітки [25].

Нижче наведено список основних функцій XML:

1. Розширюваний і зрозумілий людині. Навіть якщо додати додаткові дані, більшість XML-застосунків продовжуватимуть працювати за призначенням.

2. Загальна простота. Доступність даних, модифікація платформи, передача даних і обмін – все це стає простіше завдяки XML. Не втрачаючи даних, XML спрощує оновлення або розширення для нових операційних систем, програм або браузерів. Доступ до даних може отримати широкий спектр «пристроїв для читання», включаючи людей, комп'ютери, голосові помічники, стрічки новин тощо.

3. Відокремлює дані від HTML. За допомогою XML дані можна зберігати в різноманітних файлах XML. Таким чином, вам не потрібно буде оновлювати HTML, щоб вносити зміни в базові дані, що дозволить вам зосередитися на використанні HTML/CSS для відображення та стилю.

4. Дозволяє перевірку XML. Документ XML можна перевірити за допомогою DTD або схеми XML. Завдяки цьому XML-документ гарантовано буде синтаксично дійсним, і можна уникнути будь-яких проблем, спричинених недоліками XML.

5. XML підтримує Юнікод. Оскільки XML сумісний з Юнікодом, він може передавати практично будь-яку інформацію будь-якою письмовою людською мовою.

6. Використовується для створення нових мов. Багато нових мов Інтернету з'явилися в результаті XML. WSDL можна використовувати для опису доступних веб-служб. WAP і WML використовуються як мови розмітки портативних пристроїв. Для стрічок новин використовуються мови RSS [27].

XML працює кількома способами, одним із яких є передбачуваний формат даних. Через жорсткі вимоги до форматування XML програми, які обробляють або відображають закодовані дані, вироблятимуть помилку, якщо форматування неправильне.

XML-документ має бути дійсним XML-кодом, щоб він вважався правильно сформованим, тобто таким, що відповідає синтаксису XML і може бути прочитаний і зрозумілий аналізатором XML [30].

#### **2.4 Реляційна система керування базами даних SQLite**

SQLite – це компактна програмна бібліотека або база даних, яка інтегрована в численні програми. Однією з його відмінних рис є відсутність клієнт-серверної структури. Альтернативи SQLite включають MySQL, MongoDB і PostgreSQL [4].

Назва «SQLite» складається з «SQL», що є скороченням від «Structured Query Language», і «lite», розмовного терміна, який використовується тут для вказівки на те, що база даних є легшою версією, яка потребує менше пам'яті. Назва вказує на те, що, на відміну від інших поширених баз даних, реляційна база даних SQLite не вимагає додаткового серверного програмного забезпечення [16].

Крім того, легка система має невеликий розмір лише кілька сотень кілобайт і зберігає всю базу даних в одному файлі.

Бібліотека SQLite є однією з найкомпактніших у цьому списку, де розмір бібліотеки може бути менше 600 КБ. Завдяки своєму дуже малому розміру та

характеру RDBMS, вона дуже добре підходить для IoT та вбудованих пристроїв. Іншими хорошими випадками використання є веб-сайти з низьким і середнім трафіком (прибл. 100 тис. запитів на день), тестування та цілі внутрішньої розробки, аналіз даних за допомогою Tcl або Python та освітні цілі (легко налаштувати та можна використовувати для навчання SQL концепціям для студентів).

Однією з головних переваг SQLite є можливість виступати в якості додаткового рішення для клієнт-серверної корпоративної СУБД. Наприклад, вона може локально кешувати дані клієнта/сервера RDBMS і таким чином зменшувати затримку для запитів і підтримувати кінцеву програму в робочому стані в разі збоїв корпоративної RDBMS. SQLite – це база даних з відкритим кодом, яку підтримує група розробників. Для підтримки спільноти пропонується загальнодоступний список розсилки, а також надається платна професійна підтримка. Одними з ключових клієнтів SQLite є Facebook, Google і Apple.

Окрім наявності кількох корпоративних функцій, ще однією основною відмінністю між MySQL і SQLite є підтримка MySQL функцій для кількох користувачів. Це разом із корпоративними функціями та масштабованістю робить його ідеальним кандидатом для розподілених програм. MySQL має перевагу над PostgreSQL щодо простих і важких операцій читання, коли йдеться про пропускну здатність і продуктивність. Він також набагато простіший у встановленні та використанні та має ширшу спільноту порівняно з PostgreSQL.

Простота використання та керованість робить його чудовим інструментом для вебсайтів, вебзастосунків, OLTP- застосунків і є важливою частиною стеку LAMP. У MySQL 8 також представлено функції NoSQL, які прирівнюють її до PostgreSQL. SQLite завдяки перевагам здобув шанобливу репутацію як серед особистих користувачів, так і серед розробників комерційного програмного забезпечення.

Ось що робить реляційну систему таким популярним вибором:

- Область застосування. Низькі вимоги до пам'яті є основною перевагою системи баз даних. Маючи лише кілька сотень кілобайт, бібліотека займає мало

місця. Це позитивно впливає на швидкість роботи. SQLite виправдовує свою назву та виявляється ефективним рішенням як для великих, так і для малих програм.

- Немає необхідності в додатковому програмному забезпеченні. SQLite не потребує серверного програмного забезпечення, і його можна легко інтегрувати. Додаткові програми або зовнішні пакети не потрібні. Надана продуктивність сервера чітко відображається, і нею легко керувати.

- Універсальність. SQLite підтримує більшість команд мови SQL і сумісний із більшістю популярних сховищ даних. Ось чому бібліотека використовується в багатьох програмах, таких як Facebook і WhatsApp, а також у всіх основних браузерах, комерційних програмах і клієнтських операційних системах. Оскільки обмін між двома різними системами спрощений, багато розробників покладаються на економічне рішення як на підструктуру свого програмного забезпечення.

- Портативність. Згадані вище аспекти також пов'язані з тим фактом, що файли SQLite легко створювати резервні копії та транспортувати. Оскільки файли SQLite не залежать від серверних програм і не потрібно налаштовувати базу даних, можна використовувати SQLite на різних системах без будь-яких проблем.

- Надійність. Уніфікований доступ до даних, низьке споживання ресурсів і незалежність від структури сервер-клієнт роблять SQLite безпечним варіантом. Помилки пам'яті або проблеми, спричинені недостатньою кількістю оперативної пам'яті, є меншою проблемою для цієї бази даних [24].

- Публічний домен. SQLite доступний без ліцензії.

За допомогою SQLite дані можна зберігати в різних форматах. Це також означає, що перетворення не обов'язково потрібно для подальшої обробки в різних системах. Можливі типи даних, які можна зберегти та зберігати:

1. INTEGER: охоплює всі цілі значення.

2. REAL: Це включає наближене представлення дійсного числа у вигляді плаваючої коми.

3. TEXT: цей тип даних містить увесь звичайний текстовий вміст.

4. BLOB : цей термін розшифровується як двійковий великий об'єкт і стосується даних у форматі двійкового значення, таких як мініатюри або файли конфігурації в XML.

5. NULL: це дозволяє зіставляти нульові значення в SQLite.

Це означає, що всі типові типи даних можна зберігати в SQLite [29].

## **2.5 Висновок до другого розділу**

В другому розділі кваліфікаційної роботи розглянуто засоби, необхідні для розробки мобільних застосунків на платформі Android. Детально проаналізовано інструменти, такі як Android Studio, мова програмування Kotlin, особливості мови розмітки XML та реляційна система керування базами даних SQLite. Цей аналіз відображає актуальні можливості та інструменти, що використовуються для створення мобільних застосунків для платформи Android. Знання про ці засоби є ключовим для ефективної розробки програмного забезпечення з метою підтримки профілактики цукрового діабету та захворювань щитоподібної залози через мобільний застосунок.

## РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПЕРСОНАЛЬНОГО МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ПРОФІЛАКТИКИ ЦУКРОВОГО ДІАБЕТУ ТА ЗАХВОРЮВАНЬ ЩИТОПОДІБНОЇ ЗАЛОЗИ

### 3.1 Створення реєстраційного меню персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози

Перш за все, слід створити новий проєкт. Для цього потрібно натиснути на кнопку «File» у панелі інструментів, після цього з'явиться випадаюче меню у якому слід вибрати «New», а після цього вибрати «New project». На рисунку 3.1 зображено панель інструментів Android Studio.

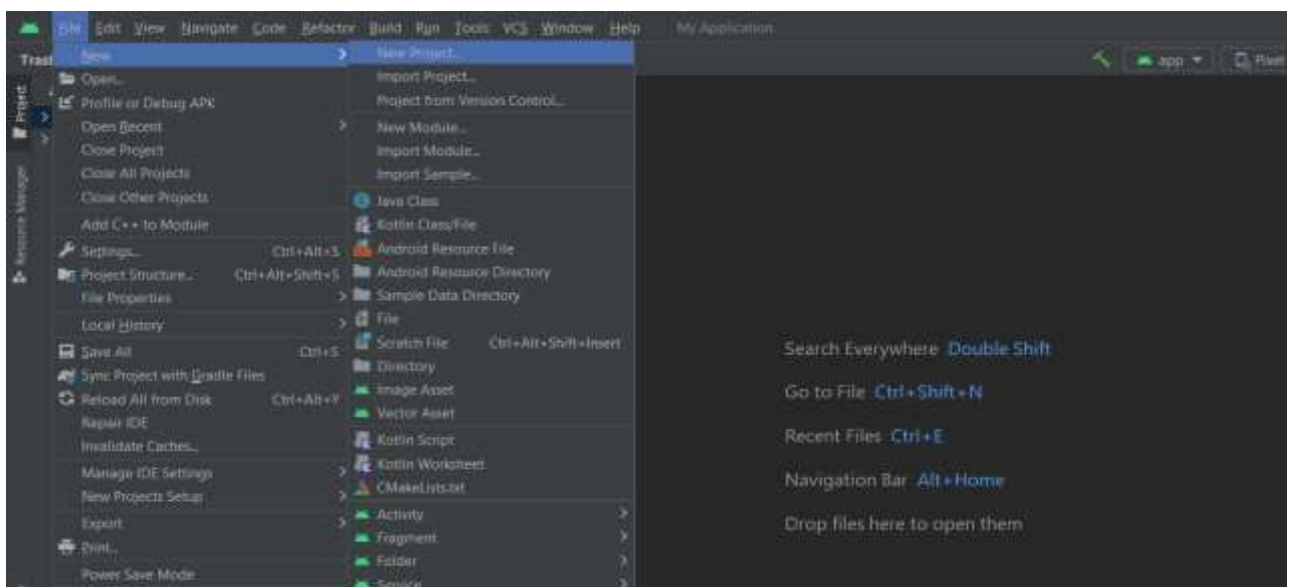


Рисунок 3.1 – Панель інструментів Android Studio

У результаті буде відкрито вікно «New project» у якому слід вибрати пункт «Empty project», для продовження слід натиснути на кнопку «Next», це зображено на рисунку 3.2.

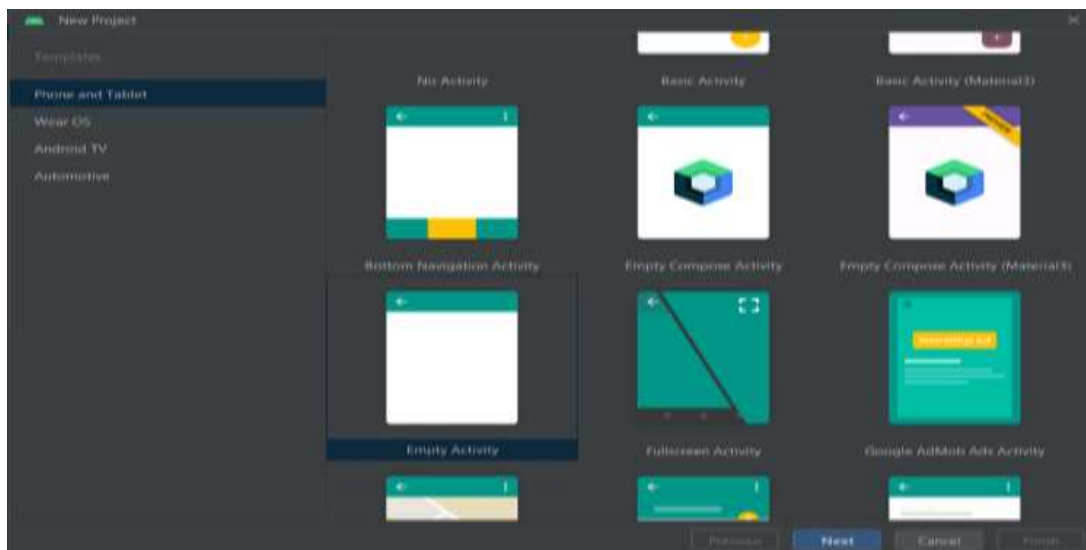


Рисунок 3.2 – Вікно «New project»

У наступному вікні буде відкрито панель конфігурації проєкту мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози, у якому слід створити назву та вибрати розташування файлу. Для завершення процесу створення нового проєкту слід натиснути на кнопку «Finish». На рисунку 3.3 зображено вікно конфігурації проєкту.

