

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЧЕРНІГІВСЬКА ПОЛІТЕХНІКА»

Навчально-науковий інститут електронних та інформаційних технологій
Кафедра інформаційних та комп'ютерних систем

Допущено до захисту

Завідувач кафедри ІКС
к.е.н., доцент Базилевич В.М.

" ____ " _____ 2020р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

**МОБІЛЬНИЙ ДОДАТОК ДЛЯ СПОРТЗАЛУ НА БАЗІ ОС ANDROID 3
ВИКОРИСТАННЯМ ФРЕЙМВОРКУ LUMEN – FRONT-END**

Спеціальність: 123 – Комп'ютерна інженерія

Виконавець:

студент гр. КІ-162

Патай А. В. _____

(підпис)

Керівник:

ст. викладач

Казнадій С.П. _____

(підпис)

Чернігів 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЧЕРНІГІВСЬКА ПОЛІТЕХНІКА»

Навчально-науковий інститут електронних та інформаційних технологій
Кафедра інформаційних та комп'ютерних систем

ЗАТВЕРДЖУЮ:
Завідувач кафедри ІКС
к.е.н., доцент Базилевич В.М.

“ ____ ” _____ 2020 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ (ПРОЕКТ) ЗДОБУВАЧА
ВИЩОЇ ОСВІТИ

Патай Альона Валеріївна

Тема роботи: Мобільний додаток для спортзалу на базі ОС Android з
використанням фреймворку Lumen – front-end

*Тему затверджено наказом ректора
від " ____ " _____ 2020р. № _____*

1. Вхідні дані до роботи

Наукові дослідження провідних вчених світу в області розробки мобільних додатків, технічна документація додатків під ОС Android, технічна документація середовища розробки Android Studio.

2. Зміст розрахунково-пояснювальної записки

Аналіз завдання «Розробка Android додатку для спортзалу» включає в себе: аналіз предметної області, аналіз подібних систем, постановку завдання на розробку мобільного додатку.

3. Перелік графічного матеріалу (у разі необхідності)

11 слайдів презентації

4. Календарний план

№	Назва етапів роботи	Термін виконання	Примітки
1.	Визначитися з тематикою роботи	10.02.2020	
2.	Знайомство з проблемою, інформаційними джерелами, аналіз існуючих рішень	06.03.2020	
3.	Аналіз задачі створення системи	01.04.2020	Звіт
4.	Розробка сайту	18.05.2020	
5.	Залік з переддипломної практики	18.05.2020	Звіт
6.	Реалізація системи	01.06.2020	
7.	Підготовка текстової частини і слайдів	01.06.2020	Звіт
8.	Попередній захист дипломної роботи	01.06.2020	
9.	Завершення оформлення, рецензування, перевірка на плагіат	15.06.2020	
10.	Захист дипломної роботи	З 15.06.2020	

Завдання підготував:

керівник _____ Казнадей С.П.

(підпис) (прізвище, ім'я, по батькові)

«__» _____ 2020р.

Завдання одержав:

студент _____ Патай А.В.

(підпис) (прізвище, ім'я, по батькові)

«__» _____ 2020р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання кваліфікаційної роботи бакалавра

ст. гр. КІ-162 Патай А. В.

Тема роботи: Мобільний додаток для спортзалу на базі ОС Android з використанням фреймворку Lumen – front-end

Запропоновані технічні та експлуатаційні результати роботи

Необхідно розробити клієнтську частину мобільного додатку для спортзалу, який повинен забезпечити користувача наступним функціоналом:

- зручний та зрозумілий веб-інтерфейс;
- можливість авторизації та входу в особистий кабінет;
- перегляд доступного обладнання;
- можливість обрати тренера;
- підбір індивідуального плану тренувань.

Клієнтська частина мобільного додатку буде розроблена за допомогою середовища розробки Android Studio та взаємодітиме з серверною частиною.

Обсяг текстової та графічної інформації

Робота обсягом 50-80 стор. Формату А4 і 10-12 електронних слайдів

Орієнтовна трудомісткість робіт – 80 людино-годин.

Планові терміни по етапах

Попередній захист з наданням чистового варіанту роздрукованих текстів та ілюстративного матеріалу в повному обсязі .

Плановий термін захисту роботи

з 15 червня 2020 р.

Робота планується до захисту на засіданні ЕК.

Керівник роботи

Казнадій С.П.

Дата видачі завдання

15 березня 2020 р

АНОТАЦІЯ

Кваліфікаційна робота бакалавра, 56 с., 19 рис., 2 табл., 7 джерел.

Об'єктом розробки було обрано мобільний додаток, а саме клієнтську частину для додатку спортзалу.

Додаток було розроблено для операційної системи Android, з використанням технологій Java, також застосовувалися інструменти Android Studio.

Метою даної роботи є створення мобільного додатку для спортзалу. Завдяки йому стане можливим перегляд можливостей спортзалу без фізичної присутності в ньому. Також можна буде переглядати всі послуги, що доступні для замовлення, і безпосередньо, робити замовлення.

Результатом роботи є реалізація таких можливостей додатку:

- авторизація у власному кабінеті;
- вибір тренера;
- вибір дієти;
- додавання тренувань за власним графіком;
- додаткові функції роботи з особистим кабінетом користувача (реєстрація, зміна пароллю, перегляд обраного).

Подальша розвиток можливий в сторону розширення функціональних можливостей, покращення інтерфейсу та розробки серверної частини додатку.

Робота має практичну цінність. Розрахунок економічної цінності не проводився.

ANDROID SDK, JAVA, API, АНДРОІД ДОДАТОК, POSTGRESQL

ABSTRACT

Bachelor's qualification work, 56 p., 19 images, 2 tables, 7 sources.

The object of development was a mobile application, namely the client part for the gym application.

The application was developed for the Android operating system, using Java technologies, and Android Studio tools were also used.

The purpose of this work is to create a mobile application for the gym. Thanks to him, it will be possible to view the capabilities of the gym without a physical presence in it. You can also view all the services available for ordering and place orders directly.

The result is the implementation of the following features of the application:

- Authorization in own office;
- selection of a coach;
- choice of diet;
- adding trainings according to your own schedule;
- additional functions for working with the user's personal account (registration, password change, viewing favorites).

Further development is possible in the direction of expanding functionality, improving the interface and developing the server part of the application.

The work has practical value. The calculation of economic value was not performed.

ANDROID SDK, JAVA, API, ANDROID APPLICATION, POSTGRES SQL

ЗМІСТ

ВСТУП.....	9
1.1 Аналіз предметної області.....	11
1.1.1 Поняття «додатки».....	11
1.1.2 WEB додаток	11
1.1.3 Мобільні додатки	13
1.2 Вивчення існуючих систем	14
1.2.1 Порівняння існуючих додатків.....	14
1.2.2 Уточнення загальної моделі	18
1.3 Побудова базової моделі предметної області.....	19
1.3.1 Опис проблеми	19
1.3.2 Концептуальна модель предметної області	19
1.3.3 Аналіз вимог до інтерфейсу системи	20
1.3.4 Завдання додатку	20
1.4 Постановка задачі	21
1.4.1 Функціональні вимоги	22
1.4.2 Нефункціональні вимоги	22
1.4.3 Обмеження задачі.....	22
1.5 Висновки до розділу 1	22
2.1 Вибір технічних засобів побудови системи.....	23
2.1.1 Вибір мови програмування	23
2.1.2 Вибір середовища розробки	26
2.1.3 Вибір СУБД.....	29
2.2 Архітектура ІКС	30
2.3 Розробка програмної системи.....	31
2.3.1 Структура програмної підсистеми	31
2.3.2 Розробка бази даних.....	31

2.3.3 Розробка класів сутностей системи.....	33
2.3.4 Розробка WEB-компоненту системи.....	33
2.4 Висновки до розділу 2.....	35
3 Реалізація додатку.....	36
3.1 Результати розробки пакетів.....	36
3.2 Результати розробки класів.....	37
3.2.1 Клас MainActivity.....	37
3.2.2 Клас AccountActivity.....	39
3.2.3 Клас TrainerActivity.....	39
3.2.4 Клас DietActivity.....	39
3.2.5 Клас InventarActivity.....	39
3.2.7 Клас Info.....	40
3.2.8 Клас User.....	40
3.3 Реалізація інтерфейсу користувача.....	41
3.3.1 Екран запуску.....	41
3.3.2 Особистий кабінет.....	44
3.3.3 Екран «Дієти».....	46
3.3.4 Екран «Тренери».....	49
3.3.5 Екран відображення інформації про тренера.....	52
3.3.6 Екран «Інфо».....	53
3.4 Висновки до розділу 3.....	54
ВИСНОВКИ.....	55
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56

ВСТУП

Мобільний додаток – це певне програмне забезпечення, що призначене для роботи на гаджетах, планшетах та інших мобільних пристроях. Більшість мобільних застосунків мають бути встановлені на самому пристрої або ж можуть бути завантажені на нього з інтернет-магазинів мобільних додатків. Вони можуть бути застосовані для покращення чи пришвидшення виконання тих або інших задач.

В Україні, як і в усьому світі в наші дні, «спортивний» бізнес має досить велику популярність та немалу конкуренцію, а одним з методів заохочення клієнтів та посилення іміджу спортзалів та фітнес-клубів є модернізація обслуговування. Згідно з дослідженнями компанії з вимірювання та аналітики ЗМІ – ComScore, що надає маркетингові дані та аналітику для підприємств, у 2019 році користувачі витрачають 85% свого «мобільного» часу на використання застосунків у порівнянні з мобільними або веб-браузерами. Також зазначимо, що додаток забезпечує прямий доступ від екрану мобільного пристрою до певного сервісу, і взаємодія з ним, як правило, є простішою і швидшою, у порівнянні з веб-додатком.

