КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

Кваліфікаційна робота
на здобуття освітнього рівня магістра

за спеціальністю 121 Інженерія програмного забезпечення
на тему:

РОЗРОБКА ANDROID ЗАСТОСУНКУ ДЛЯ ВІДСТЕЖЕННЯ ФІЗИЧНИХ ВПРАВ У РІЗНОМАНІТНИХ УМОВАХ

Виконав студент 2-го курсу магістратури
Владислав ТОЧАНЕНКО _____
(підпис)

Науковий керівник:
доцент
Костянтин ЖЕРЕБ _____
(підпис)

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Роботу розглянуто й допущено до захисту на засіданні кафедри інтелектуальних програмних систем
«__»_____2023 р.,
протокол №___
Завідувач кафедри
Олександр ПРОВОТАР _____
(підпис)

Київ – 2023

РЕФЕРАТ

Обсяг роботи – 60 сторінок, 22 використаних джерела, 19 рисунків, 1 таблиця.

ANDROID ЗАСТОСУНОК, ANDROID JETPACK, MATERIAL DESIGN, МАКЕТИ, KOTLIN.

Об’єктом випускної кваліфікаційної роботи магістра є розробка застосунків для операційної системи Android, який відповідає вимогам до сучасних застосунків як до архітектури, так і до графічного користувацького інтерфейсу. Розробка ведеться із використанням сучасних інструментів для розробки застосунків для ОС Android.

Мета випускної кваліфікаційної роботи магістра полягає у створені застосунку для відслідковування виконання фізичних вправ у різноманітних умовах. Застосунок повинен бути розробленим для операційної Android, використовуючи сучасні інструменти та бібліотеки для створення застосунків для ОС Android, повинен задовольняти вимогам дизайну сучасних застосунків для ОС Android. Застосунок повинен надавати користувачеві необхідний функціонал для створення, перегляду та редагування фізичних вправ та зарядок, що складаються з комплексів вправ, а також надавати користувачеві доступ до перегляду статистики за процесом виконання вправ. Застосунок повинен бути розміщеним у відкритому доступі на веб сервісі для спільної розробки програм GitHub.

Інструменти розробки: мова програмування Kotlin, мова розмітки XML, мова зборки проєктів Groovy Kotlin DSL, колекція бібліотек Android Jetpack, комплекс бібліотек Android Jetpack, система контролю версій Git, веб сервіс GitHub, сервіс розробки макетів Figma, сервіс розробки графіків draw.io, бібліотека Material 3 Design Kit.

Результати роботи: у ході роботи над випускною кваліфікаційною роботою було виконано огляд сучасних інструментів та методів розробки застосунків для операційної системи Android. Розроблено макети застосунку, розроблена його архітектура та модель бази даних. Розроблено застосунок для операційної системи

Android, що дозволяє користувачеві відслідковувати виконані фізичні вправи у будь-яких умовах.

Розроблений застосунок може використовуватись як застосунок для відслідковування виконання фізичних вправ, так і для ознайомлення з процесом сучасним процесом розробки застосунків для операційної системи Android за рахунок розміщення програмного коду застосунку у відкритому доступі.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

ОС – операційна система;

DSL (Domain-Specific Language) – мова програмування, яка спеціалізована для певної прикладної області;

JVM (Java Virtual Machine) – віртуальна машина мови програмування Java;

UML (Unified Modeling Language) – уніфікована мова моделювання;

XML (Extensible Markup Language) – мова розмітки для збереження даних.

ВСТУП

Оцінка сучасного стану об’єкта розробки. Операційна система Android є найпопулярнішою операційною системою для смартфонів у світі. За рахунок цього розробка застосунків для операційної системи Android залишається однією з найпопулярніших сфер розробки програмного забезпечення. Постійний розвиток архітектури та функціоналу системи Android, а також покращення системи дизайну Material Design, сприяють появі нових застосунків, що втілюють у життя проєкти, які мають сучасний дизайн та є зручними для користування.

За останні роки процес розробки застосунків для операційної системи Android зазнав значних змін. Було розроблено низку бібліотек, що роблять архітектуру застосунків більш гнучкою та безпечною, було розроблено нову систему дизайну Material Design 3.0, що робить користування застосунками більш інтерактивним та зрозумілим для будь-якого користувача. Через це велика кількість застосунків потребують змін, або повного переписування задля відповідності сучасним вимогам щодо реалізації застосунків для операційної системи Android.

Актуальність роботи та підстави для її виконання. За останні роки велика кількість користувачів була змушена змінити свій спосіб життя через карантин, що був пов’язаний з пандемією COVID-19, та через повномасштабне вторгнення Росії на територію України. Через наведені причини велика кількість користувачів мобільних пристроїв почала шукати застосунки, які б дозволили відслідковувати виконання фізичних вправ у будь-яких умовах, адже недостатня кількість фізичних вправ можуть привезти до фізичних та психічних захворювань. Користувачам необхідно мати застосунок, який працював би без під’єднання до мережі інтернет, надавав би зручний інтерфейс до відслідковування виконаних фізичних вправ, та надавав інтерфейс для перегляду виконаних фізичних вправ.

Наразі існують застосунки, що дозволяють користувачам відслідковувати виконані фізичні вправи, проте жоден з цих застосунків не дає зручний інтерфейс,

який працював би без підключення до мережі інтернет та був би безкоштовним для використання.

Мета й завдання роботи. Метою випускної кваліфікаційної роботи є реалізація власного застосунку для відслідковування виконання фізичних вправ у будь-яких умовах. Для реалізації застосунку необхідно використовувати сучасні бібліотеки та принципи розробки застосунків для операційної системи Android. Для досягнення мети було поставлено такі завдання:

- ознайомитись з сучасними інструментами розробки застосунків для операційної системи Android;
- дослідити наявні застосунки з функціоналом відслідковування виконання фізичних вправ;
- розробити макет графічного інтерфейсу користувача;
- навести функціональні вимоги до застосунку та розробити діаграму прецедентів;
- розробити модель бази даних застосунку;
- розробити застосунок для ОС Android, що відповідає функціональним вимогам та моделі бази даних;
- провести відповідність застосунку до функціональних вимог;
- опублікувати застосунок в магазині застосунків.

Об’єкт, методи й засоби розробки. Об’єктом випускної кваліфікаційної роботи магістра є розробка застосунків для операційної системи Android, який відповідає вимогам до сучасних застосунків як до архітектури, так і до графічного користувацького інтерфейсу. Розробка ведеться із використанням сучасних інструментів для розробки застосунків для ОС Android. Застосунок має такі функціональні можливості, як створення, перегляд та зміна вправ, створення, перегляд та зміна зарядок, що складаються з комплексів вправ, а також перегляд статистики виконання фізичних вправ.

Для розробки застосунку була обрана розробка застосунку на мові програмування Kotlin у інтегрованому середовищі програмування Android Studio.

Допоміжними інструментами у процесі розробки були Git, Figma, Draw.io, колекція бібліотек Android Jetpack, бібліотека графічного інтерфейсу користувача Material Design та інші інструменти.

Можливі сфери застосування. Розроблений застосунок можна використовувати будь-де для відслідковування виконаних фізичних вправ. Таким чином користувач зможе використовувати застосунок для покращення свого фізичного стану та краще слідкувати за процесом виконання фізичних вправ.

Проєкт знаходиться у відкритому доступі, що дає змогу стороннім розробникам доповнювати проєкт, або створювати свій проєкт на основі розробленого. За рахунок того, що під час розробки проєкту використовувались найсучасніші методи та інструменти розробки застосунків для операційної системи Android, розроблений проєкт також можна використовувати як навчальний проєкт для огляду сучасних технологій.

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМИ ВІДСЛІДКОВУВАННЯ ВИКОНАННЯ ФІЗИЧНИХ ВПРАВ

1.1 Огляд існуючих застосунків

Наразі наявна велика кількість застосунків для відслідковування виконання фізичних вправ. Проте кожен з них має низку недоліків, які роблять процес користування застосунком не надто зручним або корисним досвідом. Застосунок для відслідковування виконання фізичних вправ повинен надавати користувачеві швидкий та зручний метод записування вправ. Розглянемо найпопулярніші з наявних застосунків, що мають функціонал, схожий на запис тренувань та вправ.

1.1.1 Застосунок Fitbod

Застосунок Fitbod [1] має сучасний дизайн, дозволяє фільтрувати вправи за м’язами, що тренуються під час виконання вправи. У застосунку наявний темний інтерфейс. Користувач має можливість передивлятись історію виконання вправ та зарядок.

Найбільшими мінусами такого застосунку є модель монетизації та робота застосунка без під’єднання до мережі Інтернет. Користувач має можливість записати тільки шість зарядок, після чого необхідно щомісяця сплачувати фіксовану суму. Розробники застосунка Fitbod ставляться зневажливо до своїх користувачів, бо не надають інформацію про суму, яку саме необхідно буде сплачувати щомісяця для отримання доступу до запису більшої кількості зарядок. Також застосунок не працює без під’єднання до мережі Інтернет, що робить його користування в умовах застосування віялових вимкнень світла. Застосунок не надає жодного контенту, що оновлюється, тому відсутні будь-які причини для наявності підписки у застосунку Fitbod.

Зовнішній вигляд застосунку наведено на рисунку 2.1.

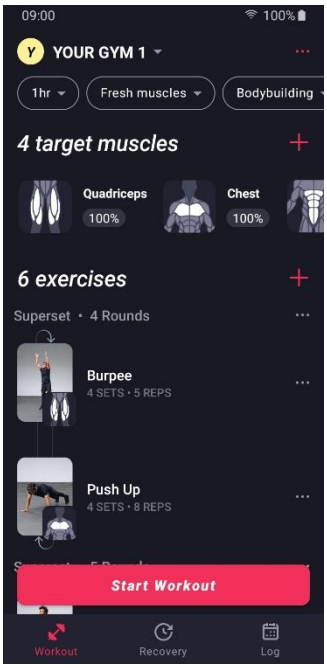

Рисунок 2.1 – Зовнішній вигляд застосунку Fitbod

1.1.2 Застосунок GymKeeper

Застосунок GymKeeper [2] має застарілий та перевантажений дизайн, який виконаний за стандартами дизайну Material Design 2015 року. Велика кількість елементів графічного інтерфейсу користувача не мають чіткого виділення серед інших, що робить навігацію застосунком незручною для користувача. Так само, як і застосунок Fitbot, застосунок GymKeeper обмежує кількість зарядок, які користувач може записати, проте у цьому застосунку замість підписки користувач повинен заплатити 350 гривень одноразово для розблокування повного доступу до застосунку. Застосунок не відрізняється ані зовнішнім видом, ані функціоналом від інших подібних застосунків.

Зовнішній вигляд застосунку наведено на рисунку 2.2.

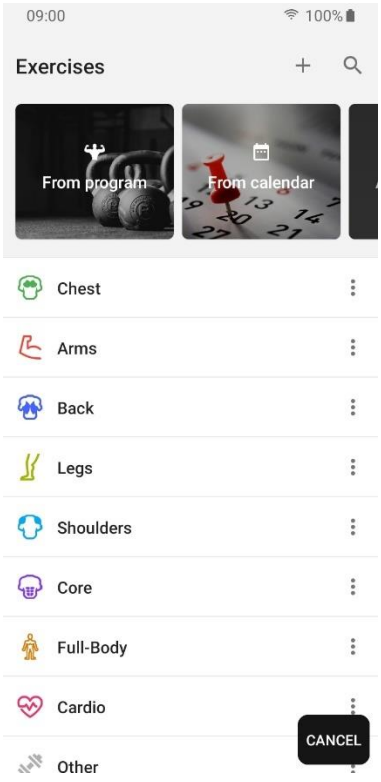

Рисунок 2.2 – Зовнішній вигляд застосунку GymKeeper

1.1.3 Застосунок GymRun

Застосунок RymRun [3] має сучасний дизайн, темну тему інтерфейсу. Проте інтерфейс занадто складний для розуміння, що робить користування застосунком незручним та збільшує час користування застосунком.

Так само як і застосунки Fitbot чи GymKeeper, функціонал застосунку обмежений, поки користувач не придбає повну версію застосунку. На відміну від інших застосунків, кількість записаних вправ та зарядок не обмежена, проте обмежений інший функціонал, який є необхідним під час відслідковування виконаних фізичних вправ. До такого функціоналу відноситься перегляд статистики виконаних вправ та зарядок. Щоб отримати доступ до повного функціоналу необхідно сплачувати по 90 гривень на місяць. Застосунок не надає жодного контенту, що оновлюється, тому відсутні будь-які причини для наявності підписки у застосунку GymRun.