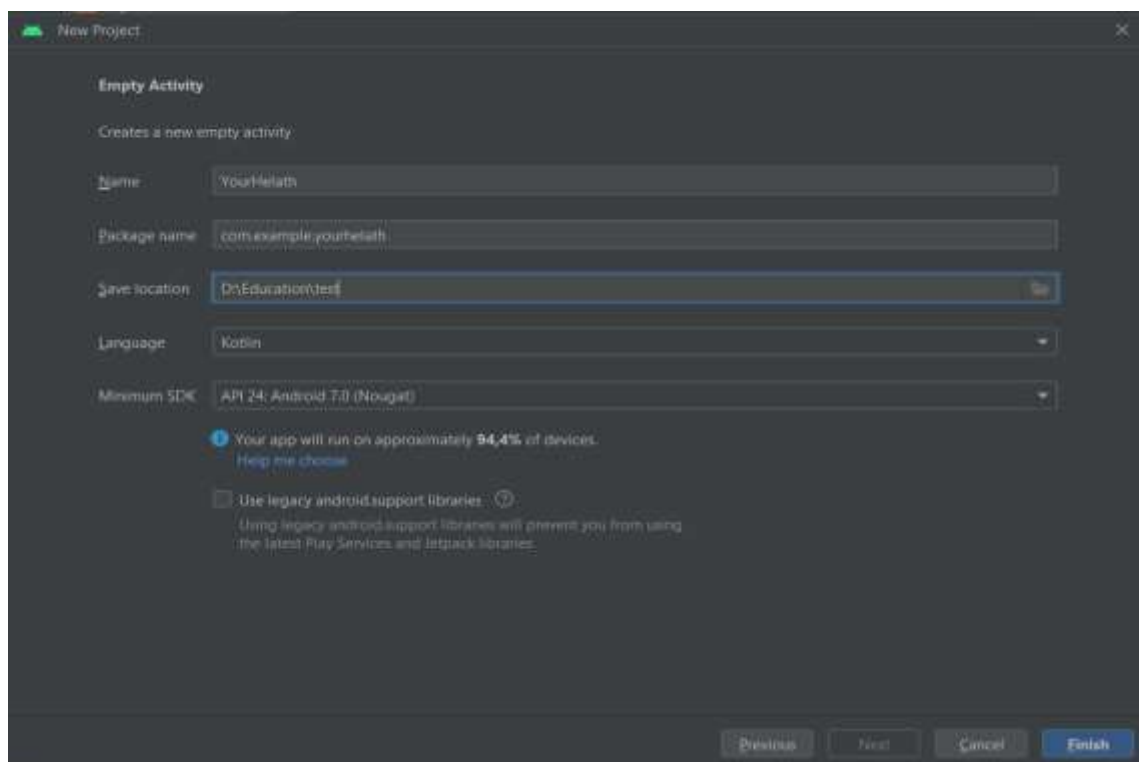


Рисунок 3.3 – Вікно конфігурації проєкту



Після цих дій проєкт мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози буде успішно створений. Наступним кроком слід створити нові фрагменти: для цього потрібно натиснути на кореневу папку «Java» та вибрати у меню пункт «New» а далі команду «Fragment», а у ньому «Fragment (blank)». На рисунку 3.4 зображено створення нового фрагменту мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози.

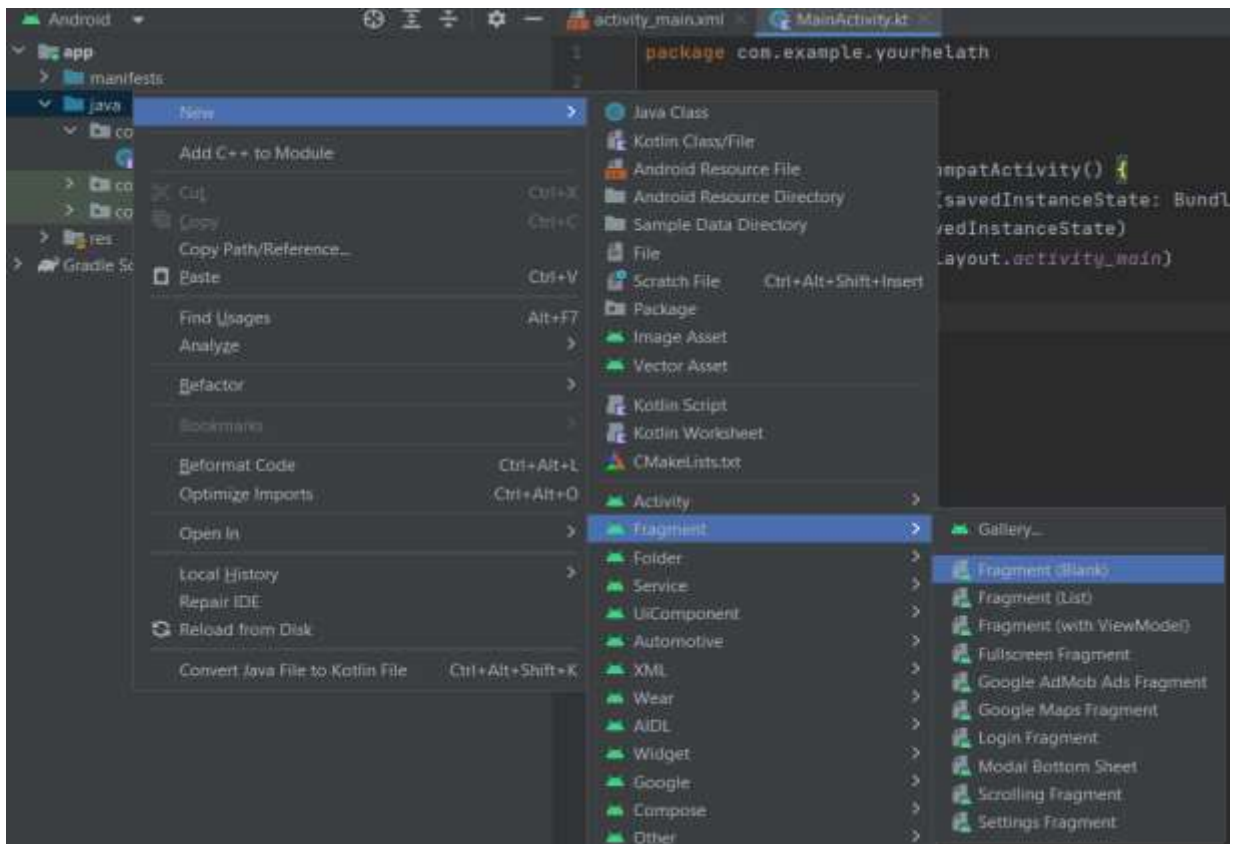


Рисунок 3.4 – Створення нового фрагменту мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози

У вікні конфігурації слід вказати тільки назву «Entrance1», для завершення слід натиснути на кнопку «Finish». Це зображено на рисунку 3.5. Таких фрагментів слід створити шість, вони матимуть назву: «Entrance2», «Entrance3», «Entrance4», «Entrance5» та «Result» відповідно.

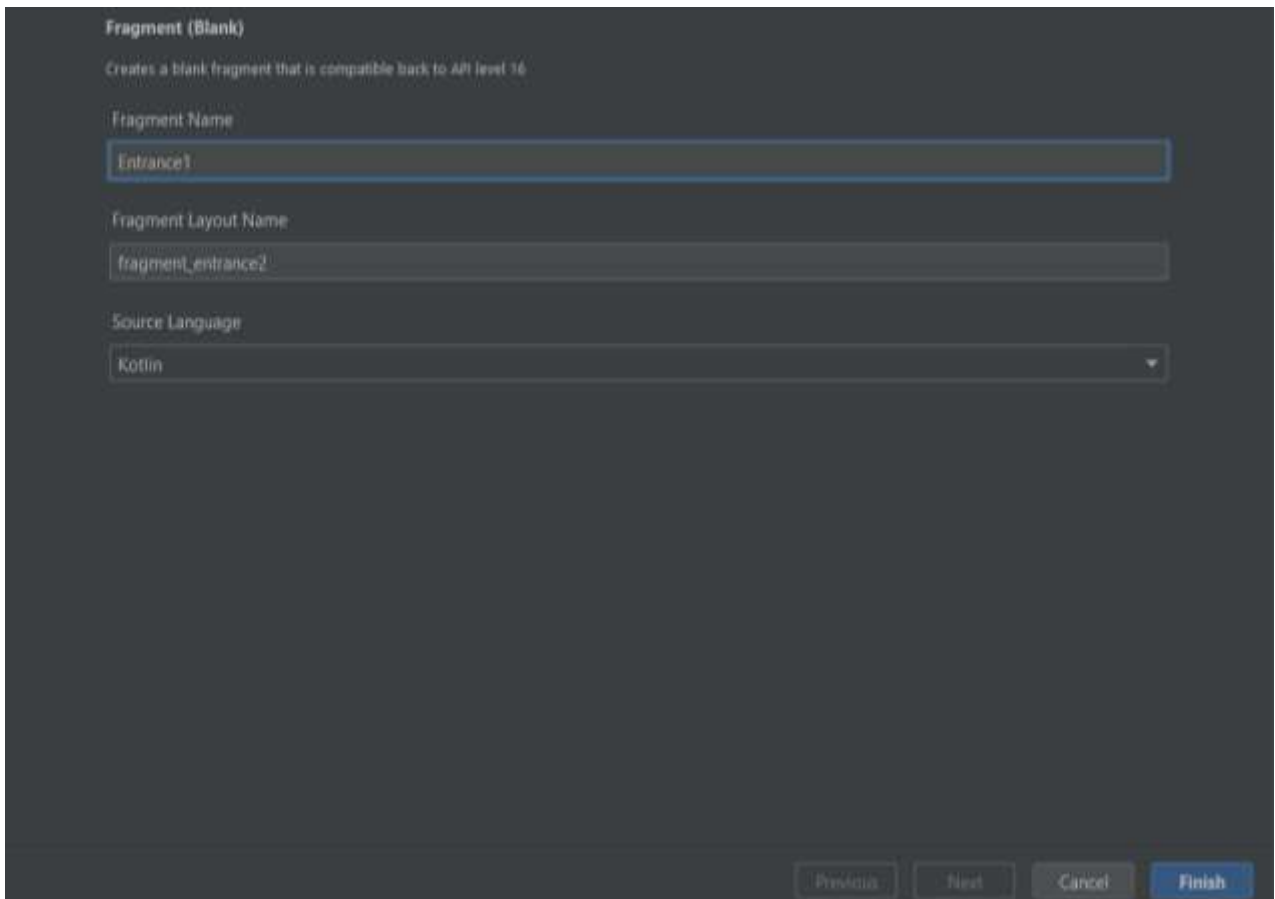


Рисунок 3.5 – Вікно конфігурації фрагменту мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози

Далі треба перейти у файл «`fragment_entrance1`» мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози, який був створений автоматично при створенні фрагменту. У цьому файлі створюється весь графічний інтерфейс мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози. Для початку за допомогою тегу «`ImageView`», створюється задній фон для першої реєстраційної сторінки. Цей тег матиме такі атрибути «`android:layout_width`» зі значенням «`match_parent`», для встановлення ширини зображення на увесь екран, «`android:layout_height`» зі значенням «`match_parent`», для встановлення висоти на увесь екран.

