

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних технологій

Кваліфікаційна робота бакалавра

на тему: Розробка кросплатформених додатків
за допомогою Flutter

Виконав студент групи К-19
спеціальності 122 Комп'ютерні науки
Захарчук Нікіта Володимирович

Керівник д.т.н., професор
Казакова Надія Феліксівна

Консультант _____

Рецензент Копиченко І.Ю.,
регіональний координатор
програми EGAP

Одеса 2023

ЗМІСТ

Перелік скорочень.....	5
ВСТУП	6
1 ВИБІР АРХІТЕКТУРИ ТА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ	
ЗАДАЧ	8
1.1 Аналіз найпоширеніших фреймворків	8
1.2 Аналіз архітектурни фреймворків.....	9
1.3 Система контролю версій.....	12
1.4 Сховище даних / авторизація.....	12
1.5 Програмні засоби	13
1.6 Середовище розробки.....	13
1.7 Фреймворк	14
1.8 Мова програмування.....	15
1.9 Потреби для встановлення ПЗ	17
1.10 Архітектура.....	17
2 FLUTTER ДОДАТОК	21
2.1.Порівняння Flutter з нативом	21
2.2 Hello World!	21
2.3 Stateless та Stateful Widgets	23
3 АУДІОКАЗКИ ПЛАН РОЗРОБКИ	25
3.1 Загальні положення.....	25
3.2 Структура мобільного додатку.....	25
3.3 Опис об'єктів моделей та функціоналу	30
4 ОПИС ПРАКТИЧНОЇ РЕАЛІЗАЦІЇ	31
4.1 Загальні положення.....	30
4.2 Firebase сховище	51
4.3 Ринок Flutter розробників.....	54
ВИСНОВКИ	58
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	59

ДОДАТОК А..... 60

ТЕРМІНИ, СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

Snapshot – дословно означає “знімок”, в системі контролю версій характеризує стан, версію додатку.

Аватар – фото, яке використовується для відображення в аккаунті або профілю користувача.

Репозиторій (англ. repository) – місце, де зберігаються й підтримуються які-небудь дані [1]. Найчастіше дані в репозиторії зберігаються у вигляді файлів, доступних для подальшого розповсюдження по мережі. Термін репозиторій можна дослівно перекласти, як сховище.

Шейрити (англ. share) – відсилати інформацію у додатку.

Android (укр. Андроїд) — операційна система і платформа для мобільних телефонів та планшетних комп'ютерів, створена компанією Google на базі ядра Linux.

Push – перехід на наступну/вказану сторінку.

Push-повідомлення – повідомлення з спливаючим вікном тексту.

Real-Time – зміни які відбуваються наглядно, швидко та на очах.

Route (укр. Шлях) – рути, це різні шляхи до екранів додатку.

State Manager – підхід до відстежування бізнес-процесів додатку.

Subscription (укр. Підписка) – підписка, або слідкування за якоюсь подією, при зміні якої приходить повідомлення та інформація про цю подію.

ОС – Операційна система.

ПЗ – Програмне забезпечення.

ПК – Персональний комп'ютер.

API – Application Programming Interface.

BaaS – Backend as a Service.

Натив – з англійського “Native”. Мова програмування/додаток заточений під написання програм специфічної платформи.

ВСТУП

Кількість компаній які потребують додаток для декількох платформ з кожним днем зростає. З ростом такої потреби зростає необхідність в рішеннях які дозволять швидко і вигідно розробити додаток для бізнесу. Найпоширенішою комбінацією для мобільних пристроїв являється iOS та Android. Раніше, для цього потрібно було наймати дві команди розробників та створювати абсолютно однакові додатки. І навіть тоді, коли компанія приступає до розробки додатків, в силу специфіки платформ буває так, що ідентично не вдається відтворити створений план, та макет застосунків.

Одним з рішень, яке дозволяє компаніям розробляти додатки для декількох платформ швидко і ефективно, є використання кросплатформених технологій. Ці технології дозволяють розробникам створювати додатки, які працюють на різних платформах, використовуючи один і той же код. Це має значно скоротити час та витрати на розробку, а також полегшити підтримку та оновлення додатків.

Тема розробки кросплатформених додатків являється неймовірно актуальною на даному етапі розвитку не тільки мобільних, а й технологій взагалі. Дипломна робота має за мету аналіз кросплатформених рішень, їх прорівняння з традиційними засобами розробки, та між собою, аналіз технологій необхідних для розробки функціонального додатку.

У цій роботі буде розроблено повноцінний кросплатформений мобільний додаток, який дозволяє записувати, прослухувати та зберігати аудіо, розглянуті проблеми які виникають у процесі розробки, та їх вирішення. Буде застосовано та проаналізовано систему контролю версій, яка є необхідною для відстеження процесу проекту та його збереження на серверах GitHub'у. Буде використано технологію Firebase для реєстрації через номер телефону та хмарне сховище яке дозволить зберігати користувацькі данні. Буде проведений аналіз доступних рішень середовищ розробки, які між ними відмінності, для яких цілей краще підходять. Основною відмінністю

кросплатформених фреймворків являється мова програмування, буде визначена специфіка цих мов, та поглиблений аналіз мови застосованого фреймворку.

Дана бакаларська кваліфікаційна робота складається зі вступу, 6-ти розділів, висновків, переліку посилань з найменувань та 1 додатка. Повний обсяг роботи становить 59 сторінок, 43 рисунок та 11 посилань.

1 ВИБІР АРХІТЕКТУРИ ТА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ЗАДАЧ

1.1 Аналіз найпоширеніших фреймворків

Flutter – крос-платформний SDK з відкритим вихідним кодом, розроблений компанією Google. Дозволяє розробникам створювати високоякісні флаттер-додатки для різних платформ, включаючи iOS, Android, Linux, Windows, Mac та Google Fuchsia, полегшує крос-платформну розробку веб-додатків.

Переваги Flutter:

1. Багатий вибір віджетів з адаптованим дизайном. Компоненти інтерфейсу Flutter легко впроваджувати та використовувати, оскільки він дуже схожий на Material Design.
2. Активна, зростаюча спільнота.
3. Відмінна документація та підтримка від команди розробників.
4. Легке вивчення та використання фреймворку, мови програмування.
5. Можливість створювати версії Flutter-додатків для різних платформ з одного вихідного коду.
6. Швидкий запуск продукту на ринок.
7. Модульне підключення до вже наявних додатків.

React Native – випущений Facebook у 2015 році. Фреймворк з відкритим кодом, дозволяє розробникам створювати додатки для iOS, Android, Windows та macOS, використовуючи єдину кодову базу. Використовуючи JavaScript та архітектуру React, розробка на React Native також полегшує створення веб-версій цих додатків, що ще більше розширює їхню універсальність.

Переваги React Native:

1. Використовуючи нативні компоненти, фреймворк забезпечує нативний досвід, гарантуючи плавну анімацію, чуйні інтерфейси та чудову загальну продуктивність.

2. Завдяки великій та активній спільноті розробників, React Native отримує вигоду від постійної підтримки та оновлень.
3. Мова JavaScript дозволяє більшості Front-End розробникам з легкістю почати користуватись React Native.

Xamarin – популярна платформа для розробки мобільних додатків з відкритим вихідним кодом, призначена для розробників, які працюють з C#. Будучи розширенням платформи розробки .NET, Xamarin дозволяє розробникам створювати додатки для операційних систем iOS, Android та Windows.

Переваги Xamarin:

1. Єдиний технологічний стек, дозволяє розробникам використовувати єдиний технологічний стек, що базується переважно на C#.
2. Нативний користувацький досвід, забезпечує нативний користувацький інтерфейс, використовуючи нативні API та можливості кожної платформи.

1.2 Аналіз архітектури фреймворків

Flutter - це розширювана, багат шарова система, що базується на серії незалежних модулів. Кожен з них залежить від базового рівня. Кожен елемент рівня фреймворку є необов'язковим, який можна замінити.

Flutter надає доступ до функцій Hot Reload та Hot Restart, що дозволяють в живому часі бачити зміни на екрані, замість виконання процесу збірки кожен раз з початку.