Об'єктом розробки є мобільний додаток для комунікації з спортзалом. Головний принцип роботи даного додатку полягає в наступному: кожен клієнт залу може зареєструватися та отримати доступ до вибору власного тренера, дієти, бажаного інвентарю та до перегляду клієнтської інформації залу, а також налаштування власного календаря тренувань. Всі дані, що відображаються, завантажуються з бази даних, та можуть бути змінені тільки розробником.

Мобільний додаток для залу в змозі оптимізувати робочі процеси і зменшити витрати на робочий персонал, примножити клієнтську базу; використання додатку зменшить навантаження на персонал та час для

обслуговування клієнтів, це, у свою чергу, збільшить кількість замовлень та доходи шляхом збільшення клієнтів та зменшення витрат.

1. АНАЛІЗ ЗАДАЧІ РОЗРОБКИ ДОДАТКУ

1.1 Аналіз предметної області

1.1.1 Поняття «додатки»

Додатками вважають багатофункціональні застосунки, що можуть бути використані людиною в будь-якій сфері її діяльності, для пришвидшення чи автоматизації роботи. При цьому, як показує практика, зменшуються затрати на персонал та покращується ефективність виконаної роботи.

В сучасному світі кожен має змогу користуватися можливостями, що надає «всесвітня павутина». І використання смартфонів – повсякденна річ, адже сьогодні смартфон – не лише пристрій для комунікації, але і плеєр, камера, жорсткий диск, пошта, доступ до Інтернету. Отже, є кілька варіантів використання додатків – тобто адаптація сайту до розмірів і розширення екрану гаджету або мобільний додаток, який і було обрано об'єктом розробки.

1.1.2 WEB додаток

Отже, адаптивний сайт або ще часом називають WEB додаток – це просто сайт, який автоматично пропорційно змінює свій стиль та відображення залежності від розмірів екрану гаджета. Однак, добре адаптувати сайт під мобільні пристрої, не означає просто вмістити увесь його вміст на невеличкий екран, потрібно щоб у повній мірі зберігалися функціональність та юзабіліті сайту для відвідувачів.

Необхідність мобільного сайту – недоречно навіть обговорювати, тому що поширення доступних мобільних пристроїв, зручні мультисенсорні екрани, вільний, часом безкоштовний, вихід в інтернет, легкий обмін інформацією – разом, неймовірно збільшило аудиторію потенційних клієнтів будь-якого бізнесу. З цих самих причин неадаптивний сайт сьогодні порівнюють з доісторичним архаїзмом, і в кого він помічений – відкрито буде висміяний.[1]

Плюси WEB додатку:

- Веб-програми набагато легше підтримувати, оскільки вони мають загальну базу кодів на кількох мобільних платформах.
- Веб-додатки можуть бути оброблені таким чином, щоб бути сумісними з будь-яким старим мобільним пристроєм.
- Веб-додатки не вимагають від розробників додавати свою програму в будь-який магазин додатків для схвалення. Звичайно, Apple App Store містить список веб-програм.
- Оскільки веб-програми не повинні бути схвалені на ринку додатків, вони можуть бути випущені в будь-який час і в будь-якій формі відповідно до налаштувань розробника.
- Користувачам не потрібно відвідувати магазини додатків, щоб завантажити або оновлювати веб-додатки, оскільки найновіша версія завжди завантажується, коли користувач отримує доступ до веб-програми.

Мінуси WEB додатку:

- Мобільні веб-програми мають обмежений доступ до функцій пристрою.
- Веб-додатки, які підтримують кілька мобільних веб-переглядачів, можуть виявитись дорогою пропозицією для розробників, тому що це підвищує витрати на їх розробку та підтримку.
- Різні користувачі працюють з різними мобільними браузерами. Це може ускладнити для вас ведення обліку своїх моделей використання, а також забезпечити підтримку для всіх.
- Користувачам може виявитись важко знайти веб-додаток, оскільки його не систематично перелічено в жодному магазині додатків.

– Оскільки для веб-програм не існує регуляризованої системи контролю якості, користувачам не завжди можна гарантувати безпеку та безпеку додатка[2].

1.1.3 Мобільні додатки

Мобільний або нативний додаток зберігається на самому пристрої, і, хоча більшість людей більше знайомі з веб-переглядачами та завантаженням своїх програм, розробник не може гарантувати, що кожен користувач завжди використовує ту ж версію програми[2]. Після установки на смартфон, додаток стає певним інструментом для безпосередньої роботи з користувачем, адже він розрахований на неодноразове використання. Деякі додатки можуть значно пришвидшити виконання повсякденних справ, деякі – виконання важливої роботи. При цьому, ми прекрасно розуміємо, якщо додаток закриває яку-небудь потребу користувача, то він набуває широкої популярності серед аудиторії.

До плюсів мобільного додатку можна віднести:

- можливість взаємодіяти з додатком різними способами;
- об'єднання або навпаки, розмежування, доступу до функціональних частин пристрою;
- можливість збереження контенту , призначеного для користувача, проведення ускладнених обчислень, чим забезпечений високий рівень персоналізації;
- використання «офф-лайн» - без використання Інтернету.

Проте є і мінуси:

- вимоги програми до ОС пристрою, на який необхідно встановити додаток;
- необхідність установки перед використанням.

Зважаючи на наведені аргументи, можна зробити висновок, що мобільний додаток – просто незамінна річ в повсякденному житті кожної людини, але

необхідно з розумінням ставитися до розробки таких додатків і з максимальною точністю виконувати поставлені перед собою цілі у розробці, адже користувачі мають отримувати якісний продукт, що відповідає їх вимогам.

1.2 Вивчення існуючих систем

На даний момент уже розроблені декілька програмних продуктів для спортзалів, проте більшість з них розроблені у вигляді особистого спортзалу в смартфоні, тобто для занять вдома. Ті ж програми, що є саме мобільними версіями спортзалу, є закордонного виробництва, а значить абсолютно не орієнтовані на клієнтів спортзалів України. Для прикладу наведемо три програми, які мають найбільшу популярність.

1.2.1 Додаток «SworKit Fitness»

Мобільний додаток для Android та iOS «SworKit Fitness» (див. рис. 1.2) надає тренування для жінок, чоловіків, дітей без тренажерних залів (що уже є недоліком в порівнянні з програмою, що розроблюється), натомість фітнесом можна займатися вдома, присутні програми для тренувань. До переваг можна віднести велику кількість комбінацій вправ, засікати час тренування, можна створювати свій план тренувань і дотримуватись його, відео-ролики з тренуваннями для повторень. Основними недоліками застосунку, зважаючи на відгуки клієнтів, є необхідність оплачувати ліцензію після пробної 30-денної версії, а також нарікання на тимчасову чи повну недоступність відео-вправ.

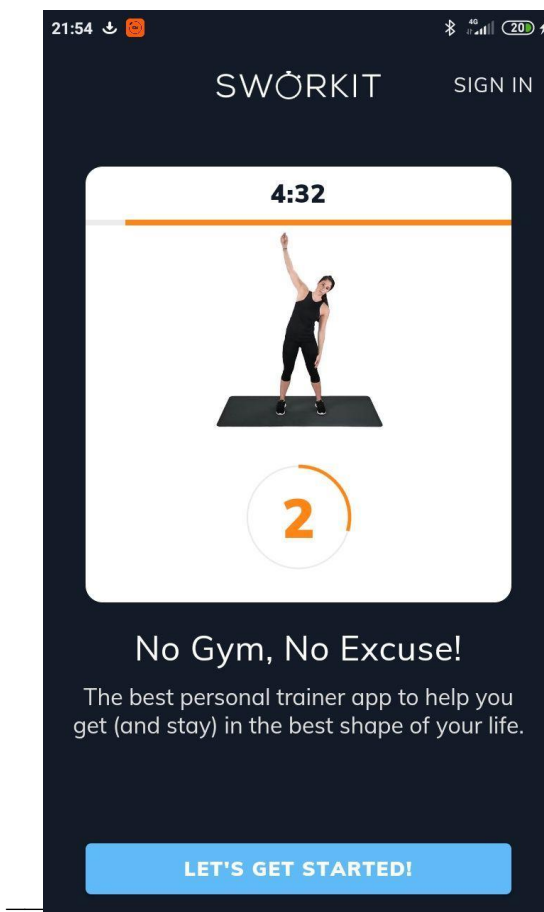


Рисунок 1.1 – Додаток «Sworkit Fitness»

1.2.2 Додаток «Fitness&Bodybuilding»

«Fitness&Bodybuilding» (див. рис. 1.3) теж можна віднести до застосунків без використання спортзалу. В ньому поєднаний зручний інтерфейс, можливість створити свою програму тренувань, присутні відео-ролики для виконання вправ, вбудований таймер тощо. Серед недоліків можна виділити необхідність покупки платної ліцензії, відсутність української мови, часті збої в роботі програми та анімацій.