Для правильного кадрування використовується тег «`android:scaleType`» з параметром «`centerCrop`». Скрипт, що відповідає за задній фон мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози наведено на лістингу 3.1.

Лістинг 3.1 – Скрипт, що відповідає за задній фон мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози

```
<ImageView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:scaleType="centerCrop"
    android:src="@drawable/background"/>
```

Створюється тег «LinearLayout» з атрибутами:

«android:layout\_width» з параметром «match\_parent»;

«android:layout\_height» з параметром «match\_parent»;

«android:gravity» з параметром «center» встановить розташування всіх об'єктів чітко посередині області визначеної тегом;

«android:orientation» з параметром «vertical» встановить орієнтацію об'єктів всередині тегу вертикальною.

Всередині цього тегу створюється ще один тег «com.google.android.material.card.MaterialCardView».

Тег «MaterialCardView» використовується для створення візуальної картки. Його атрибути включають «android:layout\_width» зі значенням «292sp» та «android:layout\_height» зі значенням «78sp», щоб встановити ширину та висоту карточки. Атрибути «app:cardCornerRadius» та «app:cardBackgroundColor» встановлюють радіус закруглення кутів та колір фону відповідно.

У внутрішньому тезі «LinearLayout» атрибути «android:layout\_width» та «android:layout\_height» встановлюються на «match\_parent», щоб зайняти всю доступну площу в межах картки. Скрипт наведено на лістингу 3.2.

Лістинг 3.2 – Скрипт, що відповідає за текстові поля у картці мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```

android:gravity="center"
android:orientation="vertical">
    <com.google.android.material.card.MaterialCardView
        android:layout_width="292sp"
        android:layout_height="78sp"
        app:cardCornerRadius="12dp"
        app:cardBackgroundColor="#59FFFFFF">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:gravity="center"
            android:orientation="horizontal">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:fontFamily="@font/rubik_bold"
                android:text="Your"
                android:textColor="@color/black"
                android:textSize="48sp" />
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="2dp"
                android:fontFamily="@font/rubik_bold"
                android:text="Health"
                android:textColor="#64C9A5"
                android:textSize="48sp" />
            </LinearLayout>
        </com.google.android.material.card.MaterialCardView>

```

Створюються три ідентичні поля для введення інформації за допомогою тегу «TextInputLayout».

Для задання стилю використовується атрибут «style», який посилається на зовнішній ресурс з параметром «?attr/textInputOutlinedStyle». Ширина блоку встановлюється на 292 пікселів, висота – «wrap\_content». Задається відступ від верхнього краю екрану на 83 пікселів.

Встановлюється параметри для задання закругленості кутів блоку:

```

«app:boxCornerRadiusBottomStart»; «app:boxCornerRadiusBottomEnd»;
«app:boxCornerRadiusTopStart»;
«app:boxCornerRadiusTopEnd».

```

Встановлюється колір фону блоку – «app:boxBackgroundColor», колір тексту віджету підказки – «app:hintTextColor», колір обрамлення блоку – «app:boxStrokeColor», колір тексту підказки – «android:textColorHint».

У блок додається вбудований компонент «TextInputEditText», що відповідає за введення тексту. Для нього встановлюються наступні атрибути: ідентифікатор «android:id», ширина – «match\_parent», висота – «wrap\_content», підказка атрибут «android:hint» – «Username», максимальна кількість ліній «android:maxLines» – «1», колір тексту – «android:textColor» з параметром «@color/black». У наступних текстових полях змінюватиметься тільки атрибут, що відповідає за id та за підказки. Скрипт, що відповідає за ці текстові поля знаходиться у додатку А.

Після цього слід створити кнопку «Get Start». Для створення використовується тег «Button».

Атрибут «android:fontFamily» матиме значення «@font/rubik\_bold». Розмір тексту встановлюється атрибутом «android:textSize» зі значенням «25dp», а «app:cornerRadius» зі значенням «12dp» (див. лістинг 3.3).

Лістинг 3.3 – Скрипт, що відповідає за кнопку «Get start» мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози

```
<Button
```

```
    android:layout_marginTop="67dp"
    android:id="@+id/button1"
    style="@style/Widget.Material3.Button"
    android:layout_width="200dp"
    android:layout_height="70dp"
    android:backgroundTint="#5BFDFDFD"
    android:text="Get start"
    android:textColor="@color/black"
    android:fontFamily="@font/rubik_bold"
    android:textSize="25dp"
    app:cornerRadius="12dp" />
```

Результат створеного першого вікна реєстрації мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози продемонстровано у додатку Б.

Створюється тег «RelativeLayout» з атрибутами: «android:layout\_width» з параметром «match\_parent», «android:layout\_height» з параметром «match\_parent» та «android:gravity» з параметром «center\_horizontal». Внутрішній тег «Button» містить наступні атрибути: «android:id» з параметром «@+id/button2», «android:layout\_marginTop» з параметром «67dp», «style» з параметром «@style/Widget.Material3.Button», «android:layout\_gravity» з параметром «center\_horizontal», «android:layout\_width» з параметром «200dp», «android:layout\_height» з параметром «70dp», «android:backgroundTint» з параметром «#3E886E», «android:text» з параметром «NEXT», «android:textColor» з параметром «@color/white», «android:fontFamily» з параметром «@font/rubik\_bold», «android:textSize» з параметром «25dp», «app:cornerRadius» з параметром «12dp», «android:layout\_alignParentBottom» з параметром «true», «android:layout\_marginBottom» з параметром «100dp» та «android:foregroundGravity» з параметром «center». Скрипт, що описує ці об'єкти розміщено у додатку В. Зображення щойно створеної сторінки розміщено у додатку Д.

Додається кнопка для руху до наступної сторінки, вона є ідентичною, що була описана для сторінки «fragment\_entrance2». Наступним кроком буде створення сторінки на яку буде перенаправлено користувача, коли він обрав пункт, що аналізу не здано. Скрипт, що відповідає за текстові поля для введення даних мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози розміщено у додатку Е.

Перш за все, копіюється два теги з двома текстовими полями «YourHealth». Також створюється ідентична картка з текстовим полем, що інформує про те що аналізу не здано.

Після цього створюються два теги «MaterialCardView» з атрибутами: «android:id» з параметрами «@+id/howtoanalyses» та «@+id/addanalyses»

відповідно; «android:layout\_width» з параметром «match\_parent»; «android:layout\_height» з параметром «wrap\_content»; «android:layout\_gravity» з параметром «center»; «app:cardCornerRadius» з параметром «12dp»; «app:cardElevation» з параметром «7dp»; «android:layout\_marginTop» з параметром «50sp» для першого тегу та «35sp» для другого; «app:cardBackgroundColor» з параметром «@color/white»; «android:layout\_marginStart» з параметром «10dp»; «android:layout\_marginEnd» з параметром «10dp».

Лістинг, у якому розміщений увесь скрипт, що відповідає за сторінку на яку буде перенаправлено користувача, якщо він обрав пункт, що аналізи не здано знаходиться у додатку Ж. Зображення щойно створеної сторінки розміщено у додатку З.

Після цього створюється надписи та картки на яких буде розміщена проаналізована інформація. Для проаналізованої інформації про діабет та про щитовидну залозу ці поля будуть ідентичними, відрізнятиметься тільки надпис.

Перш за все, створюється тег «LinearLayout» з такими атрибутами: «android:layout\_width» з параметром «match\_parent», «android:layout\_height» з параметром «wrap\_content» та «android:orientation» з параметром «vertical». В середині цього тегу є дочірній елемент «TextView», що відповідає за надпис над картою. Він має такі атрибути: «android:layout\_width» з параметром «wrap\_content», «android:layout\_height» з параметром «wrap\_content», «android:layout\_gravity» з параметром «center\_horizontal», «android:layout\_marginTop» з параметром «30dp», «android:fontFamily» з параметром «@font/rubik\_bold», «android:text» з параметром «Щитовидна залоза», «android:textColor» з параметром «@color/black» та «android:textSize» з параметром «24sp». Скрипт, що відповідає за користувацький інтерфейс розміщено у додатку К.

Створення бази даних SQLite. Спочатку, створюється клас «DataBaseHelper», що наслідує клас «SQLiteOpenHelper», всередині цього класу створюється об'єкт «companion object», для того аби потім поля, що були

записані туди, можна було б використовувати з інших класів. У цьому об'єкті для створення полів використовуються такі ключові слова «private const val» та «const val», ці слова вказують, що поля є константними.

Будуть створені такі поля: «DB\_NAME», «DB\_VERSION», «TABLE\_THYROID», «THYROID\_ID», «T3», «T4», «TABLE\_DIABETS», «THYROID\_STIMULATING\_HORMONE», «DIABET\_ID», «HOMA\_IR», «DATE», «GLYCATED\_HEMOGLOBIN», «LIPIDOGRAM», «BMI», «TABLE\_WEIGHT», «WEIGHT\_ID», «WEIGHT».

Після цього створюється метод «onCreate» у якому буде створено необхідні таблиці. Для створення таблиці, перш за все, створюється змінна, у якій буде записано SQL-запит. Ключовим словом «CREATE TABLE», створюється таблиця, а через кому додаються необхідні поля для цієї таблиці.

Також створюється метод «onUpgrade», який буде перевіряти чи існує таблиця, якщо ні, то він її створить, для цього викликається метод «execSQL» для об'єкта «db», в параметри якого передається набір ключових слів та назва таблиці «DROP TABLE IF EXISTS \$TABLE\_THYROID». Скрипт, що відповідає за клас «DataBaseHelper» розміщений у додатку Л.

Для взаємодії з базою даних створюється ще один клас «MyDbAdapter», у ньому створюється об'єкт класу «SQLiteDatabase», що був створений раніше. У параметрах вказується аргумент «writableDatabase». Після цього створюються функції для додавання чи отримання даних, для всіх таблиць вони є ідентичними, відрізняється тільки назва таблиці та поля з яких беруться дані. Для створення функції для додавання інформації про щитовидну залозу, перш за все, пишеться ключове слово «fun», а тоді назва функції «add\_thyroid», приймати у аргументи буде об'єкт класу «Thyroid», що буде створений пізніше, а повертати бути число формату «Long». Створюється змінна «values», якій присвоюється значення функції «ContentValues», до якої було викликано ще одну функцію «apply». Всередині цієї функції викликається ще одна функція «put», у параметри до якої передається назва стовпця якому буде належати значення, та саме значення, це виконується для кожної із змінних. Після цього слід повернути об'єкт класу



«database» до якого викликано метод «insert» з такими параметрами: «DataBaseHelper.TABLE\_THYROID», «null», «values» (див. лістинг 3.4).

Лістинг 3.4 – Скрипт, що відповідає за клас, який надає доступ до бази даних мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози

```
class MyDbAdapter(private val context: Context) {
    private val database: SQLiteDatabase by lazy {
        DataBaseHelper(context).writableDatabase
    }
    fun add_thyroid(thyroid: Thyroid): Long {
        val values = ContentValues().apply {
            put(DataBaseHelper.THYROID_STIMULATING_HORMONE,
thyroid.thyroid_stimulating_hormone)
            put(DataBaseHelper.T3, thyroid.t3)
            put(DataBaseHelper.T4, thyroid.t4)
            put(DataBaseHelper.DATE, thyroid.date)
        }
        return database.insert(DataBaseHelper.TABLE_THYROID,
null, values)
    }
}
```

Створюється функція, що повертатиме список з даними мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози. Перш за все, створюється пустий список «thyroids» нічим не ініціалізований, після цього створюється змінна у якій буде міститися запит до бази даних. Потім функція виконує запит SQL до таблиці, вибираючи всі рядки з цієї таблиці.

Після того, як запит виконано, функція перевіряє, чи є дані в результатах запиту. Якщо є, то вона проходить по кожному рядку та створює новий об'єкт класу «Thyroid», використовуючи дані з кожного рядка. Далі, цей об'єкт додається до списку «thyroids».

Після того, як всі рядки були оброблені, функція повертає список «thyroids» з усіма об'єктами класу «Thyroid», які були зчитані з таблиці в локальній базі даних (див. лістинг 3.5).

## Лістинг 3.5 – Скрипт, що відповідає за функцію, що повертає дані з таблиці