Зовнішній вид застосунку наведено на рисунку 2.3.

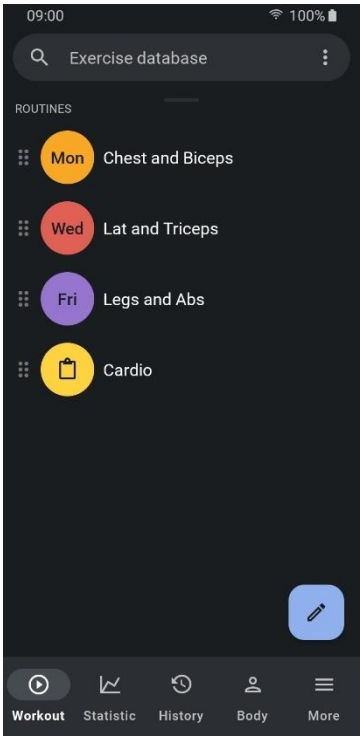

Рисунок 2.2 – Зовнішній вигляд застосунку GymRun

1.1.4 Застосунок Hevy

Застосунок Hevy [4] має сучасний мінімалістичний інтерфейс та темну тему інтерфейсу. Слід зауважити, що дизайн застосунку не відповідає сучасним вимогам до застосунків, що розробляються для операційної системи Android, адже дизайн застосунку Hevy виконаний із дотриманням вимог до графічного інтерфейсу користувача застосунків, що розробляються для операційної системи iOS. Функціонал застосунку не надає можливості користувачеві відслідковувати виконання фізичних вправ. Перевагою застосунка Hevy є наявність функціоналу соціальної мережі, що робить користування застосунком більш інтерактивним.

Одним з мінусів цього застосунку є наявність обмеження кількості зарядок. Для розблокування більшої кількості функціоналу, користувач повинен сплачувати по 580 гривень на рік, або одноразово 2600 гривень. Ціна занадто висока, якщо враховувати кількість функціоналу, який надає застосунок Hevy. Проте з урахуванням того, що у застосунку присутній функціонал соціальної мережі, що

дозволяє користувачеві поділитись з друзями прогресом виконання фізичних вправ, використання моделі монетизації у виді підписки цілком обґрунтована.

Зовнішній вид застосунку наведено на рисунку 2.4.

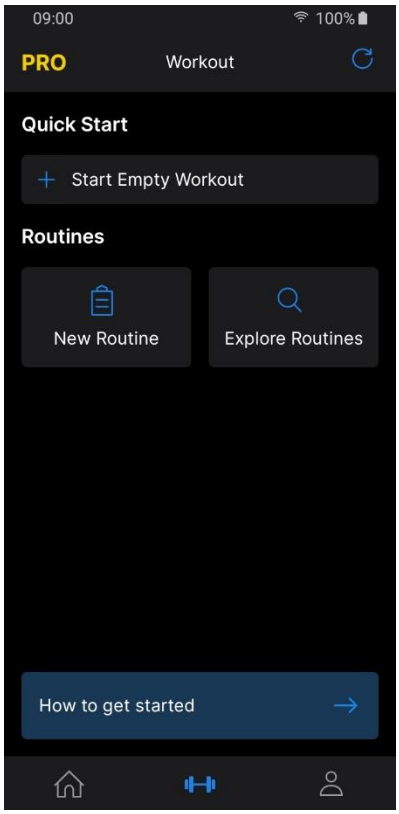

Рисунок 2.4 – Зовнішній вигляд застосунку Hevy

1.1.5 Застосунок Training Diary

Застосунок Training Diary [5] є застосунком із найбільш складним та застарілим графічним інтерфейсом користувача із усіх розглянутих застосунків. Цей застосунок має дуже заплутаний інтерфейс, що робить користування застосунком незручним та повільним. Дизайн застосунку порушує сучасні принципи дизайну Material Design. До таких порушень можна віднести порушення щодо відстаней між елементами, розмір іконок у списках, неправильне використання верхніх навігаційних панелей, де у верхній навігаційній панелі присутні елементи користувацького інтерфейсу, що виконують функціонал редагування вправи, відсутні позначки сторінок з пустим наповненням та багато іншого. Цей застосунок є єдиним застосунком, що має рекламу та розділ з купонами

у різноманітних магазинах Європи. Застосунок Training Diary має обширний функціонал, проте значна частина цього функціоналу також потребує купівлі платної версії застосунку. Цей застосунок є найдешевшим із розглянутих – для розблокування повного функціоналу застосунку необхідно одноразово придбати повну версію за 120 гривень.

Зовнішній вид застосунку наведено на рисунку 2.5.

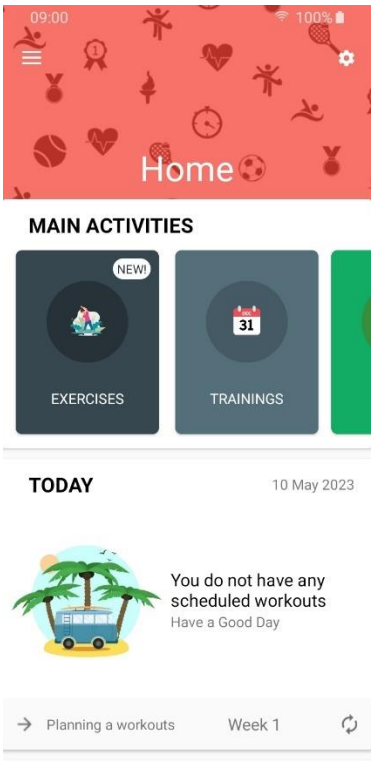

Рисунок 2.5 – Зовнішній вигляд застосунку Training Diary

1.1.6 Застосунок FitHero

Застосунок FitHero [6] має найбільш привабливий та мінімалістичний інтерфейс з розглянутих застосунків. Присутні як світла, так і темна теми інтерфейсу. Цей застосунок дає можливість відслідковувати виконання вправ та переглядати історію їх виконання. Мінусом є відсутність можливості групувати вправи у зарядки. Певна частина функціоналу застосунка так само є прихованою за обов’язковою оплатою. Так само, як і застосунок GymRun, застосунок FitHero має тип підписки. Для отримання доступу до повного функціоналу застосунку, необхідно оформити підписку на 500 гривень на рік. Так само як і інші застосунки,

застосунок FitHero не має обґрунтованого використання типу підписки, бо не має контенту, що постійно оновлюється.

Зовнішній вид застосунку наведено на рисунку 2.6.

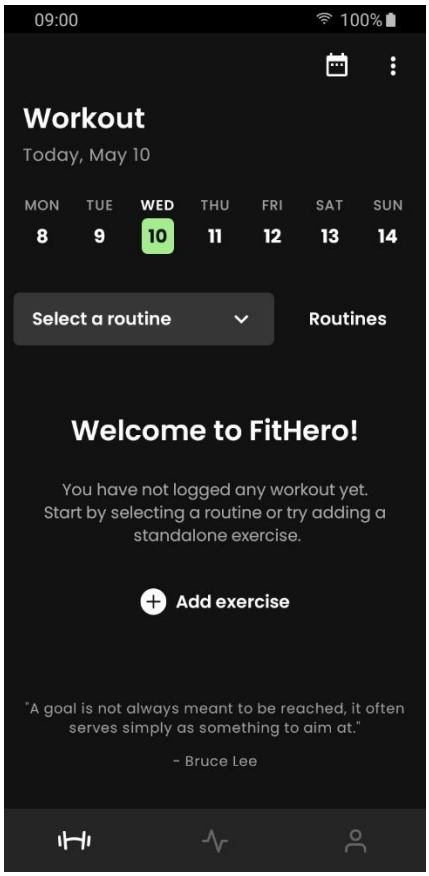

Рисунок 2.6 – Зовнішній вигляд застосунку FitHero

1.2 Порівняння наведених застосунків

Розглянуті застосунки, які дозволяють відслідковувати виконання фізичних вправ, не розроблені із врахуванням зручності графічного інтерфейсу користувача, який дозволить швидко і зручно записати виконані вправи і відслідковувати їх кількість.

Кожен з розглянутих застосунків має певні обмеження, що передбачають розблокування за допомогою одноразовими покупками або підписками, проте вони не є обґрунтованими, за виключенням застосунку Hevy. Зазвичай застосунки, які поставляються із моделлю підписки, надають користувачеві сервіс, що має динамічні дані. До таких сервісів відносяться YouTube Music, Netflix, Pocket Casts

та інші. Якщо ж брати за увагу наявні застосунки для відслідковування вправ, окрім застосунку Hevy, то використання моделі підписки не є необхідним, адже такі застосунки лише зберігають дані користувача та не надають йому оновлені. Наприклад, застосунок Hevy реалізовує функціонал соціальної мережі, який потребує наявності серверної частини, що повинна бути розміщена на віддаленому сервері. Для інших застосунків, можливо, наявність моделі монетизації у виді підписки може бути обґрунтована, якщо для їх функціоналу необхідно використовувати платний прикладний програмний інтерфейс, наприклад, для отримання списку вправ, їх опису і інструкцій з виконання. Проте такий список вправ можна легко наповнити за допомогою особистого досвіду, або навіть отримати з відкритих джерел в мережі Інтернет.

Більшість програм потребують створення зарядок, які складаються з комплексів вправ, що збільшує час користування застосунком і зменшує його користь для користувача. Таким чином якщо користувач захоче відслідкувати лише одну вправу, він повинен буде витратити більше часу від необхідного. Таким чином необхідно розробити застосунок, який надаватиме можливість одночасно записувати як окремі вправи, так і їх комплекси.

Також однією з основних проблем наявних застосунків для відслідковування вправ є відсутність української локалізації. Зазвичай студенти в Україні мають високий рівень володіння англійською мовою технічного рівня. Проте для користування застосунками для відслідковування фізичних вправ, що мають лише англійську локалізацію, багатьом користувачам доведеться перекладати велику кількість слів, які пов’язані з назвами груп м’язів людини, українською мовою.

Також жоден з наведених застосунків не надає можливості користувачеві обрати графічне зображення структури тіла. Таким чином будь-які малюнки у застосунку, що схематично зображують м’язи, які тренуються під час виконання вправ, мають лише один вигляд. Зазвичай, це структура м’язів дорослої людини чоловічої статі. Застосунком для відслідковування виконання фізичних вправ можуть користуватись користувачі різної статі та різних вікових груп. Тому необхідно надавати користувачеві можливість обрати зовнішній вид схематичного

зображення м’язів, щоб забезпечити відповідність схем у застосунку до структури тіла користувача.

Повне порівняння застосунків наведено у таблиці:

	Fitbod	GymKeeper	GymRun	Hevy	Training Diary	FitHero
Українська локалізація	Відсутня	Відсутня	Відсутня	Відсутня	Відсутня	Відсутня
Модель використання	Підписка	Одноразовий платіж	Підписка	Одноразовий платіж або підписка	Одноразовий платіж	Підписка
Коштовність покупки	Приховано від користувача	350 грн.	90 грн. / міс.	580 грн. / рік або 2600 грн.	120 грн.	500 грн. / рік
Записування вправ	Присутня	Присутня	Присутня	Присутня	Присутня	Присутня
Записування зарядок	Присутня	Присутня	Присутня	Присутня	Присутня	Відсутня
Зображення м’язів	Присутня	Відсутня	Відсутня	Відсутня	Відсутня	Присутня
Дизайн	Сучасний	Застарілий	Сучасний	Сучасний	Застарілий	Сучасний
Обирання структури тіла	Відсутня	Відсутня	Відсутня	Відсутня	Відсутня	Відсутня

РОЗДІЛ 2. ОБҐРУНТУВАННЯ ЗАСОБІВ РОЗРОБКИ

2.1 Мова програмування Kotlin

Однією з мов програмування, яка стрімко набирає популярність щороку, є мова програмування Kotlin. Kotlin є мовою програмування зі строгою типізацією, що компілює вихідний код у код, що може бути запущеним на віртуальній машині Java. Мова програмування Kotlin розроблена компанією JetBrains у 2011 році, а у 2015 стала основною мовою розробки застосунків для операційної системи Android [7].

Для розробки застосунка, що дає можливість відслідковувати виконані фізичні вправи, було обрано використання саме мови програмування Kotlin, адже велика кількість сучасної інформації, що стосується розробки Android застосунків, написані з використанням мови програмування Kotlin.