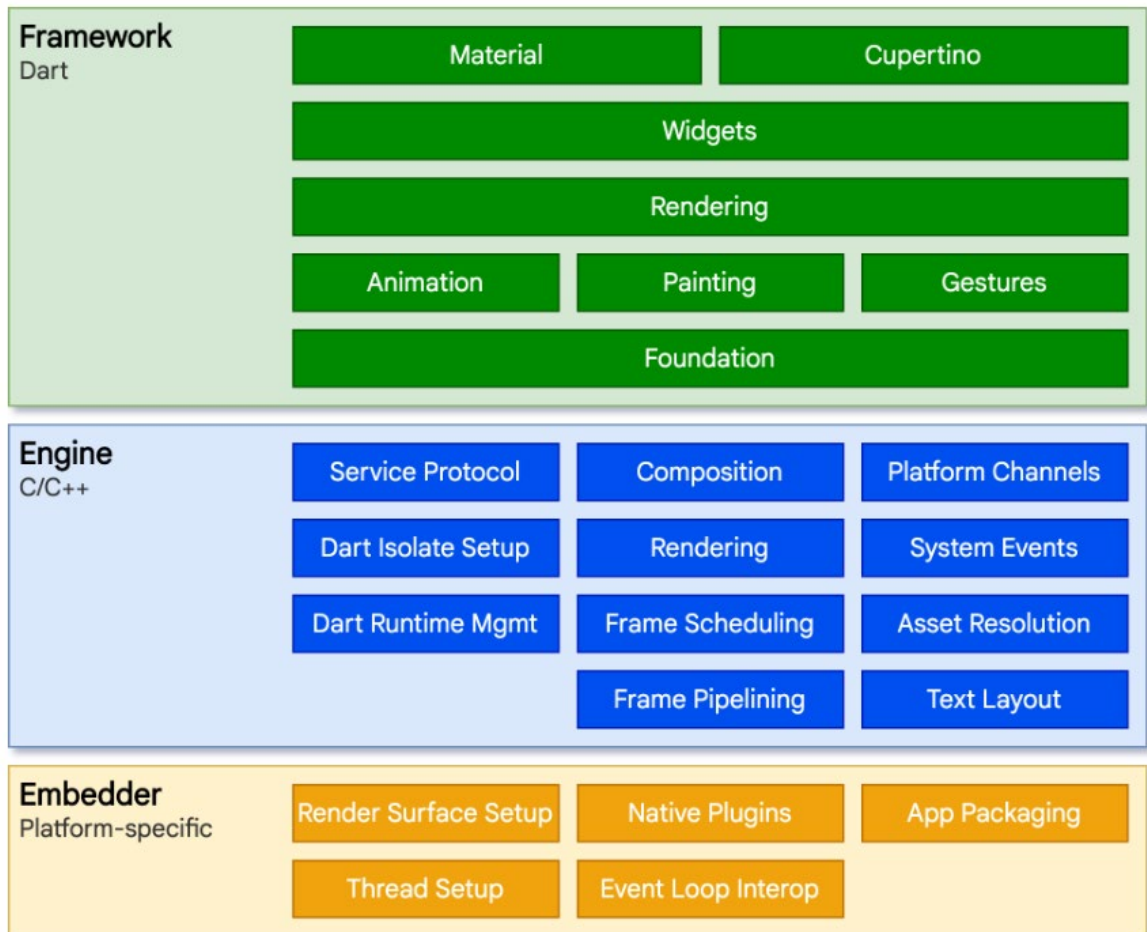


Рисунок 1.1 – Архітектура фреймворку Flutter

Архітектура React Native розкривається через взаємодію декількох ключових компонентів та процесів. Основні складові включають:

1. Код React: це основний розроблений розробниками код, який описує структуру та поведінку інтерфейсу користувача.
2. JavaScript Runtime: інтерпретує JavaScript код для взаємодії з нативними компонентами.
3. Мости (Bridges): мости – це елементи, які дозволяють JavaScript-коду звертатися до нативних API та функцій.
4. Нативні компоненти: передвизначені компоненти інтерфейсу користувача специфічні для платформи iOS або Android.
5. Native Modules: нативні модулі - це спеціальні JavaScript файли, які дозволяють JavaScript взаємодіяти з нативним кодом платформи.

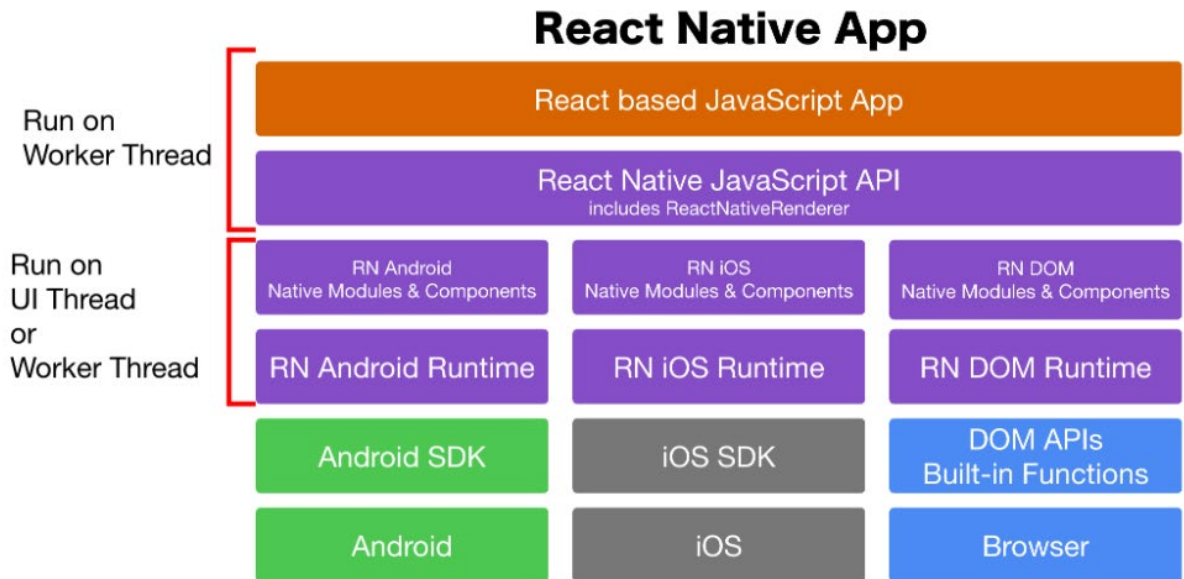


Рисунок 1.2 – Архітектура фреймворку React Native

Xamarin базується на NET. Тому він автоматично вирішує завдання розподілу пам'яті, сумісності та збору сміття з базовими платформами. Xamarin дозволяє створювати нативний інтерфейс на кожній платформі та використовувати C# для розробки бізнес-логіки за допомогою цього фреймворку.

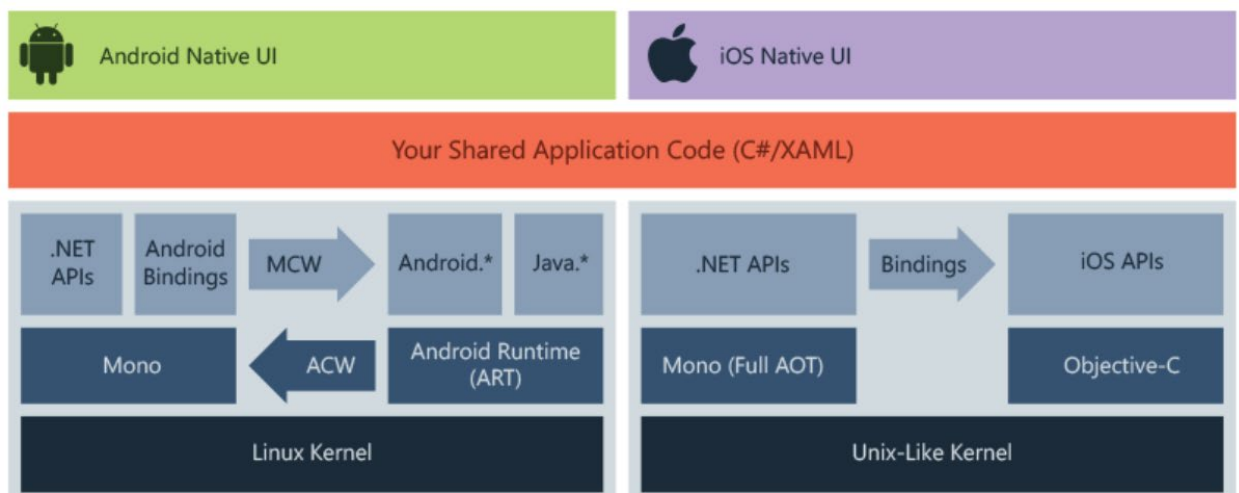


Рисунок 1.3 – Архітектура фреймворку Xamarin

1.3 Система контролю версій

Git – це система для керування та контролю версіями. Це найпопулярніший та безкоштовний інструмент, в якому зберігається код та історія його змін.

До основних завдань системи контролю версій Git належить:

1. збереження коду та історії змін;
2. збереження інформації про користувачів, які змінюють код;
3. можливість відкотити код до будь-якої версії;
4. можливість об'єднувати різні версії, зміни версій;
5. підготовка кінцевого коду до релізу.

GitHub є одним із сервісів на основі Git, саме він був використаний для зберігання, та транспортування коду між моїми пристроями для підтримки сучасної версії та розробки.

1.4 Сховище даних / авторизація

Портал AvadaMedia дає таке визначення Firebase [2]. Firebase – це платформа розробки мобільних додатків, яка дозволяє розробнику не відволікатися на створення бекенд, тобто прихованої від користувача програмної частини проекту. І це спрощує і прискорює створення мобільних додатків, дає можливість повністю зосередитися саме на UX / UI, тобто, на призначеному для користувача інтерфейсі і досвіді. Саме зв'язка Firebase з фреймворком Flutter дозволяє створювати швидкі програми для Android і iOS. Firebase – це одне з BaaS-рішень, яке дає розробнику масу можливостей:

1. управління базою даних;
2. хмарне сховище (для вмісту, створеного користувачами);
3. аутентифікація користувача;
4. push-повідомлення;
5. дистанційне оновлення;

6. хостинг;

7. інші функціональні можливості платформи або постачальника, наприклад, Firebase пропонує індексацію пошуку Google;

Firestore Realtime Database надає розробникам API, який синхронізує дані додатки між клієнтами і зберігає їх в хмарному сховищі. Додаток підключається до бази даних через WebSocket, який відповідає за синхронізацію даних протягом усього сеансу.

1.5 Програмні засоби

Для роботи з розширеним функціоналом, необхідно під'єднати наступні пакети з pub.dev [3].

`flutter_svg` – дозволяє використовувати .svg малюнки у вигляді іконок;

`flutter_sound` – впроваджує гнучку роботу з аудіо записами;

`GetX` – це надзвичайно легке та потужне рішення для Flutter. Яке поєднує в собі високопродуктивне керування станом за допомогою контролерів, інтелектуальне впровадження залежностей та керування маршрутами швидко й практично;

`GetX` має величезну екосистему, яка також може працювати з тим самим кодом на Android, iOS, Web, Mac, Linux, Windows;

1.6 Середовище розробки

Android Studio [5] – інтегроване середовище розробки (IDE) виробництва Google, за допомогою якого розробникам стають доступні інструменти для створення програм на платформі Android OS. Android Studio можна встановити на Windows, Mac та Linux. Android Studio створене з урахуванням IntelliJ IDEA.