```

fun get_thyroid():MutableList<Thyroid>{
    var thyroids = mutableListOf<Thyroid>()
    val query = "SELECT * FROM
${DataBaseHelper.TABLE_THYROID}"
    val cursor = database.rawQuery(query, null)
    if (cursor.moveToFirst()) {
        do {
            val thyroid_stimulating_hormone =
cursor.getDouble(cursor.getColumnIndex(DataBaseHelper.THYROID_
STIMULATING_HORMONE))
            val t3 =
cursor.getDouble(cursor.getColumnIndex(DataBaseHelper.T3))
            val t4 =
cursor.getDouble(cursor.getColumnIndex(DataBaseHelper.T4))
            val date =
cursor.getString(cursor.getColumnIndex(DataBaseHelper.DATE))

thyroids.add(Thyroid(thyroid_stimulating_hormone, t3, t4,
date))
        } while (cursor.moveToNext())
    }
    return thyroids
}

```

Після цього база даних мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози буде створена. Наступний крок – це створення функціональної частини та перехід між фрагментами. Вся функціональна частина писатиметься у фрагментах, що були створені раніше. Перш за все, у методі «onCreateView» слід створити об’єкт класу «MyDbAdapter». Створюється змінна «viewPager» їй присвоюється id від об’єкту «ViewPager2». Після цього за допомогою методу «setOnClickListener» зчитується натиснення кнопки, якщо було зроблене натиснення зчитуються дані з полів, за допомогою методів «findViewById» у параметрах якого вказується id об’єкта, що був створений у файлі XML. Це виконується для кожного з полів, після цього всі ці дані записуються у базу даних за допомогою створених раніше методів та виконується перехід до наступної сторінки за допомогою зміни значення поля «currentItem» (див. лістинг 3.6).

Лістинг 3.6 – Скрипт, що відповідає за зчитування даних з полів на першій сторінці реєстрації мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози

```
var view = inflater.inflate(R.layout.fragment_entrancel,
container, false)

        var db = MyDbAdapter(requireContext())

        var viewPager =
activity?.findViewById<ViewPager2>(R.id.viewPager)

view.findViewById<Button>(R.id.button1)?.setOnClickListener {
    var username: String =
view.findViewById<TextInputEditText>(R.id.username).text.toString()

        var weight: Double =
view.findViewById<TextInputEditText>(R.id.weight).text.toString()
.toDouble()
        var height: Double =
view.findViewById<TextInputEditText>(R.id.height).text.toString()
.toDouble()
        db.add_username(User(username, height))
        db.add_weight(Weight(weight, (weight/(height /
100).pow(2.0)), getDateAsString()))
        viewPager?.currentItem = 1
    }
return view
```

Для створення функціональної частини для другої сторінки реєстрації, перш за все, оголошуються об'єкти з XML-файлу, за допомогою методу «findViewById», у параметри передається id необхідного об'єкта. Після цього зчитується інформація про натискання на об'єкт «RadioButton» за допомогою методу «setOnClickListener», якщо цей об'єкт вже був натиснутий, то нічого не слід змінювати, але якщо не був натиснутий, то слід змінити стани за допомогою зміни значення поля «isChecked». Аналогічно все виконується і для наступного об'єкту.

Після цього зчитується натискання на кнопку та перевіряються стани об'єктів «RadioButton», від цих значень залежить куди буде перенаправлено користувача (див. лістинг 3.7).

### Лістинг 3.7 – Скрипт, що відповідає за функціональну частину другої сторінки реєстрації

```

var view = inflater.inflate(R.layout.fragment_entrance2, container,
false)

        var                                rb1                                =
view.findViewById<RadioButton>(R.id.radio_button_1)
        var                                rb2                                =
view.findViewById<RadioButton>(R.id.radio_button_2)
        rb1.setOnClickListener {
            if(rb1.isChecked) {
                rb1.isChecked = true
                rb2.isChecked = false
            }
        }
        rb2.setOnClickListener {
            if(rb1.isChecked) {
                rb1.isChecked = false
                rb2.isChecked = true
            }
        }

        var                                viewPager                            =
activity?.findViewById<ViewPager2>(R.id.viewPager)
        view.findViewById<Button>(R.id.button2).setOnClickListener
{
            if(rb1.isChecked) {
                viewPager?.currentItem = 2
            } else {
                viewPager?.currentItem = 5
            }
        }
        return view

```

Після цього на сторінках «entrance3» та «entrance4» зчитуються дані з полів аналогічним чином, як це було на «entrance1».

На сторінці з результатами «Results», перш за все, створюється об'єкт класу «MyDbAdapter». Після цього за допомогою методів «get\_diabet» та «get\_thyroid» отримується список зі всіма записами у базу даних, а після цього дістається останній елемент за індексом, це робиться за допомогою такого запису «db.get\_diabet().size-1». Створюються змінні, що міститимуть у собі id об'єктів, що відповідають за текст у файлі XML за допомогою методів «findViewById». Потім за допомогою блоків «if» та «if else» перевіряються дані та виводиться інформація про результати у текстові поля. Після цього зчитуються натискання

кнопки та виконується перехід на основну сторінку мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози. Скрипт, що відповідає за цей код розміщено у додатку М. Після цього функціональна частина реєстраційного меню буде зроблена.

### **3.2 Створення основної сторінки персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози**

Перш за все, для створення графічного інтерфейсу основної сторінки встановлюється колір заднього фону за допомогою атрибуту «android:background» зі значенням «#E9E8EC». Після цього створюється тег «RelativeLayout» з атрибутами, що відповідають за розмір: «android:layout\_width» з параметром «match\_parent», «android:layout\_height» з параметром «match\_parent». Після цього копіюються текстові поля з попередніх фрагментів, що відповідають за логотип. Створюється схожі текстові поля, тільки відрізнятиметься розташування та текст, атрибут «android:text» матиме значення «"Привіт, "», а інше текстове поле значення «username». Після цього створюється тег «MaterialCardView» з наступними атрибутами: «android:layout\_width» з параметром «328dp»; «android:layout\_height» з параметром «140dp»; «app:cardCornerRadius» з параметром «12dp»; «app:cardElevation» з параметром «7dp»; «android:layout\_gravity» з параметром «center»; «android:layout\_marginTop» з параметром «50sp»; «app:cardBackgroundColor» з параметром «@color/white». Детальніше це описано у додатку Н.

Необхідно створити дві такі картки, оскільки в першій міститиметься інформація про дані щитовидної залози, а в другій дані про діабет. Для створення графічного інтерфейсу меню, перш за все, створюється новий XML-файл під назвою «bottom\_sheet». Після цього створюється тег «FrameLayout» з атрибутами: «android:id» зі значенням «@+id/standard\_bottom\_sheet»; «style» зі

значенням «@style/Widget.Material3.BottomSheet»; «android:layout\_width» з параметром «match\_parent»; «android:layout\_height» з параметром «match\_parent».

У цьому внутрішньому тегу міститься внутрішній тег «LinearLayout» з атрибутами: «android:layout\_width» з параметром «wrap\_content»; «android:layout\_height» з параметром «wrap\_content»; «android:orientation» з параметром «vertical»; «android:layout\_gravity» з параметром «center\_horizontal». В цьому тегу також міститься тег «View», що відповідає за кнопку згортання, маю такі атрибути: «android:layout\_width» з параметром «90dp»; «android:layout\_height» з параметром «4dp»; «android:background» зі значенням «@drawable/rectangle»; «android:layout\_gravity» з параметром «center\_horizontal»; «android:layout\_marginTop» з параметром «10dp».

Зображення користувацького інтерфейсу головної сторінки показано у додатку П.

Далі міститься тег «com.google.android.material.card.MaterialCardView» з атрибутами: «android:id» зі значенням «@+id/add\_thyroid»; «android:layout\_width» з параметром «328dp»; «android:layout\_height» з параметром «66dp»; «android:layout\_marginTop» з параметром «20dp»; «app:cardCornerRadius» з параметром «12dp»; «app:cardBackgroundColor» з параметром «@color/white»; «android:backgroundTint» з параметром «#D9D9D9». Також створюється тег «TextView» з атрибутами: «android:layout\_width» з параметром «match\_parent»; «android:layout\_height» з параметром «match\_parent»; «android:text» з параметром «Щитовидна залоза»; «android:textColor» з параметром «@color/black»; «android:textSize» з параметром «20dp»; «android:fontFamily» з параметром «@font/rubik\_bold»; «android:gravity» з параметром «center», що відповідає за вирівнювання тексту по центру. Таких елементів «MaterialCard» необхідно створити ще два: один для діабету, а інший для ваги. Відрізнятимуться тільки зображення та надписи. Скрипт, що відповідає за графічний інтерфейс нижнього меню мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози розміщено у додатку Р.

Для додавання функціональності нижньому меню створюється клас «ModalBottomSheet», що наслідується від «BottomSheetDialogFragment». Створюється змінна, що міститиме у собі посилання на XML-файл з нижнім меню. У методі «onCreateView» створюються змінні для карток за допомогою методу «findViewById» з параметром, що був вказаний раніше для конкретної картки. Після цього за допомогою методу «setOnClickListener» зчитується натискання на картку, якщо натискання було зчитане, то користувача буде перенаправлено на іншу необхідну сторінку мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози (див. лістинг 3.8).

Лістинг 3.8 – Скрипт, що відповідає за клас з нижнім меню мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози

```
class ModalBottomSheet : BottomSheetDialogFragment() {

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        val view = inflater.inflate(R.layout.bottom_sheet,
            container, false)
        val add_thyroid = view.findViewById<MaterialCardView>(R.id.add_thyroid)
        val add_diabet = view.findViewById<MaterialCardView>(R.id.add_diabet)
        val add_weight = view.findViewById<MaterialCardView>(R.id.add_weight)
        add_thyroid.setOnClickListener {
            val intent = Intent(requireActivity(),
                Thyroid_data::class.java)
            startActivity(intent)
        }
        add_diabet.setOnClickListener {
            val intent = Intent(requireActivity(),
                Diabet_data::class.java)
            startActivity(intent)
        }
        add_weight.setOnClickListener {
            val intent = Intent(requireActivity(),
                Weight_data::class.java)
        }
    }
}
```

