Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем

РОЗРОБКА ANDROID-ДОДАТКУ

Текстова частина до курсової роботи
за спеціальністю «Інженерія програмного забезпечення»

Керівник курсової роботи
ст. викладач Борозенний С. О.

_____
(підпис)
“____” _____ 2022 р.
Виконав студент 3 р. н.
Макаренко Д. О.
“____” _____ 2022 р.

Київ – 2022

Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра мультимедійних систем

ЗАТВЕРДЖУЮ

Зав. кафедри мультимедійних систем

доцент, к.ф.-м.н.

О. П. Жежерун

_____2022 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студенту Макаренку Данилу Олександровичу 3-го курсу факультету інформатики

ТЕМА Розробка ANDROID-додатку

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Вступ

Опис можливостей застосунку

Технічний опис

Технічні особливості додатку

Висновок

Список літератури

Дата видачі “___” _____ 2022 р. Керівник _____

Завдання отримав _____

Тема: Розробка Android-додатку

Календарний план виконання роботи:

№	Етап	Термін виконання
1.	Отримання завдання на курсову роботу.	11.10.2021
2.	Проектування архітектури застосунку.	Листопад 2021
3.	Написання клієнтської частини застосунку.	Грудень 2022
3.	Написання серверної частини застосунку.	Лютий 2022
4.	Поєднання клієнтської та серверної частин.	Березень 2022
5.	Написання текстової частини роботи.	Травень 2022
6.	Захист курсової роботи.	

Студент Макаренко Данило Олександрович

Керівник Борозенний О. С.

“_____”_____

Зміст

Вступ

До війни я чи не щодня ходив до басейну, і мене постійно дивувало : чому у час сучасних технологій мені потрібно щоразу показувати своє паперове посвідчення на вході до басейну. Зрозуміло, що це надзвичайно незручно, адже посвідчення можна пошкодити, загубити або забути. Водночас для більшості людей мобільний телефон став невід’ємною частиною життя – пристроєм, який завжди поруч та виконує цілий спектр корисних послуг. Стикнувшись із цією проблемою, я вирішив створити максимально зручний та корисний мобільний застосунок для спортивних залів, що дозволяє користувачам назавжди позбутись незручних пластикових карток та паперових посвідчень для відвідувачів спортзалів.

Опис можливостей застосунку

Авторизація

При запуску додатку користувач бачить заставку (рисунок 1).

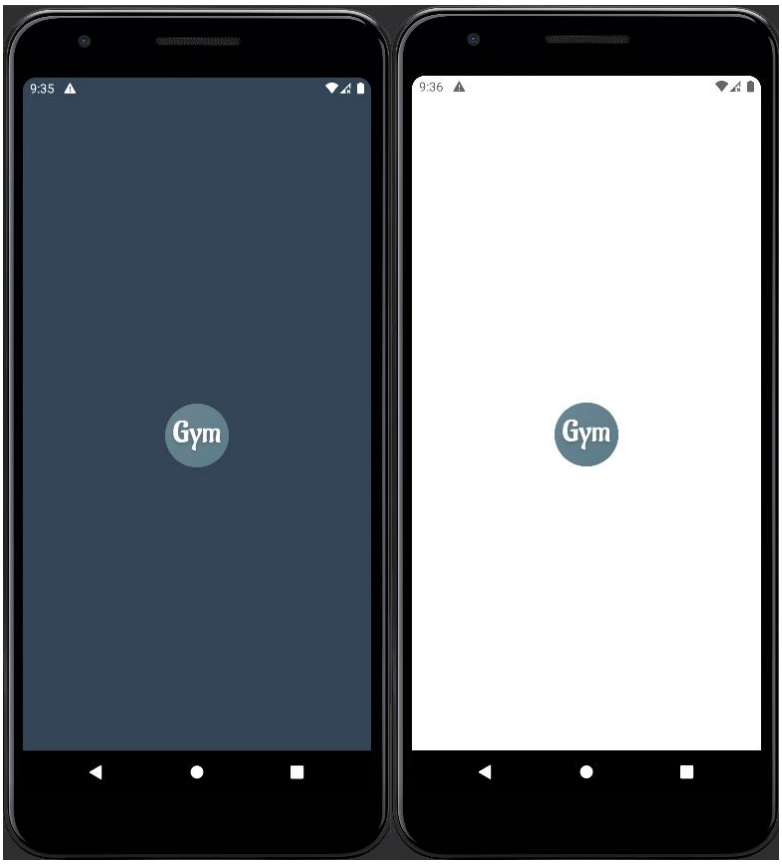


Рисунок 1. Заставка для користувачів із темною та світлої темами на телефоні

Якщо користувач авторизований, то від одразу потрапляє свій головний екран залежно від ролі. Якщо користувач вийшов із свого облікового запису або запускає застосунок уперше, то від потрапляє на екран авторизації(рисунок 2).

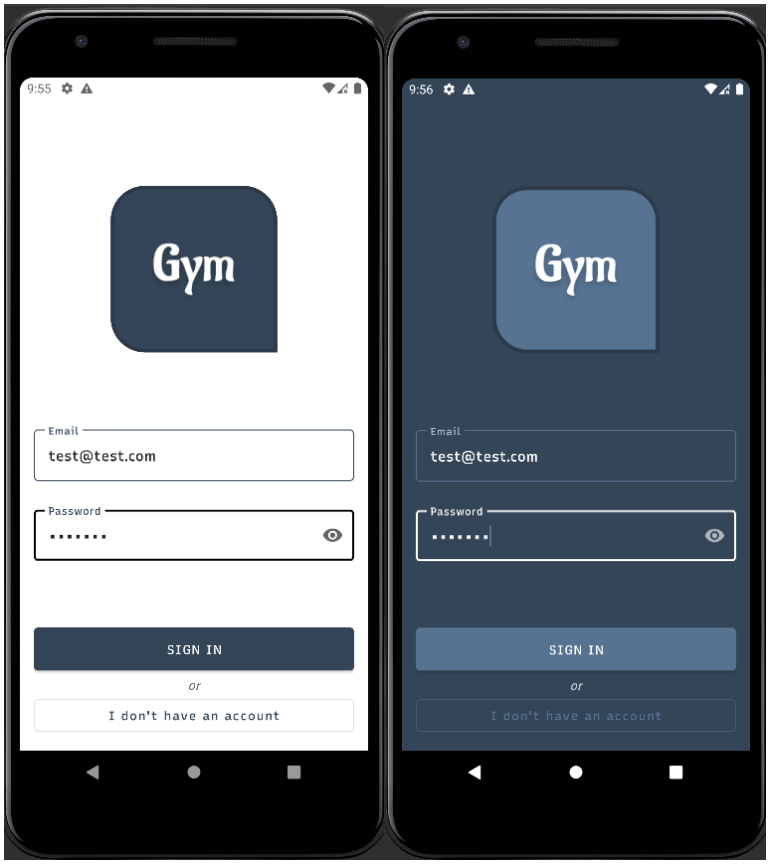


Рисунок 2. Екран авторизації

Поточний екран містить поле для вводу електронної адреси користувача та його паролю. Для здійснення авторизації користувач повинен натиснути на кнопку «Sing in» (з англ. - увійти).

Якщо користувач увів неправильні дані, то він побачить відповідні повідомлення.