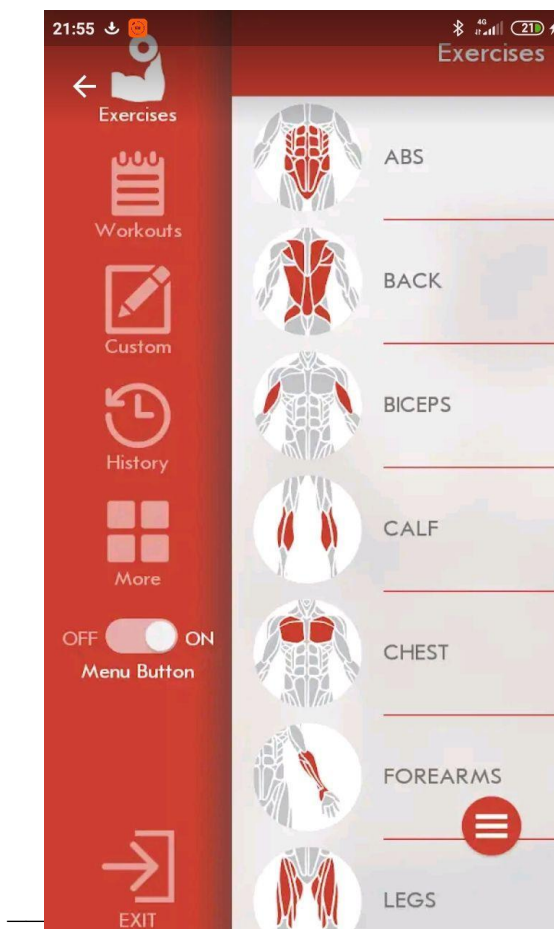


Рисунок 1.2 – Додаток «Fitness&Bodybuilding»

1.2.3 Додаток «Fitness Trainer FitPro»

Програма «Fitness Trainer FitPro»(див. рис. 1.4) схожа до вищенаведених програм. Програма дає можливість обрати вправи з понад 200 простих та ускладнених програм тренувань, різні вправи для чоловіків та жінок, відслідковування особистого результату, календар занять і інше. На відміну від попередніх програм програма адаптована під недосвідченого користувача. Основний недолік – програма англійською мовою, що не завжди є плюсом у використанні.

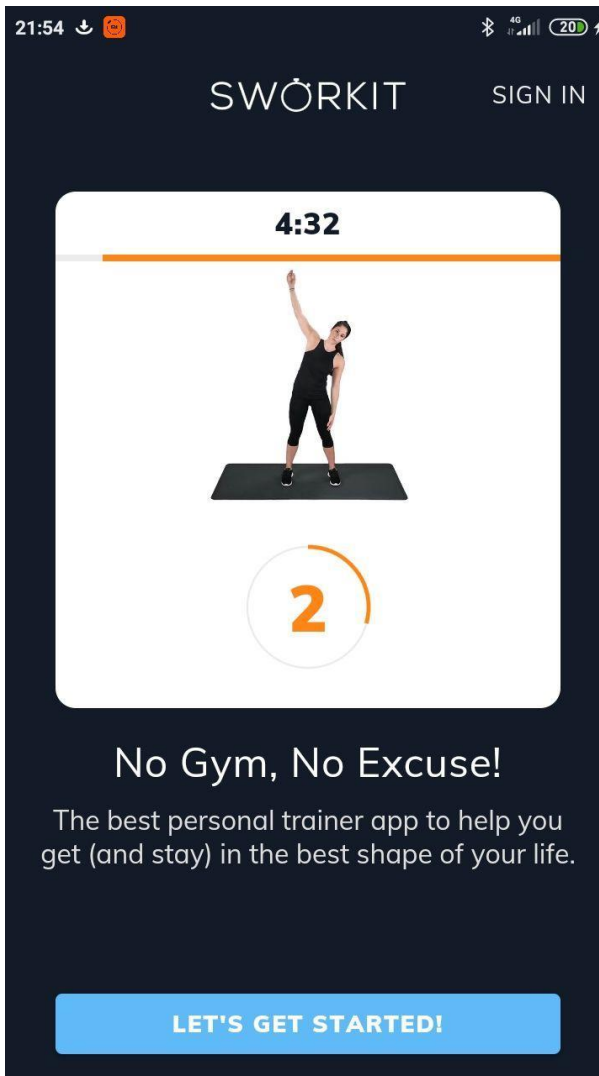


Рисунок 1.3 – «Fitness Trainer FitPro»

Порівняння розглянутих програм і запланованого проекту представлено в таблиці 1.1.

Таблиця 1.1 – Порівняння застосунків

Критерій	Sworkit Fitness	Fitness & Bodybuildi ng	Fitness Trainer FitPro	Розроблюва- ний проект
Простота користування	-	-	+	+
Прив'язка до певного спортзалу чи мережі залів	-	-	-	+
Власний кабінет	-	-	+	+
Список тренувань	+	+	+	+
Можливість займатись спортом вдома	+	+	+	+
Конфіденційність	+	-	+	+
Календар	-	-	+	+
Можливість самостійно обрати тренера	-	-	-	+

1.2.4 Уточнення загальної моделі

Проаналізувавши вже існуючі системи і порівнявши їх, можна прийти до висновку, що кожна має свої переваги і свої недоліки. Система, що розроблюється, повинна поєднати в собі їх переваги і, за можливості, не допустити схожих недоліків наскільки це можливо. Узагальнимо, що система має бути простою і зрозумілою для використання, маючи при цьому необхідний функціонал. Її користувачі з різних суспільних груп, отже необхідно забезпечити виконання основних функцій користувача додатку (клієнту

спортзалу), публічний доступ до частини інформації через Інтернет, але збереження

1.3 Побудова базової моделі предметної області

1.3.1 Опис проблеми

Отже, за умовою необхідно розробити мобільний клієнт-серверний додаток на базі ОС Android, де клієнт зможе переглядати загальну інформацію про спортзал, керувати власним кабінетом, відстежувати особисті досягнення, робити запис на тренування.

Програмний застосунок має забезпечувати наступні функціональні можливості:

1. Відображення сторінки входу в особистий кабінет з можливістю входу/реєстрації.
2. Зчитування інформації з полів вводу.
3. Додавання даних в базу даних на сервері.
4. Зчитування даних з баз даних на сервері.
5. Відображення інформації по натисканню на кнопку/маркер.

1.3.2 Концептуальна модель предметної області

В результаті аналізу предметної області створена концептуальна модель ПО, зображена на рисунку 1.1, що містить сутності та визначає зв'язки між ними. Вона створена без орієнтації на конкретні програмні та технічні засоби виконання задач ПО, які будуть додаватися в подальшому процесі проектування системи.

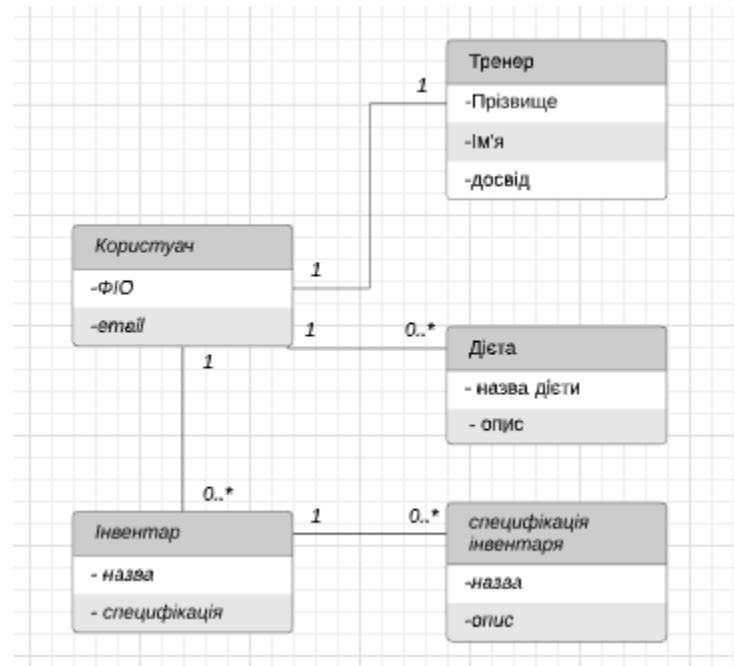


Рисунок 1.4 – Концептуальна модель предметної області

1.3.3 Аналіз вимог до інтерфейсу системи

Користувачами системи є клієнти залу, тобто люди різних вікових категорій, статі, освіти, фізичних можливостей, рівня комп'ютерного досвіду та досвіду користування смартфоном. До уваги береться і те, що система не призначена для ігор чи розваг.

Зважаючи на ці умови робимо висновок, що інтерфейс повинен бути стриманим, без зайвих деталей і простим у використанні. Текст має бути читабельним, середнього розміру, мова інтерфейсу – українська/англійська. З кожної сторінки сайту повинен бути доступ до загального меню.

1.3.4 Завдання додатку

Сформулюємо основні вимоги до системи та її функціональності. Система повинна виконувати такі функції:

- Доступ до індивідуального кабінету;
- Перегляд календаря;

- Збереження відміток в календарі;
- Доступ до меню з усіх сторінок;
- Відображення інформації про тренерів;
- Можливість запису на тренування;
- Можливість обрати дієту;
- Редагування та видалення даних;
- Пошук за запитом.

Додатком буде користуватися наступний тип користувачів:

- Клієнт – робота з календарем, додавання/видалення відміток; вибір власного тренера, можливість вибору дієти, запис на тренування; пошук за запитом. Діаграма варіантів використання для клієнта представлена на рисунку 1.2.