2.2 Система збирання проєктів Gradle Kotlin DSL

Gradle є скриптовою системою, що дозволяє компілювати, збирати, запускати та тестувати проєкти, а також управляти залежностями, які використовуються у застосунку, розподіляючи необхідні ресурси та залежності між різними задачами, які мають на меті різні види роботи з проєктом. До версії Gradle 5.0 основною мовою для написання скриптів була Groovy, проте згодом з’явилась можливість використовувати мову програмування Kotlin, що значно прискорює та полегшує процес розробки застосунків для операційної системи Android. Для системи Gradle було розроблено низку бібліотек Kotlin, що мають назву Kotlin DSL [8]. За рахунок того, що Kotlin, на відміну від Groovy, є мовою зі строгою типізацією, інтегроване середовище розробки може надавати розробнику наступні переваги під час написання Gradle скриптів: автозавершення виклику функцій та змінних, швидка

документація одночасно із використанням функцій, зручний рефактор коду скриптів та інше.

Під час розробки застосунка для відслідковування виконання фізичних вправ було вирішено використовувати систему Gradle на мові програмування Kotlin DSL, що виконує задачі компіляції програмного коду застосунка та збирання програмного коду застосунка у єдиний файл для встановлення на пристрій користувача. Система Gradle дозволяє значно прискорити процес розробки застосунку та робить процес розробки більш інтерактивним та гнучким.

2.3 Інтегроване середовище розробки Android Studio

Android Studio є інтегрованим середовищем розробки, що є основним інструментом для розробки застосунків для операційної системи Android. Android Studio розробляється компанією Google та є надбудовою над інтегрованим середовищем розробки IntelliJ IDEA [9], яке використовується для розробки програм на мовах програмування, вихідний код котрих запускається на віртуальній машині Java. Android Studio надає значну кількість інструментів, що покращують та прискорюють процес розробки застосунків для операційної системи Android. До таких інструментів можна віднести інструмент записування подій, що відбуваються на пристрої під час відтворення Android застосунку. До таких інструментів також відноситься інструмент моніторингу продуктивності виконання Android застосунку, а також відслідковування змін у базі даних, що використовується у виконуваному Android застосунку.

Також однією з основних переваг інтегрованого середовища розробки Android Studio перед іншими є наявність віртуальних пристроїв із операційною системою Android. Розробник може використовувати будь-яку кількість таких пристроїв з конфігураціями, які допоможуть провести тестування застосунку на пристроях з різними параметрами, без необхідності мати фізичні копії таких пристроїв. Наприклад, розробник може перевірити зовнішній вигляд та роботу

застосунку, що розробляється, на пристроях з різним розширенням дисплею, або з різною швидкістю підключення до мережі Internet.

Для розробки застосунку для відслідковування виконання фізичних вправ було обрано інтегроване середовище розробки Android Studio, що є кращим інструментом для розробки Android застосунків та надає обширний функціонал для прискорення процесу розробки.

2.4 Механізм з’єднання інтерфейсу та програмної реалізації

Кожному елементу, що відображається на графічному інтерфейсі користувача, необхідно надати ідентифікатор, що дає можливість однозначно визначити елемент для роботи з ним. Для роботи з елементом графічного інтерфейсу користувача необхідно отримати у коді програми його програмний об’єкт. Для цього використовується механізм отримання елементів за допомогою їх ідентифікатора. Проте у такому механізмі є недолік, через який неможливо використовувати один і той самий ідентифікатор для елементів, що мають однаковий функціонал та вигляд на різних екранах одночасно.

До сучасного механізму отримання елементу у коді програми відноситься механізм, що називається «View Binding» [10]. Цей механізм передбачає створення окремого класу, який має список програмних об’єктів елементів, що знаходяться на одному з екранів інтерфейсу користувача. Слід зауважити, що для кожного з екранів інтерфейсу користувача створюється окремий клас зі списком програмних об’єктів. Таким чином з’являється можливість надання елементам з однаковим функціоналом та зовнішнім видом однакових ідентифікаторів, що значно спрощує процес розробки Android застосунків.

Було вирішено використовувати механізм View Binding, оскільки такий механізм значно прискорить розробку застосунку для відслідковування виконання фізичних вправ, покращить його архітектуру, і є рекомендованим до використання під час розробки усіх сучасних застосунків для операційної системи Android.

2.5 Бібліотека елементів графічного інтерфейсу Material Components

Material Components є бібліотекою елементів графічного інтерфейсу користувача, яка реалізовує список елементів інтерфейсу користувача, що задовольняють усім правилам та вказівкам Material Design [11]. Цей дизайн розроблений компанією Google для операційної системи Android. Ідеєю дизайну є мінімалізм, що передбачає велику кількість анімацій для покращення інтерактивності застосунку.

Для розробки інтерактивних та сучасних застосунків для операційної системи Android та інших операційних систем, компанією Google розроблена низка бібліотек. Такі бібліотеки дозволяють розробнику швидко розробляти застосунки, що слідкують всім правилам Material дизайну. Для розробника надається низка компонентів, які дозволяють швидко розробити інтерфейс користувача, адже у бібліотеці наявні усі необхідні компоненти для розробки простих графічних інтерфейсів користувача. До таких елементів відносяться такі елементи, як картки, кнопки, списки, текстові поля та інші. Більш складні елементи користувацького інтерфейсу розробникам необхідно розробляти власноруч. Такими елементами можуть бути графіки, інтерактивні об’єкти та інше.

Для розробки застосунку для відслідковування виконання фізичних вправ було вирішено використовувати бібліотеку Material Design, яка полегшить та прискорить процес розробки застосунку. За допомогою такої бібліотеки був розроблений сучасний інтерфейс користувача, який відповідає вимогам сучасного дизайну мобільних застосунків.

2.6 Механізм серіалізації у мові програмування Kotlin

Серіалізація об’єктів є приведення об’єктів до виду, що дозволяє їх зберігати у базах даних або файлах у строковому або бінарному представленні [12]. Кожен об’єкт у будь-якій мові програмування зберігається у оперативній пам’яті комп’ютера на рівні виконання програмного коду програми. Таким чином, для

збереження даних необхідно перетворити їх у вид, який обробляється базами даних або файлами.

У той самий час для отримання об’єктів з баз даних або файлів, необхідно скористатись механізмом десеріалізації. Механізм десеріалізації дозволяє розробнику швидко і зручно отримати об’єкт з рядкового або бінарного представлення для його подальшого використання у процесі розробки застосунка.

У застосунку для відслідковування виконання фізичних вправ механізми серіалізації та десеріалізації використовується для роботи з об’єктами, що зберігаються у базі даних застосунку, та для передачі таких об’єктів між екранами застосунку.

2.7 Функції високого порядку у мові програмування Kotlin

У мові програмування Kotlin будь-яка функція є об’єктом окремого класу. Таким чином будь-яку функцію розробник може використовувати як звичайний об’єкт, тобто функцію можна передавати як параметр інших функцій та класів, що дає змогу виконувати різні дії під час виконання застосунку в залежності від різних умов. Іншими словами, можна розробляти функціонал графічних інтерфейсів користувача, що виконуватиме певні дії в залежності від вхідних параметрів, які визначаються поза межами функції, що керує графічним інтерфейсом. Таким чином виконується механізм інкапсуляції.

Функції називаються функціями високого рівня якщо використовують функцію як параметр, або надають функцію як результат виконання. [13]

Функції високого рівня використовуються під час розробки застосунку для розробки функціоналу, який дозволяє керувати базою даних з інтерфейсу користувача. Таким чином можна реалізувати видалення, зміну та додавання об’єктів з одного інтерфейсу користувача, що виконуватиме різні дії в залежності від певних параметрів запуску застосунку.

2.8 Колекція бібліотек Android Jetpack

Android Jetpack є колекцією бібліотек, які надають сучасні інструменти для розробки великої кількості функціоналу у застосунку [14]. За допомогою таких бібліотек зменшується кількість дублікатів коду, структура застосунку відповідає сучасним вимогам найкращих практик, а також будь-яка бібліотека з колекції Android Jetpack надає можливість розробляти застосунки, що працюватимуть на великій кількості версій операційної системи Android. Кожна операційна система має стрімкий темп розвитку, що створює значну кількість проблем під час розробки застосунків. Наприклад, з кожною версією операційної системи може змінюватись одна з її основних частин. До частин операційної системи Android, що зазнає великих змін, є частина, яка керує сповіщеннями користувача. Колекція бібліотек Android Jetpack дозволяє розробнику користуватись інструментами для керування сповіщеннями таким чином, щоб зміни в операційній системі Android не впливали на процес розробки застосунку.

До появи Android Jetpack кожен розробник застосунків для операційної системи Android використовував схожий на Android Jetpack набір бібліотек, що має назву Android Support Library. Проте велика кількість бібліотек з цього набору є застарілими і значна кількість вже не може бути використана для розробки застосунків, що працюють на пристроях з сучасними версіями операційної системи Android. Також частиною Android Jetpack є бібліотека Jetifier, яка надає можливість змінювати бібліотеки сторонніх розробників, що були розроблені із використанням Android Support Library, з заміною використаних бібліотек на їх еквівалентні бібліотеки з Android Jetpack. Таким чином сторонні бібліотеки, що були розроблені для розробки застосунків для застарілих версій операційної системи Android, можна використовувати паралельно і з новими бібліотеками з набору бібліотек Android Jetpack.

Android Jetpack є невід’ємною частиною кожного сучасного застосунка Android, тому майже увесь функціонал застосунку для відслідковування виконання

вправ був розроблений із використанням однієї або декількох бібліотек з колекції бібліотек Android Jetpack.

2.9 Бібліотека роботи з базами даних

Однією з бібліотек колекції Android Jetpack є бібліотека Room, що значно спрощує роботу з базами даних у мобільних застосунках для операційної системи Android [15]. Бібліотека Room надає сучасний інтерфейс роботи з базами даних, що передбачає використання різноманітних конструкцій мови програмування Kotlin. Бібліотека Android Room розроблена на основі бібліотеки SQLite, яка реалізує інтерфейс роботи з базами даних у вигляді мови програмування SQL.

Бібліотека Room також дозволяє робити версіювання бази, даючи інструмент для відслідковування версії бази даних та операцій оновлення записів бази даних зі старої схеми до нової. Таким чином, якщо на пристрої користувача встановлена версія застосунка зі старою схемою бази даних, розробник може розробити метод оновлення старої бази даних під нову версію. Завдяки цьому користувач зможе продовжити користуватись застосунком, зберігаючи раніше створені записи у базі даних.

Бібліотека для роботи з базами даних Room використовується для збереження даних на пристрої користувача для їх подальшого використання під час роботи програми. Наприклад, у розробленому застосунку для відслідковування виконання фізичних вправ у базі даних зберігаються такі об’єкти, як вправи, комплекси вправ, зарядки та дані про виконання фізичних вправ та зарядок.

2.10 Сервіс розробки макетів Figma

Figma є сервісом для розробки макетів мобільних застосунків, веб-сайтів та інших видів користувацьких інтерфейсів [16]. Кожен макет складається з окремих елементів графічного інтерфейсу користувача, які створюються за допомогою векторних та растрових зображень. Векторні зображення виступають головними

блоками для створення елементів інтерфейсу, адже обмежень на їх редагування майже немає. Растрові зображення редагуються, змінюючи кольорові характеристики зображення та його розмір. Векторні зображення є колекціями кривими Безьє, тобто таким зображенням можна надавати будь-яку форму, забарвлення, а також надавати певні ефекти. Поєднуючи векторні та растрові зображення можна розробляти макети користувацького інтерфейсу, до якого можна буде легко вносити необхідні зміни.

Для сервісу Figma існує велика кількість сторонніх розширень та наборів елементів, які також спрощують розробку макетів. Прикладом може стати Material 3 Design Figma Kit [17], що надає елементи користувацького інтерфейсу, які розроблені із виконанням правил і вимог дизайну Material Design.

Також однією з переваг сервісу Figma є створення інтерактивних прототипів, що дозволяють створити приклад користування застосунком без його розробки. Таким чином можна демонструвати як користувач буде користуватись застосунком та як застосунок реагуватиме на певні дії користувача.

Сервіс Figma використовується для розробки макетів графічного інтерфейсу користувача застосунку для відслідковування виконання фізичних вправ. За допомогою Figma значно спростився процес створення макетів. Також сервіс Figma надав низку інструментів для покращення макетів, у тому числі і створення інтерактивних прототипів, що надають інформацію про принцип користування застосунком.