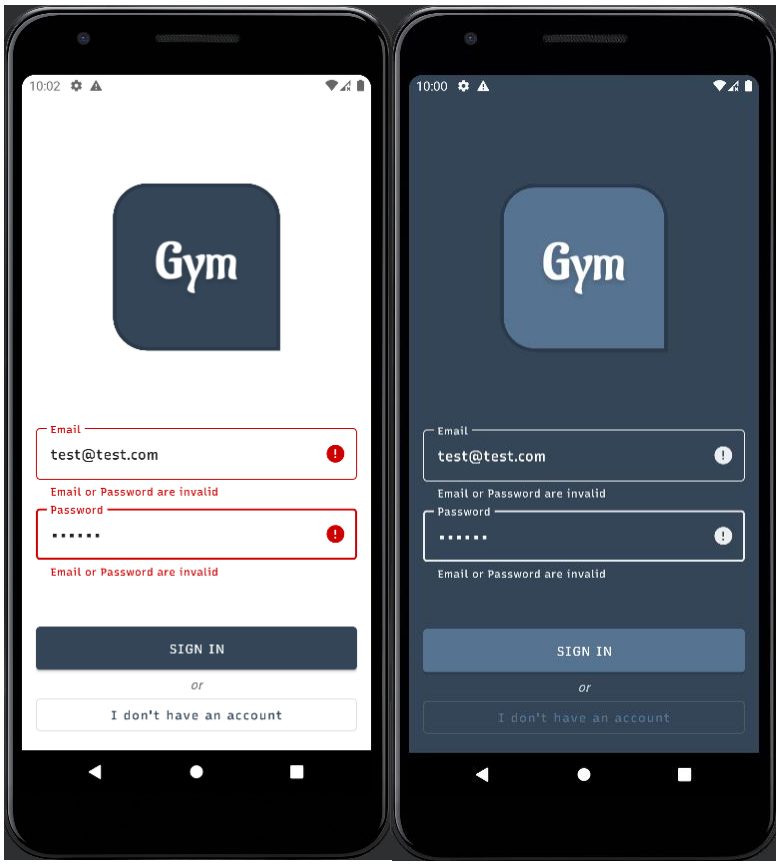


Рисунок 3. Повідомлення про введення неправильного паролю для авторизації

З екрану авторизації користувач може перейти на екран створення облікового запису.

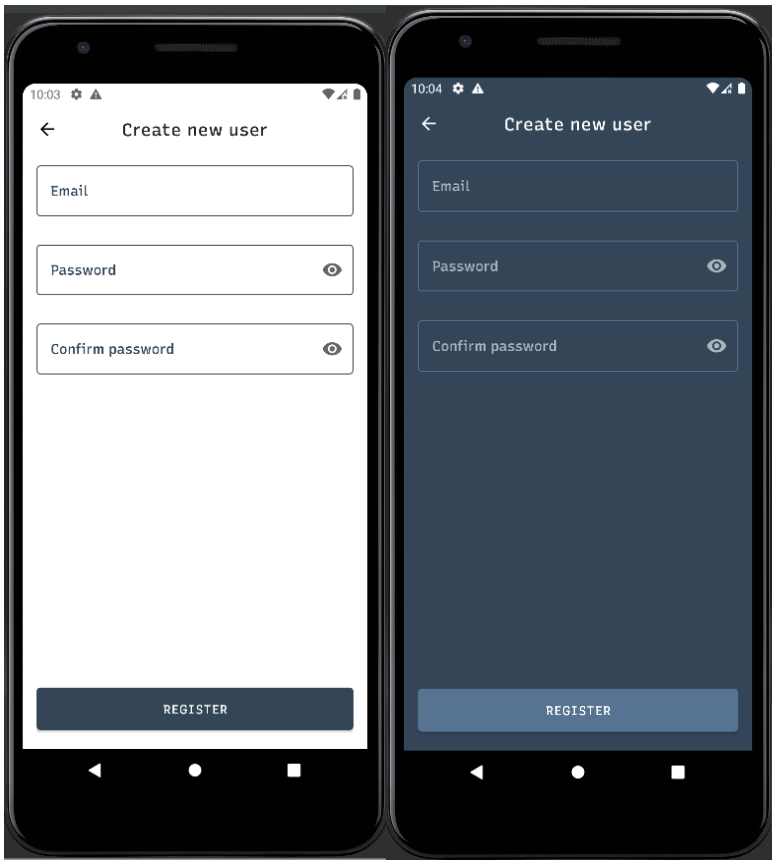


Рисунок 4. Екран створення облікового запису

Для створення облікового запису користувач має ввести :

- електронну адресу
- пароль
- підтвердження паролю
- натиснути на кнопку «Register» (з англ. - реєстрація)

Якщо користувач увів неправильні дані, то він побачить відповідне повідомлення(рисунок 5)

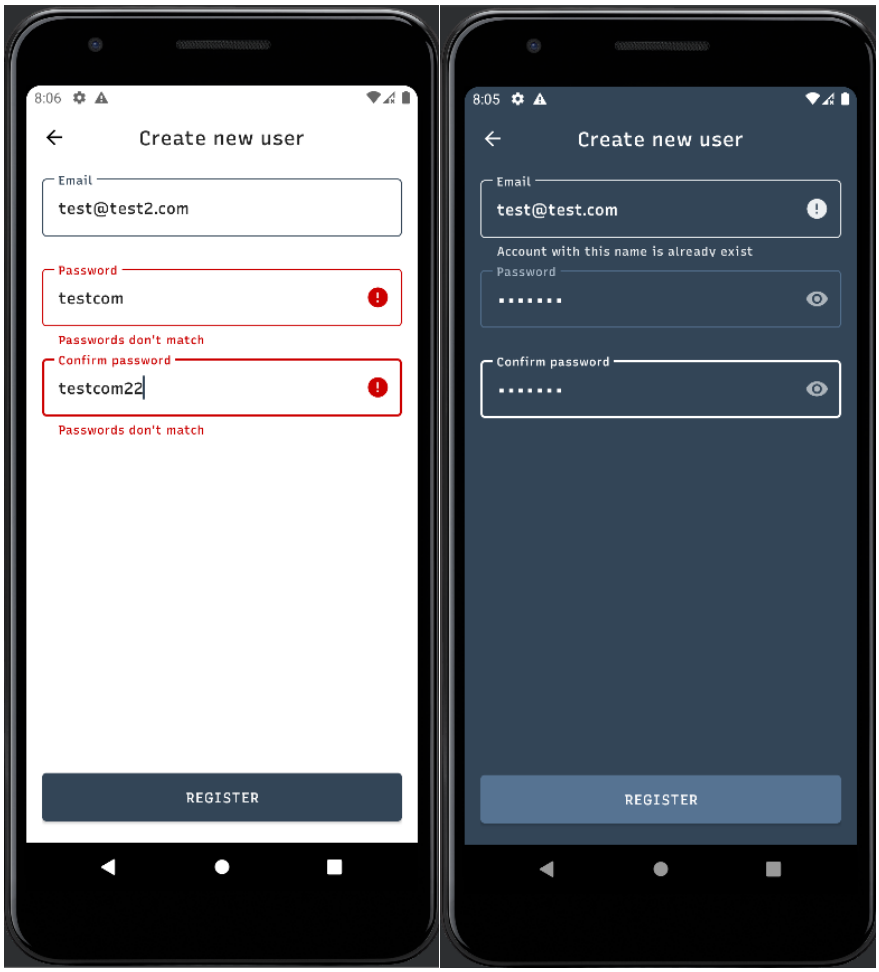


Рисунок 5. Відображення помилок при створенні облікового запису користувача

Важливо, що через екран облікового запису можливо створити лише акаунт для відвідувача спортзалу. Усі акаунти адміністраторів створюються та редагуються через Firebase Console.

Додаток поєднує в собі функціонал для відвідувача спортзалу та адміністратора. При авторизації в застосунку, залежно від ролі, обирається навігаційний граф із усією необхідною для поточного користувача застосунку функціональністю.