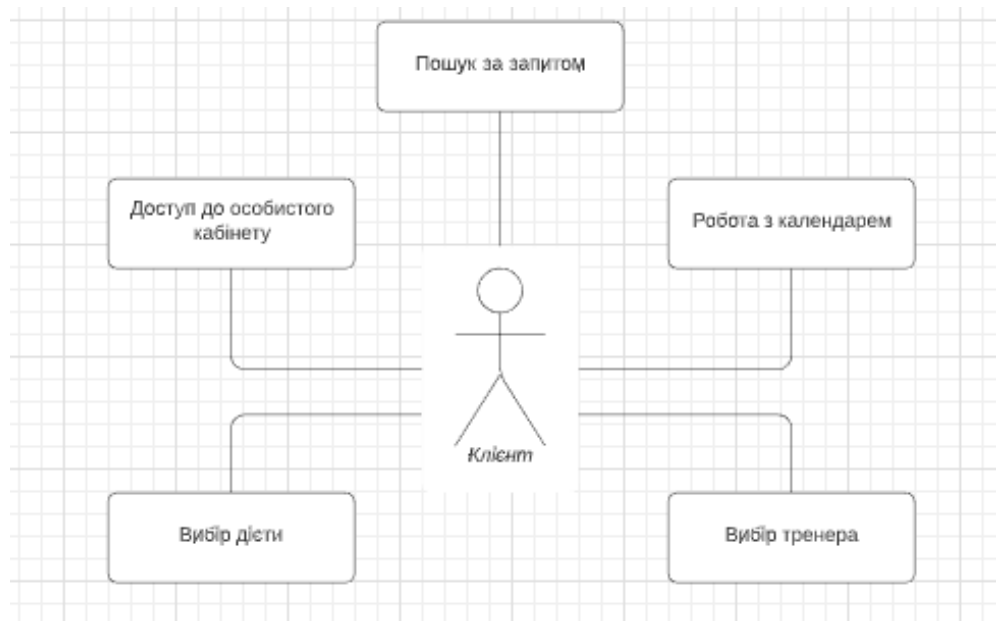


Рисунок 1.5 – Діаграма варіантів використання для клієнта

1.4 Постановка задачі

Після проведеного аналізу предметної області та огляду вже існуючих застосунків, необхідно визначити мету і завдання роботи.

1.4.1 Функціональні вимоги

Згідно з аналізом поставленого завдання впливає, що:

- система повинна відображати всі дані;
- можливість обрати тренера;
- можливість керувати власним кабінетом;
- зручний інтерфейс.

1.4.2 Нефункціональні вимоги

Згідно з аналізом поставленого завдання можна виокремити нефункціональні вимоги для розширення:

- оптимальна швидкодія застосунку;
- логічне функціональне та структурне розміщення складових на сторінці застосунку;

1.4.3 Обмеження задачі

Поставлена задача має певні обмеження. Застосунок має бути написаний в середовищі розробки Android Studio. Необхідно підтримувати гаджети з версією Android 6.0 та вище.

1.5 Висновки до розділу 1

При роботі з першим розділом був проведений аналіз задачі розробки застосунку, виконано порівняння аналогів з виокремленням недоліків та переваг. За результатами проведеного аналізу було розглянуто предметну область роботи, а також сформовані вимоги до створюваної системи.

2 РОЗРОБКА ДОДАТКУ

2.1 Вибір технічних засобів побудови системи

Вибір технічних засобів включає в себе вибір мови програмування та вибір середовища розробки мобільного додатку. Розглянемо їх.

2.1.1 Вибір мови програмування

За даними загальнодоступного ресурсу Вікіпедії, мов програмування на сьогоднішній день налічується майже 240 різновидів, кожна з яких має свої беззаперечні переваги та прикрі недоліки. Проте не всі вони підходять для написання мобільних додатків. До найбільш підходящих мов можна віднести Java та Kotlin, їх і розглянемо.

2.1.1.1 Java

Java – об'єктно-орієнтована мова програмування, розроблена компанією Sun Microsystems. В офіційній реалізації Java-програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи[2]. Програми на Java транслуються в байт-код, що виконується віртуальною машиною Java (JVM) – програмою, яка оброблює байт код і передає інструкції обладнанню як інтерпретатор.

Перевага подібного способу виконання програм в повній незалежності байт-коду від операційної системи і устаткування, що дозволяє виконувати Java-застосунки на будь-якому пристрої, для якого існує відповідна віртуальна машина. Іншою важливою особливістю технології Java є гнучка система безпеки завдяки тому, що виконання програми повністю контролюється віртуальною машиною. Будь-які операції, які перевищують встановлені повноваження програми (наприклад, спроба несанкціонованого доступу до даних або з'єднання з іншим комп'ютером) викликають негайне переривання.

Часто до недоліків концепції віртуальної машини відносять те, що виконання байт-коду віртуальною машиною може знижувати продуктивність програм і алгоритмів, реалізованих на мові Java. Дане твердження було справедливо для перших версій віртуальної машини Java, проте останнім часом воно практично втратило актуальність. Цьому сприяв ряд удосконалень:

- застосування технології трансляції байт-коду в машинний код безпосередньо під час роботи програми (JIT-технологія) з можливістю збереження версій класу в машинному кодї;

- широке використання переносних орієнтованого коду (native-код) в стандартних бібліотеках;

- апаратні засоби, що забезпечують прискорену обробку байт-коду

Основні можливості мови Java:

- автоматичне керування пам'яттю;
- розширені можливості обробки виняткових ситуацій;
- багатий набір засобів фільтрації введення / виведення;
- набір стандартних колекцій, таких як масив, список, стек;
- наявність простих засобів створення мережеских застосунків;
- наявність класів, що дозволяють виконувати HTTP-запити і обробляти відповіді;
- вбудовані в мову засоби створення багатопоточних застосунків;
- уніфікований доступ до баз даних: на рівні окремих SQL-запитів – на основі JDBC, SQLJ, на рівні концепції об'єктів, що володіють здатністю до зберігання в базі даних – на основі Java Data Objects і Java Persistence API;
- підтримка шаблонів (починаючи з версії 1.5);
- паралельне виконання програм.

Java була обрана в якості мови програмування для розробки застосунку, адже вона являється офіційною мовою розробки для платформи Android та має гарну документацію.

2.1.1.2 Kotlin

Kotlin – статично типізована мова програмування, що працює поверх JVM і розробляється компанією JetBrains. Також компілюється в JavaScript[3].

Автори ставили перед собою ціль створити лаконічнішу та типобезпечнішу мову, ніж Java, і простішу, ніж Scala. Наслідками спрощення, порівняно з Scala стали також швидша компіляція та краща підтримка мови середовищем розробки.

Мова розробляється з 2010 року, публічно представлена в липні 2011. Сирцевий код було відкрито в лютому 2012. В лютому було випущено milestone 1, який містив плагін для IDEA. У червні — milestone 2 з підтримкою Android. У грудні 2012 року вийшов milestone 4 та забезпечив підтримку Java 7. В грудні 2015 року з'явився реліз-кандидат версії 1.0, а 15 лютого 2016 року відбувся реліз версії 1.0.

Синтаксис мови схожий на Pascal, TypeScript, Nahe, PL / SQL, F #, Go і Scala, C ++, Java, C # і D. При оголошенні змінних і параметрів, типи даних вказуються після назви (роздільник двокрапка). Крапка з комою як роздільник операторів є не обов'язковою, також як в Scala і Groovy, в більшості випадків перекладу рядка досить щоб компілятор зрозумів, що вираз закінчилося. Крім об'єктно-орієнтованого підходу, Kotlin також підтримує процедурний стиль з використанням функцій. Як і в мовах C / C ++ / D, точкою входу в програмі є функція “main”, яка приймає масив параметрів командного рядка. Програми на Kotlin також підтримують Perl і Unix / Linux shell стиль інтерпретації рядків.

З 17 травня 2017 року входить в список офіційно підтримуваних мов для розробки застосунків для платформи Android.

Отже, Kotlin зовсім нещодавно став другою офіційною мовою для платформи Android, але він має гарні перспективи в майбутньому.

2.1.2 Вибір середовища розробки

Середовище розробки грає настільки важливу роль в розробці, як і мова програмування. Для того, щоб зробити вдалий вибір, необхідно враховувати особливості об'єкту, що розроблюється.

2.1.2.1 Ключові особливості Android

Android – операційна система і платформа для мобільних телефонів та планшетних пристроїв, створена Google на базі ядра Linux. Підтримується альянсом Open Handset Alliance (ОНА)[4].

Android є найпоширенішою операційною системою (ОС) для мобільних пристроїв - телефонів і планшетів. Дана система має безліч характерних рис, які роблять її популярною і привабливою для великої кількості користувачів по всьому світу. Операційна система Android є невимогливою і здатна працювати на різних конфігураціях. Саме тому більшість світових виробників оснащують свої пристрої даної ОС, оскільки інші програмні продукти призначені для окремих апаратів, відповідних певної специфікації.

Така гнучкість Android пов'язана з тим, що система побудована на ядрі Linux, що має відкритий програмний код, що дає необмежені можливості розробникам. Android може бути запуснений на пристроях, що мають об'єм оперативної пам'яті менше 256 Мб. Найбільш нові версії системи вимагають 512 Мб оперативки, що також є невеликим значенням для сучасних апаратів. Система не вимагає наявності високопродуктивного процесора і може працювати на пристроях, оснащених ядром з частотою 600 МГц.

Операційна система дає можливість установки додатків з офіційного репозиторію Google, який надає найбільшу в світі базу програм. Це пов'язано з тим, що кожен розробник може самостійно написати будь-яку програму для

апарату і розмістити її в магазині. Можливість також реалізована завдяки відкритості операційної системи. Варто відзначити, що додатки на пристрої під управлінням Android можуть бути встановлені як безпосередньо з телефону або планшета, так і через комп'ютер шляхом завантаження файлу .apk і його подальшої установки на апараті.

«Андроїд» має досить простий і інтуїтивно зрозумілий інтерфейс. Всі потрібні програми розміщуються одночасно на головному екрані і в меню апарату, яке викликається натисканням на центральну сенсорну клавішу або відповідну кнопку на екрані. всі налаштування розташовуються в секції «Налаштування», а кожна дія користувача пояснюється коментарями і підказками при першому запуску апарату. Операційна система швидко реагує на натискання користувача і виробляє установку і скачування потрібних програм і файлів з швидкістю, яка не програє іншим сучасним мобільним ОС. Поширеність серед користувачів України ОС Android зображена на рисунку 1.1.