```

        startActivity(intent)
    }
    return view
}

companion object {
    const val TAG = "ModalBottomSheet"
}
}

```

Після цього, вже в основному класі, створюється об'єкт класу «MyDbAdapter», після цього ініціалізується змінна, що відповідатиме за взаємодію з кнопкою, за допомогою методу «findViewById» у параметр вказується id, що було створене раніше. Наступним кроком слід створити об'єкт класу «modalBottomSheet» за допомогою конструктора цього класу. За допомогою викликаного методу «setOnClickListener» зчитується нажаття у об'єкта «button», якщо ця кнопка була нажата викликається метод «show» з параметрами «supportFragmentManager» та «ModalBottomSheet.TAG» до об'єкта «modalBottomSheet». Також за допомогою функції «findViewById» та методу викликаного до неї «text» змінюється текст, що відповідає за привітання до користувача. За цим самим принципом міняється вся інша текстова інформація на сторінці. Також зчитується натиснення для об'єкту «MaterialCardView», що був створений для переадресації користувача на сторінку з інформацією.

Скрипт, що відповідає за функціональну частину основної сторінки мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози розміщено у додатку С.

### **3.3 Створення вікна з інформацією про показники цукрового діабету та захворювань щитоподібної залози**

Для створення графічного інтерфейсу мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози, необхідно скопіювати текстові поля, що відповідають за логотип. Після копіювання слід створити картку на якій буде розміщена інформація про певні показники, для



цього використовується тег «MaterialCardView», він матиме такі атрибути: «android:layout\_width» з параметром «328dp»; «android:layout\_height» з параметром «match\_parent»; «android:layout\_marginTop» з параметром «20dp»; «app:cardCornerRadius» з параметром «12dp»; «app:cardElevation» з параметром «7dp»; «android:layout\_gravity» з параметром «center»; «app:cardBackgroundColor» з параметром «@color/white»; «android:layout\_marginBottom» з параметром «80sp». До нього додається внутрішній тег «LinearLayout» з атрибутами: «android:layout\_width» з параметром «match\_parent»; «android:layout\_height» з параметром «match\_parent»; «android:orientation» з параметром «vertical». У цьому внутрішньому тезі створюється тег «TextView» з атрибутами: «android:layout\_width» з параметром «wrap\_content»; «android:layout\_height» з параметром «wrap\_content»; «android:text» зі значенням «Історія»; «android:textColor» з параметром «@color/black»; «android:textSize» з параметром «20dp»; «android:fontFamily» з параметром «@font/rubik\_medium»; «android:layout\_margin» з параметром «8dp».

У цьому внутрішньому тезі створюються чотири теги «TextView» з атрибутами: «android:layout\_width» з параметром «0dp»; «android:layout\_height» з параметром «wrap\_content»; «android:text» зі значенням «Глікований гемоглобін», «НОМА-IR», «Ліпідограма» та «Дата»; «android:textColor» з параметром «@color/black»; «android:layout\_weight» зі значеннями «1», які відповідають рівній ширині для кожного елемента; «android:gravity» з параметром «center\_horizontal»; «android:fontFamily» з параметром «@font/rubik\_medium»; «android:textSize» з параметром «13dp». Після цього графічний інтерфейс сторінки, що відповідає за інформацію про певні показники буде створено, скрипт розміщено у додатку Т.

В першу чергу необхідно створити об'єкт «MyDbAdapter», після цього за допомогою методу «get\_diabet» викликаного до об'єкта «MyDbAdapter». Також створюється об'єкти класу «diabetAdapter» та «LinearLayoutManager». Викликається метод «apply» до об'єкту «RecyclerView», у фігурних дужках

вказуються необхідні адаптери мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози.

Після цього у картці встановлюються дані у Для переходу використовується функція «startActivity» у параметрах буде щойно створений об'єкт класу «Intent» (див. лістинг 3.9). Ідентичний код слід написати для щитовидної залози та ваги.

Лістинг 3.9 – Скрипт, що відповідає за функціональну частину сторінки з показниками мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози

```
val db = MyDbAdapter(this)

    DiabetList = db.get_diabet()
    diabetAdapter = DiabetAdapter(DiabetList)
    LinearLayoutManager = LinearLayoutManager(this)

    findViewById<RecyclerView>(R.id.recyclerViewDiabet).apply {
        adapter = diabetAdapter
        layoutManager = LinearLayoutManager
    }

    findViewById<TextView>(R.id.diabet_1).text =
    DiabetList[DiabetList.size-1].glycated_hemoglobin.toString()
    findViewById<TextView>(R.id.diabet_2).text =
    DiabetList[DiabetList.size-1].homa_ir.toString()
    findViewById<TextView>(R.id.diabet_3).text =
    DiabetList[DiabetList.size-1].lipidogram.toString()

    findViewById<Button>(R.id.add_diabet_button).setOnClickListener {
        val intent = Intent(this, AddDiabet::class.java)
        startActivity(intent)
    }
```

Описано створення графічного інтерфейсу для відображення інформації про певні показники здоров'я з використанням компонентів Android, таких як MaterialCardView, LinearLayout та TextView. Крім того, продемонстровано приклад коду, який відповідає за функціональну частину, зокрема, ініціалізацію бази даних, адаптерів мобільного застосунку для профілактики цукрового

діабету та захворювань щитоподібної залози для відображення даних у RecyclerView, а також обробку натискання кнопки для переходу на іншу активність.

### 3.4 Створення сторінки з додаванням даних щодо цукрового діабету та захворювань щитоподібної залози

Перш за все, для створення графічного інтерфейсу для сторінки з додаванням даних необхідно скопіювати логотип, що представлений, як два текстові поля. Після цього створюється тег «LinearLayout» з такими атрибутами: «android:layout\_width» з параметром «match\_parent»; «android:layout\_height» з параметром «wrap\_content»; «android:orientation» з параметром «vertical». Всередині тегу «LinearLayout» створюється тег «TextView» для надпису, що інформує користувача, які дані він буде вводити. Більш детально це показано у додатку У.

Цей тег має наступні атрибути: «android:layout\_marginTop» з параметром «50dp»; «android:layout\_width» з параметром «wrap\_content»; «android:layout\_height» з параметром «wrap\_content»; «android:text» з параметром «Діабет»; «android:textColor» з параметром «@color/black»; «android:textSize» з параметром «24sp»; «android:fontFamily» з параметром «@font/rubik\_bold»; «android:layout\_gravity» з параметром «center\_horizontal» (див. лістинг 3.10).

#### Лістинг 3.10 – Скрипт, що відповідає за інформуючий текст

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <TextView
        android:layout_marginTop="50dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Діабет"
        android:textColor="@color/black"
```

```

android:textSize="24sp"
android:fontFamily="@font/rubik_bold"
android:layout_gravity="center_horizontal"/>

```

Після цього слід скопіювати поля для введення даних, що були вже описані при створенні сторінки «fragment\_entrance3».

Також додається кнопка, що повертатиме користувача на основну сторінку та записуватиме дані.

Кнопка додається за допомогою тегу «Button» з наступними атрибутами:

«android:id» з параметром «@+id/save\_btn»; «android:layout\_marginTop» з параметром «67dp»; «style» з параметром «@style/Widget.Material3.Button»; «android:layout\_gravity» з параметром «center\_horizontal»; «android:layout\_width» з параметром «200dp»; «android:layout\_height» з параметром «70dp»; «android:backgroundTint» з параметром «#3E886E»; «android:text» з параметром «SAVE»; «android:textColor» з параметром «@color/white»; «android:fontFamily» з параметром «@font/rubik\_bold»; «android:textSize» з параметром «25dp»; «app:cornerRadius» з параметром «12dp»; «android:layout\_alignParentBottom» з параметром «true»; «android:layout\_marginBottom» з параметром «100dp»; «android:foregroundGravity» з параметром «center» (див. лістинг 3.11).

Лістинг 3.11 – Скрипт, що відповідає за кнопку для збереження даних

<Button

```

android:id="@+id/save_btn"
android:layout_marginTop="67dp"
style="@style/Widget.Material3.Button"
android:layout_gravity="center_horizontal"
android:layout_width="200dp"
android:layout_height="70dp"
android:backgroundTint="#3E886E"
android:text="SAVE"
android:textColor="@color/white"
android:fontFamily="@font/rubik_bold"
android:textSize="25dp"
app:cornerRadius="12dp"
android:layout_alignParentBottom="true"
android:layout_marginBottom="100dp"
android:foregroundGravity="center" />

```

Буде створено функціональну частину мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози та здійснено запис у базу даних. Перш за все, створюється об'єкт класу «MyDbAdapter». Після цього зчитується інформація про натискання кнопки за допомогою методу «setOnClickListener» викликаного до «findViewById». Якщо відбулося нажаття необхідно взяти дані з полів, за допомогою «findViewById», також до цього методу викликаються інші методи: «text», «toString()» та «toDouble()». Для запису даних у базу даних викликається метод «add\_diabet», туди передається об'єкт класу «Diabet», що буде створено пізніше, у параметри передаються дані, що були зчитані, з полів. Для переадресації користувача на іншу сторінку використовується клас «Intent» у параметри якого передається: «this», «Diabet\_data::class.java».

Для переходу використовується функція «startActivity» у параметрах буде щойно створений об'єкт класу «Intent» (див. лістинг 3.12). Для кожного з показників буде створено ідентичний скрипт для запису.

Лістинг 3.12 – Скрипт, що відповідає за функціональну частину сторінки для додавання даних мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози

```
var db = MyDbAdapter(this)

        findViewById<Button>(R.id.save_btn).setOnClickListener
    {
        var input1 =
    findViewById<TextInputEditText>(R.id.input1).text.toString().t
    oDouble()
        var input2 =
    findViewById<TextInputEditText>(R.id.input2).text.toString().t
    oDouble()
        var input3 =
    findViewById<TextInputEditText>(R.id.input3).text.toString().t
    oDouble()
        db.add_diabet(Diabet(input1, input2, input3,
    MainPage.getCurrentDateAsString()))
        var intent = Intent(this, Diabet_data::class.java)
        startActivity(intent)
    }
```

Створюються класи, що будуть об'єктами для зберігання даних мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози. Перш за все, пишеться ключові слова «data class» та назву класу «Diabet», параметрами цього класу будуть такі поля: «glycated\_hemoglobin» типу «Double»; «homa\_ir» типу «Double»; «lipidogram» типу «Double»; «date» типу «String». Ідентичні класи необхідно створити для щитовидної залози, ваги та користувача (див. лістинг 3.13).

### Лістинг 3.13 – Скрипт, що відповідає за клас «Diabet»

```
data class Diabet(var glycated_hemoglobin: Double,
                 var homa_ir: Double,
                 var lipidogram: Double,
                 var date: String)
```

Описано процес створення графічного інтерфейсу для сторінки введення даних в Android-застосунку, зокрема додавання текстових полів і кнопки для збереження даних мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози, а також реалізація функціональної частини для зчитування введених даних і збереження їх у базу даних. Також розглядається створення класів для зберігання цих даних.

## 3.5 Створення сторінки з корисною інформацією щодо профілактики цукрового діабету та захворювань щитоподібної залози

Перш за все, необхідно скопіювати тег з двома текстовими полями «YourHealth», що являються логотипом мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози, вони будуть повністю ідентичними. Після цього створюється картка з текстовим полем, вона буде повністю аналогічною до тієї, що інформує користувача про те чи здані аналізи чи ні. Також створюється нова картка з текстовими полями, у які міститимуться поради. Для її створення використовується тег «MaterialCardView» з наступними атрибутами: «android:layout\_width» з параметром «340dp»;

«android:layout\_height» з параметром «wrap\_content»; «app:cardCornerRadius» з параметром «12dp»; «app:cardElevation» з параметром «7dp»; «android:layout\_gravity» з параметром «center»; «android:layout\_marginTop» з параметром «50sp»; «app:cardBackgroundColor» з параметром «@color/white». Внутрішній тег «LinearLayout» створюється з атрибутами: «android:layout\_width» з параметром «match\_parent»; «android:layout\_height» з параметром «match\_parent»; «android:orientation» з параметром «vertical». До Скрипт, у якому знаходить код, що відповідає за графічний інтерфейс розміщено у додатку Ф.

Далі буде створюватись функціональна частини. Функціональна частина цієї сторінки повинна виконувати перехід користувача на основну сторінку, для цього використовується метод «findViewById» параметризований значенням «Button», для цього об'єкту викликається функція «setOnClickListener». Всередині цієї функції створюється об'єкт класу «Intent» у параметри якого передається: «this», «MainPage::class.java».

Для переходу мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози використовується функція «startActivity» у параметрах буде щойно створений об'єкт класу «Intent».

```
findViewById<Button>(R.id.ok_btn).setOnClickListener {
    var intent = Intent(this, MainPage::class.java)
    startActivity(intent)
}
```

Описано створення графічного інтерфейсу для сторінки в Android-застосунку з логотипом мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози, картками для інформування користувача та надання порад, а також реалізація функціональної частини для переходу користувача на основну сторінку.

### 3.6 Висновок до третього розділу

В третьому розділі кваліфікаційної роботи була проведена практична реалізація персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози. Описано процес створення реєстраційного меню, основної сторінки, вікна з інформацією про показники, сторінки з додаванням даних та сторінки з корисною інформацією мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози. Ці кроки демонструють практичну реалізацію функціональних можливостей запропонованого мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози.

Практична частина дозволяє зрозуміти, як реалізувати концепції, описані в теоретичних розділах, у вигляді реального програмного забезпечення. Такий підхід допомагає підтвердити придатність обраного підходу до розробки застосунку для вирішення важливих медичних завдань.



## РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 4.1 Шляхи підвищення якості життя та профілактики захворювань

Підвищення якості життя та профілактика захворювань є важливими аспектами збереження здоров'я та добробуту. Для досягнення цих цілей необхідно впроваджувати збалансоване харчування, фізичну активність, підтримку психічного здоров'я, якісний сон та збалансоване співвідношення роботи та відпочинку. Збалансоване харчування передбачає вживання різноманітних продуктів, багатих на поживні речовини, зменшення споживання насичених жирів, солі та цукру, а також збільшення кількості овочів, фруктів, цільнозернових продуктів та білків. Регулярна фізична активність включає не менше 150 хвилин на тиждень помірної активності або 75 хвилин інтенсивної активності, а також вправи на розтяжку та зміцнення м'язів. Для підтримки психічного здоров'я корисні практики медитації, йоги та релаксації, збалансоване співвідношення роботи та відпочинку, підтримка соціальних зв'язків та взаємодія з близькими людьми.

Якісний сон забезпечується дотриманням режиму сну, який включає 7-9 годин сну на добу, та створенням комфортних умов для сну, таких як відповідна температура, тиша та затемнення кімнати. Збалансоване співвідношення роботи та відпочинку передбачає уникання перевтоми, планування часу для відпочинку та хобі, навчання керування стресом та розподіл обов'язків.

Для профілактики захворювань важливо проходити регулярні медичні огляди та скринінги для раннього виявлення захворювань, виконувати рекомендації лікарів та дотримуватися призначених профілактичних заходів. Вакцинація відіграє ключову роль у попередженні інфекційних захворювань, тому необхідно вчасно отримувати необхідні щеплення. Дотримання правил особистої гігієни, таких як регулярне миття рук та догляд за ротовою порожниною, а також підтримка чистоти в побуті та на робочому місці, сприяють

зниженню ризику захворювань. Відмова від шкідливих звичок, таких як куріння, надмірне споживання алкоголю та наркотиків, також є важливою складовою профілактики захворювань. Підтримка нормальної маси тіла і зменшення впливу шкідливих екологічних факторів, таких як забруднення повітря та хімічні речовини, допомагають зберігати здоров'я. У разі необхідності, слід використовувати засоби індивідуального захисту. Застосування цих рекомендацій допоможе підтримувати здоров'я на високому рівні та зменшити ризик виникнення різних захворювань [27].

#### **4.2 Загальні вимоги безпеки з охорони праці для користувачів мобільних застосунків**

Мобільні застосунки стали невід'ємною частиною сучасного життя, забезпечуючи користувачів різноманітними послугами, зокрема у сфері охорони здоров'я, освіти, фінансів та розваг. Проте тривале використання мобільних пристроїв може негативно впливати на здоров'я користувачів. Цей документ визначає загальні вимоги безпеки з охорони праці для користувачів мобільних застосунків, щоб зменшити ризики і покращити умови користування.

Ергономічні вимоги включають підтримку правильної постави під час використання мобільних пристроїв. Важливо уникати тривалого нахилу голови вперед, щоб зменшити навантаження на шию. Пристрій повинен триматися на рівні очей, щоб уникнути надмірного згинання шиї, а для зручності можна використовувати підставки. Екран пристрою має бути розташований на рівні очей, щоб зменшити навантаження на шию і очі. Рекомендується використовувати підставки або тримачі [26].

Оптимальні умови освітлення передбачають, що екран пристрою не повинен бути єдиним джерелом світла в кімнаті. Краще працювати при природному освітленні або використовувати м'яке розсіяне світло. Яскравість екрана повинна відповідати рівню освітлення в приміщенні, уникаючи надто яскравих або надто темних налаштувань.

Для режимів праці та відпочинку користувачам рекомендується робити короткі перерви кожні 20-30 хвилин для розминки та відпочинку очей. Важливо виконувати вправи для очей, фокусуючись на об'єктах різної відстані, щоб зменшити навантаження на зір. Регулярно потрібно вставати і виконувати легкі фізичні вправи, щоб покращити кровообіг і зменшити м'язове напруження.

Здоров'я очей можна підтримувати, тримаючи пристрій на відстані 40-50 см від очей. Рекомендується використовувати пристрої з високою частотою оновлення екрану, щоб зменшити мерехтіння, яке викликає втому очей.

Гігієнічні вимоги включають регулярне очищення екрану і корпусу мобільного пристрою антисептичними серветками. Користувачі повинні мити руки перед і після використання пристрою.

Інформованість користувачів передбачає надання їм інформації щодо правил безпечного використання мобільних застосунків. Також важливо забезпечити доступ до технічної підтримки для консультацій з питань безпеки та здоров'я.

#### **4.3 Оцінка ефективності мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози**

Оцінка ефективності мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози є важливим аспектом в області розробки медичного програмного забезпечення. У сфері охорони здоров'я програмування використовується для створення медичних інформаційних систем, телемедицини, пристроїв для моніторингу пацієнтів та інших технологій, які сприяють збереженню здоров'я та безпеці пацієнтів.

Для оцінки ефективності такого мобільного застосунку необхідно враховувати кілька ключових критеріїв. По-перше, це точність і своєчасність збору даних про стан здоров'я користувача. Застосунок повинен забезпечувати безперервний моніторинг рівня глюкози в крові, параметрів щитоподібної залози

та інших важливих показників, що дозволяє своєчасно виявляти відхилення та реагувати на них.

По-друге, важливою є інтеграція з іншими медичними системами та пристроями. Це включає можливість синхронізації даних з електронними медичними записами, обмін інформацією з лікарями та іншими медичними працівниками через телемедицину, а також сумісність з пристроями для вимірювання життєво важливих показників.

По-третє, застосунок повинен надавати користувачам персоналізовані рекомендації щодо способу життя, харчування та фізичної активності на основі зібраних даних. Це сприяє не лише профілактиці захворювань, але й покращенню загального стану здоров'я користувача.

Ефективність мобільного застосунку також залежить від його зручності у використанні. Інтерфейс повинен бути інтуїтивно зрозумілим, щоб користувачі різного віку та технічного рівня могли легко використовувати всі функції. Крім того, регулярні оновлення та підтримка з боку розробників забезпечують покращення функціональності та адаптацію до нових вимог і технологій.

#### **4.4 Висновок до четвертого розділу**

В четвертому розділі кваліфікаційної роботи розглянуто ключові аспекти безпеки життєдіяльності та основ охорони праці, що включають шляхи підвищення якості життя, профілактики захворювань, заходи безпеки для користувачів мобільних застосунків та оцінку ефективності таких застосунків у профілактиці цукрового діабету та захворювань щитоподібної залози.

Встановлено, що збалансоване харчування, фізична активність, психічне здоров'я та якісний сон є основними факторами, що сприяють поліпшенню якості життя. Оцінка ефективності мобільних застосунків показала їх важливість у моніторингу здоров'я, наданні персоналізованих рекомендацій та інтеграції з медичними системами, що значно сприяє профілактиці та управлінню захворюваннями.

## ВИСНОВКИ

Розробка мобільних застосунків є дуже складним завданням, яке вимагає від розробників багато знань та досвіду. Для того, щоб створити успішний застосунок, потрібно зрозуміти потреби та вимоги користувачів, а також дослідити конкурентне середовище. Крім того, важливо розуміти особливості платформ, на яких буде запущений застосунок, щоб забезпечити його ефективну роботу та зручне використання. За останні роки мобільні застосунки стали необхідністю для бізнесу, особистого життя та розваг. Вони дозволяють людям спрощувати своє життя і ефективніше використовувати свій час. Наприклад, застосунки для здоров'я і фітнесу допомагають користувачам відслідковувати свої показники, а застосунки для банківських операцій дозволяють проводити фінансові операції в будь-який час та з будь-якого місця.

Під час виконання кваліфікаційної роботи освітнього рівня «Бакалавр» було проведено аналіз предметної області, яка пов'язана з розробкою мобільних застосунків. Були вивчені засади створення користувацького інтерфейсу та досвід користувача за допомогою Android Studio, мови розмітки XML для ієрархічно структурованих даних, статично типізованої мови програмування Kotlin.

Застосунок «YourHealth» точно не є досконалим, оскільки можна додати багато інших функцій, наприклад, можливість створювати власний акаунт у якому вся інформація зберігатиметься у хмарі. Також можна додати інші показники та можливість користувачу обирати, які показники він хоче бачити на основній сторінці. Після виконання кваліфікаційної роботи освітнього рівня «Бакалавр» було опановано основні аспекти проектування та організації розробки мобільних застосунків. Було вивчено та здобуто навички розробки застосунків з використанням мови програмування Kotlin. У результаті було успішно реалізовано всі заплановані завдання.

Підсумком проведеної роботи є створений готовий продукт, який був протестований у реальних умовах.

## ПЕРЕЛІК ДЖЕРЕЛ

1. Adobe. Робота із XML [Електронний ресурс]: 2024. – Режим доступу: <https://helpx.adobe.com/ua/incopy/using/xml.html> (дата звернення 16.04.2024)
2. Android Application Development: A Brief Overview of Android Platforms and Evolution of Security Systems / A. Sarkar et al. Palladam : IEEE, 2019. 79 p.
3. Android Studio: основний посібник для розробників. [Електронний ресурс] – Режим доступу: <https://medium.com/@trulydigitalmedia/android-studio-the-essential-guide-for-developers-cd43212a1919> (дата звернення 17.03.2024).
4. Difference between MySQL and SQLite [Електронний ресурс] – Режим доступу: <https://www.geeksforgeeks.org/difference-between-mysql-and-sqlite/> (дата звернення 17.03.2024)
5. How to install Android Studio [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.android.com/studio/install> (дата звернення 16.04.2024)
6. Inspection code on Android Studio [Електронний ресурс] – Режим доступу: <https://developer.android.com/studio/write/lint/> (дата звернення 20.04.2024)
7. Iversen J., Eierman M. Learning Mobile App Development. Boston : Addison-Wesley, 2014. 441 p.
8. Java vs Kotlin [Електронний ресурс] – Режим доступу: <https://www.educba.com/java-vs-kotlin/> – (дата звернення 18.04.2024)
9. JavaRush. Поняття XML [Електронний ресурс]: 2024. – Режим доступу: <https://javarush.com/ua/groups/posts/uk.2287.xml-u-java-jsho-ce> (дата звернення 16.04.2024)
10. Joorabchi, M.E., Mesbah, A., Kruchten, P.: Real challenges in mobile app development. In: ACM / IEEE International Symposium on Empirical Software Engineering and Measurement, Baltimore, Maryland, US, October 2013
11. Kotlin docs [Електронний ресурс] – Режим доступу: <https://kotlinlang.org/docs/home.html> – (дата звернення 18.04.2024)

12. Kotlin in Android Studio [Електронний ресурс] – Режим доступу: <https://developer.android.com/kotlin> (дата звернення 16.04.2024)
13. Kotlin vs Java Comparison [Електронний ресурс] – Режим доступу: <https://www.xenonstack.com/blog/kotlin-andriod/> (дата звернення 18.04.2024)
14. Krypton. Основи Kotlin [Електронний ресурс]: 2024. – Режим доступу: [https://krypton.com.ua/tutorial/kotlin\\_base/#google\\_vignette](https://krypton.com.ua/tutorial/kotlin_base/#google_vignette) (дата звернення 15.04.2024)
15. Lemon school. Мова програмування Kotlin [Електронний ресурс]: 2024. – Режим доступу: <https://lemon.school/blog/mova-programuvannya-kotlin> (дата звернення 16.04.2024)
16. Relational Database Management System [Електронний ресурс] – Режим доступу: [https://faculty.utrgv.edu/emmett.tomai/courses/3342/04\\_db/01-sqlite.html](https://faculty.utrgv.edu/emmett.tomai/courses/3342/04_db/01-sqlite.html) (дата звернення 17.03.2024)
17. Shevtsiv N. A., Striuk A. M. Cross platform development vs native development. Kryvyi Rih : Kryvyi Rih National University, 2020. 83 p.
18. Webcase. Розробка мобільних застосунків. [Електронний ресурс]: 2024. – Режим доступу: <https://webcase.com.ua/uk/razrobotka-mobilnyh-prilozhenij/> (дата звернення 10.04.2024)
19. What Is XML (Extensible Markup Language)? Meaning, Elements, and Benefits [Електронний ресурс] – Режим доступу: <https://www.spiceworks.com/tech/tech-general/articles/what-is-xml/> (дата звернення 17.03.2024)
20. Wikipedia. Android Studio. [Електронний ресурс]: 2024. – Режим доступу: [https://uk.wikipedia.org/wiki/Android\\_Studio](https://uk.wikipedia.org/wiki/Android_Studio) (дата звернення: 14.02.2024)
21. Wikipedia. Android. [Електронний ресурс]: 2024. – Режим доступу: [https://uk.wikipedia.org/wiki/Android\\_](https://uk.wikipedia.org/wiki/Android_) (дата звернення: 15.04.2024)
22. Wikipedia. Kotlin (programming language). [Електронний ресурс]: 2024. – Режим доступу: [https://en.wikipedia.org/wiki/Kotlin\\_](https://en.wikipedia.org/wiki/Kotlin_) (перекладено з англійської, дата звернення 16.04.2024)

23. Wikipedia. Mobile app. [Електронний ресурс]: 2024. – Режим доступу: [https://en.wikipedia.org/wiki/Mobile\\_app](https://en.wikipedia.org/wiki/Mobile_app) (перекладено з англійської, дата звернення 14.02.2024)

24. Wikipedia. SQLite [Електронний ресурс]: 2024. – Режим доступу: <https://uk.wikipedia.org/wiki/SQLite> (дата звернення 16.04.2024)

25. XML Basics for Computing [Електронний ресурс] – Режим доступу: <https://medium.com/@dharshithasrimal/xml-basics-in-computingc4f94c5752e1> (дата звернення 17.03.2024)

26. ДСТУ 8604:2015 „Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги”.