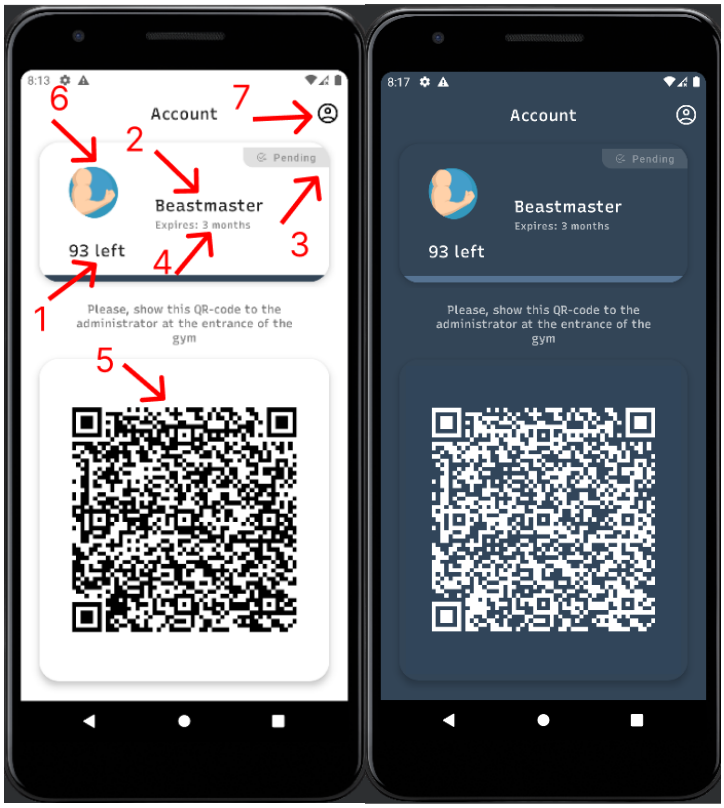
Можливості користувача

Застосунок має бути достатньо зручним та корисним, щоб користувачі обрали його замість звичайних посвідчень, тому весь необхідний функціонал для відвідувачів спортзалу мій додаток включає, а саме :

- зручна автентифікація
- інформація про поточний статус абонементу
- повна інформація про доступні абонементи та можливість їх вибору

Розгляньмо детальніше функціонал застосунку з точки зору відвідувача спортивної зали.

Головний екран



Головний екран відвідувача містить :

1. Кількість відвідувань, які залишились до закінчення терміну дії абонементу.
2. Назва абонементу. Змінюється через Firebase Console.
3. Поточний статус абонементу. Можливі значення для статусу :

- Активний (Active)
- Очікує оплати (Pending)
- Використаний (Used)

4. Дата закінчення дії абонементу. На рисунку зображено орієнтовну дату закінчення дії. Точна дата встановлюється автоматично, коли адміністратор підтверджує сплату абонементу і переводить його статус із Pending у Active.
5. Згенерований QR-код. Цей код сканує адміністратор на вході до спортивної зали. QR-код містить у собі ідентифікатор, ім’я, прізвище, номер телефону, ідентифікатор абонементу та URL зображення абонементу – це дозволяє максимально швидко відображати первинні дані адміністратору.
6. Логотип абонементу. Змінюється через Firebase Console.
7. Перехід до особистого кабінету.

Якщо користувач не має підключення до інтернету, то дані беруться з кешу, проте, якщо кеш порожній, то на головному екрані буде показано помилку.

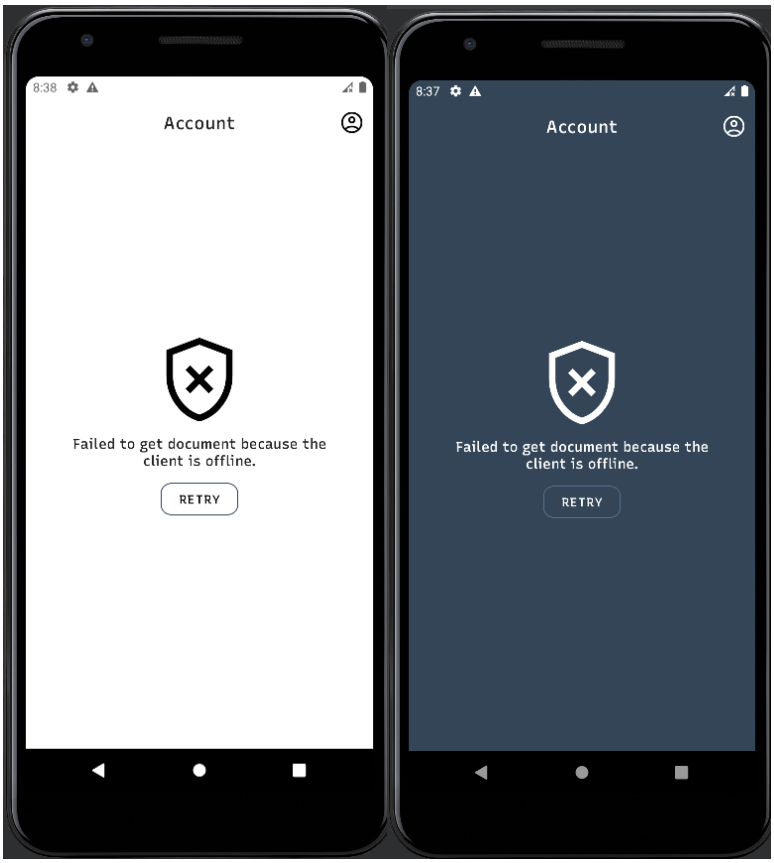


Рисунок 6. Відображення помилки на головному екрані

Користувач може спробувати завантажити головний екран ще раз, натиснувши кнопку «Retry».

Екран профілю

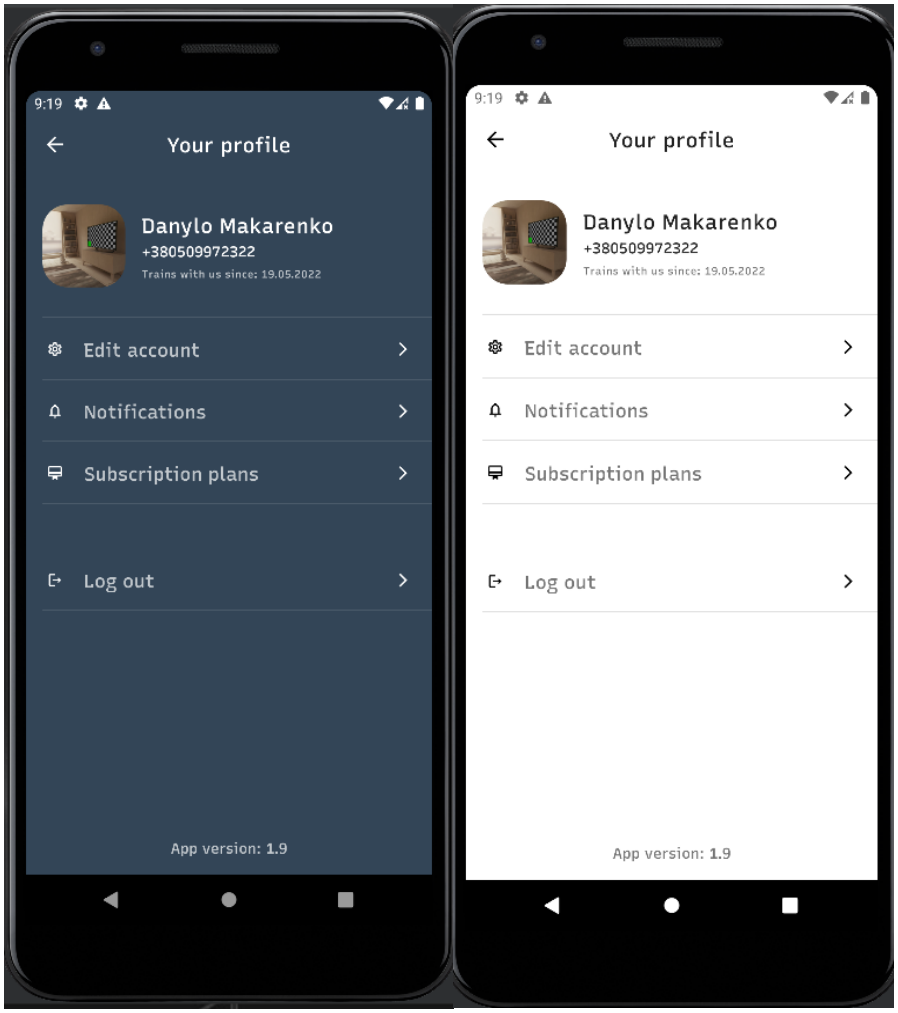


Рисунок 7. Екран профілю користувача

Екран профіль містить :

- прізвище та ім’я
- мобільний номер телефону
- дату реєстрації в додатку
- перехід до редагування особистих даних
- перехід на екран доступних абонементів
- вихід зі застосунку
- повідомлення про поточну версію застосунку