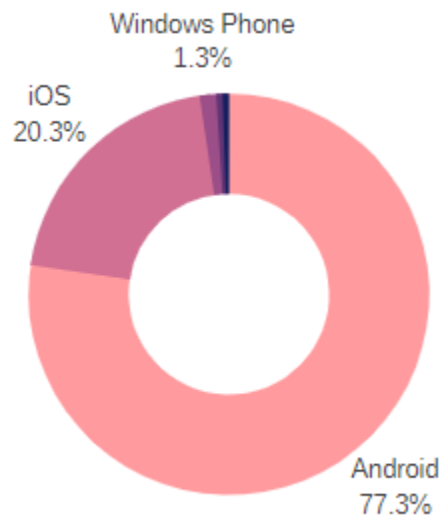


Рисунок 2.1 – Процент використання ОС Android

Зважаючи на ключові особливості ОС Android основним середовищем розробки було обрано Android Studio.

2.1.2.2 Android Studio

Android Studio – інтегроване середовище розробки (IDE) для платформи Android, представлене 16 травня 2013 року на конференції Google I/O менеджером по продукції корпорації Google – Уллі Паверс. 8 грудня 2014 року компанія Google випустила перший стабільний реліз Android Studio 1.0.

Android Studio прийшло на зміну плагіну ADT для платформи Eclipse. Середовище побудоване на базі вихідного коду продукту IntelliJ IDEA Community Edition, що розвивається компанією JetBrains. Android Studio розвивається в рамках відкритої моделі розробки та поширюється під ліцензією Apache 2.0.

Бінарні складання підготовлені для Linux (для тестування використаний Ubuntu), Mac OS X і Windows. Середовище надає засоби для розробки застосунків не тільки для смартфонів і планшетів, але і для носимих пристроїв на базі Android Wear, телевізорів (Android TV), окулярів Google Glass і автомобільних інформаційно-розважальних систем (Android Auto). Для застосунків, спочатку розроблених з використанням Eclipse і ADT Plugin, підготовлений інструмент для автоматичного імпорту існуючого проекту в Android Studio.

Середовище розробки адаптоване для виконання типових завдань, що вирішуються в процесі розробки застосунків для платформи Android. У тому числі у середовище включені засоби для спрощення тестування програм на сумісність з різними версіями платформи та інструменти для проектування застосунків, що працюють на пристроях з екранами різної роздільності (планшети, смартфони, ноутбуки, годинники, окуляри тощо). Крім можливостей, присутніх в IntelliJ IDEA, в Android Studio реалізовано кілька додаткових функцій, таких як нова уніфікована підсистема складання,

тестування і розгортання застосунків, заснована на складальному інструментарії Gradle і підтримуюча використання засобів безперервної інтеграції.

Для прискорення розробки застосунків представлена колекція типових елементів інтерфейсу і візуальний редактор для їхнього компонування, що надає зручний попередній перегляд різних станів інтерфейсу застосунку (наприклад, можна подивитися як інтерфейс буде виглядати для різних версій Android і для різних розмірів екрану). Для створення нестандартних інтерфейсів присутній майстер створення власних елементів оформлення, що підтримує використання шаблонів. У середовище вбудовані функції завантаження типових прикладів коду з GitHub.

До складу також включені пристосовані під особливості платформи Android розширені інструменти рефакторингу, перевірки сумісності з минулими випусками, виявлення проблем з продуктивністю, моніторингу споживання пам'яті та оцінки зручності використання. У редактор доданий режим швидкого внесення правок. Система підсвічування, статичного аналізу та виявлення помилок розширена підтримкою Android API. Інтегрована підтримка оптимізатора коду ProGuard. Вбудовані засоби генерації цифрових підписів. Надано інтерфейс для управління перекладами на інші мови[5].

2.1.3 Вибір СУБД

Розглянувши декілька СУБД, було зупинено вибір на PostgreSQL. PostgreSQL - це система управління базами даних на платформі Unix. Вона дозволяє додавати, вибирати та обробляти дані[6]. Запущений сервер PostgreSQL в змозі управляти великою кількістю баз даних. Також вона має можливість створення власного типу, великий вибір типів даних, вкладені підзапити для полегшення вибірки даних. СУБД підтримує бази даних великого або необмеженого розміру, а максимальний розмір однієї таблиці може бути

32Тб. Вона поступається по швидкодії MySQL, але дає вигреш у решті параметрів.

2.2 Архітектура ІКС

Для кращого представлення роботи майбутнього проекту була створена візуалізація (див. рис.2.2). Архітектура ІКС, що розроблюється, представлена на рисунку 2.3.

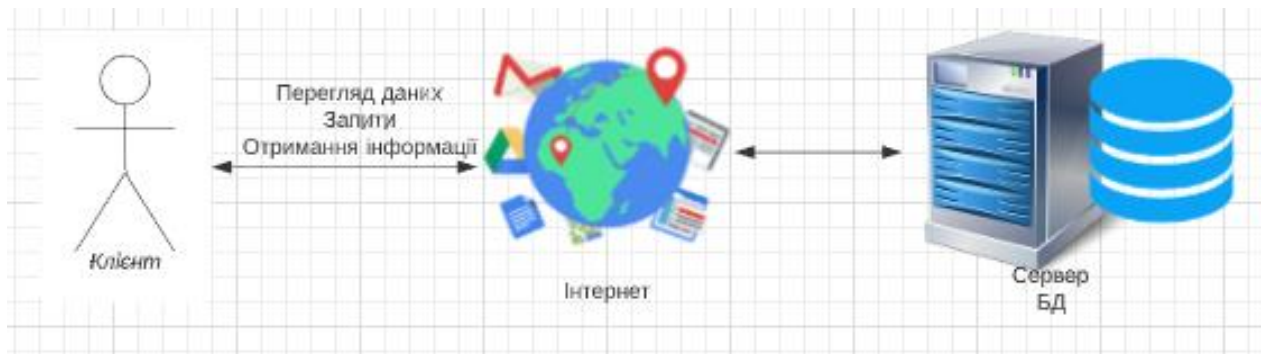


Рисунок 2.2 – Візуальне представлення ІКС



Рисунок 2.3 – Архітектура ІКС

Користувач(клієнт залу) може встановити додаток на власний смартфон з доступом до мережі Інтернет. За допомогою додатку клієнт може мати доступ до даних, які віддалено знаходяться в базах даних на сервері. Через додаток можна буде робити запити для отримання інформації, і відправляти дані для збереження і/або оновлення в базах даних. Зокрема, можна виділити перегляд власних даних, перегляд доступного інвентаря, вільних тренерів, доступних дієт та інформації про зал.

2.3 Розробка програмної системи

2.3.1 Структура програмної підсистеми

Під час проектування системи вона була розподілена на кілька частин (шарів), що дозволяє легко змінити певну не переписуючи весь код (див. рис. 2.4).

Шар доступу до даних – опис таблиць в базах даних, опис їх взаємодії між собою та основні функції що використовуються для обробки даних і зберігаються у кожній з таблиць.

Шар бізнес-логіки – тут знаходиться функціонал додатку; здійснює зв'язок між шаром доступу до даних та шаром відображення, регулюючи виконання запитів від шару відображення.

Шар відображення – дозволяє відображення інформації, надає клієнту способи управління системою за допомогою програмного інтерфейсу.



Рисунок 2.4 – Архітектура програмної системи

2.3.2 Розробка бази даних

Для системи, що розроблюється, необхідна база даних з п'ятьма таблицями: «Users» для збереження інформації про користувача, «Coachs» для зберігання інформації про тренерів, «Diets» для інформації про доступні дієти, «Trainers» для інформації про інвернтар і «Categories» для інформації про категорії інвентаря. Таблиця «Users» буде включати такі колонки: ім'я користувача, електронну адресу, пароль, id зображення. Таблиця «Coschs» міститиме інформацію про тренерів, і матиме такі поля, як порядковий номер(id) тренера, інформація про тренера, рівень досвіду, графік . Таблиця «Diets» включатиме інформацію про дієту, та матиме такі поля: порядковий id, назва дієти, інформація про дієту, id зображення. . Таблиця «Trainers» відображає доступний інвентар, містить поля: порядковий id, ім'я тренера, інформація про інвентар. . Таблиця «Categories» є таким собі розширенням таблиці інвентаря, і включає поля: id категорії, інформація про дану категорію. База даних з таблицями зображена на рисунку 2.5.