Studio містить інструменти для розробки рішень для смартфонів і планшетів, а також нові технологічні рішення для Android TV, Android Wear, Android Auto, Glass, а також емулятори цих пристроїв.

1.7 Фреймворк

В даній роботі буде використано Flutter, оскільки він є найбільш сучасним та продуктивним рішенням на даний момент, має найкращу документацію та підтримує найбільшу кількість платформ.

Flutter – це фреймворк, який використовує мову Dart. Використовується для розробки кросплатформних додатків, які можна використовувати на наступних платформах: Android, iOS, Web, Linux, macOS, Windows та Google Fuchsia. Перший раз був представлений у 2015 році, а перша версія з'явилась всього у 2017.

Flutter складається з:

1. Flutter рушій — програмний рушій для рендерингу, написаний в основному на C++ з використанням графічної бібліотеки Google Skia. Він також використовує SDK платформ Android або iOS;
2. Базової бібліотеки (Foundation library) — бібліотека складається з класів та функцій (написані на Dart), які використовують для побудови Flutter програм, для взаємодії із Flutter рушієм;
3. Віджетів. Дизайн інтерфейсу користувача у Flutter будують з віджетів.

Віджет у Flutter являє собою незмінний об'єкт, який описує частину інтерфейсу користувача. Вся графіка, текст, фігури та анімації створюють за допомогою віджетів. Складні віджети створюють шляхом об'єднання простих. На поточний час Flutter містить два набори віджетів, які відповідають відповідним принципам побудови:

1. Віджети Material Design використовують дизайн Google.
2. Віджети Cupertino імітують дизайн Apple iOS.

1.8 Мова програмування

Портал Emeritus дає таке визначення мови Dart [5]. Dart – це об'єктно-орієнтована клієнт-оптимізована мова програмування, яка використовується для створення мобільних та веб-додатків на різних платформах. Розробляється компанією Google, та спочатку позиціонувалась як мова структурованого програмування для Веб. Розробники вважали, що в довгостроковій перспективі Dart може стати прогресивною заміною JavaScript, котрий потерпає від наявних в даний час проблем з розширюваністю, продуктивністю і підтримкою розробки складних застосунків. Мова має схожий на Java синтаксис, не вимагає явного визначення типів і її можна використовувати для створення серверних та клієнтських застосунків.

У березні 2015 компанія Google представила оновлену стратегію просування Dart, у котрій вирішено не прив'язувати Dart до браузеру.

Просуванню Dart як стандарту сприяло надання компанією Google всіх пов'язаних з розробкою патентів у безоплатне використання, що не вимагає оплати відрахувань (royalty free). Тобто користування мовою є повністю безкоштовним для розробників і open source рішенням, що сприяє швидкому розвитку мови завдяки контрибуторам з спільноти користувачів.

Існує чотири способи запуску Dart-коду:

1. Web;
2. Stand-alone;
3. Ahead-of-time compiled;
4. Native;

Щоб працювати в звичайних веб-браузерах, Dart покладається на компілятор від джерела до джерела JavaScript. Згідно з сайтом проекту, Dart був «сконструйований так, щоб було легко писати інструменти розробки, добре підходив до розробки сучасних додатків і був здатний до високопродуктивних реалізацій». У веб-браузері код попередньо компілюється в JavaScript. за допомогою компілятора dart2js, що робить його

сумісним з усіма основними браузерами, при цьому браузер не повинні його використовувати. Завдяки оптимізації скомпільованого виводу JavaScript, щоб уникнути дорогих перевірок і операцій, код, написаний на Dart, може в деяких випадках працювати швидше, ніж еквівалентний код, написаний від руки в ідіомах JavaScript.

Набір для розробки програмного забезпечення Dart (SDK) постачається з окремою віртуальною машиною Dart, що дозволяє коду Dart запускатися в середовищі інтерфейсу командного рядка. Оскільки мовні засоби, включені в SDK, написані переважно на Dart, Dart VM є важливою його частиною. Ці інструменти включають компілятор `dart2js` і менеджер пакетів під назвою `pub`. Dart постачається з повною стандартною бібліотекою, яка дозволяє користувачам писати повністю працюючі системні програми, такі як власні веб-сервери.

Код Dart можна АОТ-компільувати в машинний код (нативні набори інструкцій). Програми, створені за допомогою Flutter, пакета SDK для мобільних додатків, створеного за допомогою Dart, розгортаються в магазинах додатків як код Dart, скомпільований АОТ.

Dart 2.6 включає компілятор `dart2native` для компіляції в автономний, рідний виконуваний код. До Dart 2.6 ця функція доступна лише на мобільних пристроях iOS та Android через Flutter.

1.9 Потреби для встановлення ПЗ

Для того щоб встановити, та користуватись фреймворком Flutter середовище розробки має відповідати таким мінімальним вимогам [6]:

1. ОС: Windows 10 або новішої версії (64-розрядні), на базі x86-64;
2. місце на диску: 1,64 ГБ (не включає простір для IDE/інструментів);
3. інструменти: Flutter залежить від доступності цих інструментів у вашому середовищі: Windows PowerShell 5.0 або новішої версії (це

попередньо встановлено в Windows 10), Git для Windows з опцією Use Git з командного рядка Windows;

Встановлення відбувається за гайдлайном наданим в книзі “Flutter Cookbook” [7] та допомогою завантаження архіву, або клонування репозиторію з github [8].

1.10 Архітектура

Flutter не надає базового рішення для архітектури проекту та його структури. Тим не менш найпопулярнішим рішенням є підхід “Чистої архітектури” [9].

Принципи чистої архітектури не залежать від бібліотек та фреймворку який використовується, тому є універсальним засобом для будь якого проекту.

На даному графіку можна побачити залежності та зв'язки між шарами.

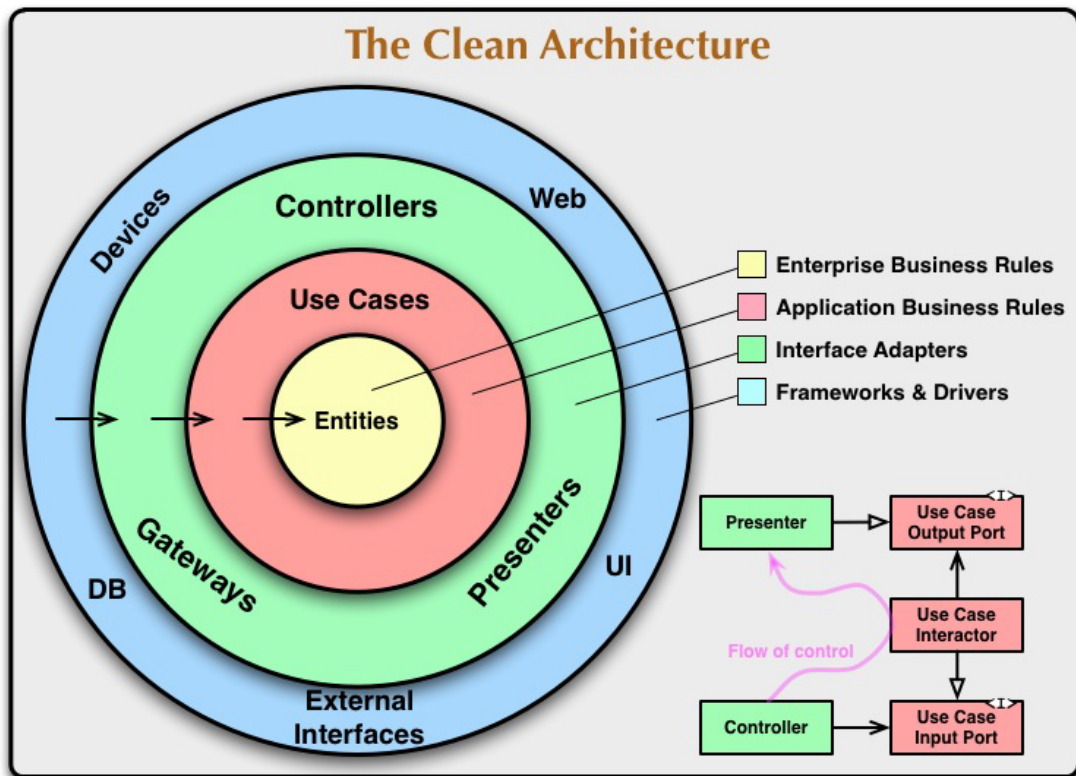


Рисунок 1.4 – Clean Architecture графік залежностей

В чистій архітектурі, ні один слой не має доступу до слоїв які знаходяться ззовні. Таким чином, архітектура відповідає наступним принципам.

Правило залежності (Dependency Rule). Концентричні кола представляють з себе різні області програмного забезпечення проекту. Загалом, чим далі вглиб, тим вищим стає рівень програмного забезпечення. Зовнішні кола - це механізми. Внутрішні кола - це політики. Таким чином це правило означає, що залежності у вихідному коді можуть вказувати лише всередину. Ніщо у внутрішньому колі не може знати нічого про будь що у зовнішньому колі. Зокрема, імена змінних та об'єктів, оголошених у зовнішньому колі, не повинні згадуватися у коді внутрішнього кола. Це стосується функцій, класів, змінних або будь-яких інших іменованих програмних об'єктів.

Сутності (Entities). Сутності інкапсують бізнес-правила та логіку проекту. Сутність може бути об'єктом з методами або набором структур даних і функцій. Це не має значення, бо вони можуть застосовуватись і у інших додатках, або проектах. Якщо сутності використовуються лише в рамках одного проекту, то ці сутності є бізнес-об'єктами додатку. Вони інкапсують найбільш загальні та високорівневі правила. Вони найменш схильні до змін, коли щось змінюється ззовні. На ці об'єкти не вплине зміна навігації по сторінці або безпеки. Жодні операційні зміни в конкретному додатку не повинні впливати на рівень сутностей.