Екран редагування особистих даних

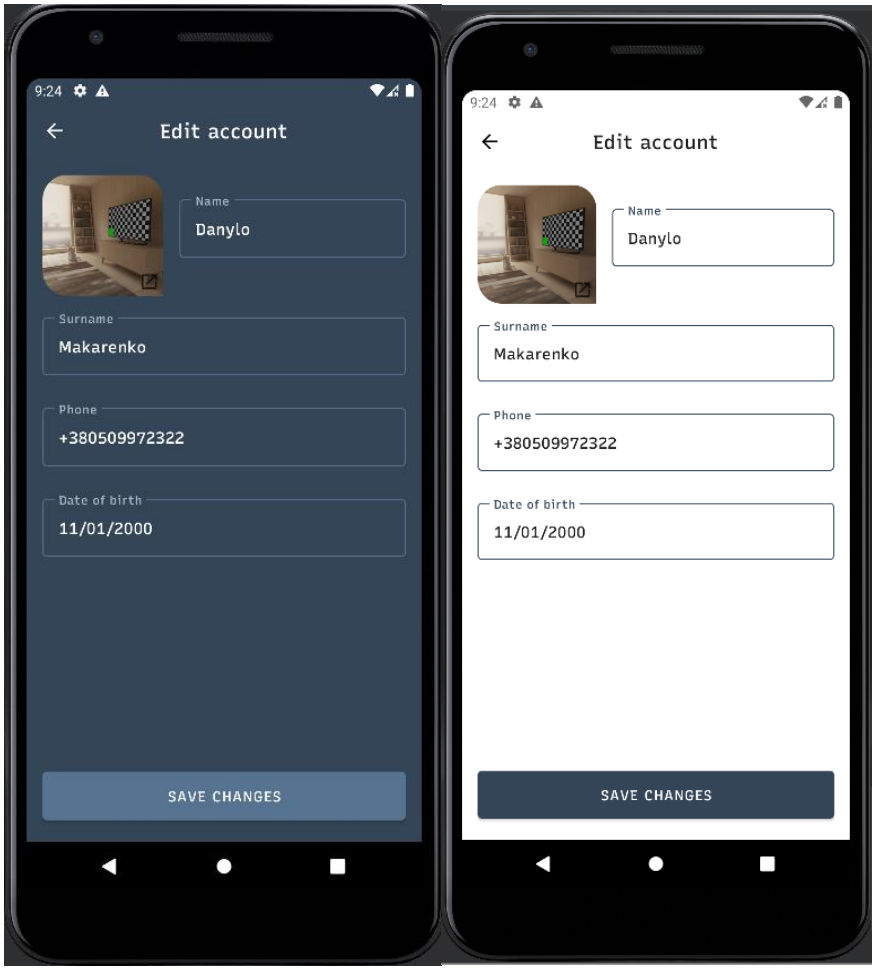


Рисунок 8. Екран редагування особистих даних користувача

Для зміни фото профілю, користувач повинен натиснути на поточне зображення. Якщо користувач ще не заповнив дані, то буде відображено стандартні(рисунок 9).

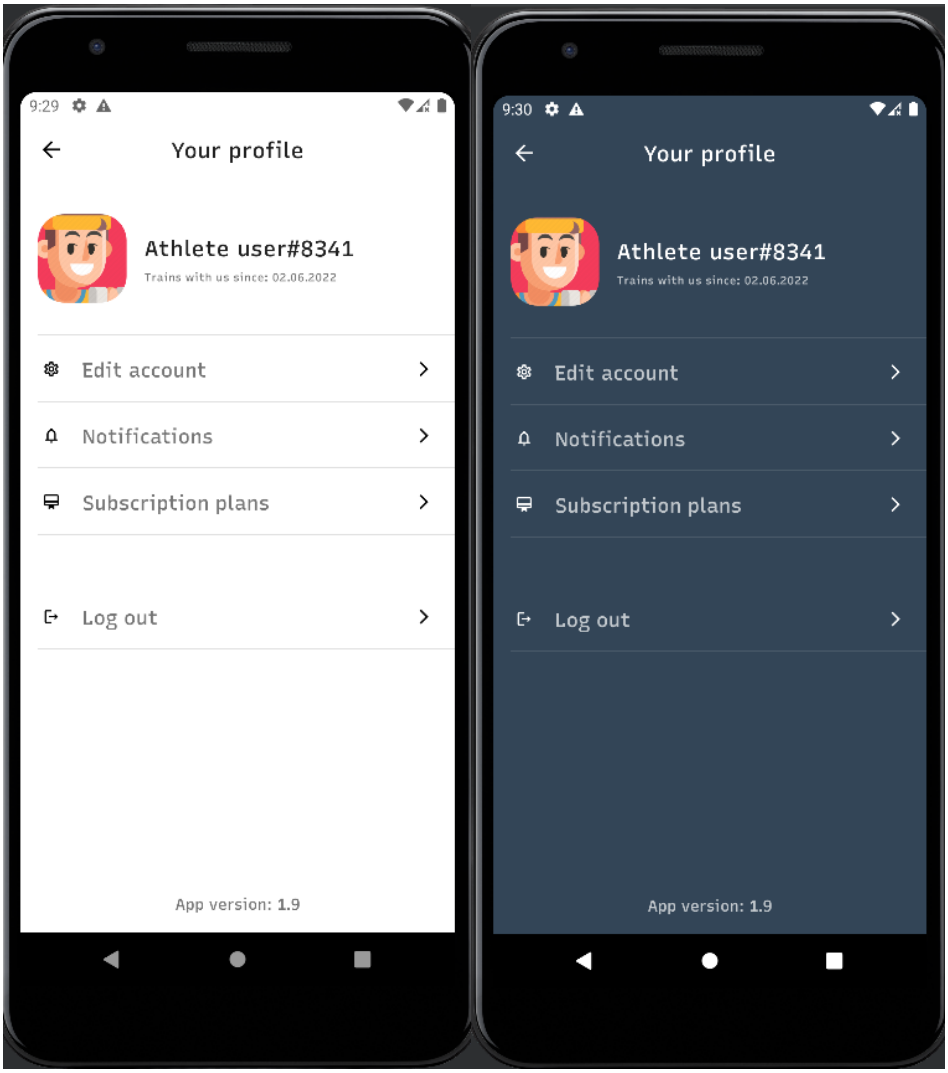


Рисунок 9. Початкові дані нового користувача

Новостворений користувач отримує стандартне ім’я - Athlete (з англ. - Атлет) та прізвище, що складається зі слова «user» та чотирьох випадкових цифр.

Екран абонементів

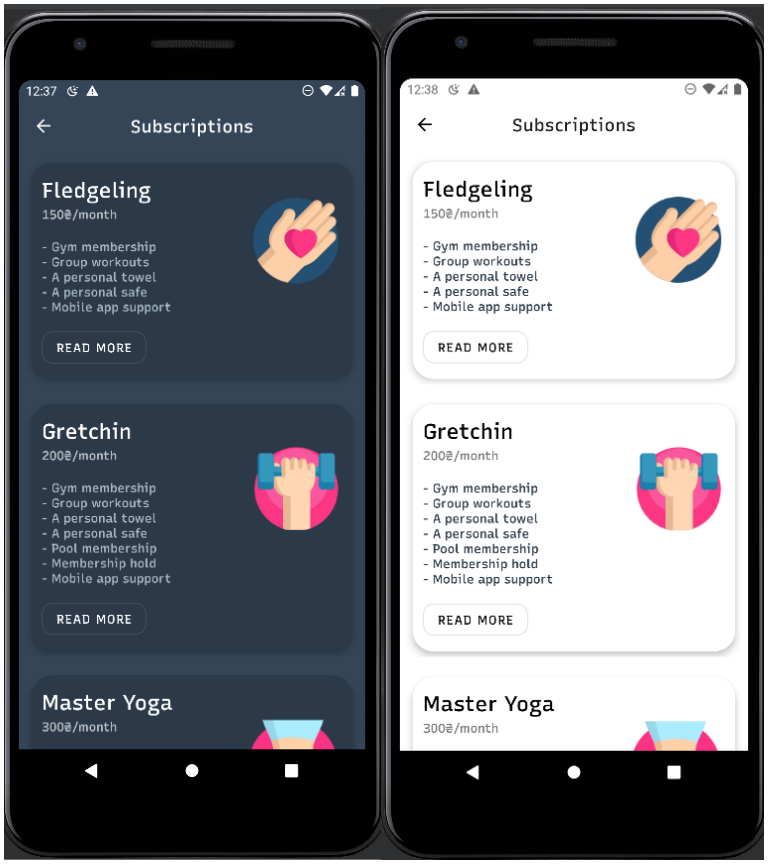


Рисунок 10. Екран усіх абонементів

Екран абонементів містить перелік усіх доступних абонементів. Кожен абонемент містить :

- назву
- зображення
- ціну за місяць
- переваги
- кнопку для переходу до деталей

Редагування абонементів виконується через Firebase Console. Із цього екрану користувач може перейти на екран детального опису абонементу.