27. Зеркалов Д.В. Безпека життєдіяльності та основи охорони праці. Навчальний посібник. К.: «Основа». 2016. – 267 с.

28. Історія, функції, версії та переваги ОС Android. [Електронний ресурс] – Режим доступу: <https://www.spiceworks.com/tech/tech-general/articles/android-os/> (дата звернення 20.04.2024)

29. Огляд рішення реляційної бази даних SQLite. [Електронний ресурс] – Режим доступу: <https://www.ionos.com/digitalguide/websites/web-development/sqlite/> (дата звернення 20.04.2024)

30. Особливості XML [Електронний ресурс] – Режим доступу: <https://www.atatus.com/glossary/xml/> (дата звернення 20.04.2024)

31. Переваги та недоліки кросплатформної та нативної розробки мобільних застосунків [Електронний ресурс] – Режим доступу: <https://merehead.com/ua/blog/cross-platform-native-mobile-development/> (дата звернення 20.04.2024)

32. Різновид мобільних застосунків та особливості їх розробки [Електронний ресурс] – Режим доступу: <https://gahov.com/university/types-of-apps/> – Дата доступу: (дата звернення 15.03.2024)

33. Типи мобільних застосунків. [Електронний ресурс] – Режим доступу: <https://smile-ukraine.com/ua/mobile-apps/mobile-apps-types> (дата звернення 15.03.2024)



34. Що таке Android Studio? [Електронний ресурс] – Режим доступу: <https://www.geeksforgeeks.org/overview-of-android-studio/> (дата звернення 15.03.2024)

35. Що таке Kotlin? [Електронний ресурс] – Режим доступу: [https://www.w3schools.com/kotlin/kotlin\\_intro.php](https://www.w3schools.com/kotlin/kotlin_intro.php) (дата звернення 17.03.2024)

36. Kramar, Taras, et al. "Peculiarities of Augmented Reality Usage in a Mobile Application: the Case of the Ivan Puluj Digital Museum." ІТТАР. 2022.

37. Duda, Oleksii, et al. "Existing City Assessment Systems." 2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT). Vol. 2. IEEE, 2020.

38. Kunanets, Nataliia, et al. "Using React and Fuzzy Expert Systems for Better Travel Experience in Local Route Planning." MoMLeT+ DS. 2023.

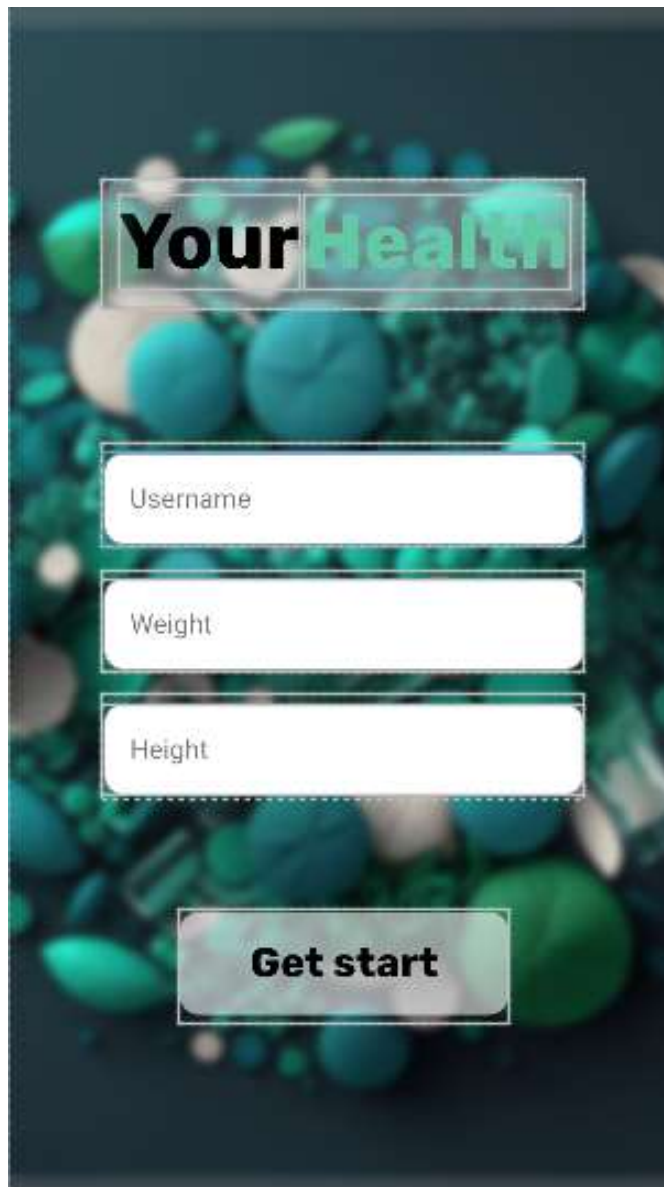
# ДОДАТКИ

**Скрипт, що відповідає за текстові поля для вводу даних  
персонального мобільного застосунку для профілактики цукрового діабету  
та захворювань щитоподібної залози**

```
<com.google.android.material.textfield.TextInputLayout
    style="?attr/textInputOutlinedStyle"
    android:layout_width="292sp"
    android:layout_height="wrap_content"
    android:layout_marginTop="83dp"
    app:boxCornerRadiusBottomStart="12dp"
    app:boxCornerRadiusBottomEnd="12dp"
    app:boxCornerRadiusTopStart="12dp"
    app:boxCornerRadiusTopEnd="12dp"
    app:boxBackgroundColor="@color/white"
    app:hintTextColor="@color/black"
    app:boxStrokeColor = "#85000000"
    android:textColorHint="#85000000">
    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/username"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Username"
        android:maxLines="1"
        android:textColor="@color/black" />
    </com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    style="?attr/textInputOutlinedStyle"
    android:layout_width="292sp"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    app:boxCornerRadiusBottomStart="12dp"
    app:boxCornerRadiusBottomEnd="12dp"
    app:boxCornerRadiusTopStart="12dp"
    app:boxCornerRadiusTopEnd="12dp"
    app:boxBackgroundColor="@color/white"
    app:hintTextColor="@color/black"
    app:boxStrokeColor = "#85000000"
    android:textColorHint="#85000000">
    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/weight"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
```

```
        android:hint="Weight"
        android:inputType="numberDecimal"
        android:maxLines="1"
        android:textColor="@color/black"/>
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    style="?attr/textInputOutlinedStyle"
    android:layout_width="292sp"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    app:boxCornerRadiusBottomStart="12dp"
    app:boxCornerRadiusBottomEnd="12dp"
```

**Зображення першого вікна реєстрації персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози**



**Скрипт, що відповідає за об'єкти для вибору позиції та кнопки  
персонального мобільного застосунку для профілактики цукрового діабету  
та захворювань щитоподібної залози**

```
<RadioGroup
    android:id="@+id/radioGroup"
    android:checkedButton="@+id/radio_button_1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="70dp"
        app:cardCornerRadius="12dp"
        app:cardElevation = "7dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="50sp"
        android:layout_marginStart="10dp"
        android:layout_marginEnd="10dp"
        app:cardBackgroundColor="@color/white">
        <RadioButton
            android:id="@+id/radio_button_1"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="Аналізи вже здані"
            android:textSize="24sp"
            android:textColor="@color/black"
            android:fontFamily="@font/rubik_light"/>
    </androidx.cardview.widget.CardView>

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="70dp"
        app:cardCornerRadius="12dp"
        app:cardElevation = "7dp"
        android:layout_marginTop="20dp"
        android:layout_marginBottom="15dp"
        android:layout_marginStart="10dp"
        android:layout_marginEnd="10dp"
        android:layout_gravity="center_horizontal"
        app:cardBackgroundColor="@color/white">
        <RadioButton
            android:id="@+id/radio_button_2"
```

```
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:text="Аналізи не здані"  
        android:textSize="24sp"  
        android:textColor="@color/black"  
        android:fontFamily="@font/rubik_light"/>  
    </androidx.cardview.widget.CardView>  
</RadioGroup>  
</LinearLayout>  
</LinearLayout>  
<RelativeLayout  
    android:layout_width="match_parent"
```

**Зображення сторінки «fragment\_entrance2» персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози**

**YourHealth**

Перш за все, вам слід здати мінімальний набір аналізів.

Аналізи вже здані

Аналізи не здані

**NEXT**



**Скрипт, що відповідає за поля для введення даних на сторінці  
«fragment\_entrance3» та «fragment\_entrance4» персонального мобільного  
застосунку для профілактики цукрового діабету та захворювань  
щитоподібної залози**

```
<com.google.android.material.textfield.TextInputLayout
    style="?attr/textInputOutlinedStyle"
    android:layout_marginTop="20dp"
    android:layout_marginStart="10dp"
    android:layout_marginEnd="10dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:boxCornerRadiusBottomStart="12dp"
    app:boxCornerRadiusBottomEnd="12dp"
    app:boxCornerRadiusTopStart="12dp"
    app:boxCornerRadiusTopEnd="12dp"
    app:boxBackgroundColor="@color/white"
    app:hintTextColor="@color/black"
    app:boxStrokeColor = "#85000000"
    android:textColorHint="#85000000">
<com.google.android.material.textfield.TextInputEditText
    android:id="@+id/input1" android:layout_width="match_parent"
    android:layout_height="wrap_content" android:hint="ТТГ
    (Тиреотропний гормон)" android:maxLines="1"
        android:inputType="numberDecimal"
        android:textColor="@color/black" />
</com.google.android.material.textfield.TextInputLayout>
    <com.google.android.material.textfield.TextInputLayout
        style="?attr/textInputOutlinedStyle"
        android:layout_marginStart="10dp"
        android:layout_marginEnd="10dp"
        android:layout_marginTop="15dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:boxCornerRadiusBottomStart="12dp"
        app:boxCornerRadiusBottomEnd="12dp"
        app:boxCornerRadiusTopStart="12dp"
        app:boxCornerRadiusTopEnd="12dp"
        app:boxBackgroundColor="@color/white"
        app:hintTextColor="@color/black"
        app:boxStrokeColor = "#85000000"
        android:textColorHint="#85000000">
<com.google.android.material.textfield.TextInputEditText
    android:id="@+id/input2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Т3 вільний (Трийодтиронін)"
    android:maxLines="1"
```