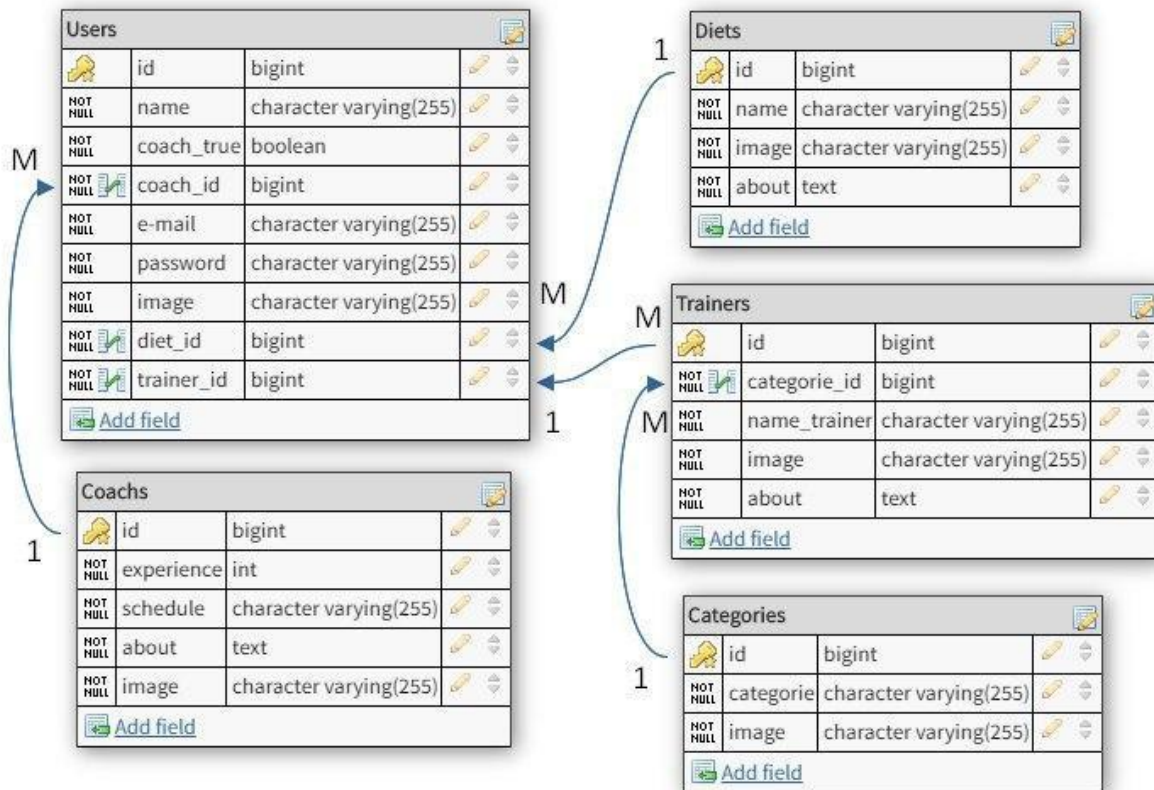


Рисунок 2.5 – База даних

2.3.3 Розробка класів сутностей системи

Було спроектовано 1 основний пакет диплома, де зберігаються основні класи.

На рисунку 2.6 зображено класи, які входять в пакет.

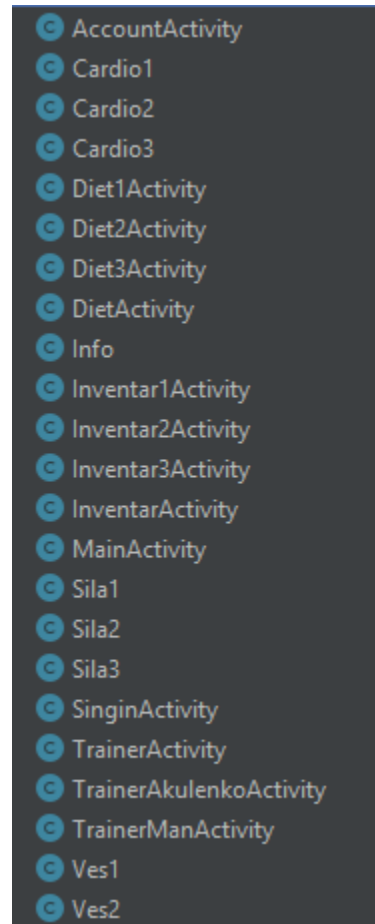


Рисунок 2.6 – Класи в пакеті ui

2.3.4 Розробка WEB-компоненту системи

Користувачі даної системи мають різні вікові категорії, стать, розумові навички, навички користування смартфоном. Інтерфейс буде доволі простий, без зайвих деталей, зрозумілий на інтуїтивному рівні і в той же час буде функціональним.

Інтерфейс буде представлений у єдиному варіанті – для користувача(клієнта залу). Більшу частину екрану займатиме доступна/обрана інформація, знизу екрану розташоване меню з можливістю вибору сторінки. Приблизний інтерфейс для користувача зображений на рисунку 2.7.



Рисунок 2.7 – Ескіз інтерфейсу для користувача

Карту сайту теж розглядаємо в єдиному варіанті – для користувача. Користувач має можливість ввійти до особистого кабінету, в меню обрати сторінку, на яку бажає перейти(тренери, дієти, інвентар, інформація про зал). На кожній сторінці є можливість вибору, в результаті якого відбудеться перехід далі або показ інформації(див. рис. 2.8).

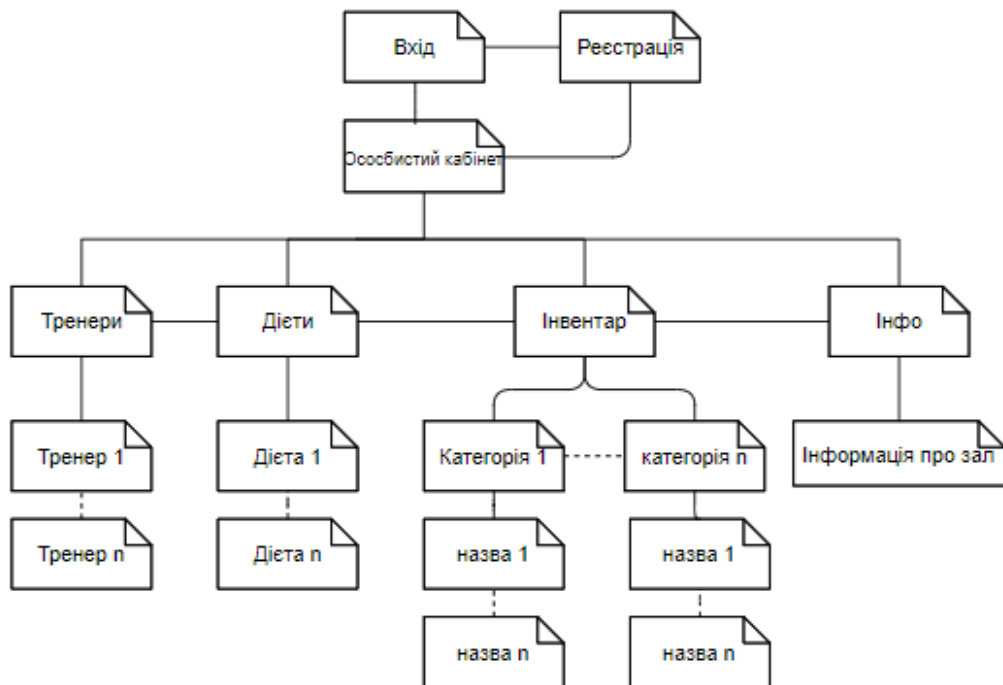


Рисунок 2.8 – Карта сайту для клієнта

2.4 Висновки до розділу 2

В якості основної мови програмування було обрано Java, бо вона являється офіційною мовою для платформи Android, і доволі часто використовується і для інших платформ.

Для розробки додатку було обрано Android Studio, адже вона розроблена на базі IntelliJ Idea, її можна безкоштовно встановити і до того ж має найбільшу підтримку платформи Android.

В розділі спроектована архітектура, вказані пакети та класи, спроектований додаток.

3 РЕАЛІЗАЦІЯ ДОДАТКУ

3.1 Результати розробки пакетів

Для зручності знаходження класів був створений пакет diploma. У таблиці 3.1 представлений список класів.

Таблиця 3.1 – Опис пакету

Назва пакету	Список класів	Призначення
diploma	AccountActivity.java Cardio1.java Cardio2.java Cardio3.java Diet1Activity.java Diet2Activity.java DietActivity.java Info.java InventarActivity.java Inventar1Activity.java Inventar2Activity.java MainActivity.java Sila1.java Sila2.java SinginActivity.java TrainerActivity.java TrainerAkulenko.java TrainerManActivity.java Ves1.java User.java	Відображення інформації для користувача

3.2 Результати розробки класів

Як було вказано вище, класи знаходяться в пакеті diploma, і, відповідно, в кожному класі реалізовані методи , за допомогою яких реалізована програма загалом, а також початкові способи обробки і відображення інформації.

3.2.1 Клас MainActivity

В цьому класі описано екран, який відкривається при запуску додатку. В нашому випадку, це екран авторизації, з можливістю авторизуватись або зареєструватись, якщо ви ще не користувач. На лістингу 3.1 наведена частину коду класу MainActivity.java, що відповідає за вхід і реєстрацію користувача.

Лістинг 3.1 – Частина класу MainActivity.

```
private void showRegisterWinndow() {
    AlertDialog.Builder dialog = new AlertDialog.Builder(this); //создание
диалогового окна

    LayoutInflater inflater = LayoutInflater.from(this);

    View register_window = inflater.inflate(R.layout.register_window, null);

    //получение полей

    final EditText email = register_window.findViewById(R.id.email);// final -
потому что данная переменная вызывается извне и final делает ее константой

    final EditText password = register_window.findViewById(R.id.password);

    dialog.setPositiveButton("Sing In", new DialogInterface.OnClickListener() {

        @Override

        public void onClick(DialogInterface dialogInterface, int which) {

            if (TextUtils.isEmpty(email.getText().toString())) {

                //Snackbar дозволяє виводити помилки, MAKE - створює вікно
повідомлення
            }
        }
    });
}
```

```

                Snackbar.make (root,                "Введите                почту",
Snackbar.LENGTH_SHORT) .show ();

                return; }

                if (password.getText ().toString ().length () < 6) {

                //Snackbar позволяет выводить ошибки, MAKE - СОЗДАЕТ ОКНО
УВЕДОМЛЕНИЯ

                Snackbar.make (root,                "Введите                пароль",
Snackbar.LENGTH_SHORT) .show ();

                return;

                }

                //Регистрация пользователя

                auth.createUserWithEmailAndPassword (email.getText ().toString (),
password.getText ().toString ()) .addOnSuccessListener (new
OnSuccessListener <AuthResult> () {

                @Override

                public void onSuccess (AuthResult authResult) {

                User user = new User ();

                //создаем объекты Юзер, передаем необходимые значения

                user.setEmail (email.getText ().toString ());
                user.setPassword (password.getText ().toString ());

                //передача в БД инфо

                user.child (FirebaseAuth.getInstance ().getCurrentUser ().getUid ()) //ключ
по которому идентифицируем пользователя / емейл

                .setValue (user); //передал в табличку Юзер все данные

                .addOnSuccessListener (new OnSuccessListener <Void> () {

                @Override

                public void onSuccess (Void aVoid) {

```

```

                Snackbar.make (root,                "Добавление                успешно!",
Snackbar.LENGTH_SHORT) .show ();}

        });}

    });

}

    dialog.show ();}}    }

```

3.2.2 Клас AccountActivity

Цей клас призначений для відтворення основних дій , що здійснюються користувачем на сторінці «Особистий кабінет», а також доступу до елементів цієї сторінки.