Екран деталей про абонемент

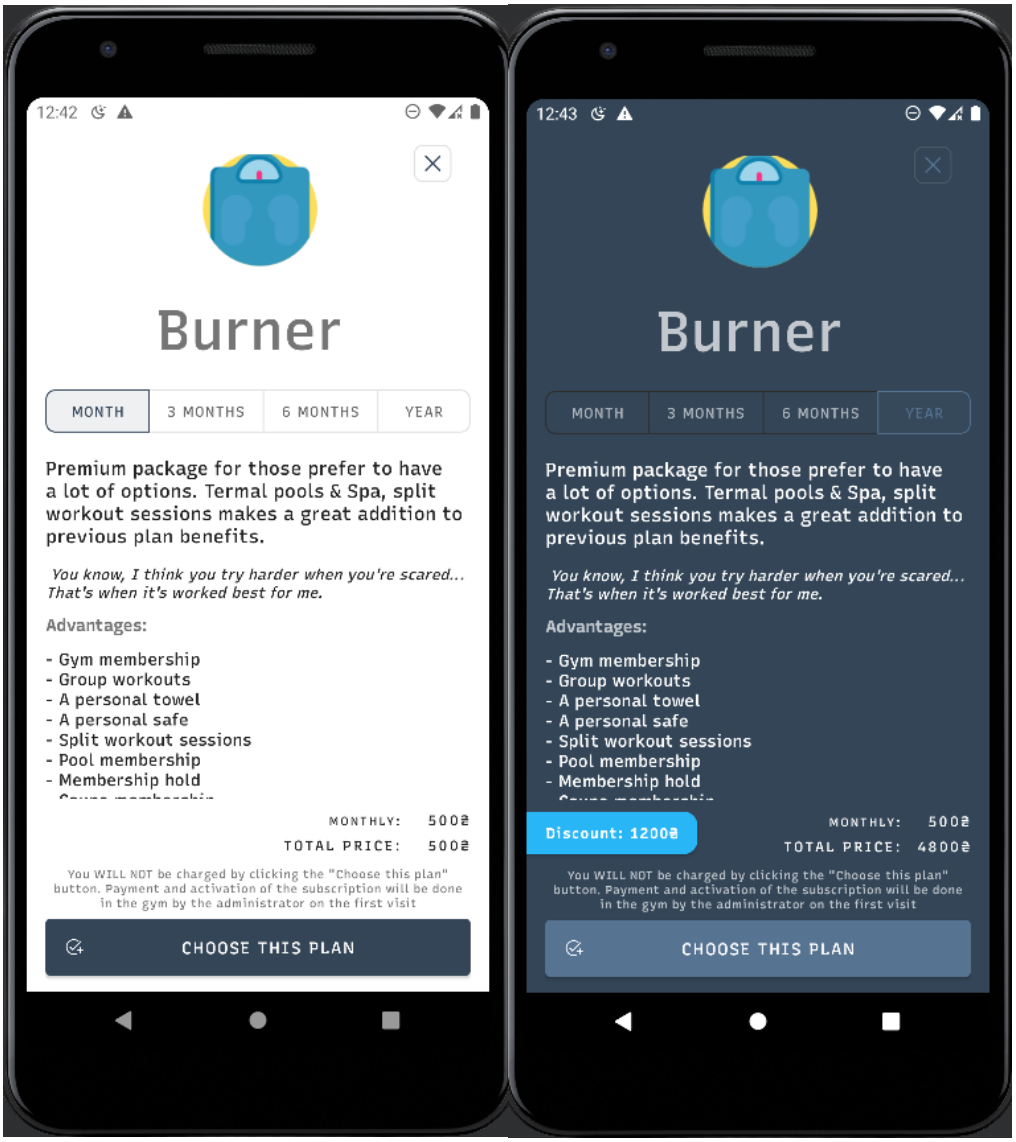


Рисунок 11. Деталі про абонемент

На екрані деталей про абонемент користувач може переглянути :

- опис абонементу
- його переваги
- обрати необхідну кількість місяців
- обрати поточний план

Користувач при виборі річного абонементу побачить відповідне повідомлення про знижку(рис 8.2).

Можливості адміністратора

Адміністратор перевіряє відвідувачів лише через QR-код. Після авторизації адміністратор потрапляє на свій головний екран. Редагування даних про адміністратора виконується через Firebase Console.

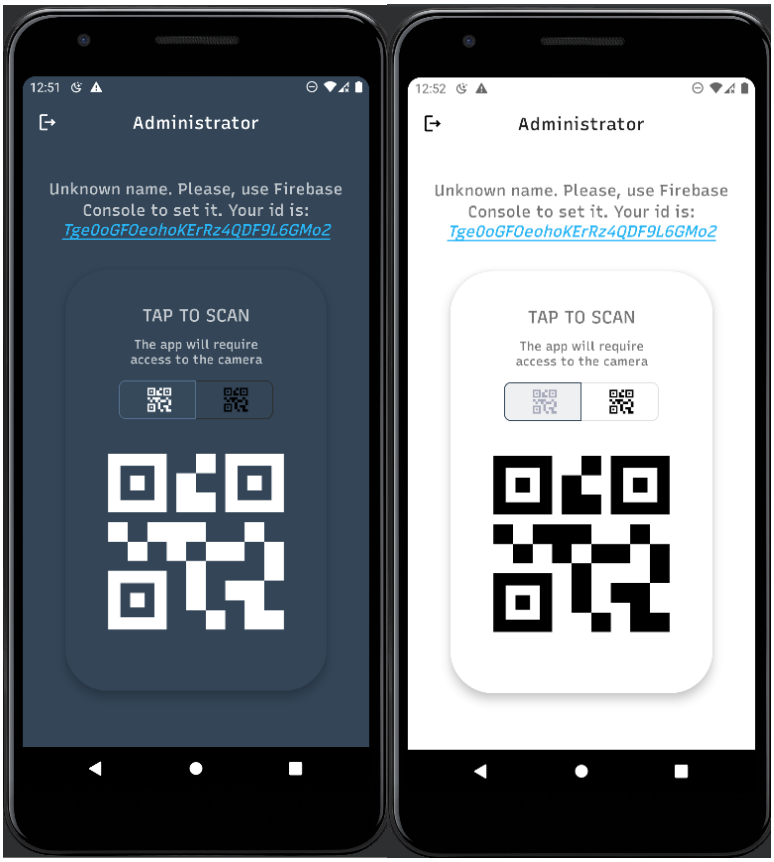


Рисунок 12. Головний екран адміністратора

При натисканні на зображення QR-коду відкривається сканер. Перед натисканням, адміністратор обрає : який саме колір має QR-код користувача.