Способи використання (Use Cases). Цей рівень містить бізнес-правила для конкретних застосунків або додатків. Він інкапсулює та реалізує всі варіанти використання системи. Ці варіанти використання організують потік даних до і від сутностей та спрямовують ці сутності на використання їх бізнес-правил для досягнення цілей їх використання. Зміни на цьому рівні не впливають на сутності. Так само на цей рівень не впливають зміни у зовнішніх середовищах. Цей рівень ізольований від зовнішніх змін.

Інтерфейсні адаптери (Interface Adapters). Програмне забезпечення цього рівня являє собою набір адаптерів, які перетворюють дані з певного формату, на формат зручний для використання в зовнішніх шарах, таких як база даних. Цей рівень реалізує архітектуру MVC (Model-View-Controller). До цього шару належать презентатори, представлення та контролери. Моделі, швидше за все, є просто структурами даних, які передаються від контролерів до варіантів використання, а потім назад від варіантів використання до презентерів і представлень. Жоден код всередині цього кола не повинен знати нічого про базу даних. На цьому рівні також знаходяться адаптери та мапери.

Фреймворки та драйвери (Frameworks and Drivers). Зовнішній шар який відносить до себе базу даних, фреймворк, зовнішні інтерфейси. Також цей шар імплементує інтерфейс UI. Цей шар є найвищим.

Дане архітектурне рішення є комплексним і дозволить уникнути будь яких проблем між взаємозв'язками коду. Ця система дуже легко піддається тестуванню та заміні елементів.

2 FLUTTER ДОДАТОК

2.1 Порівняння Flutter з нативом

Як зазначалося у вступі, Dart дуже схожий по синтаксису на C-подібні мови програмування, розглянемо наступний приклад Java та Dart кодів.

Java код

```
public static String getWord(String[] stringArray){
    StringBuffer result = new StringBuffer();
    for (int i = 0; i < stringArray.length; i++){
        if (i != stringArray.length - 1) {
            result.append(stringArray[i] + " ");
        } else{
            result.append(stringArray[i] + ".");
        }
    }
    return result.toString();
}
```

Dart код

```
static String makeSentence(List<String> stringArray) {
    StringBuffer result = StringBuffer();
    for (int i = 0; i < stringArray.length; i++) {
        if (i != stringArray.length) {
            result.write(stringArray[i] + ' ');
        } else {
            result.write(stringArray[i] + '.');
        }
    }
    return result.toString();
}
```

На прикладі цього коду можна зробити висновок, що через те, що синтаксис дуже схожий на C-подібні мови програмування, і у програмістів які

працювали з Java, C#, або іншими мовами, не буде ніяких проблем у вивченні цієї мови.

Також, завдяки розробнику Араону Оетрелу (Aaron Oertel), який переписав свій нативний Java застосунок на Flutter було проаналізовано на скільки зменшився написаний їм код [10].

```

Meta-Data for the native Android app
http://cloc.sourceforge.net v 1.60 T=0.42 s (431.4 files/s, 37607.1 lines/s)
-----
Language                files      blank      comment      code
-----
Java                    83         2405        512          8599
XML                     96         478         28           3577
Bourne Again Shell      1          19          20           121
DOS Batch                1          24          2            64
IDL                      1          2           0            15
-----
SUM:                    182        2928        562         12376

```

Рисунок 2.1 – Аналітика по Android Native додатку

```

Meta-Date for the Flutter app
http://cloc.sourceforge.net v 1.60 T=0.16 s (247.5 files/s, 14905.1 lines/s)
-----
Language                files      blank      comment      code
-----
Dart                    31         263         39           1735
Bourne Again Shell      1          19          20           121
DOS Batch                1          24          2            64
XML                      3          3           22           35
YAML                    1          9           9            17
Objective C             2          4           1            16
C/C++ Header            1          2           0            4
-----
SUM:                    40         324         93           1992

```

Рисунок 2.2 – Аналітика по Flutter додатку

З наведеного графіку можна побачити, що Android застосунок має 83 Java файли, та 8599 строк коду, у свій час Flutter має 31 файл, та 1735 строк, але не треба забувати про xml макети у Java застосунках, що робить даний рахунок 179/12176 проти 31/1735, вже дивлячись на це, можна зрозуміти

колосальну витрату часу на написання такої кількості коду, або навіть верстання xml-макету. Ознайомитись з даною статтею можна за посиланням №3.

Приклади продуктів, де використовується Flutter:

- Toyota розробили додаток для автомобілів з комп'ютерними системами на борту та смартфонів;
- Google Pay;
- BMW розробили додаток для електромобілів на Flutter;
- Alibaba Group, китайська найбільша платформа для продажу власних товарів;
- Епіцентр, найпопулярніший магазин будівельних та побутових речей в Україні розробив додаток для менеджерів, водіїв та кур'єрів;

2.2 Hello World!

Після встановлення, Flutter вітає наступним кодом [11].

```
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
    ),
  ),
}
```

```

        home: const MyHomePage(title: 'Flutter Demo Home Page'),
    );
}
}

```

```

class MyHomePage extends StatefulWidget {
  const MyHomePage({Key? key, required this.title}) : super(key: key);
  final String title;

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

```

```

class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(widget.title),
    ),
    body: Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          const Text(
            'You have pushed the button this many times:',

```

```
    ),  
    Text(  
      '$_counter',  
      style: Theme.of(context).textTheme.headline4,  
    ),  
  ],  
),  
),  
floatingActionButton: FloatingActionButton(  
  onPressed: _incrementCounter,  
  tooltip: 'Increment',  
  child: const Icon(Icons.add),  
),  
);  
}  
}
```

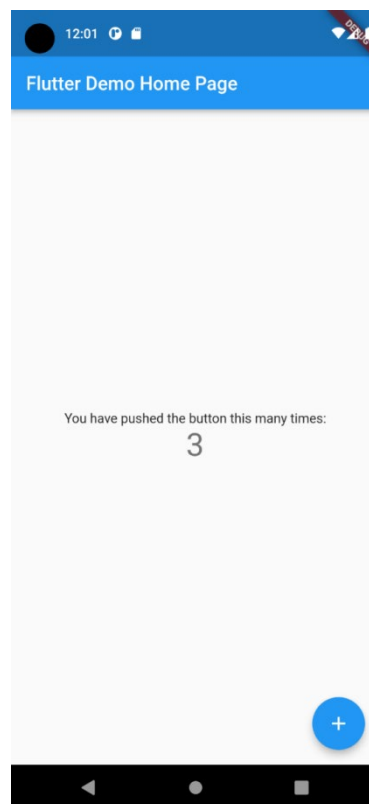


Рисунок 2.3 – Базовий Flutter застосунок

Для розуміння базової структури фреймворку треба створити “Hello World” застосунок. Спочатку треба імпортувати пакет flutter/material, що дозволить користуватись набором стандартних віджетів з Material Design. Далі, у методі main() скористатись функцією runApp(), попередньо вклавши в неї наш віджет MyApp, const ідентифікатор додається для пришвидшення виконання програми, явно вказуючи на те, що екземпляр класу не буде міняти вхідні параметри. У @override (перевизначеному) методі build(BuildContext context) повертається MaterialApp (головний віджет необхідний для відмальовки Material віджетів), який в параметрі home приймає холст Scaffold. Далі, завдяки віджету Center відмальовується Text(‘Hello World’) по центру.

```
import 'package:flutter/material.dart'; // підключаємо базовий набір віджетів

void main() {
  runApp(const MyApp()); // функція runApp запускає Flutter
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return const MaterialApp( // позначає, що ми використовуємо Material Design віджети
      home: Scaffold( // пустий холст, на якому ми будемо малювати
        body: Center( // віджет, який вирівнює вміст по центру
          child: Text('Hello World'), // можна догадатись ;)
        ), // Center
      ), // Scaffold
    ); // MaterialApp
  }
}
```

Рисунок 2.4 – Приклад коду Hello World

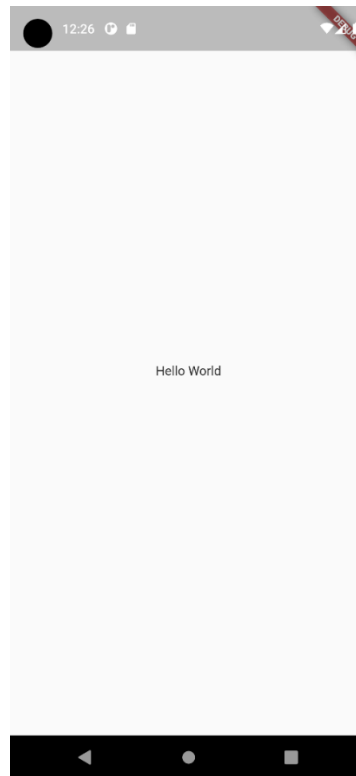


Рисунок 2.5 – Вивід на екран тестового застосунку з текстом “Hello World”

2.3 Stateless та Stateful Widgets

Створення Stateless та Stateful віджетів являється однією з основних особливостей створення UI у Flutter. Часто застосовується фраза “Everything is a Widget” (усе є віджетом) при описі Flutter додатку.

- StatelessWidget – віджет який не має змінного стану.
- StatefulWidget – віджет який оновлює стан в процесі виконання додатку.

StatefulWidget з перекладу означає “Віджет без стану”, але це не означає що його зовсім нема, адже віджети це просто класи, і їх властивості можуть бути змінені. Просто після того, як віджет буде намальований – зміни його стану не призведуть до оновлення цього віджета у UI. Такі віджети можна назвати константними.

StatefulWidget для реалізації свого стану потребують допоміжний клас State (стан). Стан — це інформація, яку можна прочитати синхронно під час створення віджета і можна змінити протягом життя віджета.