**Скрипт, що відповідає за сторінку на яку буде перенаправлено користувача, якщо він обрав пункт, що аналізи не здано**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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"
xmlns:app="http://schemas.android.com/apk/res-auto"
tools:context=".onboarding.frames.Entrance5"
android:background="#E9E8EC">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="30dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:orientation = "vertical">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:fontFamily="@font/rubik_bold"
    android:text="Your"
    android:textColor="@color/black"
    android:textSize="36sp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="2dp"

    android:fontFamily="@font/rubik_bold"
    android:outlineAmbientShadowColor="@color/black"
    android:text="Health"
    android:textColor="#64C9A5"
    android:textSize="36sp" />
</LinearLayout>
<com.google.android.material.card.MaterialCardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    app:cardCornerRadius="12dp"
```

**Зображення сторінки при незданих аналізах персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози**



## Скрипт, що відповідає за інтерфейс користувача сторінки про нездані аналізи

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="30dp"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:orientation = "vertical">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:fontFamily="@font/rubik_bold"
            android:text="Your"
            android:textColor="@color/black"
            android:textSize="36sp" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="2dp"
            android:fontFamily="@font/rubik_bold"
            android:outlineAmbientShadowColor="@color/black"
            android:text="Health"
            android:textColor="#64C9A5"
            android:textSize="36sp" />
    </LinearLayout>
    <com.google.android.material.card.MaterialCardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        app:cardCornerRadius="12dp"
        app:cardElevation="7dp"
        android:layout_marginTop="20sp"
        app:cardBackgroundColor="@color/white"
        android:layout_marginStart="10dp"
        android:layout_marginEnd="10dp">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_margin="15dp">
            <ImageView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:background="@drawable/info"/>
```

**Скрипт, що відповідає за клас «DataBaseHelper» персонального  
мобільного застосунку для профілактики цукрового діабету та  
захворювань щитоподібної залози**

```

class DataBaseHelper(context: Context): SQLiteOpenHelper(context,
DB_NAME, null, DB_VERSION) {
    companion object {
        private const val DB_NAME = "analyzes.db"
        private const val DB_VERSION = 3
        const val TABLE_THYROID = "thyroid_table"
        const val THYROID_ID = "thyroid_id"
        const val THYROID_STIMULATING_HORMONE =
"thyroid_stimulating_hormone"
        const val T3 = "t3"
        const val T4 = "t4"
        const val TABLE_DIABETS = "diabets_table"
        const val DIABET_ID = "diabet_id"
        const val GLYCATED_HEMOGLOBIN = "glycated_hemoglobin"
        const val HOMA_IR = "homa_ir"
        const val LIPIDOGRAM = "lipidogram"
        const val DATE = "date"
        const val TABLE_WEIGHT = "weight_table"
        const val WEIGHT_ID = "weight_id"
        const val WEIGHT = "weight"
        const val BMI = "bmi"
    }
    override fun onCreate(db: SQLiteDatabase?) {
        val createTable_THYROID = "CREATE TABLE $TABLE_THYROID " +
            "($THYROID_ID INTEGER PRIMARY KEY AUTOINCREMENT, " +
            "$THYROID_STIMULATING_HORMONE REAL, " +
            "$T3 REAL, " +
            "$T4 REAL, " +
            "$DATE TEXT" + ")"
        val createTable_DIABET = "CREATE TABLE $TABLE_DIABETS " +
            "($DIABET_ID INTEGER PRIMARY KEY AUTOINCREMENT, " +
            "$GLYCATED_HEMOGLOBIN REAL, " +
            "$HOMA_IR REAL, " +
            "$LIPIDOGRAM REAL, " +
            "$DATE TEXT" + ")"
        val createTable_WEIGHT = "CREATE TABLE $TABLE_WEIGHT " +
            "($WEIGHT_ID INTEGER PRIMARY KEY AUTOINCREMENT, " +
            "$WEIGHT REAL, " +
            "$BMI REAL, " +
            "$DATE TEXT" + ")"
        val createTable_Name = "CREATE TABLE USER " +
            "(username TEXT, " +
            "height REAL" + ")"
        db?.execSQL(createTable_THYROID)
        db?.execSQL(createTable_DIABET)
        db?.execSQL(createTable_WEIGHT; db?.execSQL(createTable_Name)}

```

**Скрипт, що відповідає за функціональну частину сторінки персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози з результатами**

```

var db = MyDbAdapter(this)
    var diabet = db.get_diabet()[db.get_diabet().size-1]
    var thyroid = db.get_thyroid()[db.get_thyroid().size-1]
    var textDiabet = findViewById<TextView>(R.id.inform_2)
    var textThyroid = findViewById<TextView>(R.id.inform_1)
    if (diabet.glycated_hemoglobin in 111.0..141.0 &&
diabet.homa_ir in 4.1..6.1) {
        textDiabet.text = "Все в нормі!"
    } else if(diabet.glycated_hemoglobin>141 && diabet.homa_ir
in 4.1..6.1) {
        textDiabet.text = "Ви харчуєтеся дуже великою кількістю
солодкого та випічки."
    } else if(diabet.glycated_hemoglobin in 111.0..141.0 &&
diabet.homa_ir>6.1){
        textDiabet.text = "Діабет ще не діагностовано, але вже
будуть проблеми із зайвою вагою. Підшлункова залоза працює на межі."
    } else if (diabet.glycated_hemoglobin>141 &&
diabet.homa_ir>6.1){
        textDiabet.text = "Слід піти до ендокринолога. Скоріше
за все, діабет 2 буде діагностовано."
    } else { textDiabet.text = "Слід звернутися до ендокринолога"}
    if(thyroid.hyroid_stimulating_hormone in 0.29..4.9 &&
thyroid.t3 in 4.0..8.3 && thyroid.t4 in 10.3..24.5){
        textThyroid.text = "Все в нормі!"
    } else if(thyroid.hyroid_stimulating_hormone>4.9 &&
thyroid.t3 in 4.0..8.3 && thyroid.t4 in 10.3..24.5){
        textThyroid.text = "Щитовидна залоза працює собі на
шкоду. Поки що проблем немає."
    } else if(thyroid.hyroid_stimulating_hormone>4.9 &&
thyroid.t3<4.0 && thyroid.t4<10.3){
        textThyroid.text = "Гіпотиреоз. Рекомендується здати
додатковий аналіз і записатися до ендокринолога."
    } else if(thyroid.t3>8.3 && thyroid.t4>24.5){
        textThyroid.text = "Гіпертиреоз. Нічого можна не здавати,
обов'язково слід записатися до ендокринолога."
    } else if(thyroid.t4 in 10.3..24.5 && thyroid.t3<4.0){
        textThyroid.text = "Необхідно здати додаткові
дослідження. Дивитися на йод, і мікроелементи, і вітамін В6."
    } else {
        textThyroid.text = "Слід звернутися до ендокринолога"
    }
    findViewById<Button>(R.id.button5).setOnClickListener {
        var intent = Intent(this, MainPage::class.java)
        onBoardingfinished()
        startActivity(intent)
    }
}

```

**Скрипт, що відповідає за інтерфейс користувача головної сторінки  
персонального мобільного застосунку для профілактики цукрового діабету  
та захворювань щитоподібної залози**

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="30dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:orientation = "vertical">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:fontFamily="@font/rubik_bold"
                android:text="Your"
                android:textColor="@color/black"
                android:textSize="36sp" />
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="2dp"
                android:fontFamily="@font/rubik_bold"
                android:outlineAmbientShadowColor="@color/black"
                android:text="Health"
                android:textColor="#64C9A5"
                android:textSize="36sp" />
        </LinearLayout>
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:gravity="right">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:fontFamily="@font/rubik_bold"
                android:text="Привіт, "
                android:textColor="@color/black"
                android:textSize="24sp" />
            <TextView
                android:id="@+id/username"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="2dp
```

**Зображення користувацького інтерфейсу головної сторінки  
персонального мобільного застосунку для профілактики цукрового  
діабету та захворювань щитоподібної залози**





**Скрипт, що відповідає за нижнє меню персонального мобільного застосунку для профілактики цукрового діабету та захворювань щитоподібної залози**

```

<FrameLayout
    android:id="@+id/standard_bottom_sheet"
    style="@style/Widget.Material3.BottomSheet"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <FrameLayout
        android:id="@+id/bottom_sheet_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/white">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_gravity="center_horizontal"><View
                android:layout_width="90dp"
                android:layout_height="4dp"
                android:background="@drawable/rectangle"
                android:layout_gravity="center_horizontal"
                android:layout_marginTop="10dp"/>
    <com.google.android.material.card.MaterialCardView
        android:id="@+id/add_thyroid"
        android:layout_width="328dp"
        android:layout_height="66dp"
        android:layout_marginTop="20dp"
        app:cardCornerRadius="12dp"
        app:cardBackgroundColor="@color/white"
        android:backgroundTint="#D9D9D9"><LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal"
            android:layout_marginLeft="5dp">
            <ImageView
                android:layout_width="50dp"
                android:layout_height="50dp"
                android:layout_gravity="center"
                android:src="@drawable/thyroid"/>
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="Щитовидна залоза"
                android:textColor="@color/black"
                android:textSize="20dp"
            android:fontFamily="@font/rubik_bold"
            android:gravity="center"/>

```

**Скрипт, що відповідає за функціональну частину основної сторінки  
персонального мобільного застосунку для профілактики цукрового діабету  
та захворювань щитоподібної залози**

```
class ModalBottomSheet : BottomSheetDialogFragment() {
    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        val view = inflater.inflate(R.layout.bottom_sheet,
            container, false)
        val add_thyroid =
            view.findViewById<MaterialCardView>(R.id.add_thyroid)
        val add_diabet =
            view.findViewById<MaterialCardView>(R.id.add_diabet)
        val add_weight =
            view.findViewById<MaterialCardView>(R.id.add_weight)
        add_thyroid.setOnClickListener {
            val intent = Intent(requireActivity(),
                Thyroid_data::class.java)
            startActivity(intent)
        }
        add_diabet.setOnClickListener {
            val intent = Intent(requireActivity(),
                Diabet_data::class.java)
            startActivity(intent)
        }
        add_weight.setOnClickListener {
            val intent = Intent(requireActivity(),
                Weight_data::class.java)
            startActivity(intent)
        }
        return view    }
    companion object {
        const val TAG = "ModalBottomSheet"
    }
}

class MainPage : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main_page)
        var db = MyDbAdapter(this)
        var button = findViewById<Button>(R.id.add_button)
        val modalBottomSheet = ModalBottomSheet()
        button.setOnClickListener {
            modalBottomSheet.show(supportFragmentManager,
                ModalBottomSheet.TAG)
        }
    }
}
```

## Скрипт, що відповідає за графічний інтерфейс сторінки з інформацією про певні показники

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:layout_marginTop="30dp"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:fontFamily="@font/rubik_bold"
                android:text="Your"
                android:textColor="@color/black"
                android:textSize="36sp"/>
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginLeft="2dp"
                android:fontFamily="@font/rubik_bold"

android:outlineAmbientShadowColor="@color/black"
        android:text="Health"
        android:textColor="#64C9A5"
        android:textSize="36sp" />
    </LinearLayout>

    <com.google.android.material.card.MaterialCardView
        android:layout_width="328dp"
        android:layout_height="140dp"
        android:layout_marginTop="20dp"
        app:cardCornerRadius="12dp"
        app:cardElevation="7dp"
        android:layout_gravity="center"
        app:cardBackgroundColor="@color/white">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:gravity="center">
            <TextView
                android:layout_width="wrap_content"
```

**Зображення сторінки для додавання нових даних**

**YourHealth**

**Діабет**

Глікований гемоглобін

НОМА-ІR

Ліпідограма

**SAVE**

**Скрипт, що відповідає за графічний інтерфейс сторінки з інформацією про  
здачу аналізів персонального мобільного застосунку для профілактики  
цукрового діабету та захворювань щитоподібної залози**

```
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".Information_activity"
  android:background="#E9E8EC">
  <LinearLayout android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="30sp"
    android:layout_marginLeft="20sp"
    android:layout_marginRight="20sp"
    android:orientation = "vertical">
    <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:orientation="horizontal">
      <TextView  android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="@font/rubik_bold"
        android:text="Your"
        android:textColor="@color/black"
        android:textSize="36sp"/>
      <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="2dp"
        android:fontFamily="@font/rubik_bold"
        android:outlineAmbientShadowColor="@color/black"
        android:text="Health"
        android:textColor="#64C9A5"
        android:textSize="36sp" />
    </LinearLayout>
    <com.google.android.material.card.MaterialCardView
      android:layout_width="340dp"
      android:layout_height="wrap_content"
      app:cardCornerRadius="12dp"
      app:cardElevation="7dp"
      android:layout_gravity="center"
      android:layout_marginTop="50sp"
      app:cardBackgroundColor="@color/white">
      <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
```