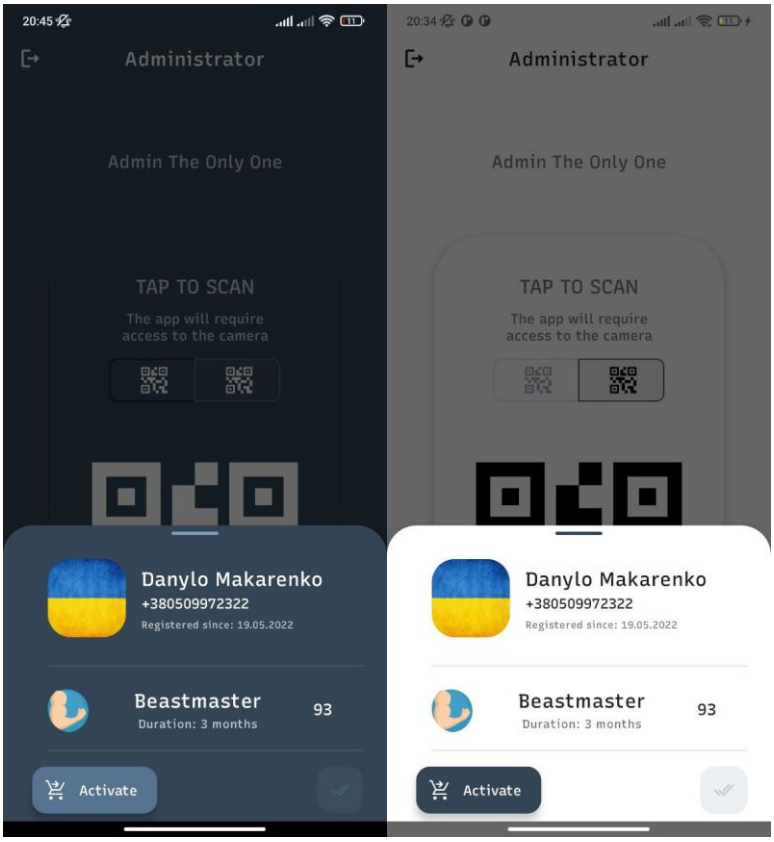


Рисунок 13. Екран результату сканування неактивованого абонементу

Після сканування QR-коду адміністратор бачить ім’я, прізвище, телефон, зображення профілю, та повну інформацію про поточний абонемент користувача.

Якщо статус абонемента «використаний» або «очікує сплати», то адміністратор спочатку має виконати активацію абонементу.

Наразі клієнт може оплатити абонемент лише на касі. Коли адміністратор отримує оплату, то він активує абонемент користувача в застосунку.

Якщо статус абонемента «активний», то адміністратор лише підтверджує відвідування спортивної зали, зменшуючи залишок відвідувань на один.

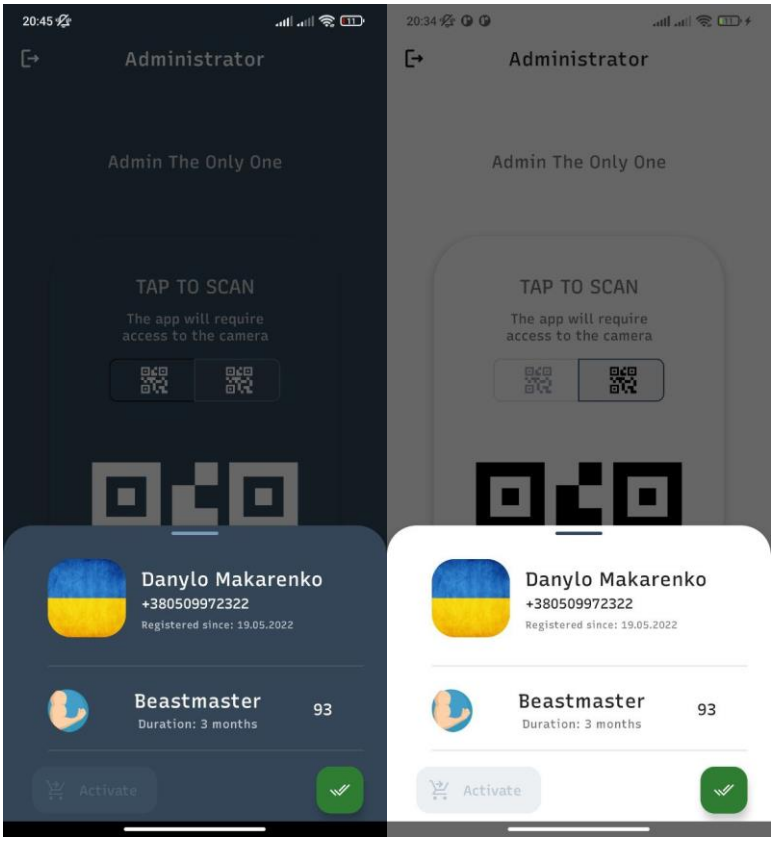


Рисунок 14. Екран підтвердження входу

Якщо адміністратор підтверджує використання останнього доступного відвідування, то статус абонемента автоматично стає «використаний». Після чого клієнт знов повинен обрати новий чи сплатити вже існуючий абонемент.

Клієнт може змінювати статус абонементу, що має статус «використаний» або «очікує сплати». Змінити активний абонемент через застосунок неможливо, але, якщо це необхідно, то це можна зробити на касі через Firebase Console.

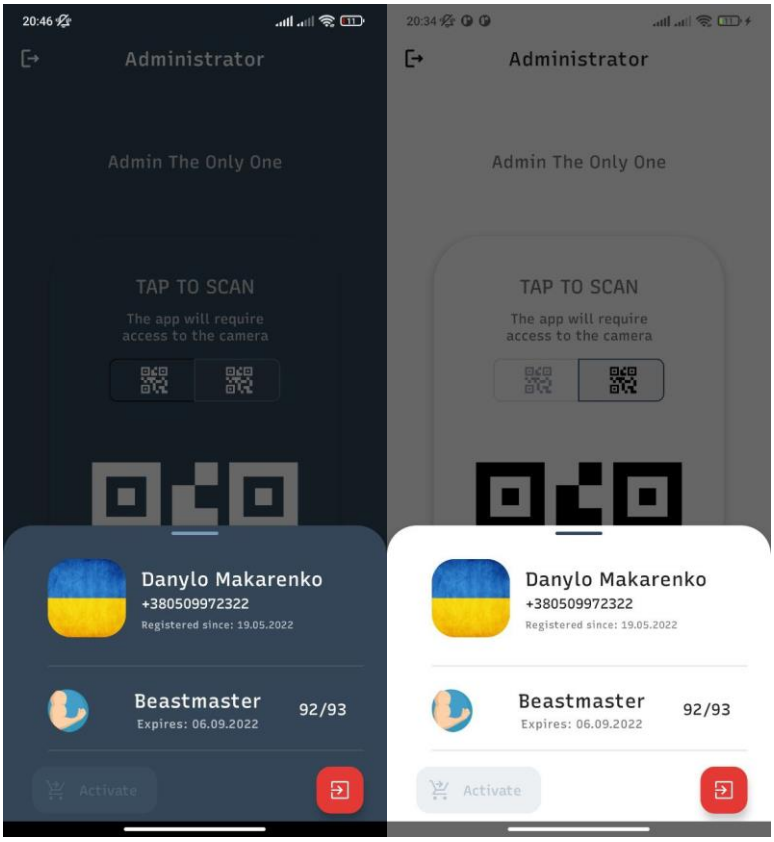


Рисунок 15. Екран підтвердження виходу

Адміністратор сканує QR-код не тільки на вході, а ще й на виході. Це дозволить у майбутньому додати історію відвідувань клієнта.

Технічний опис застосунку

Усі візуальні компоненти застосунку розроблені з використання дизайн-патерну Material Design, що був розроблений компанією Google в 2014 році. Дотримання принципів Material Design зробило вигляд мого застосунку сучасним та приємним.

Застосунок написаний із використання мови програмування Kotlin, що є офіційно рекомендованою мовою розробки під операційну систему Android. Kotlin - сучасна мова програмування, що має багато переваг над Java. Наведу деякі з них:

- Null-safety - дозволяє зменшити до мінімуму кількість помилок через відсутність значення
- Extension functions – дозволяють розширювати функціонал уже існуючих класів без їх модифікації
- String templates – дозволяє позбутись явної незручної конкатенації стрічок
- Operator overloading – дозволяє перевизначити деякі оператори, що робить синтаксис більш лаконічним
- Data classes – зручні класи, що перевизначають стандартні методи, такі як : toString(), hashCode(), equals(), тощо
- Розділення колекцій на змінні та незмінні

Під час розробки та проектування додатку я користувався принципами Clean Architecture, розробленими Робертом Мартіном. Мій застосунок розділений на наступні шари :

- Domain layer – шар домену – логіка застосунку
- Data layer – шар даних – взаємодія з даними

- Presentation layer – шар представлення – взаємодія з користувачем

Принципи чистої архітектури я поєднав із патерном MVVM(Model-View-ViewModel), що забезпечило :

- максимальну масштабованість та гнучкість застосунку
- можливість легкого тестування

Система Android для відображення всіх елементів застосунку використовує головний потік. Якщо виконувати якісь довготривалі операції на головному потоці, то система припинить роботу застосунку, тому всі операції з мережею та базою даних телефону виконуються в фоновому потоці. Для зручної роботи з багатопоточністю я використовував бібліотеку RxJava, що є надзвичайно популярною серед Android-розробників. Популярними є також корутини(coroutines), проте я хотів покращити свої знання саме з RxJava.