Приклади використання Stateless та Stateful віджетів:

Stateless Widget

```
class MyStatelessWidget extends StatelessWidget {
  const MyStatelessWidget({Key? key}) : super(key: key);

  // анотація @override потрібна для оптимізації,
  // що перевизначений метод батьківського класу ми використовуємо
  // не будемо, тому компілятор може викинути його
  @override
  Widget build(BuildContext context) {
    return const Text('Hello!');
  }
}
```

Stateful Widget

```
class Bird extends StatefulWidget {
  const Bird({
    Key? key,
    this.color = const Color(0xFFFFE306),
    this.child,
  }) : super(key: key);

  final Color color;
  final Widget? child;

  @override
  State<Bird> createState() => _BirdState();
}

class _BirdState extends State<Bird> {
  double _size = 1.0;

  void grow() {
```

```
    setState(() { _size += 0.1; });  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return Container(  
      color: widget.color,  
      transform: Matrix4.diagonal3Values(_size, _size, 1.0),  
      child: widget.child,  
    );  
  }  
}
```

3 АУДІОКАЗКИ ПЛАН РОЗРОБКИ

3.1 Загальні положення

Проект являється застосунком Memory Box для особистого користування. Додаток дозволяє записувати, програвати, систематизувати та зберігати аудіозаписи.

Основна функціональність - це можливість записувати, відтворювати, систематизувати та зберігати аудіозаписи. Зберігання аудіозаписів здійснюється наступним чином:

1. Без реєстрації запис зберігається локально на пристроях.
2. Зареєстровані користувачі доступні для розвантаження аудіозаписів у хмарне сховище з можливістю подальшого завантаження на пристрій.
3. Хмарне зберігання до 500 Мб доступне безкоштовно, можна придбати додаткове місце на диску

Приклад: Створення memory box, в якому людина може написати казку своїй дитині, створити аудіозапис із історією з життя, а потім, з часом, цінність таких записів лише збільшується.

3.2 Структура мобільного додатку

В даному підрозділі буде розглянуто структуру мобільного додатку, та зображено послідовність дій в додатку для досягнення певної мети.

Екран авторизації складається з:

1. Екран привітання;
2. Екран реєстрації;

Можна використовувати заявку відразу після завантаження, реєстрація не є обов'язковою. Під час першого запису з'являється екран з натяком на те, що реєстрація дозволяє зберігати аудіо в хмарному сховищі. Перед реєстрацією всі аудіофайли зберігаються локально на пристрої. Реєстрація можлива в будь-

який час використання програми і проводиться через підтвердження SMS - генерування короткого пароля. Подальші дзвінки до програми відбуваються без запиту на підтвердження, аж до виходу користувача через форму "Вихід із програми". Детальна інформація не вимагається під час реєстрації та заповнюється в будь-який час на екрані "профілю".

Бокове ліве меню викликається натисканням на кнопку "меню" у верхньому лівому куті та має наступні поля:

1. головний екран;
2. профіль;
3. підбірки;
4. усі аудіо-файли;
5. пошук;
6. нещодавно видалені;
7. написати в техпідтримку;

Головний екран – це вікно програми з основними функціональними елементами:

1. вихід до меню через бокову панель/кнопку меню;
2. список колекцій;
3. список аудіо-файлів;
4. кнопка аудіозапису;

Профіль складається з:

1. фото;
2. ім'я;
3. телефон;
4. видалення акаунта;
5. кнопка виходу із додатка;

Екран "Підбірки" представляє основний елемент сортування аудіо-файлів:

1. назва;
2. аватар;

3. короткий опис (необов'язкове поле);
4. номери: кількість файлів, загальна тривалість;
5. список відповідних аудіозаписів;

Підбірки можна:

1. створити (створюється папка, заповнюються поля, аудіозаписи вибираються із загального списку). Один аудіозапис може бути включений до різних підбірок;
2. формувати (заповняти аудіофайлами);
3. перейменовувати;
4. видалити (у цьому випадку аудіозаписи не видаляються, а перестають відповідати певному вибору). Важливо! Аудіо-запис може не відповідати одному вибору та бути доступним у "Усі аудіо-записи" /через "Пошук";
5. відтворити у випадковому порядку або відповідно до серійних кількості аудіозаписів. Підбірки можна поставити на повтор та програвати по колу;

Екран Усі аудіо-файли являється списком усіх аудіозаписів із іменами та аватарами та має наступні дії:

1. відтворити/зупинити;
2. вилучити;
3. відкрийте екран "Детальніше про аудіозаписи";

Екран "Детальніше про аудіозапис" відкривається при натисканні на функцію "Детальніше" при відтворенні аудіозаписів:

1. редагування інформаційних полей (ім'я, опис, аватар);
2. вступ та показ коментаря/опису;
3. додайте до вибору/видалення з вибору;
4. функція надсилання аудіозаписів до месенджера;
5. функція перенесення в "Нещодавно видалені аудіо-файли";

Пошук – це екран з лінією пошуку. Видає список аудіо-файлів за ключовими словами.

Нещодавно видалені аудіо-файли має наступні елементи:

1. список видалених аудіо-файлів;
2. назва та дата видалення;
3. можливість відновити;
4. можливість безповоротного видалення файлів з натиском та pop-up попередження;

Екран Написати в підтримку викликає поштовий клієнт з адресою електронної пошти. Екран запису аудіо-файлу відкривається після натиску кнопки “Запис” в центрі нижньої частини екрану.

1. назва аудіозапису. За замовчуванням “Запис №_”. Змінити назву можна пізніше;
2. тривалість запису – лічильник 0:00:00, з оновленням в онлайн режимі;
3. візуалізація звукової доріжки;
4. кнопка стоп. При натисканні запис зупиняється та зберігається;

Екран швидкого відтворювання нових аудіозаписів відкривається при збереженні файла та має наступні пункти:

1. назва з можливістю редагування при натиску;
2. поле для вводу текстового коментаря;
3. поле для завантаження аватарки запису;
4. тривалість у форматі 0:00;
5. шкала таймлайн;
6. кнопка play/pause;
7. іконка смітника;
8. іконка групування (три точки);

При натисканні іконки смітника з’являється повідомлення о видаленні, яке треба підтвердити. Якщо підтверджено, з’являється запис про те, що вайл перенесений у папку “Нещодавно видалені аудіо-файли”. З цієї папки можна або видалити, або відновити файл.

При натисканні на іконку групування з’являється екран вибору підбірки. Окрім цього, з’являється можливість створити нову підбірку і додати аудіо до

неї. Екран відтворення (плеєр) відкривається при натиску на аудіозапис. Відтворення файлу можливе у фоновому режимі.

Має наступні поля:

1. назва підбірки;
2. аватар аудіозапису;
3. назва;
4. кнопки play/pause;
5. назад/вперед (перелистування аудіозаписів в підбірці);
6. кнопка перемотки на 15 секунд назад/вперед;

Навігація по застосунку відбувається завдяки меню навігації зліва являється наслідуємим крізь увесь додаток, доступно для відкриття по свайпу зліва, або завдяки кліку на необхідний елемент (кнопка меню). Сторінки додатку можуть мати локальну навігацію.

3.3 Опис об'єктів моделей та функціоналу

Користувачі додатку можуть бути двох типів, не зареєстровані, або зареєстровані та авторизовані у додатку за допомогою номера телефону, які мають приватний набір аудіозаписів, що зберігаються на сервері. Аудіозаписи створюються шляхом запису голосу із вбудованого диктофона. Користувачі можуть:

1. створювати (записувати) аудіозаписи;
2. сортувати аудіозаписи;
3. відтворювати аудіозаписи;
4. зберігати аудіозаписи;
5. видаляти аудіозаписи;
6. шерити (надсилати через месенджери) аудіозаписи;

Аудіо-файли у форматі .mp3, або .асс, створені шляхом запису голосу з вбудованого диктофона. Аудіозапис має обмеження на тривалість - 60 хвилин. Підбірки є основним елементом сортування аудіо-файлів..