3.2.3 Клас TrainerActivity

Цей клас призначений для відтворення основних дій , що здійснюються користувачем на сторінці «Тренери», а також доступу до елементів цієї сторінки.

3.2.4 Клас DietActivity

Цей клас призначений для відтворення основних дій , що здійснюються користувачем на сторінці «Дієти», а також доступу до елементів цієї сторінки.

3.2.5 Клас InventarActivity

Цей клас призначений для відтворення основних дій , що здійснюються користувачем на сторінці «Інвентар», а також доступу до елементів цієї сторінки.

3.2.6 Клас InventarActivity

Цей клас призначений для відтворення основних дій , що здійснюються користувачем на сторінці «Інвентар», а також доступу до елементів цієї сторінки.

3.2.7 Клас Info

Цей клас призначений для відтворення основних дій , що здійснюються користувачем на сторінці «Інвентар», а також доступу до елементів цієї сторінки.

3.2.8 Клас User

Даний клас призначений для опису взаємодії списку подій з елементами класу MainActivity.java

Код класу User показано на лістингу 3.2.

Лістинг 3.2 – Клас User

```
public class User extends MainActivity {

    public class User{
        private String email, password;

        public User(){}
        public User(String email, String password){
            this.email=email;
            this.password=password;
        }

        public String getEmail(){
            return email;
        }
        public void setEmail(String email){
            this.email=email;
        }

        public String getPassword(){
            return password;
        }
    }
}
```



```
    }  
    public void setPassword(String password) {  
        this.password=password;  
    }  
}}
```

3.3 Реалізація інтерфейсу користувача

Як було визначено, за обробку і доступ до даних відповідають класи, та методи, що в них визначені. А за відображення даних та інформації відповідають файли розмітки, отже надалі розглянемо інтерфейс саме з цими файлами.

3.3.1 Екран запуску

Найперше, розглянемо екран запуску. Екрану запуску в нашій програмі відповідає сторінка авторизації/реєстрації(див. рис. 3.1). Тут є два поля для вводу даних для авторизації, а також функціональна кнопка “Sing IN” , що викликає вікно реєстрації нового користувача. За відображення інтерфейсу даної сторінки відповідає файл розмітки activity_main(лістинг 3.3)

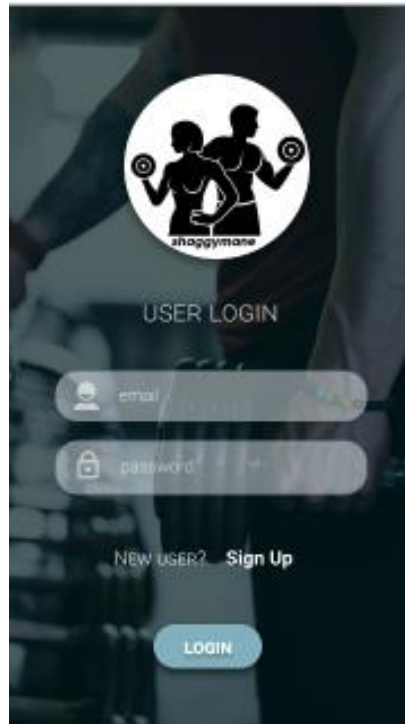


Рисунок 3.1 – Початкова сторінка

Лістинг 3.3 – Лістинг activity_main

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:background="@drawable/fonlekran"
    tools:context=".MainActivity">

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        tools:layout_editor_absoluteX="64dp"
        tools:layout_editor_absoluteY="242dp">

        <ImageView
            android:id="@+id/imageView3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20sp"
            android:visibility="visible"
            app:srcCompat="@drawable/emblem"

```

```

        tools:visibility="visible" />

<TextView
    android:id="@+id/create_new_account"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20sp"
    android:layout_marginBottom="20sp"
    android:fontFamily="sans-serif-smallcaps"
    android:text="user" name="
    android:textAlignment="center"
    android:textColor="#98FFFFFF"
    android:textSize="24sp" />

<EditText
    android:id="@+id/email"
    android:layout_width="match_parent"
    android:layout_height="44dp"
    android:layout_marginLeft="40sp"
    android:layout_marginRight="40sp"
    android:background="#60FFFFFF"
    android:ems="10"
    android:hint="email"
    android:inputType="textEmailAddress"
    android:padding="10sp" />

<EditText
    android:id="@+id/password"
    android:layout_width="match_parent"
    android:layout_height="44dp"
    android:layout_marginLeft="40sp"
    android:layout_marginTop="20sp"
    android:layout_marginRight="40sp"
    android:background="#60FFFFFF"
    android:ems="10"
    android:hint="password"
    android:inputType="textPassword"
    android:padding="10sp" />

<TextView
    android:id="@+id/new_user"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10sp"
    android:fontFamily="sans-serif-smallcaps"
    android:paddingTop="30sp"
    android:text="New User?"
    android:textAlignment="center"
    android:textColor="#98FFFFFF"
    android:textSize="18sp"
    android:textStyle="bold" />

```

```

        <Button
            android:id="@+id/btn_singin"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="100sp"
            android:layout_marginRight="100sp"
            android:background="#6782B1BE"
            android:text="sing
            android:textColor="#A7FFFFFF"
            android:textIsSelectable="false"
            android:textSize="18sp"
        in"

        <Button
            android:id="@+id/btn_login"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="100sp"
            android:layout_marginTop="30sp"
            android:layout_marginRight="100sp"
            android:background="#82B1BE"
            android:text="login"
            android:textColor="#D6E0F1"
            android:textIsSelectable="false"
            android:textSize="18sp"
        />

        <TextView
            android:id="@+id/textView3"
            android:layout_width="match_parent"
            android:layout_height="148dp"
        />
    </LinearLayout>
</RelativeLayout>

```

3.3.2 Особистий кабінет

Далі розглянемо особистий кабінет – сторінка, що відображається у випадку успішної авторизації(див. рис. 3.2). Тут клієнт може редагувати інформацію про себе, вийти з профілю, є доступ до функціонального меню(лістинг 3.4) з можливістю переходу між екранами. До речі, меню доступне з кожного екрану, крім початкової сторінки, а це дозволяє без проблем, напряму звертатись до різних екранів.

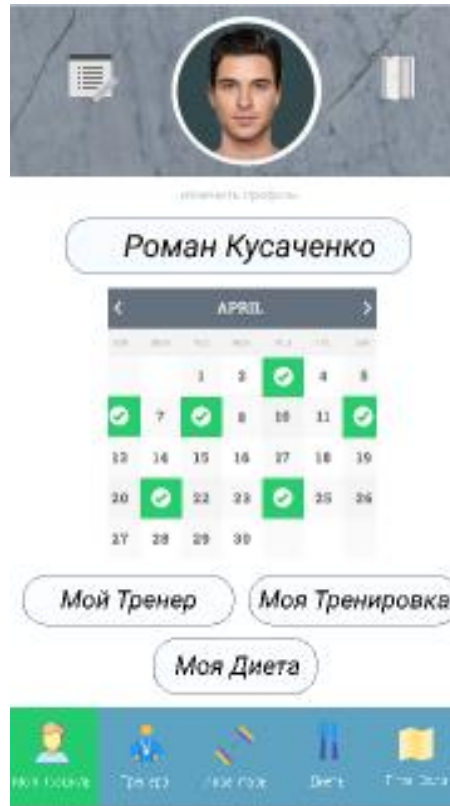


Рисунок 3.2 – Особистий кабінет

Лістинг 3.4 – Лістинг файлу розмітки меню bottom_nav_menu.xml

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/item_account"
        android:icon="@drawable/ic_person_black_24dp"
        android:title="Профиль"/>

    <item
        android:id="@+id/item_trainer"
        android:icon="@drawable/ic_trainers"
        android:title="Тренера"/>

    <item
        android:id="@+id/item_inventar"
        android:icon="@drawable/ic_shtanga"
        android:title="Инвентарь"/>

    <item
        android:id="@+id/item_diet"
        android:icon="@drawable/ic_diet"
        android:title="Диеты"/>

    <item

```

```

android:id="@+id/item_info"
android:icon="@drawable/ic_map"
android:title="Карта

```

зала"/>

</menu>

3.3.3 Екран «Дієти»

Даний екран відображає доступні дієти для вибору (див. рис.3.3). Розмітка даного екрану на лістингу 3.4.