Одним із принципів SOLID є принцип єдиної відповідальності. Сутність не повинна бути відповідальною за створення чи пошук своїх залежностей. Зручної бібліотекою для ін’єкції залежностей є Koin. Koin повністю написана мовою програмування Kotlin, що дозволяє активно користуватись усіма перевагами цієї мови, такими як, наприклад, Kotlin DSL (Domain-Specific Language) (з англ. – предметно-орієнтована мова). Популярним та рекомендованим компанією Google є Dagger Hilt – ще одна бібліотека для ін’єкції залежностей, проте, після детального аналізу, я зробив висновок, що для мого застосунку більше підходить Koin, адже Dagger містить багато потужних механізмів, які корисні у великих проектах, проте мій додаток – це відносно малий проект, тож для пришвидшення швидності розробки я обрав саме Koin.

Для роботи із зображеннями я використовував бібліотеку Glide, що дозволяє швидко та зручно із можливістю кешування працювати із зображеннями.

Бібліотека ZXing Embedded генерує та зчитує QR-коди.

Jetpack Navigation у поєднанні з Safe Args допомагає зручно змінювати екрани з можливістю передавати дані через аргументи.

Firebase

Firebase – це платформа для розробки веб та мобільних застосунків, що поєднує в собі ціле різноманіття інструментів для швидкої та якісної розробки. Найбільш цікавою для мене є саме Android розробка, то ж я вирішив використати Firebase, а не писати власну серверну частину.

Збереження даних користувачів здійснюється через Firestore Database. Firebase Database є нереляційною базою даних, що забезпечує масштабованість, гнучкість та високу продуктивність.

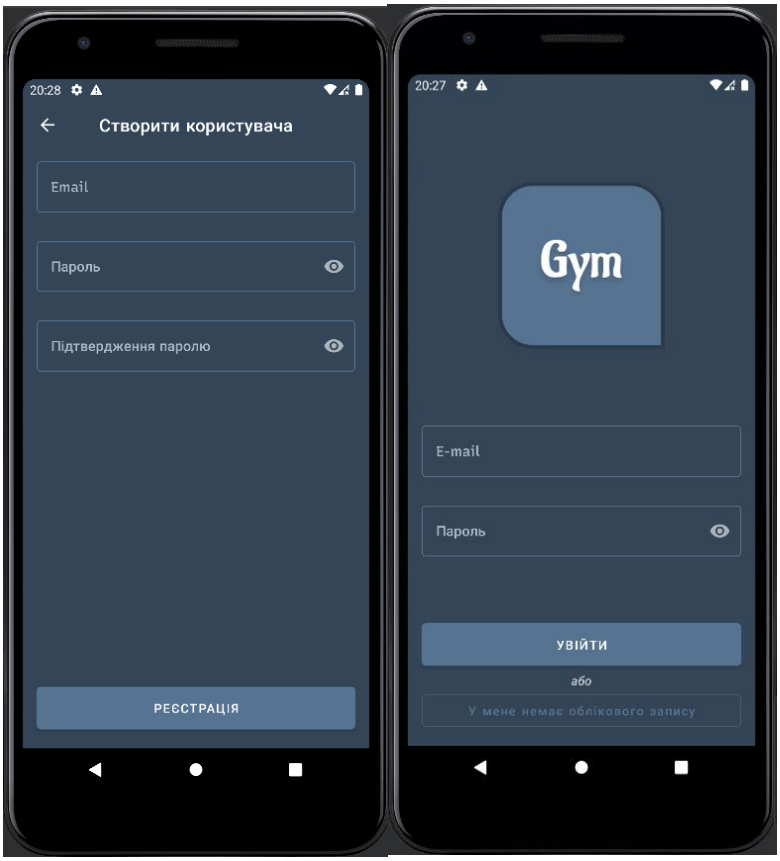
Авторизація забезпечується за допомогою Firebase Auth, що надійно зберігає всі паролі користувачів.

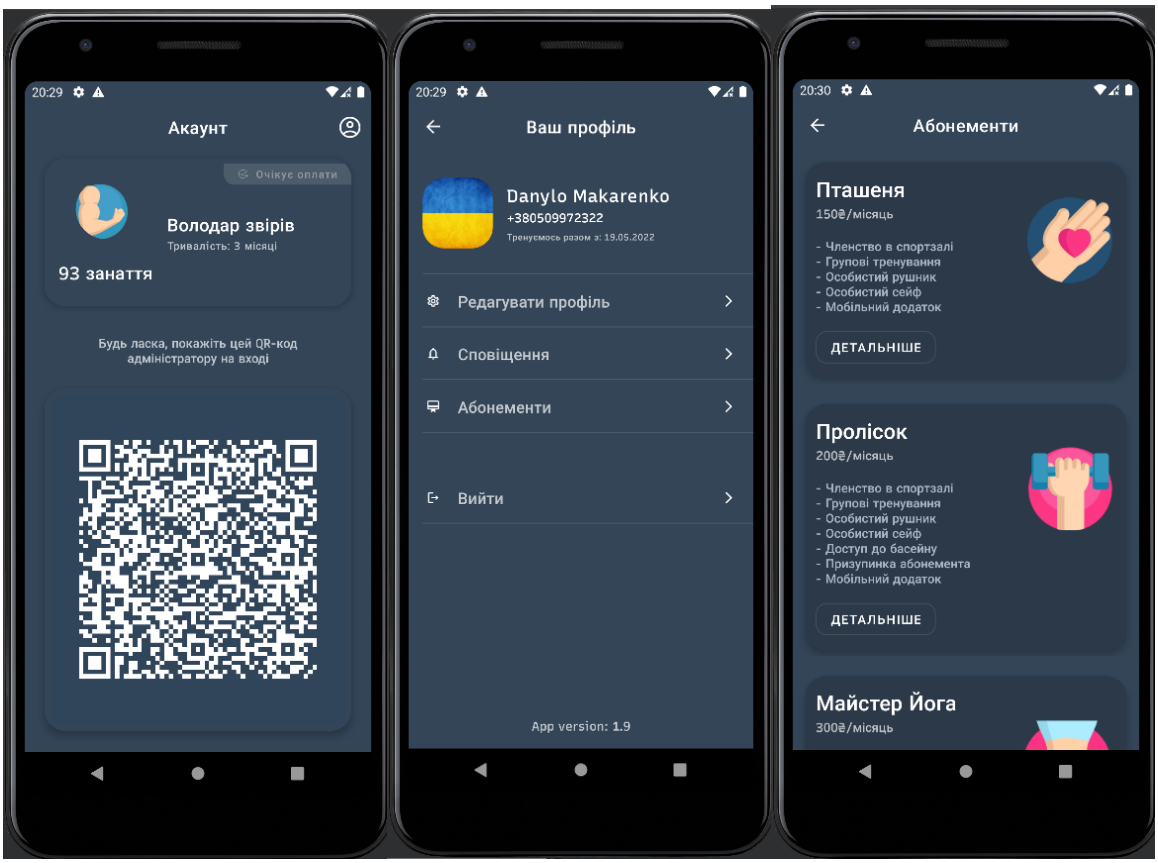
Зображення профілів користувачів та абонементів зберігаються в Firebase Storage.