2.11 Бібліотека елементів Material 3 Design Kit для Figma

Оскільки Figma підтримує використання сторонніх бібліотек елементів користувацького інтерфейсу, розробники з компанії Google розробили бібліотеку, яка має усі наявні елементи користувацького інтерфейсу, що мають описану специфікацію у правилах Material Design. Така бібліотека має назву Material 3 Design Kit та присутня у відкритому доступі. Бібліотека має низку елементів користувацького інтерфейсу, що дозволяють швидко розробити будь-який макет

користувацького інтерфейсу. За рахунок того, що кожен елемент також наданий у світлій і темній темі, розробник має можливість легко розробити макет застосунку, що буде мати функціонал перемикання тем в залежності від налаштувань системи користувача.

Для розробки макетів за допомогою сервісу Figma було вирішено використовувати бібліотеку елементів графічного інтерфейсу користувача Material 3 Design Kit. Використання бібліотеки Material 3 Design Kit дозволить розробити макети графічного інтерфейсу користувача, що відповідають сучасним стандартам розробки застосунків для операційної системи Android.

2.12 Розширення для роботи з іконками Material Design Icons

Сервіс для розробки макетів користувацького інтерфейсу Figma також має підтримку сторонніх розширень, які допомагають швидше та зручніше розробляти макети графічного інтерфейсу користувача. Одним з таких розширень є розширення Material Design Icons, що розроблене компанією Icons8 [18]. Розширення Material Design Icons є безкоштовним і надає розробнику обширний список векторних зображень, що мають високу роль у розробці застосунків для операційної системи Android. За допомогою таких зображень можна надати користувачеві необхідну інформацію про функціонал елементу. Наприклад, якщо на кнопці присутнє зображення, що нагадує «плюс», то це означає, що функціонал цієї кнопки надає функціонал додавання певного елементу. Якщо ж на кнопці присутнє зображення, що нагадує «пензлик», то це означає, що функціонал цієї кнопки надає функціонал редагування певного елементу.

Тому було вирішено використовувати розширення Material Design Icons для сервісу Figma. За допомогою розширення Material Design Icons вдалось розробити макети графічного інтерфейсу користувача, що мають векторні зображення, які допомагають користувачеві дізнатись про функціонал певних елементів графічного інтерфейсу.

2.13 Сервіс Draw.io для створення графічного представлення архітектури застосунку

Сервіс Draw.io використовується для створення графічного представлення моделей даних, схем роботи застосунку та інших діаграм, що використовуються у процесі розробки програмного забезпечення, у тому числі і застосунків для операційної системи Android. Цей сервіс надає низку інструментів, за допомогою яких можна легко створити UML діаграму [19].

Створення графічного представлення даних та схем користування застосунком є однією з основних частин розробки застосунку, що надає розробнику інформацію у більш зручному вигляді для подальшої її реалізації.

2.14 Сервіс Firebase для створення серверного функціоналу

Firebase є сервісом, що дає можливість швидко і зручно розробити функції застосунка для роботи яких необхідно було б створювати серверний застосунок. Цей сервіс дає інтерфейс для користування такими функціями, як авторизація користувачів, зберігання даних, відправка сповіщень та інші, без знання розробки серверних застосунків.

2.15 Система контроля версій Git та сервіс GitHub

Зміни на кожному сучасному проєкті відслідковуються за допомогою систем контролю версій. До такої системи відноситься також і система Git. Вона дозволяє розробнику контролювати зміни у проєкті, ділитись проєктом з іншими розробниками та працювати над проєктом у команді. За рахунок того, що кожен проєкт, який відслідковується системою контроля версій Git, має повну історію змін, розробники мають можливість швидко повернутись до минулих версій проєкту. Таким чином з’являється можливість роботи над проєктом паралельно. У кожного розробника є одна чи декілька гілок, що складається з окремих комітів.

Коміт – є різницею всіх файлів проєкту між поточним станом та минулим комітом. При завершенні роботи над певною частиною проєкту, зміни розробника поєднуються зі стабільною версією проєкту, та надаються у доступ іншим розробникам.

GitHub є веб сервісом, який дозволяє будь-якому розробнику завантажувати свій проєкт на сервіс для того, щоб розробники мали змогу отримати доступ до проєкту та його історії з будь-якого пристрою.

Для розробки проєкту було використано систему контроля версій Git, а також проєкт був викладений у відкритий доступ на сервісі GitHub. Це прискорило та спростило процес розробки застосунку за рахунок зручного внесення змін до програмного коду застосунку.

РОЗДІЛ 3. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Розробка сучасного застосунку для операційної системи Android є складною та комплексною задачею. Процес розробки такого застосунку передбачає перш за все розробку його архітектури, макетів, діаграми бази даних та інших графічних даних, що спростять процес розробки за рахунок визначення функціоналу та архітектури застосунку ще до початку його реалізації.

Розробка застосунку почалась з визначення функціональних вимог застосунку. Таким чином було визначено перелік функціоналу, який необхідно відобразити на макетах графічного інтерфейсу користувача.

3.1 Функціональні вимоги застосунку та діаграма прецедентів

Застосунок для відслідковування виконання фізичних вправ повинен містити базовий функціонал роботи з фізичними вправами та зарядками. Фізична вправа є серією рухів, які необхідно виконати певну кількість разів або певну кількість часу. Зарядка є комплексом фізичних вправ. Користувач повинен мати можливість записати або окремо необхідні фізичні вправи, або зарядку, що складається з декількох комплексів фізичних вправ. Також користувачеві повинно бути наданий певний список завчасно записаних фізичних вправ, щоб користувач мав змогу користуватись застосунком одразу після його встановлення, без необхідності створення загальних фізичних вправ, таких як підтягування, віджимання та інші.

Користувач повинен мати змогу додавати, редагувати та видаляти окремі фізичні вправи та їх комплекси. Кожна вправа та зарядка також можуть мати малюнок, який допоможе користувачеві швидше зрозуміти як виконувати вправи, або буде зручнішим для знаходження необхідної вправи або зарядки зі списку існуючих. Такий малюнок можна або завантажити з пристрою користувача, або зробити свій за допомогою камери з пристрою користувача.

У кожній вправі присутній також список м’язів, що тренуються. Таким чином користувач може створювати зарядки, які будуть охоплювати комплекс м’язів задля кращого тренування усього тіла.

Користувач повинен мати змогу подивитись історію виконання фізичних вправ та зарядок, а також подивитись на графік, що показує прогрес за кількістю чи часом виконання фізичних вправ. За допомогою такого графіку користувач може зрозуміти які м’язи у нього тренуються найчастіше, за рахунок чого він зможе краще розробити для себе програму зарядки. Інтерфейс історії виконання фізичних вправ і зарядок також повинен мати вибір дати, за яким користувач зможе фільтрувати історію за необхідним проміжком часу.

Застосунок повинен мати підтримку різних мов інтерфейсу, наприклад, української та англійської, а також різних тем застосунку, таких як світла і темна теми. У майбутньому функціонал мов інтерфейсу та його оформлення можна доповнювати.

Також кожна вправа та зарядка повинні мати поле, що зберігає чи потрібно використовувати спеціальні інструменти на кшталт гантелей або килимка для фітнесу.

Керуючись описаними функціональними вимогами було розроблено діаграму прецедентів. Таким чином було зображено основний функціонал для користувача застосунка, де єдиним актором є звичайний користувач. Діаграму прецедентів наведено на рисунку 3.1.

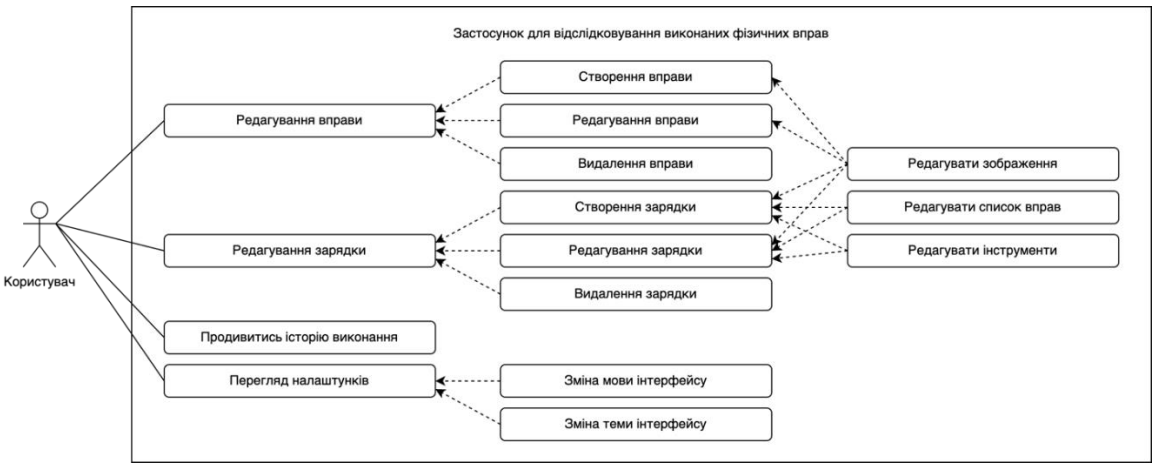

Рисунок 3.1 Діаграма прецедентів застосунку

3.2 Створення макету застосунку у Figma

Сучасний процес розробки застосунків для операційної системи Android передбачає створення макетів інтерфейсу користувача за допомогою сервісів на кшталт Figma. Такі макети дозволяють розробити дизайн застосунку ще до початку його реалізації, що спрощує сам процес реалізації. Через те, що усі елементи інтерфейсу мають чітке положення на екрані та мають чітко визначений дизайн, розробнику мобільного застосунка необхідно лише розробити дизайн інтерфейсу користувача згідно наданим умовам.

Макетом є певне графічне представлення зовнішнього виду одного з екранів застосунку. Він містить чітке розташування елементів, їх розміри, кольори та стиль тексту. Макети застосунку повинні надавати вигляд інтерфейсу користувача для кожного з екранів, який може бачити користувач. Тому для застосунка, що буде допомагати користувачеві відслідковувати виконання фізичних вправ, необхідно розробити наступні макети:

- головна сторінка;
- сторінка вправ;
- сторінка зарядок;
- сторінка створення та зміни вправи;
- сторінка створення та зміни зарядки;
- сторінка перегляду історії виконання вправ та зарядок;
- сторінка налаштувань;
- сторінка «Про застосунок».

Також необхідно заповнити макети певною інформацією, яка зробить макети схожими на зображення з реального розробленого застосунку. Слід зауважити, що макети екранів застосунку повинні також мати і наступні елементи користувацького інтерфейсу:

- бокове меню навігації;
- верхня навігаційна панель;
- діалог обирання зображення;
- діалог обирання м’язів під час створення та редагування вправи;
- діалог підтвердження видалення вправи, зарядки та їх записів;
- та інші елементи користувацького інтерфейсу.

Для досягнення кращого ефекту необхідно надавати макети у рамці, що нагадує сучасний телефон, що працює на операційній системі Android. Рамка такого виду повинна мати статус панель, і навігаційну панель.

3.2.1 Розробка стилю дизайну застосунка

Кожен мобільний застосунок, що розробляється на операційної системі Android, має певний стиль, який розробник може розробити на початковому етапі розробки застосунка. До частин стилю відноситься кольорова тема, стилі шрифтів, стилі стандартних елементів та інше.

Кольорова тема повинна мати спокійне забарвлення. До таких кольорів відносяться синій та зелений. Синій колір асоціюється з соціальними мережами, такими як Twitter, Facebook [21] та іншими, тому для застосунку для відслідковування виконання фізичних вправ було вирішено обрати зелену тему.

При наявності певних списків у застосунку необхідно кожному елементу з цих списків надавати окремий колір, що буде допомагати користувачеві швидше знайти необхідний елемент з цього списку. За допомогою такої ідентифікації за кольором можна зробити дизайн застосунку більш персоналізованим, адже на сторінках, що мають більш детальну інформацію про певний об’єкт, можна використовувати такий колір для позначення основної інформації про цей об’єкт. Прикладом може бути сторінка відображення інформації про зарядку. На такому екрані забарвлення, що притаманне для обраної зарядки, можна також використовувати і в функціональних елементах, таких як кнопки.