4 ОПИС ПРАКТИЧНОЇ РЕАЛІЗАЦІЇ

4.1 Загальні положення

При першому запуску з'являється вітальне вікно

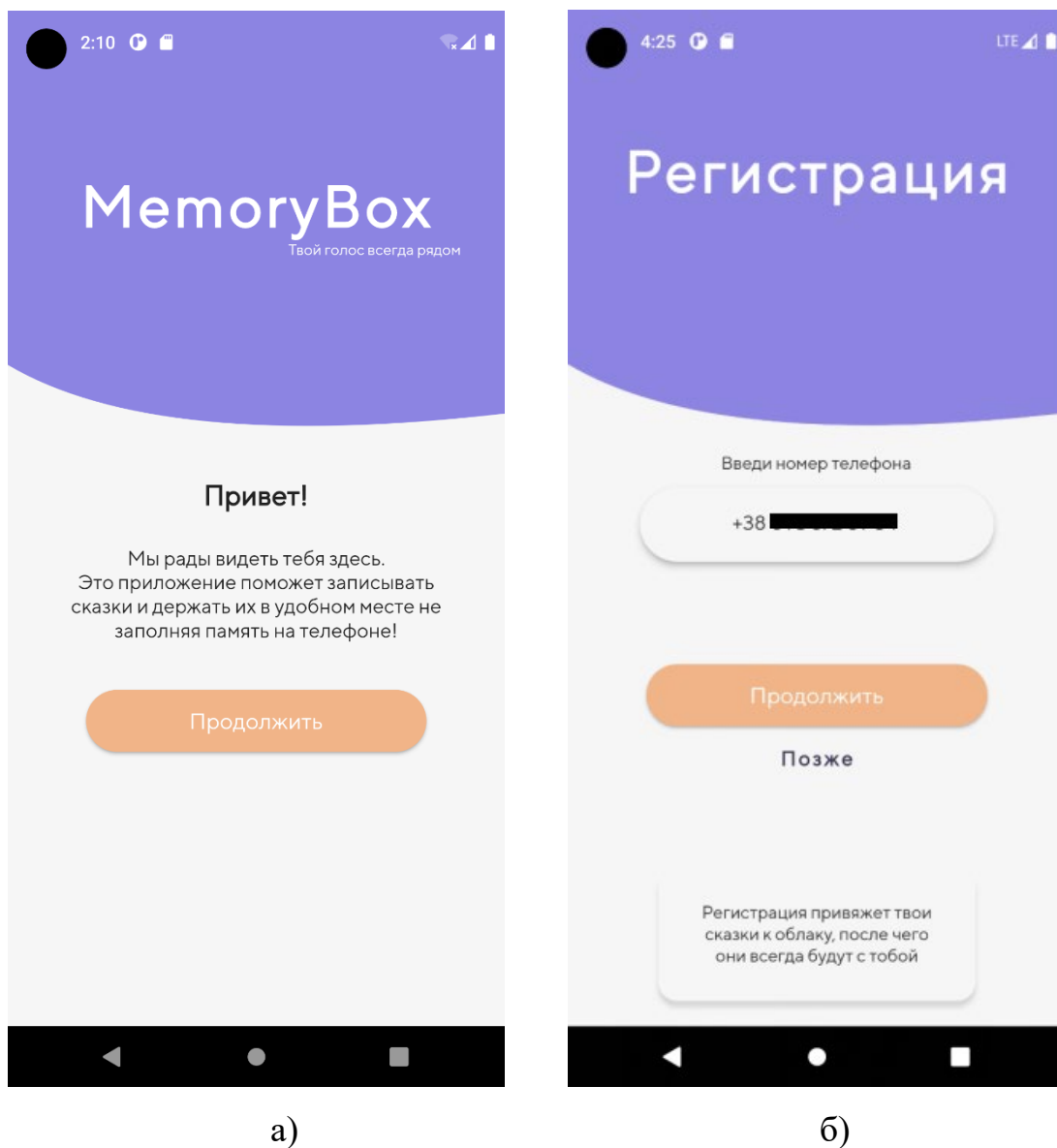


Рисунок 4.1 – а) Стартовий екран; б) Реєстрація за номером телефону

Після натискання кнопки “Продовжити”, яка називається `ElevatedButton` у Flutter коді і має параметр `onPressed`, в якому додана наступна функція:

```
Get.to(
  () => Registration(),
  transition: Transition.cupertino,
  duration: Duration(seconds: 1),
);
```

Здійснюється перехід на екран Реєстрації, з анімацією характерною для iOS пристроїв. Після цього, в наступному вікні вводиться номер телефону для підтвердження авторизації.

Після цього виконується перевірка:

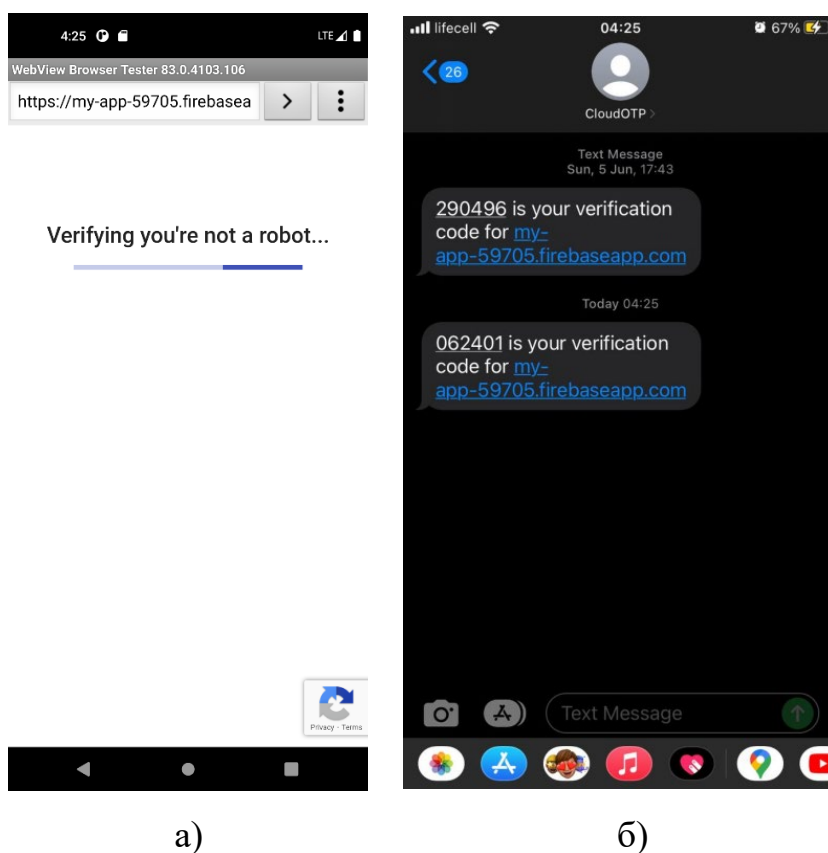


Рисунок 4.2 – а) Перевірка на робота; б) Повідомлення з смс

Після перевірки на телефон приходить повідомлення з кодом підтвердження.

Вікно для вводу коду з повідомлення:

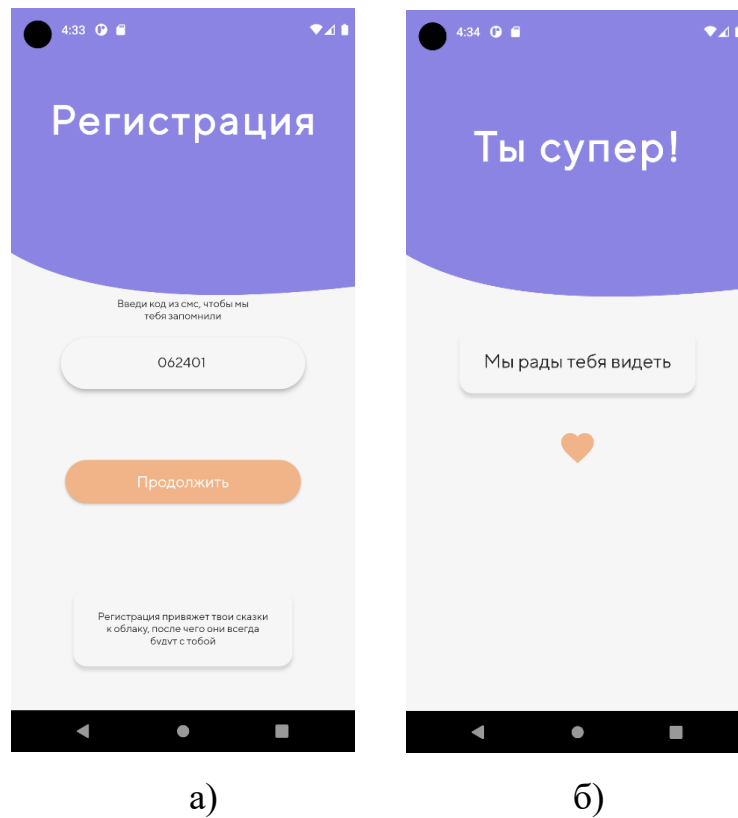


Рисунок 4.3 – а) Экран реєстрації; б) Экран привітання після реєстрації

Після переходу з реєстрації на 2 секунди з'являється екран привітання.
Після завершення попереднього таймеру відкривається головний екран.

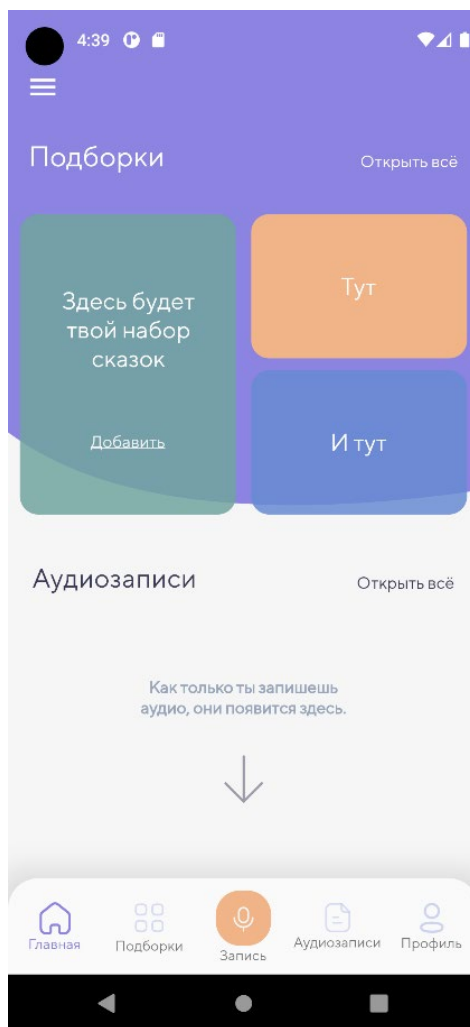


Рисунок 4.4 – Головний екран

При першому переході на екран запису з'являється діалогове вікно для запиту доступу на користування мікрофоном, де можна дозволити при використанні додатку, тільки цей раз, або відмовитись надавати доступ.