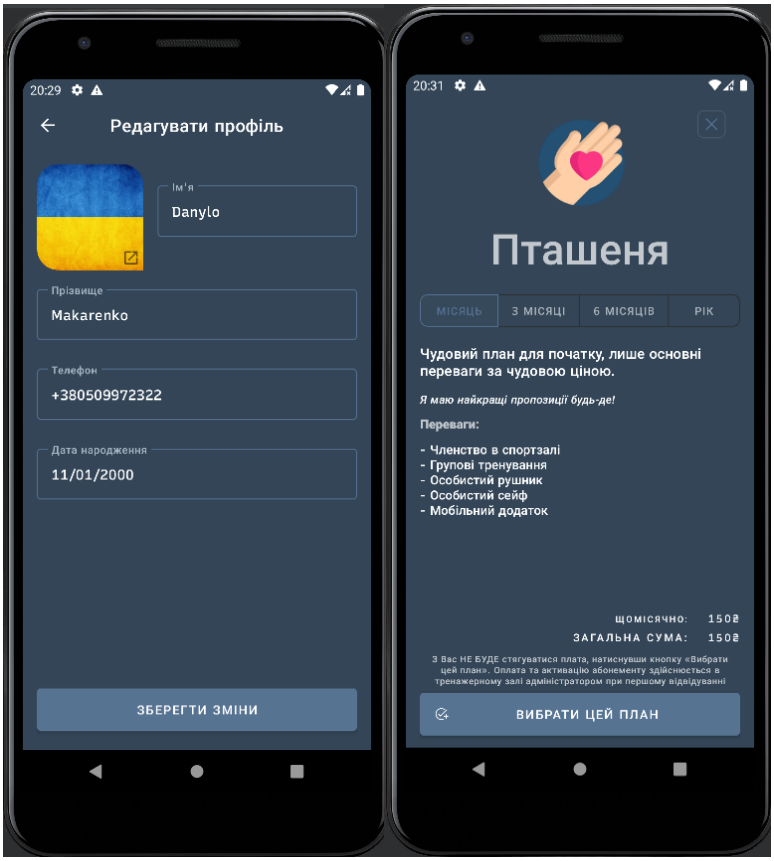
Технічні особливості застосунку

1. Застосунок підтримує українську локалізацію.

Якщо користувач обрав мовою системи українську, то застосунок буде відображати весь текст українською.




20:29
Акаунт
Очікує оплати
Володар звірів
Тривалість: 3 місяці
93 заняття
Будь ласка, покажіть цей QR-код адміністратору на вході
20:29
Ваш профіль
Danylo Makarenko
+380509972322
Тренуємось разом з: 19.05.2022
Редагувати профіль
Сповіщення
Абонементи
Вийти
App version: 1.9
20:30
Абонементи
Пташеня
150₴/місяць
- Членство в спортзалі
- Групові тренування
- Особистий рушник
- Особистий сейф
- Мобільний додаток
ДЕТАЛЬНІШЕ
Пролісок
200₴/місяць
- Членство в спортзалі
- Групові тренування
- Особистий рушник
- Особистий сейф
- Доступ до басейну
- Призупинка абонемента
- Мобільний додаток
ДЕТАЛЬНІШЕ
Майстер Йога
300₴/місяць


20:29
Редагувати профіль
Ім'я
Danylo
Прізвище
Makarenko
Телефон
+380509972322
Дата народження
11/01/2000
ЗБЕРЕГТИ ЗМІНИ
20:31
Пташеня
МІСЯЦЬ
3 МІСЯЦІ
6 МІСЯЦІВ
РІК
Чудовий план для початку, лише основні переваги за чудовою ціною.
Я маю найкращі пропозиції будь-де!
Переваги:
- Членство в спортзалі
- Групові тренування
- Особистий рушник
- Особистий сейф
- Мобільний додаток
ЩОМІСЯЧНО: 150₴
ЗАГАЛЬНА СУМА: 150₴
З Вас НЕ БУДЕ стягуватися плата, натиснувши кнопку «Вибрати цей план». Оплата та активацію абонементу здійснюється в тренажерному залі адміністратором при першому відвідуванні
ВИБРАТИ ЦЕЙ ПЛАН

2. Застосунок підтримує темну та світлу тему на телефоні.
3. Застосунок активно та ефективно використовує кешування даних. Під час кожного запуску застосунку виконується запит на сервер для синхронізації даних користувача. Надалі всі дані користувача зберігаються в зашифрованому вигляді в пам’яті телефону. Кешування дозволяє:

- уникнути зайвих витрат інтернету користувача
- зменшити навантаження на сервер
- пришвидшити роботу застосунку

4. Застосунок використовує Palette API, що дозволяє залежно від фону автоматично підібрати контрастний колір для зображення.

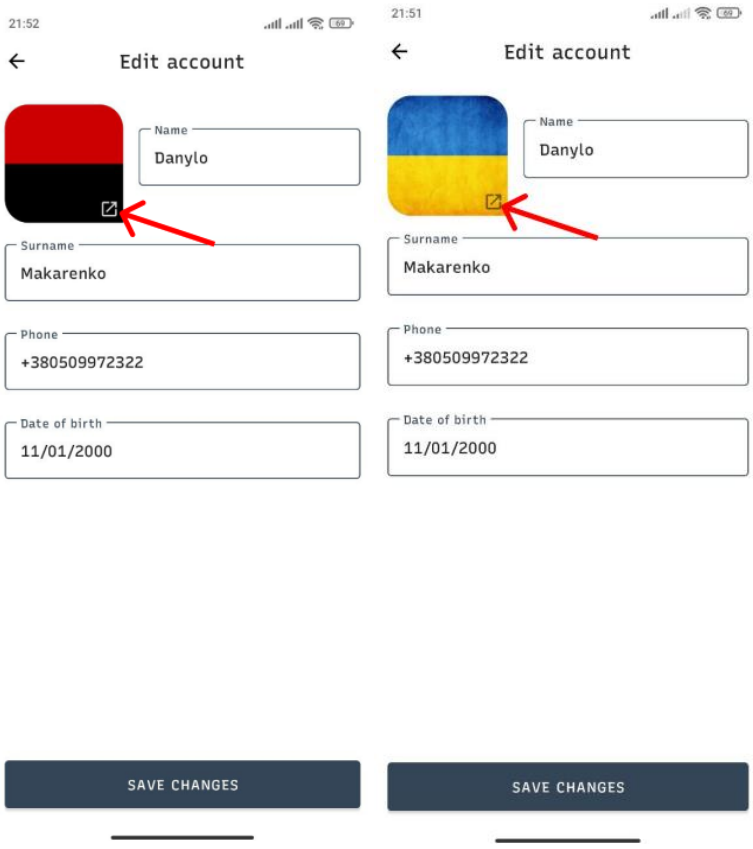


Рисунок 16. Використання Palette API

Висновок

Використання широкого спектру технологій дозволило мені детально, не лише в теорії, а й на практиці закріпити набуті знання.

Застосунок написаний із використанням основних сучасних технологій мобільної розробки під операційну систему Android та враховує принципи чистої архітектури, що робить його масштабованим та відповідно відкриває простір можливостей для подальших покращень та досліджень.

Таким чином мені вдалося розробити повністю готовий до використання програмний продукт, який дозволяє зробити відвідування спортивної зали не тільки корисним, а ще й зручним та простим.

Список літератури

1. MVVM Pattern [https://docs.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/mvvm]
2. Firestore [https://firebase.google.com/docs/firestore/quickstart]
3. Firebase Auth [https://firebase.google.com/docs/auth/android/start]
4. Firebase Storage [https://firebase.google.com/docs/storage/android/start]
5. Glide [https://github.com/bumptech/glide/blob/master/README.md]
6. Jetpack Navigation [https://developer.android.com/guide/navigation]
7. RxJava [https://reactivex.io/documentation]
8. Koin [https://insert-koin.io/docs/reference/koin-core/start-koin]
9. XZing Embedded [https://github.com/journeyapps/zxing-android-embedded/blob/master/README.md]
10. Material Design [https://material.io/design]
11. Palette API [https://developer.android.com/training/material/palette-colors]
12. Shared Preferences [https://www.tutorialspoint.com/android/android_shared_preferences.htm] [https://developer.android.com/training/data-storage/shared-preferences]
13. Kotlin [https://kotlinlang.org/docs]
14. Dark theme [https://developer.android.com/guide/topics/ui/look-and-feel/darktheme]
15. Localization [https://developer.android.com/guide/topics/resources/localization]
16. Clean Architecture: A Craftsman's Guide to Software Structure and Design / Robert C. Martin - Prentice Hall, 2018 – 1432 с.