Також для розробки макетів графічного інтерфейсу користувача було використане розширення Material Design Icons. За його допомогою була розроблена низка елементів графічного інтерфейсу користувача, що мають пояснення до функціоналу цього елементу у вигляді векторного зображення. Таким чином користувач зможе розпізнати необхідні елементи та їх функціонал лише за його дизайном, не вчитуючись у текст.

3.2.2 Розробка іконки та назви застосунку

Одними з основних елементів дизайну застосунку для операційної системи Android є його іконка та назва.

Спираючись на те, що метою роботи є створення застосунку для відслідковування виконання вправ, його назва повинна мати відношення до спорту. Також назва кожного застосунку повинна бути такою, яку користувач легко запам’ятає. Якщо назва застосунку відповідатиме наведеним критеріям, його можна буде легко знайти у магазині застосунків Google Play Market, а також користувач, який зацікавлений у застосунках для відслідковування виконання вправ, з більшою імовірністю встановить застосунок, що матиме назву, яка відноситься до відслідковування виконання вправ.

За назву було вирішено обрати частину відомої фрази «Хто не скаче». Таким чином назву розробленого застосунку для відслідковування фізичних вправ буде легко запам’ятати користувачам, особливо які знаходяться в Україні або знають українську мову. Вислів «Хто не скаче» спонукає користувачів до виконання фізичних вправ, що, відповідно, спонукає користувача до використання розробленого застосунку.

Іконкою застосунку є растрове зображення, що дозволяє користувачеві легко знайти посилання на застосунок на мобільному пристрої користувача. Якщо застосунок має іконку, що легко впізнається користувачем, то користувач матиме більше бажання повернутись до використання такого застосунку. Таким чином іконка застосунку повинна мати зображення, що дозволить користувачу легко

ідентифікувати її серед інших застосунків. Опираючись на назву застосунку, «Хто не скаче», та українське походження джерела назви застосунку, було вирішено розробити іконку, на фоні якої буде зображений прапор України, а на передньому плані міститься схематичне зображення людини, яка виконує вправу на здійснення стрибків, як зображено на рисунку 3.2.

Рисунок 3.2. Іконка застосунку для відстеження виконання вправ

3.2.3 Розробка макету головної сторінки застосунку

Головна сторінка застосунка для відслідковування виконання фізичних вправ повинна надавати користувачеві функціонал для швидкого і зручного записування виконаних фізичних вправ, а також надавати статистику за виконаними фізичними вправами за день.

Для швидкого і зручного записування виконаних фізичних вправ розроблено секцію «закріплені вправи», яка дозволяє користувачеві розміщувати улюблені вправи для швидкого доступу до них. Кожна така вправа має вигляд карточки певного кольору, що містить малюнок вправи, її назву та статистику виконання за поточний день.

Для перегляду статистики за поточний день було розроблено секцію «останнє тренування», що дозволяє користувачеві швидко переглянути статистику виконання вправ за день. Ця статистика складається зі списку виконаних вправ, кожна з яких

представлена у вигляді «пігулки». Кожна вправа містить назву вправи, кількість виконання та певний колір, який асоціюється з цією вправою. Також присутнє посилання для переходу до екрану перегляду більш детальної сторінки не тільки за поточний день, а і за певний проміжок часу, який вказує користувач.

Також одним з елементів головної сторінки є привітання користувача та відображення поточної дати.

Розроблений макет головного екрану наведено на рисунку 3.3.

Рисунок 3.3. Макет головного екрану застосунку

3.2.4 Розробка макету навігаційної панелі застосунку

Кожен застосунок, що розробляється для операційної системи Android має певний вид навігації застосунком. Така навігація необхідна для того, щоб користувач мав змогу переміщуватись між екранами застосунку, а також мати швидкий доступ до функціоналу, який частіше за все використовується користувачами застосунку. До такого функціоналу у розробленому застосунку для відслідковування виконаних фізичних вправ відноситься функціонал для перегляду списку вправ, функціонал для перегляду списку зарядок, функціонал для перегляду історії виконання вправ, функціонал для перегляду налаштувань застосунку та функціонал для перегляду короткої інформації про застосунок та його розробника. Розроблений макет навігаційної панелі наведено на рисунку 3.4.

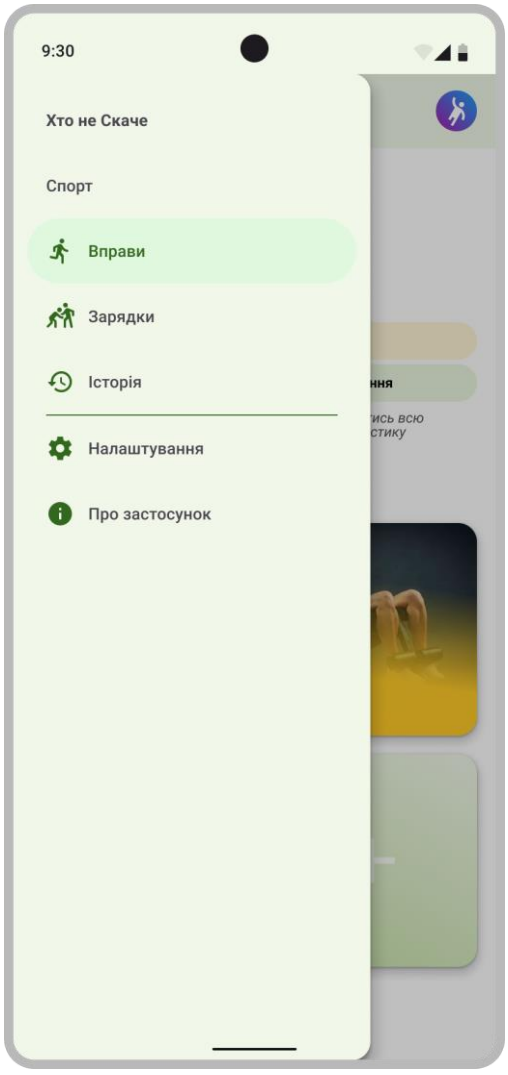

Рисунок 3.4. Макет навігаційної панелі

Зовнішній вигляд навігації розроблено у виді навігаційної панелі, яку користувач може відкрити жестом від лівого краю дисплею пристрою, або натиснувши на кнопку, що знаходиться у лівому верхньому куті головної сторінки застосунку та має вигляд трьох горизонтальних рисок. Такий вигляд кнопки називаються «бургером».

3.2.5 Розробка макету сторінки користувача

Розроблений застосунок для відслідковування виконання фізичних вправ надає кожному користувачу персоналізований досвід користування застосунком. Тому для створення такого досвіду користування застосунком була розроблена сторінка користувача. Розроблений макет сторінки користувача наведено на рисунку 3.5.

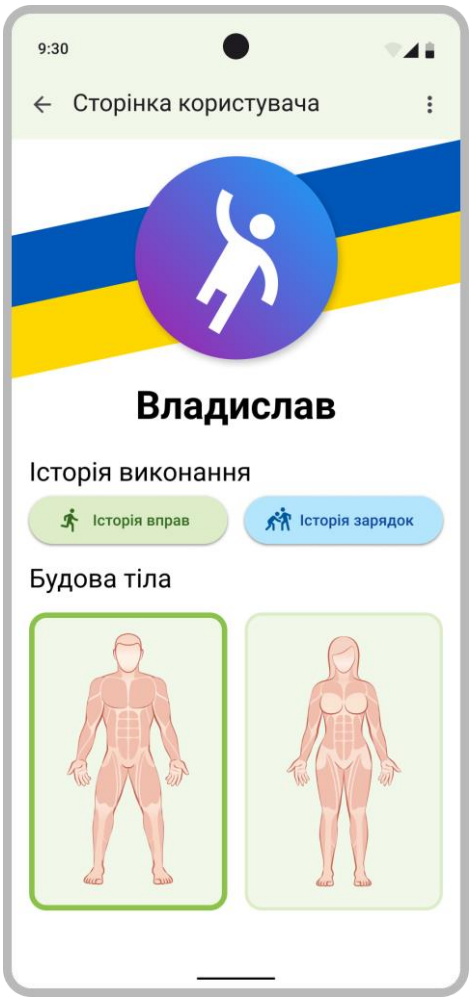

Рисунок 3.5. Макет сторінки користувача

На цій сторінці можна побачити зображення, яке користувач використовує для персоналізації застосунку, прапор України для україномовної версії застосунку, кнопки для переходу на сторінки перегляду історії виконання вправ та зарядок, а також обирання будови тіла. Обирання будови тіла необхідне для того, щоб відображати м’язи, що тренуються під час виконання певного виду вправ. Було вирішено використовувати саме обирання будови тіла за зображенням, щоб не прив’язуватись до статі користувача.

3.2.6 Розробка макету списку вправ

Однією з основних частин застосунку для відслідковування виконання фізичних вправ є відображення списку створених вправ. Кожна вправа з такого списку повинна надавати загальну інформацію про вправу, таку як її назва, опис, фотографія та список м’язів, які тренуються під час виконання вправи. Кожна така вправа позначається певним кольором для того, щоб користувач міг швидко знайти необхідну вправу з усього списку. Також у правому нижньому куті екрану розташована кнопка з текстом «Додати вправу» та іконкою «плюса», що дозволяє користувачеві перейти на екран створення нової вправи. Розроблений макет сторінки списку вправ наведено на рисунку 4.6.

Рисунок 3.6. Макет сторінки списку вправ

3.2.7 Розробка макетів огляду, створення та редагування вправ

Для перегляду детальної інформації про кожну вправу розроблено сторінку для перегляду усієї інформації про вправу. На сторінці перегляду присутні такі елементи графічного інтерфейсу користувача, як зображення вправи, назва вправи, опис вправи, кнопка для переходу статисти виконання вправи та список м'язів, що тренуються під час виконання поточної вправи. Також у правому нижньому куті розташована кнопка, що дозволяє перейти на екран редагування вправи. Макет сторінки огляду вправи наведено на рисунку 3.7 а).

Для створення та редагування вправи створено сторінку, що дозволяє користувачеві встановити зображення вправи, її назву, опис і редагувати список м’язів, які тренуються під час виконання поточної вправи. Для редагування списку м’язів використовуються кнопки додавання м’язів та видалення присутніх м’язів. Також у правому нижньому куту екрану знаходиться кнопка, що дозволяє зберегти зміни, що були внесені до вправи. Макет сторінки створення та редагування вправи наведено на рисунку 3.7 б).

Слід зауважити, що кольорова схема деяких елементів користувацького інтерфейсу співпадає з кольором, що асоціюється з поточною вправою, що дозволяє користувачеві легше орієнтуватись у яку саме вправу вносяться зміни.