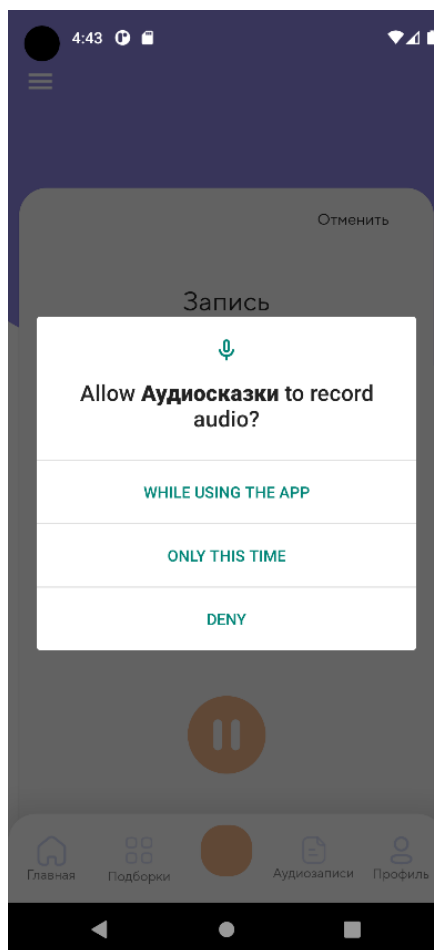


Рисунок 4.5 – Запит на надання доступу до мікрофона

Для того щоб мікрофон емулятора захоплював звук з комп'ютера треба увімкнути “Virtual microphone uses host audio input” у настройках емулятора.

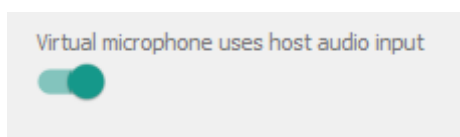


Рисунок 4.6 – Доступ до мікрофону комп'ютера

Після цього почнеться запис з візуалізацією гучності мікрофона.

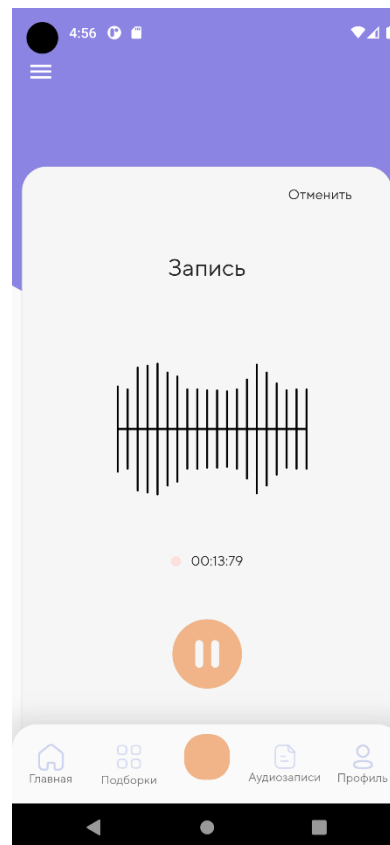


Рисунок 4.7 – Візуалізація запису

Реалізація алгоритму відображення (динамічна анімація):

```
class _ShapePainter extends CustomPainter {
  List maxPoints = [];

  _ShapePainter(this.maxPoints);

  @override
  void paint(Canvas canvas, Size size) {
    // maxPoint <= 0 ? maxPoint = 0 : null;

    Paint paint = Paint()
      ..color = Colors.black
      ..strokeWidth = 2
      ..strokeCap = StrokeCap.round;
  }
}
```

```

for (int i = 0; i < 20; i++) {
    Offset startPoint = Offset(size.width / 2 + i * 10 - 90,
size.height / 2);
    Offset endPoint =
        Offset(size.width / 2 + i * 10 - 90, size.height / 2 +
maxPoints[i]);

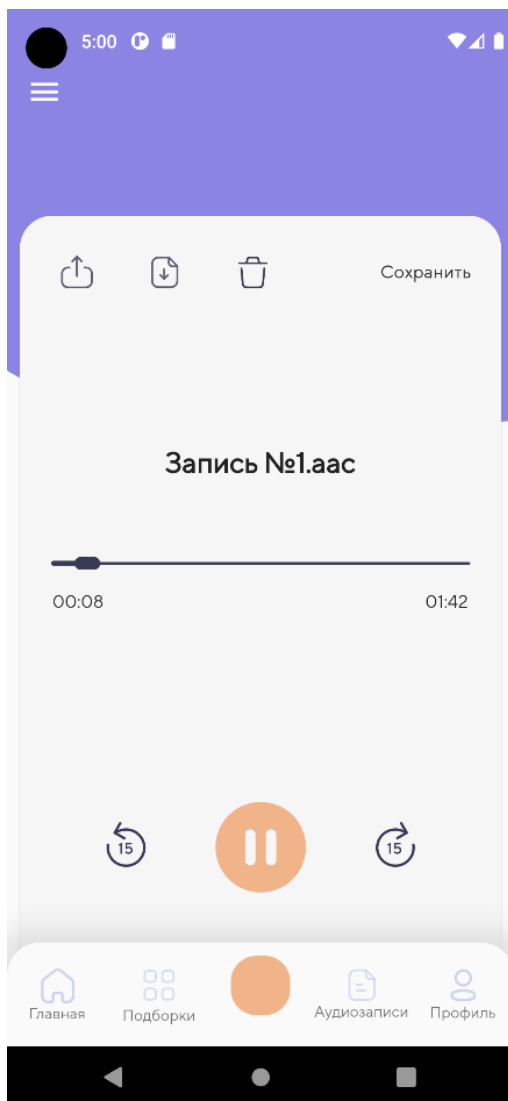
    canvas.drawLine(startPoint, endPoint, paint);
    canvas.drawLine(startPoint, endPoint.scale(1, -1), paint);
}
canvas.drawLine(Offset(size.width / 2 - 90, size.height / 2),
    Offset(size.width / 2 + 100, size.height / 2), paint);
}

@Override
bool shouldRepaint(covariant CustomPainter oldDelegate) {
    return false;
}
}

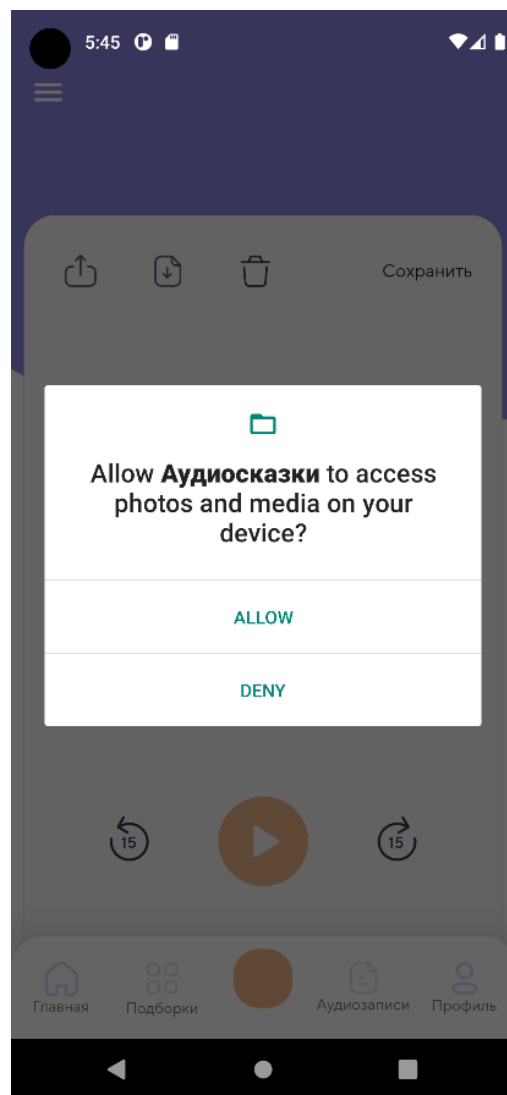
```

Даний клас ShapePainter приймає на вхід лист точок “maxPoints”, по яким відмальовує графік гучності в онлайн режимі.

Далі, після запису аудіо відкривається екран прослуховування запису.



а)

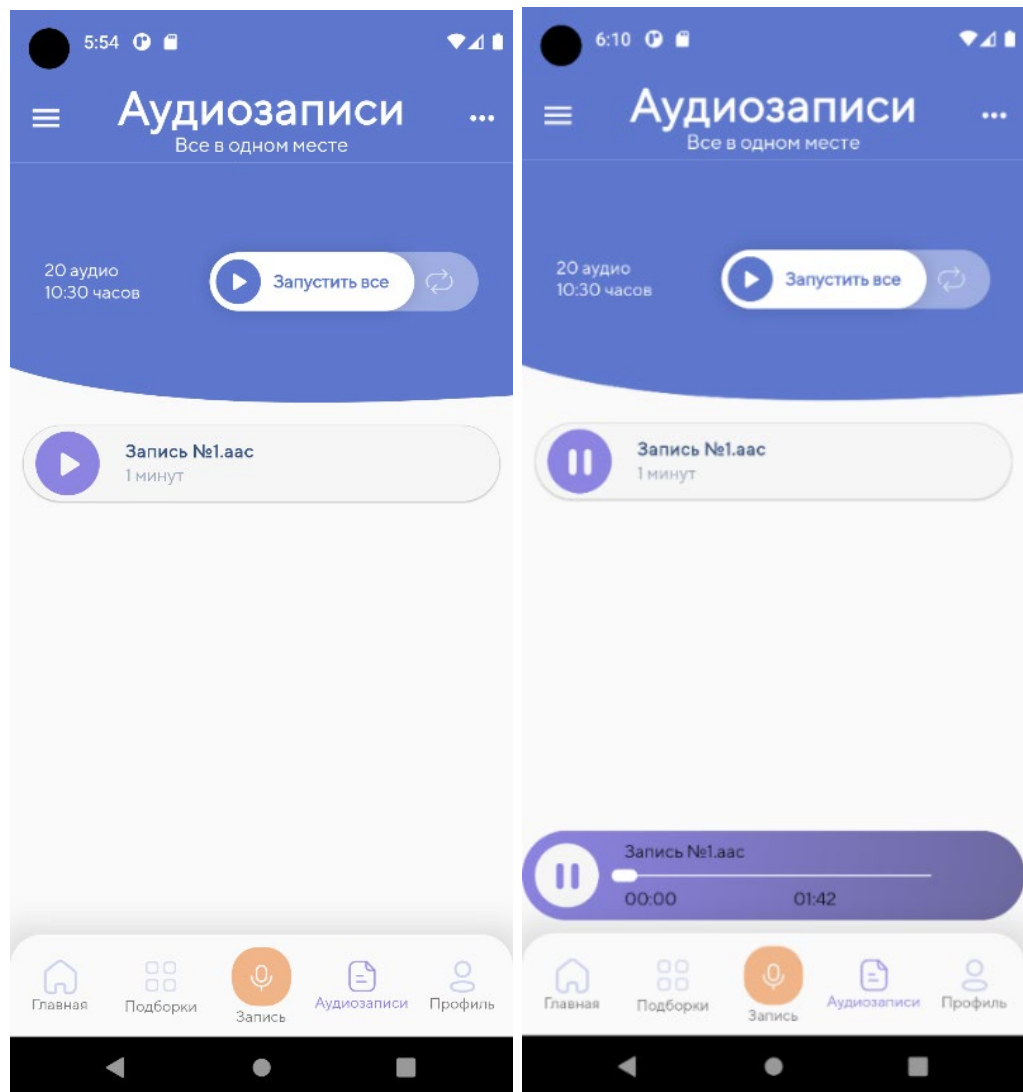


б)