Рисунок 3.3 – Екран «Дієти»

Лістинг 3.5 – Лістинг файлу розмітки для екрану «Дієти»

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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=".DietActivity"
    android:background="@color/fon">

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

```

```
android:orientation="vertical"
android:layout_above="@+id/bottom_nav_menu"
android:layout_marginLeft="20dp"
android:layout_marginRight="20dp">

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="150dp"
    android:orientation="horizontal"
    android:layout_marginTop="50dp"
    android:layout_marginBottom="30dp"
    android:background="@drawable/btn_obvodka_black"
    android:padding="10dp">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="0dp"
        android:layout_weight="50"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/diet1" />

    <Button
        android:id="@+id/btn_diet1"
        android:layout_width="0dp"
        android:layout_weight="50"
        android:layout_height="match_parent"
        android:background="#00FFFFFF"
        android:text="Diet1"
        android:textAlignment="center"
        android:textColor="#FFFFFF"
        android:textSize="18sp"
        android:textStyle="italic" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="150dp"
    android:orientation="horizontal"
    android:background="@drawable/btn_obvodka_black"
    android:padding="10dp"
    android:layout_marginBottom="30dp"
    >

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="0dp"
        android:layout_weight="50"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/diet2"
        />
```

```
        <Button
            android:id="@+id/btn_diet2"
            android:layout_width="0dp"
            android:layout_weight="50"
            android:layout_height="match_parent"
            android:background="#00FFFFFF"
            android:text="Diet2"
            android:textAlignment="center"
            android:textColor="#FFFFFF"
            android:textSize="18sp"
            android:textStyle="italic" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="150dp"
        android:orientation="horizontal"
        android:background="@drawable/btn_obvodka_black"
        android:padding="10dp"
    >

        <ImageView
            android:id="@+id/diet3"
            android:layout_width="0dp"
            android:layout_weight="50"
            android:layout_height="wrap_content"
            app:srcCompat="@drawable/diet3"
        />

        <Button
            android:id="@+id/btn_diet3"
            android:layout_width="0dp"
            android:layout_weight="50"
            android:layout_height="match_parent"
            android:background="#00FFFFFF"
            android:text="Diet3"
            android:textAlignment="center"
            android:textColor="#FFFFFF"
            android:textSize="18sp"
            android:textStyle="italic" />

    </LinearLayout>

</LinearLayout>

<com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/bottom_nav_menu"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```



```

        android:layout_alignParentBottom="true"
        android:background="@color/fon_menu"
        app:itemIconTint="@drawable/selector"
        app:itemTextColor="@drawable/selector"
        app:menu="@menu/bottom_nav_menu" />
</RelativeLayout>

```

3.3.4 Екран «Тренери»

Даний екран ідентичний за розміткою до попереднього, проте відображає інший розділ – «Тренери».

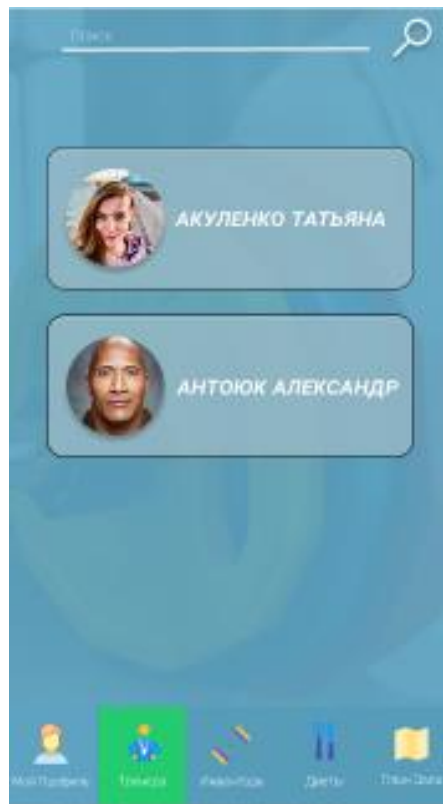


Рисунок 3.4 – Екран «Тренери»

Лістинг 3.6 – Лістинг файлу розмітки trainer_activity.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:background="@color/fon"

```

```
tools:context=".TrainerActivity">
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_above="@+id/bottom_nav_menu"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp">
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="150dp"
    android:orientation="horizontal"
    android:layout_marginTop="50dp"
    android:layout_marginBottom="30dp"
    android:background="@drawable/btn_obvodka_black"
    android:padding="10dp">
```

```
<ImageView
```

```
    android:id="@+id/imageView"
    android:layout_width="0dp"
    android:layout_weight="50"
    android:layout_height="wrap_content"
    app:srcCompat="@drawable/trainer1" />
```

```
<Button
```

```
    android:id="@+id/btn_trainer_akulenko"
    android:layout_width="0dp"
    android:layout_weight="50"
    android:layout_height="match_parent"
    android:background="#00FFFFFF"
    android:text="Татьяна
    Андрей"
    android:textAlignment="center"
    android:textColor="#FFFFFF"
    android:textSize="18sp"
    android:textStyle="italic" />
```

Акуленко"

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="150dp"
    android:orientation="horizontal"
    android:background="@drawable/btn_obvodka_black"
    android:padding="10dp"
    >

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="0dp"
        android:layout_weight="50"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/trainer2"
        />

    <Button
        android:id="@+id/btn_trainer2"
        android:layout_width="0dp"
        android:layout_weight="50"
        android:layout_height="match_parent"
        android:background="#00FFFFFF"
        android:text="Антонюк"
        android:textAlignment="center"
        android:textColor="#FFFFFF"
        android:textSize="18sp"
        android:textStyle="italic"
        />

</LinearLayout>

</LinearLayout>

<com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/bottom_nav_menu"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:background="@color/fon_menu"
```

```

app:itemIconTint="@drawable/selector"
app:itemTextColor="@drawable/selector"
app:menu="@menu/bottom_nav_menu"
/>
</RelativeLayout>

```

3.3.5 Экран відображення інформації про тренера

На цю сторінку відбувається перехід, якщо на попередній натискаємо на тренера. Доступний перегляд інформації, кнопка назад, меню(рис. 3.4).



Рисунок 3.5 – Экран «Тренер»

3.3.6 Екран «Інфо»

Цей екран відображає загальну інформацію про зал, сторінка статична і ніяких дій виконати не можна, окрім перегляду(рис. 3.4).



Рисунок 3.46 – Екран «Інфо»

Файл розмітки info_activity відображений в лістингу 3.5.

Лістинг 3.7 – Файл розмітки info_activity.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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=".Info"
    android:background="@color/fon">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_above="@+id/bottom_nav_menu"
        android:layout_margin="20dp">
```

```

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="10"
            android:text="Немного о нас"
            android:textColor="@color/white"
            android:textSize="30sp"
            android:gravity="center"/>
        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="0dp"
            android:layout_weight="40"
            android:src="@drawable/plan_zala"
            android:layout_gravity="center"/>
        <TextView
            android:layout_width="match_parent"
            android:text="@string/info"
            android:textColor="@color/white"
            android:textSize="18sp"/>

    </LinearLayout>
</ScrollView>
</LinearLayout>

<com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/bottom_nav_menu"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:background="@color/fon_menu"
    app:itemIconTint="@drawable/selector"
    app:itemTextColor="@drawable/selector"
    app:menu="@menu/bottom_nav_menu" />
</RelativeLayout>

```

3.4 Висновки до розділу 3

В результаті роботи було створено пакет з класами, що реалізують додаток. Також було розроблено екрани додатку за встановленими ескізами – екран запуску, реєстрації, власний кабінет, екран з відображенням дієт, тренерів, інвентарю тощо.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи, згідно з технічним завданням було спроектовано і створено додаток для ОС Android, призначений для доступу до послуг у спортзалі.

У першому розділі було проведено аналіз розробки додатку, у другому розділі було спроектовано кваліфікаційну роботу (проекування архітектури та розроблені діаграми пакетів та класів), відповідно в третьому розділі додаток був реалізований. При створенні даної інформаційної системи були враховані основні тенденції розвитку мобільних застосунків для платформи Android, також враховані переваги та недоліки.

Система реалізована за допомогою середовища розробки Android Studio з використанням такої мови програмування як Java та Android SDK. Удосконалення системи можливо в сторону розширення функціональних можливостей, розробки серверної частини додатку, а також можливе покращення надійності та швидкості роботи

Проект має практичну цінність та актуальність, оскільки його використання призведе до зменшення витрат на обслуговуючий персонал, підвищить швидкість та якість обслуговування, та сприятиме покращенню іміджу закладу.

Проаналізувавши все вищевказане, можна зробити висновок, що додаток має свої недоліки, що можуть бути ліквідовані у майбутньому. Проте додаток реалізований згідно з технічним завданням і тому результат, отриманий в даній кваліфікаційній роботі, вважаю таким, що відповідає поставленим задачам.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) Мобільний додаток – внесемо розуміння у значення терміну [Електронний ресурс]. - Режим доступу: URL: <https://www.quality-assurance-group.com/mobilnyj-dodatok-vnosymo-rozuminnya-u-znachennya-terminu/> - Назва з екрану.

2) Зробити більше: веб програма проти власного додатка: плюси і мінуси – 2020 [Електронний ресурс]. - Режим доступу: URL: <https://uk.go-travels.com/97634-pros-and-cons-of-native-apps-and-mobile-web-apps-2373173-2292572> - Назва з екрану.

3) Java - Вікіпедія [Електронний ресурс]. - Режим доступу: URL: <https://uk.wikipedia.org/wiki/Java> - Назва з екрану.

4) Kotlin - Вікіпедія [Електронний ресурс]. - Режим доступу: URL: <https://uk.wikipedia.org/wiki/Kotlin> - Назва з екрану.

5) Android - Вікіпедія [Електронний ресурс]. - Режим доступу: URL: <https://uk.wikipedia.org/wiki/Android> - Назва з екрану.

6) Android Studio [Електронний ресурс]. - Режим доступу: URL: http://www.tadviser.ru/index.php/Android_Studio - Назва з екрану.

7) База даних PostgreSQL [Електронний ресурс]. - Режим доступу: URL: http://ukrhosting.ua/bazi_danih_postgresql-p-263951.html- Назва з екрану.