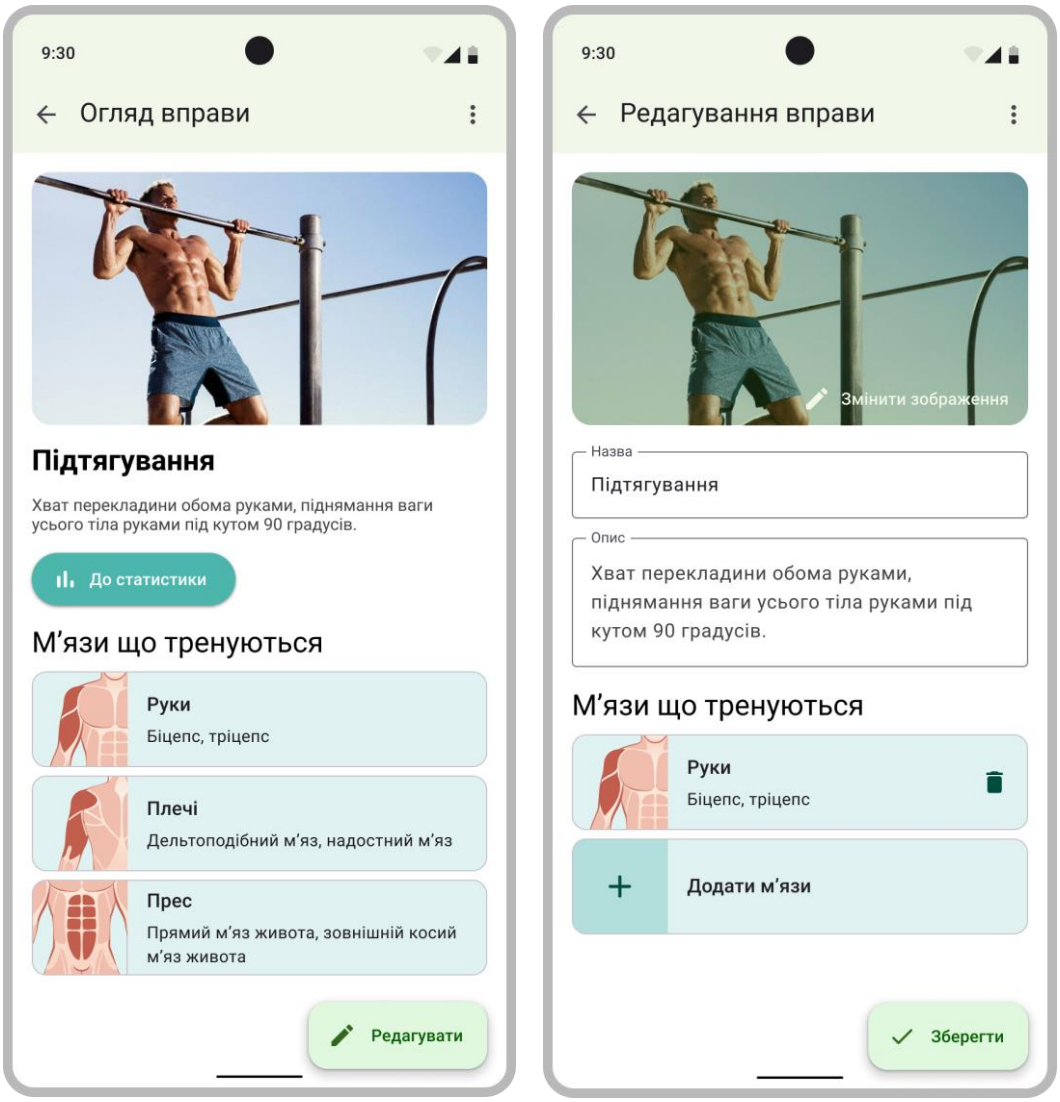

Рисунок 3.7 а) макет сторінки огляду вправи, б) макет сторінки створення та редагування вправи

3.2.8 Розробка макету списку зарядок

Однією з основних частин застосунку для відслідковування виконання фізичних вправ є перегляд списку наявних зарядок. Зарядки дозволяють швидко і зручно додавати комплекси вправ за одне натискання. Таким чином користувачу для записування виконаної зарядки, яку користувач виконує регулярно, не потрібно буде щоразу записувати вправи та їх кількість. За допомогою використання зарядок користувач уникає виконання зайвих дій. Макет списку зарядок наведений на рисунку 3.8.

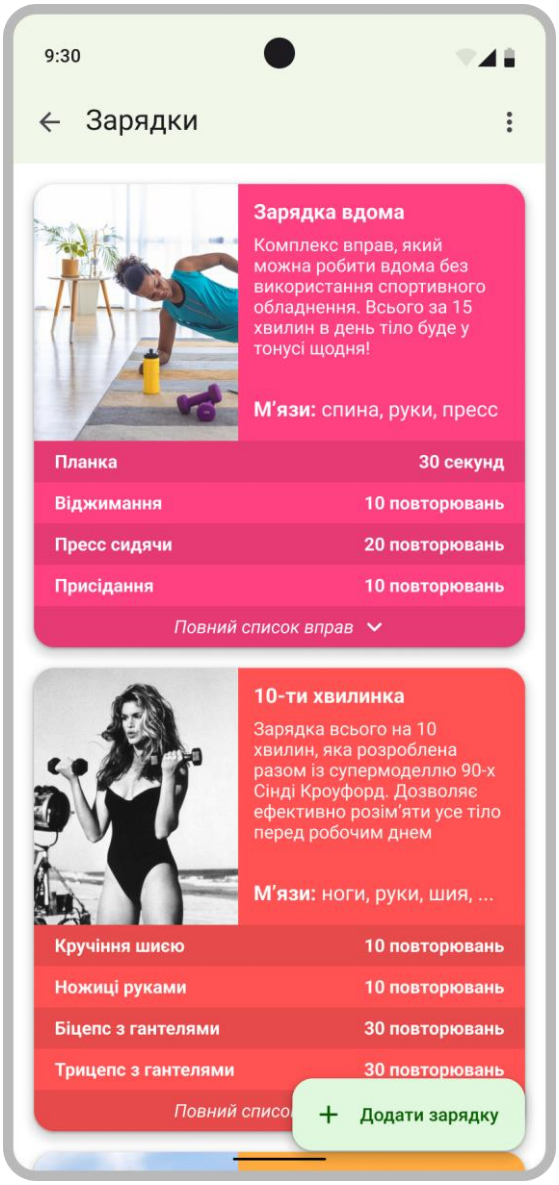

Рисунок 3.8. Макет сторінки списку зарядок

На екрані перегляду списку зарядок відображається список карток, кожна з яких надає інформацію про зарядку. До такої інформації відноситься назва зарядки, її опис, м’язи, що тренуються під час виконання зарядки, та список з вправ, з яких складається зарядка. Кожна така вправа відображає назву вправи та її тривалість або кількість повторювань. У випадку, якщо кількість вправ перебільшує 4, у самому низу картки зарядки відображається кнопка, яка дозволяє передивитись повний список вправ, що містяться у поточній зарядці.

У правому нижньому куті екрану знаходиться кнопка, що дозволяє користувачеві перейти на екран створення нової зарядки.

3.2.9 Розробка макетів огляду, створення та редагування зарядок

Для перегляду детальної інформації про кожну зарядку було розроблено сторінку, що відображає детальну інформацію, таку як зображення зарядки, назву зарядки, опис зарядки та список вправ, з яких складається зарядка. Також на сторінці перегляду інформації про зарядку міститься кнопка, що дозволяє користувачеві перейти до статистики виконання зарядки. У правому нижньому куті сторінки міститься кнопка, що дозволяє користувачеві перейти у режим редагування зарядки.

Список вправ, з яких складається зарядка, представляє собою список карток, кожна з яких позначає окрему вправу. Така картка містить наступну інформацію про вправу: назва вправи, зображення вправи та кількість секунд або повторювань виконання поточної вправи.

Макет сторінки огляду детальної інформації зарядки наведено на рисунку 3.9 а).

Для створення та редагування зарядки створено окрему сторінку, яка дозволяє редагувати наступну інформацію, що стосується зарядки: зображення, назву, опис та список вправ, з яких складається зарядка. Для редагування списку вправ використовується кнопка видалення вправи та кнопка додавання нової вправи. У правому нижньому куті сторінки розташовано кнопку, що дозволяє

зберегти зміни, що були внесені у зарядку. Макет сторінки створення та редагування зарядки наведено на рисунку 3.9 б).

Слід зауважити, що так само, як і з вправами, кожній з зарядок надається певний колір, який впливає на відображення деяких елементів користувацького інтерфейсу на екранах перегляду, створення та редагування зарядки.

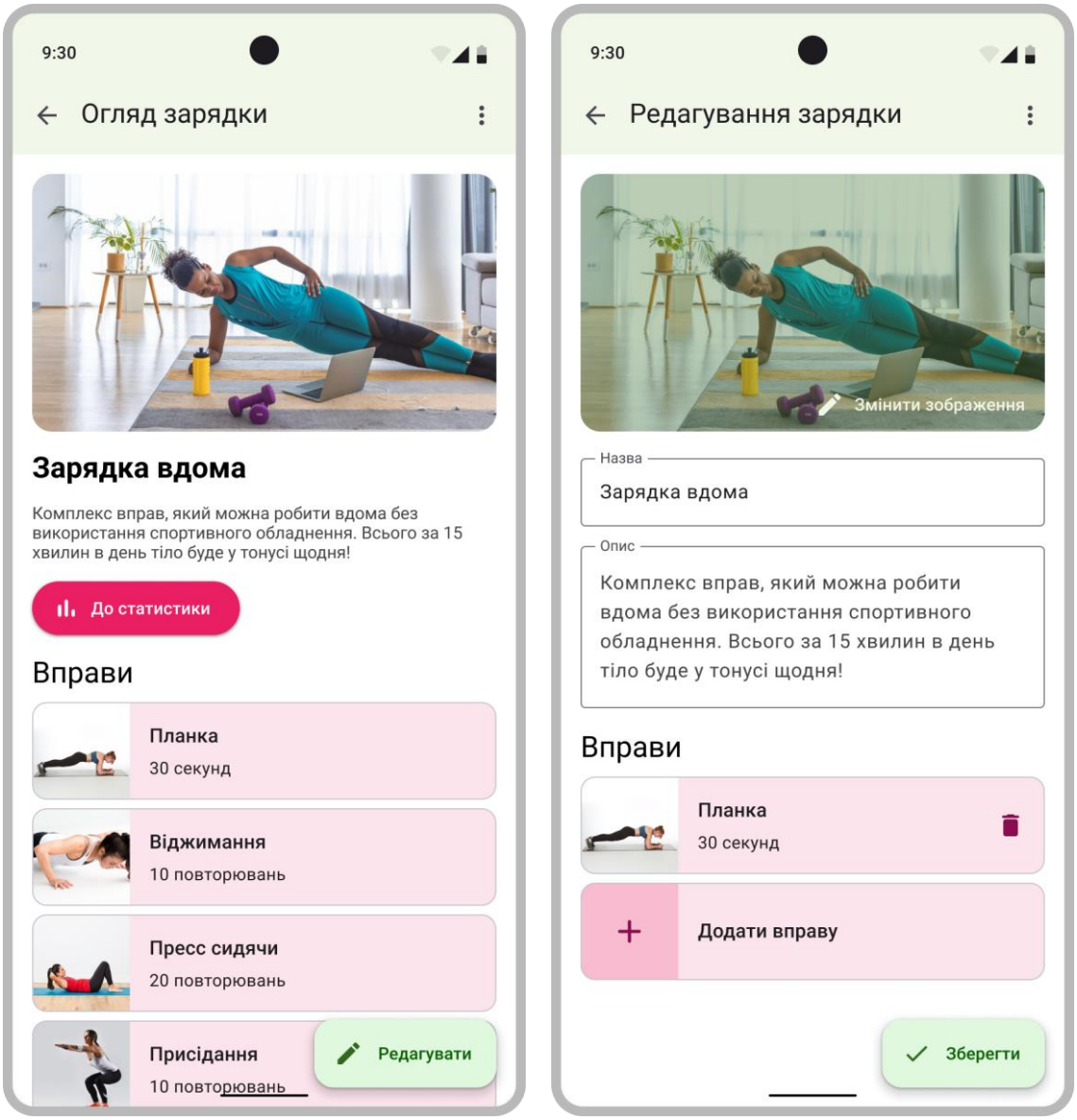

Рисунок 3.9 а) макет сторінки огляду зарядки, б) макет сторінки створення та редагування зарядки

3.2.10 Розробка макету перегляду статистики

Для перегляду статистики виконання вправ та зарядок розроблено дизайн відповідної сторінки. Вона містить календар, на якому кольоровими кружечками позначено дні, коли користувач займався певною вправою. Також за допомогою календаря можна керувати переглядом за місяцями. Під календарем наведено список днів виконання фізичної вправи та кількість їх повторювань. Макет сторінки перегляду статистики наведено на рисунку 3.10.

Рисунок 3.10 Макет сторінки перегляду статистики

3.3 Розробка моделі даних «сутність-зв’язок»

Для розробки застосунку для відслідковування виконання фізичних вправ необхідно розробити модель бази даних, яка буде відображати механізм збереження вправ та зарядок, які користувач створюватиме під час користування застосунком. Було розроблено модель бази даних з графічним представленням у вигляді воронячої лапки [22]. База даних складається з восьми сутностей: сутності окремих об’єктів «Вправа», «Зарядка», «М’язи»; сутності відношення об’єктів «Комплекс Вправ», «Вправи для Зарядки», «М’язи для Зарядки»; сутності для відслідковування виконання вправ та зарядок «Виконані Вправи для Зарядки» та «Виконані Зарядки».

Сутність «Вправа» необхідна для зберігання вправ. Таблиця для збереження вправ має наступні поля:

- особистий ідентифікаційний номер, за яким можна однозначно визначити вправу;
- назва вправи;
- опис вправи;
- посилання на зображення вправи.

Сутність «Зарядка» необхідна для зберігання зарядок. Таблиця для зберігання зарядок має наступні поля:

- особистий ідентифікаційний номер, за яким можна однозначно визначити зарядку;
- назва зарядки;
- опис зарядки.

Сутність «М’язи» необхідна для зберігання м’язів, які тренуються вправами. Користувач не може змінювати рядки таблиці м’язів, проте використовує їх під час створення та редагування вправ. Таблиця для зберігання м’язів має наступні поля:

- особистий ідентифікаційний номер, за яким можна однозначно визначити м’язи;

- назви м’язів;
- опис м’язів.

Сутність «Комплекс Вправ» необхідна для зберігання окремого комплексу вправи, що зазначає кількість її виконання у секундах або повтореннях. Таблиця для зберігання комплексів вправ має наступні поля:

- особистий ідентифікаційний номер, за яким можна однозначно визначити комплекс вправ;
- ідентифікатор вправи, кількість виконання якої вказується;
- кількість секунд, яку виконується вправа, якщо вона дорівнює 0, то використовується кількість повторювань виконання вправи;
- кількість повторювань, яку виконується вправа, якщо вона дорівнює 0, то використовується кількість секунд виконання вправи.

Сутність «Вправи для Зарядки» необхідна для збереження списку вправ, що відносяться до певної зарядки. Таблиця для зберігання вправ для зарядки має наступні поля:

- ідентифікатор вправи;
- ідентифікатор зарядки.

Сутність «М’язи для зарядки» необхідна для зберігання списку м’язів, які тренуються під час виконання певної вправи. Таблиця для зберігання м’язів для зарядок має наступні поля:

- ідентифікатор м’язів;
- ідентифікатор зарядки.

Сутність «Виконані вправи для зарядки» необхідна для зберігання виконаних комплексів вправ, а саме їх час виконання. Таблиця для зберігання виконаних вправ для зарядки має наступні поля:

- особистий ідентифікаційний номер, за яким можна однозначно визначити виконані вправи для зарядки;
- ідентифікатор комплексу вправ;
- час виконання у Міллі секундах, що пройшов з першого січня 1970 року.

Сутність «Виконані зарядки» необхідна для зберігання виконаних зарядок, а саме їх час виконання. Таблиця для зберігання виконаних зарядок має наступні поля:

– особистий ідентифікаційний номер, за яким можна однозначно визначити виконану зарядку;
– ідентифікатор зарядки;
– час виконання у мілісекундах, що пройшов з першого січня 1970 року.

Детальна діаграма моделі «сутність-зв’язок» наведено на рисунку 3.11.

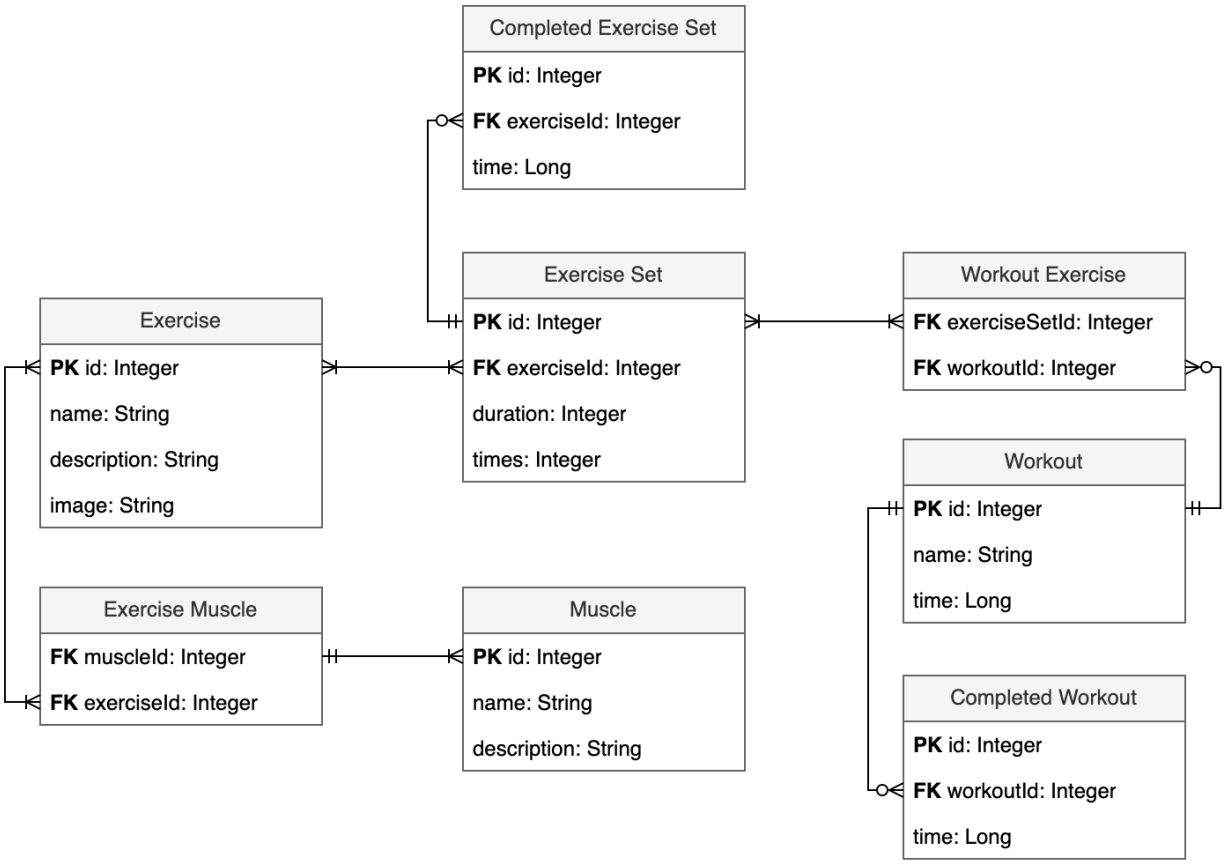

Рисунок 3.11. Діаграма моделі бази даних

Така модель бази даних задовольняє вимогам першої нормальної форми, тобто ця модель задовольняє реляційним принципам: кожне значення є нероздільним та кожне поле містить лише одне значення.

Також така модель задовольняє вимогам другої нормальної форми, адже будь-яке значення з будь-якого рядка будь-якої таблиці наведеної моделі функціонально залежать від ключа, за яким можна ідентифікувати необхідний рядок.

Наведена модель задовольняє вимогам третьої нормальної форми, адже кожен атрибут з кожного рядка кожної бази даних або є ключем, або доповнює інформацію про об’єкт, який можна визначити за допомогою ключа.

3.4 Створення мобільного застосунку

3.4.1 Архітектура застосунку та бази даних

Архітектура сучасного мобільного застосунку для операційної системи Android передбачає розділення програмної реалізації на частини, що виконують певний вид робіт. Архітектура розробленого застосунка для відслідковування виконання фізичних вправ розділена два пакети, перший відповідає за роботу з базою даних, а другий – з роботою з графічним інтерфейсом користувача.

Пакет роботи з базою даних має два класи, перший з яких відповідає за створення єдиного об’єкту бази даних, тобто реалізовує шаблон проєктування «одинак», а другий містить низку методів, які необхідні для переведення об’єктів мови програмування у рядковий тип даних. Для такого переведення використовується механізм серіалізації та десеріалізації даних. Рядковим представленням методів виступає формат даних JSON, що дозволяє швидко і зручно змінювати записані дані без необхідності їх десеріалізації.

Також частиною пакету роботи з базою даних є два вкладених пакети. Перший пакет містить об’єкти доступу до даних (англ. DAO), а другий містить описи сутностей і таблиць у базі даних.

Об’єкти доступу до даних реалізовують методи для збереження, оновлення та видалення записів, а також для отримання записів за допомогою SQL запитів. За допомогою бібліотеки Android Room Database розробник має можливість швидко і зручно розробити методи роботи з базою даних. Для цього використовуються

анотації для створення (@Insert), оновлення (@Update), видалення (@Delete), оновлення зі створенням у разі відсутності (@Upsert) та отримання (@Query) рядків бази даних. Методи отримання рядків з бази даних також можуть отримувати деякі вхідні параметри, за рахунок яких можна виконувати динамічні SQL запити.

Будь-який застосунок, що працює на операційній системі Android та зберігає дані користувача у постійну пам’ять, використовує бази даних для збереження такого виду даних. Інструментом для роботи з реляційними базами даних під час розробки застосунків для операційної системи Android була база даних SQLite. Проте її використання передбачало створення великої кількості повторюваного коду, який значно ускладнював та сповільнював процес розробки застосунку. Сучасний метод розробки застосунків із використанням баз даних передбачає використання бібліотеки з набору бібліотек Android Jetpack, а саме бібліотеки Android Room Database.

Також великою перевагою використання бібліотеки Android Room Database є те, що кожна операція виконується паралельно із потоком, що працює з графічним інтерфейсом користувача, що унеможливлює зменшення плавності користування застосунком користувачем.

Описи сутностей та таблиць зберігаються у пакеті, що має назву Entities. Кожна сутність описує одну окрему таблицю бази даних, вказуючи її назву, опис полів, назву полів. Також можна визначити головний ключ з таблиці за допомогою анотації @PrimaryKey. Поля класів, що позначені такою анотацією, можна також і визначити автоматично розраховуваними, що дозволить уникнути зайвої роботи із ідентифікаторами об’єктів, кожен об’єкт матиме індивідуальний номер.

Другим з основних пакетів архітектури застосунку є пакет роботи з графічним інтерфейсом користувача. Такий пакет містить три вкладені пакети, кожен з яких відповідає за певну частину графічного інтерфейсу. Перший є пакет з назвою «Activities», що має у собі певну кількість класів, кожен з яких описує логічну частину роботи користувача з застосунком. Кожен з цих класів пов’язаний виключно з однією сторінкою застосунка.

Другий вкладений пакет пакету роботи з графічним інтерфейсом користувача є пакет адаптерів, що містить класи, які реалізують роботу зі списками елементів. Наприклад, для списку вправ існує окремий адаптер, який керує створенням цього списку, його наповненням, реагування на дії користувача, та інше. Такі адаптери використовуються для створення сучасних списків елементів, що мають назву «Recycler View», тобто список, що може переробляти елементи під час його прокручення користувачем. Такий метод реалізації списків значно збільшує продуктивність застосунку та зменшує навантаження на пристрій користувача, адже у випадку, коли такий список матиме велику кількість елементів, користувачу відображатиметься тільки певна частина елементів списку, при чому їх наповнення буде змінюватись з прокрученням списку. Таким чином зникає необхідність операційній системі Android тримати велику кількість однакових елементів графічного інтерфейсу користувача у пам’яті пристрою, коли вони не відображаються на пристрої користувача.

Третій вкладений пакет роботи з графічним інтерфейсом користувача є пакет фрагментів. Фрагментом є елемент графічного інтерфейсу, що працює як окремий екран, проте є частиною іншого екрану. Це дозволяє одночасно використовувати елементи застосунку, які необхідно було б розробляти на декількох сторінках окремо. Таким чином збільшується швидкість розробки застосунків для операційної системи Android та спрощується його архітектура.

Головним класом застосунку є клас «KhtoNeSkacheApp», що є вхідною точною роботи застосунку.

Зображення архітектури застосунки наведено на рисунку 3.12.

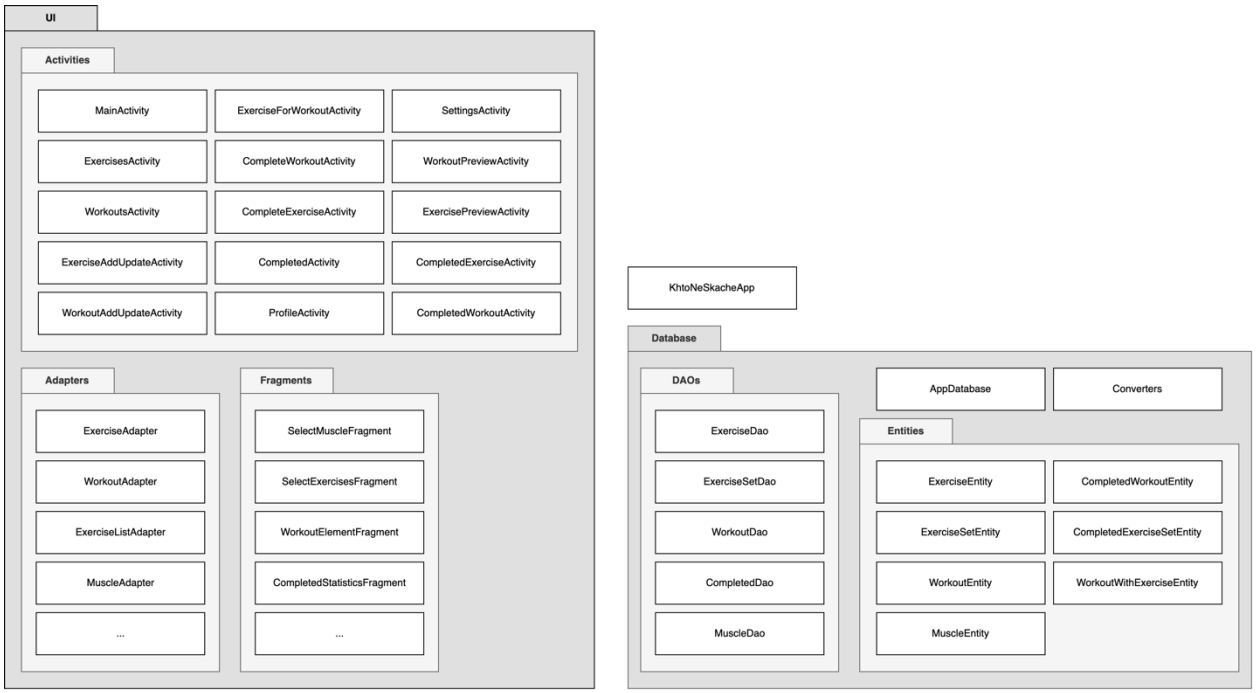

Рисунок 3.12. Архітектура застосунку

3.4.2 Розробка інтерфейсу користувача

Розробка інтерфейсу користувача для застосунків, що розробляються для операційної системи Android, виконується за допомогою використання мови розмітки XML. Кожен елемент графічного інтерфейсу користувача є «виглядом» (англ. View). Такі елементи мають низку атрибутів, що дозволяють швидко змінити зовнішній вид елемента.

Кожна сторінка, фрагмент, або об’єкт списку має окремий файл XML. Таким чином кожен графічний елемент застосунку є окремою частиною, яку можна змінювати без впливання на зміни у інших графічних елементах.

Кожен елемент графічного інтерфейсу користувача має певний зовнішній вигляд, що задається самим розробником, або використовується зовнішній вигляд елементу однієї з бібліотек. Для створення значної частини застосунку було використано бібліотеку Material Components, яка дозволила швидко та зручно розробити графічний інтерфейс користувача, що відповідає сучасним стандартам дизайну застосунків для операційної системи Android.

Також однією з невід'ємних частин застосунку є розробка його єдиного стилю. Файл colors.xml містить список кольорів, що використовуються у застосунку. Файл theme.xml керує використанням зазначених кольорів у контексті елементів програми. Наприклад, у файлі theme.xml можна визначити зовнішній вигляд елементу графічного інтерфейсу користувача, що дозволить швидко змінити вигляд цього елементу на усіх сторінках однозначно.

Кожен рядковий ресурс повинен бути виписаний у файл string.xml. За допомогою цього з'являється можливість створювати багатомовні графічні інтерфейси користувача, адже кожен рядок буде міститись у файлі, який можна легко замінити іншими, з необхідним перекладом.

Відстані між елементами та розміри шрифтів зберігаються у файлі dimens.xml. Використання цього файлу так само полегшує і прискорює розробку застосунку, адже для того, щоб змінити певну частину графічного інтерфейсу користувача, буде достатньо лише змінити змінну у файлі dimens.xml.

Папка menu містить ресурси меню, які визначають який функціонал матиме певний елемент меню. Наприклад, для розробки бокової навігаційної панелі необхідно використовувати ресурс меню, що матиме структуру з усіма необхідними посиланнями на сторінки застосунку.

Для розробки графічного інтерфейсу користувача також використана низка інструментів Android Studio, таких як додавання векторних іконок, або інструмент для відслідковування помилок під час запуску застосунку на реальному пристрої.

3.5 Використання серверного функціоналу за допомогою Firebase

Велика кількість серверних програм, які розробляються для задоволення потреби застосунків у певному функціоналі, зазвичай реалізовують схожий функціонал, такий як авторизація та збереження даних у хмарі. Щоб прискорити процес розробки було вирішено використовувати сервіс Firebase, який надає значну кількість інструментів для організації певного функціоналу без розробки серверної частини застосунку.

Сервіс Firebase було використано для створення авторизації та автоматичних завантажень знімків стану програми на випадок псування таких даних. Через часте відсутня електроживлення взимку 2022-2023 років, багато користувачів зіштовхнулось з проблемою збереження даних застосунку без підключення до мережі інтернет. Таким чином було вирішено завантажувати знімок бази даних застосунку за допомогою Firebase та автоматизованих задач.

3.6 Публікація на Play Market та закрите тестування

Play Market є сервісом для публікації та завантаження застосунків, що розробляються для операційної системи Android. Кожна людина, яка має смартфон, що працює на базі операційної системи Android, користується цим сервісом для отримання оновлень застосунків, завантаження нових та видалення непотрібних.

Для того, щоб опублікувати застосунок на сервісі Play Market необхідно отримати акант розробника. Такий акаунт необхідно активувати, заплативши компанії Google 25 доларів США. Після отримання доступу до панелі керування, розробник має можливість завантажити свій застосунок для закритого тестування. Під час тестування користувачі не можуть ставити оцінку застосунку, але можуть залишати відгуки про роботу застосунку, які побачить виключно розробник проєкту.

Для того, щоб опублікувати застосунок необхідно пройти низку перевірок. Після підтвердження публікації застосунку, розробник зможе публікувати розроблені застосунки для тестування серед певного кола користувачів.

Тестування застосунку відбувалось за допомогою групи користувачів, які погодились використовувати розроблений застосунок для відслідковування виконаних фізичних вправ. Кожен з учасників групи отримав закритий доступ до застосунку на сервісі Google Play Market, після чого мав змогу завантажувати та оновлювати застосунок, використовуючи стандартні методи роботи з застосунками.

Кожен користувач, користуючись застосунком, приділяв високу увагу до зручності користування. Такі відгуки користувачів допомогли розробити простий та зрозумілий дизайн застосунку.

Також велику роль у покращенні застосунку зіграли відгуки користувачів, що пов’язані з критичними помилками під час користуванням застосунку. До таких помилок частіше за все відноситься помилка виконання, коли програмний код застосунку має або помилкову логіки роботи, або має вади, які призводять до отримання пустих посилань на об’єкти.

РОЗДІЛ 4. МЕТОДИ ПОКРАЩЕННЯ ЗАСТОСУНКУ

Кожен мобільний застосунок повинен покращуватись, щоб збільшувати аудиторію своїх користувачів. Для цього розробники повинні покращувати якість користування застосунком та його функціонал.

4.1 Збільшення кількості мов

На даний момент застосунок має дві мови користувацького інтерфейсу – англійська та українська. Для збільшення аудиторії можна додати більше найбільш вживаних мов: німецька, французька, китайська, японська та інші.

4.2 Функціонал соціальної мережі

Цікавим функціоналом, який може зацікавити велику кількість користувачів, є функціонал соціальної мережі. Можна розробити систему, яка дозволить користувачам бачити прогрес виконання вправ іншими користувачами, запрошувати їх на «фізичні дуелі», що передбачають виконання комплексів вправ, а також можливість створювати дописи про тренування.

4.3 Покращення надання інформації про виконані вправи

Для кращого надання інформації про виконані вправи можна використати бібліотеки інтерфейсу користувача, що дозволяють розробити графіки. Наприклад, для користувача, який працює над збільшенням кількості підтягувань на перекладині, було б корисно бачити графік прогресу з початку тренувань до поточного дня.

4.4 Розробка серверної частини застосунку

Одним з покращенням, що значно розширить функціонал застосунку є створення серверної частини за допомогою бібліотек Spring або Vertx на мові програмування Kotlin. Завдяки цьому можна буде розробити прикладний програмний інтерфейс, який буде зберігати дані не тільки на пристрої користувача, а і на віддаленому сервері, що зробить можливим створення такого функціоналу, як елементи соціальної мережі.

4.5 Випуск стабільної версії застосунку

Кожен застосунок, що є опублікованим на Google Play Market, повинен правильно працювати на будь-якому пристрої, на будь-якій операційній системи з базою Android та на пристрої з будь-якою конфігурацією. Для цього необхідно провезти комплексне тестування застосунку, що потребує значну кількість часу та ресурсів.

Як тільки кількість зауважень до застосунку з боку користувачів стала мінімальною, це означає, що застосунок готовий до публікації.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи магістра було:

- досліджено сучасні методи та інструменти розробки застосунків для операційної системи Android;
- оглянуто список схожих за функціоналом застосунків та зроблений аналіз їх переваг та недоліків;
- описано функціональні вимоги до застосунку;
- розроблено макети застосунку для відслідковування виконання вправ;
- розроблено архітектуру та базу даних застосунку;
- реалізовано застосунок та розміщено його програмний код у відкритому доступі.

У процесі розробки були використані сучасні інструменти для створення застосунків для операційної системи Android. Розроблений застосунок можна використовувати не залежно від статусу підключення до мережі Інтернет, що робить його користування можливим у будь-яких умовах. Застосунок має великий простір до покращення та вдосконалення, а також для створення нового функціоналу на базі розробленого.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Застосунок Fitbod Workout & Fitness Plans – Google Play Market [Електронний ресурс] – Режим доступу: https://play.google.com/store/apps/details?id=com.fitbod.fitbod
2. Застосунок GymKeeper – Gym Workout Log – Google Play Market [Електронний ресурс] – Режим доступу: https://play.google.com/store/apps/details?id=com.kg.app.sportdiary
3. Застосунок Workout Tracker & Gym Plan Log – Google Play Market [Електронний ресурс] – Режим доступу: https://play.google.com/store/apps/details?id=com.imperon.android.gymapp
4. Застосунок Hevy – Gym Log Workout Tracker – Google Play Market [Електронний ресурс] – Режим доступу: https://play.google.com/store/apps/details?id=com.hevy
5. Застосунок Workout Diary – Trainings plan – Google Play Market [Електронний ресурс] – Режим доступу: https://play.google.com/store/apps/details?id=com.nicola.sporttrack
6. Застосунок FitHero – Gym Workout Tracker – Google Play Market [Електронний ресурс] – Режим доступу: https://play.google.com/store/apps/details?id=com.fnp.fithero
7. Develop Android apps with Kotlin [Електронний ресурс] – Режим доступу: https://developer.android.com/kotlin
8. Gradle Kotlin DSL Primer [Електронний ресурс] – Режим доступу: https://docs.gradle.org/current/userguide/kotlin_dsl.html
9. Android Developer [Електронний ресурс] – Режим доступу: https://developer.android.com/
10. View binding, Jetpack Compose [Електронний ресурс] – Режим доступу: https://developer.android.com/topic/libraries/view-binding
11. Material Components for Android [Електронний ресурс] – Режим доступу: https://github.com/material-components/material-components-android

12. Kotlin Serialization Guide [Електронний ресурс] – Режим доступу: https://github.com/Kotlin/kotlinx.serialization/blob/master/docs/serialization-guide.md
13. High-order functions and lambdas [Електронний ресурс] – Режим доступу: https://kotlinlang.org/docs/lambdas.html
14. Android Jetpack: Modern Android [Електронний ресурс] – Режим доступу: https://developer.android.com/jetpack
15. Save data in a local database using Room [Електронний ресурс] – Режим доступу: https://developer.android.com/training/data-storage/room
16. About Figma [Електронний ресурс] – Режим доступу: https://www.figma.com/about/
17. Material 3 Design Kit [Електронний ресурс] – Режим доступу: https://www.figma.com/community/file/1035203688168086460
18. Icons8: Material Design Icons [Електронний ресурс] – Режим доступу: https://www.figma.com/community/plugin/740272380439725040/Material-Design-Icons
19. Security-first diagramming for teams [Електронний ресурс] – Режим доступу: https://www.drawio.com/
20. FirebaseUI for Storage [Електронний ресурс] – Режим доступу: https://firebaseopensource.com/projects/firebase/firebaseui-android/storage/readme/
21. Бейлор Черрі – Why Are Social Media Sites Blue? How Color Psychology Drives Engagement [Електронний ресурс] – https://www.bluleadz.com/blog/why-are-social-media-sites-blue
22. Еверест Гордон. Базові моделі структури даних, пояснені на типових прикладах. – 1976. – С. 39-46.