Рисунок 4.8 – а) Прослуховування треку; б) Вікно запису доступу

При натиску іконки характеризуючу локальне збереження аудіозапису, відкривається вікно запису доступу до локального сховища.

Сторінка аудіозаписів, де відображаються всі збережені аудіофайли.



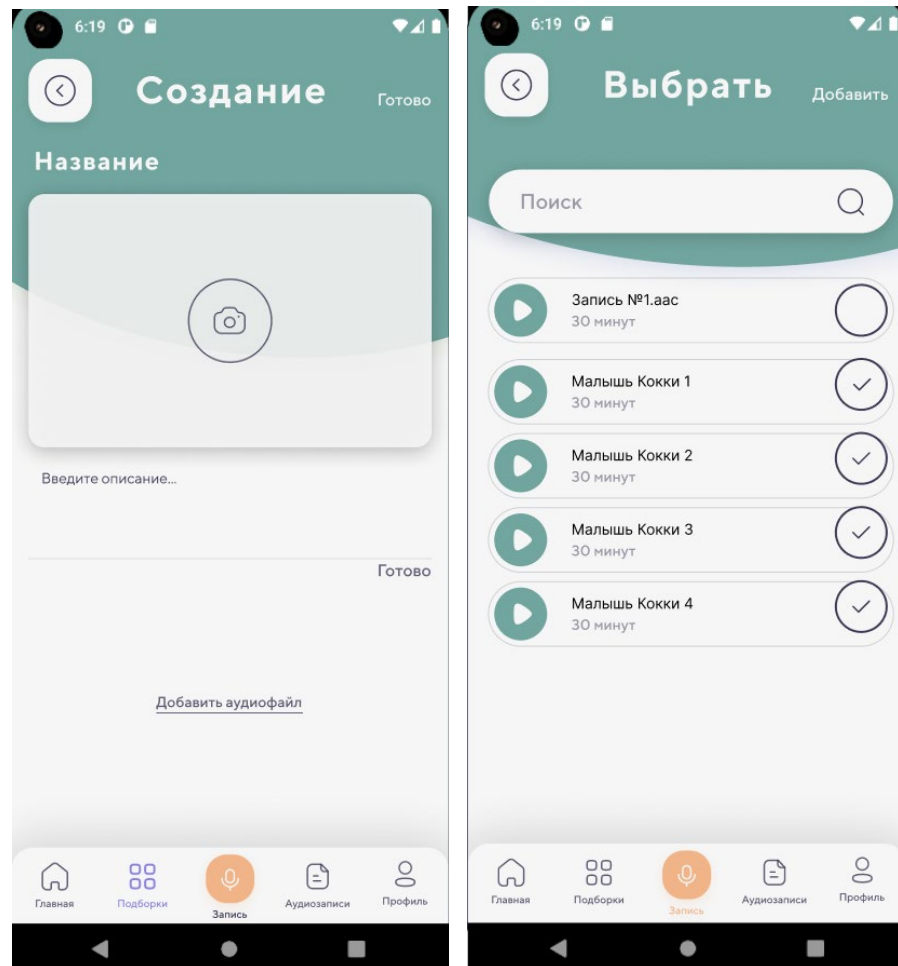
а)

б)

Рисунок 4.9 – а) Экран відображення аудіозаписів; б) Экран аудіозаписів у стану прослуховування

На цьому екрані можна прослуховувати збережені аудіо-файли.

Екран “Підбірки”, створений для збереження аудіофайлів у підбірки. Перейшовши на екран підбірок, на екрані відобразиться створення підбірки одразу ж, якщо не має ні одної збереженої підбірки. Для створення підбірки, вибирається фото, опис, та додати вибрані аудіозаписи з листу.



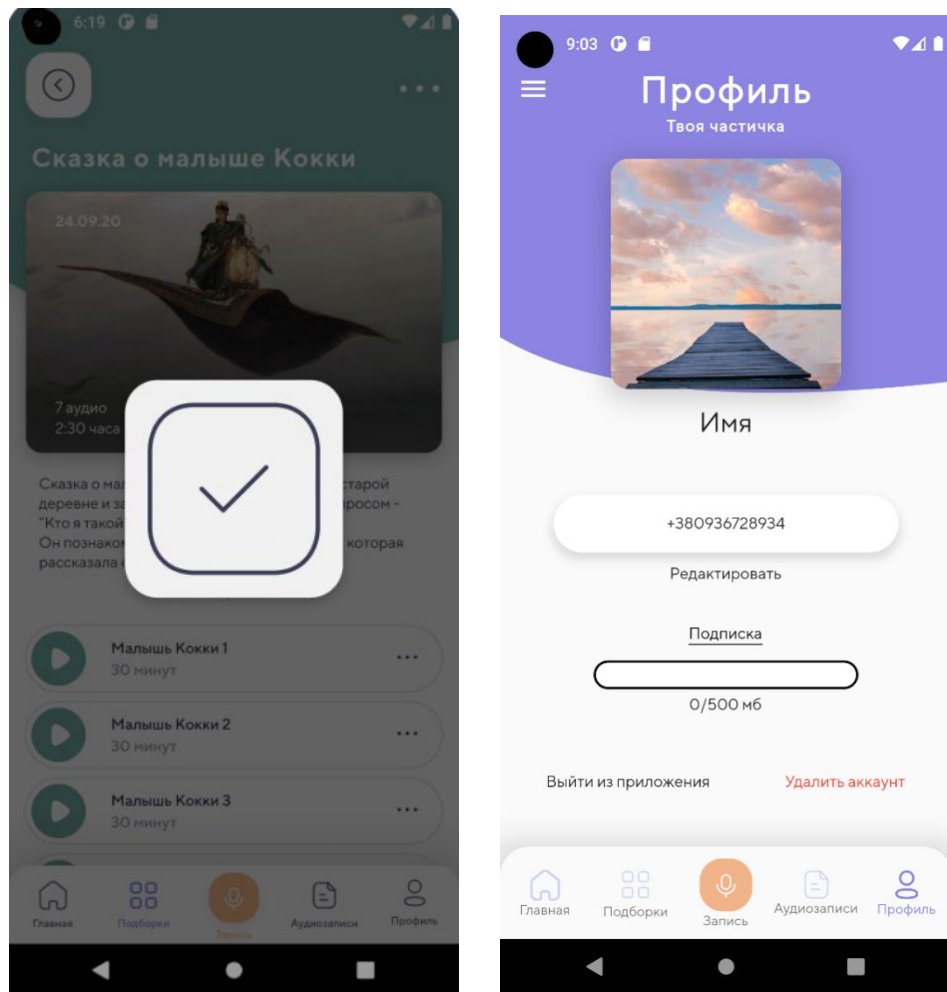
а)

б)

Рисунок 4.9 – а) Экран створення підбірок; б) Прибирання аудіофайлів

Після вибору необхідних аудіозаписів та фото, необхідно натиснути кнопку “Додати”.

Після успішного створення з’являється тимчасове вікно з підтвердженням об успішності створення підбірки.



а)

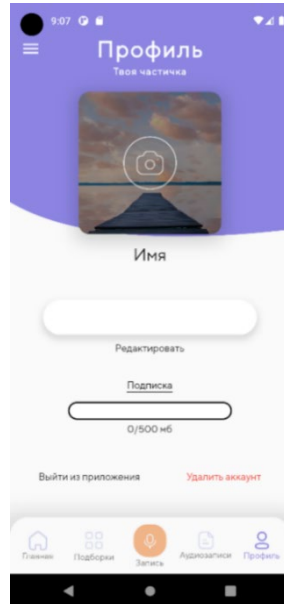
б)

Рисунок 4.10 – а) Успішне створення підбірки; б) Профіль

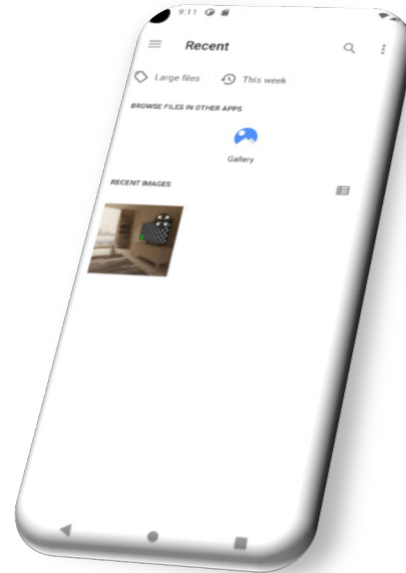
Екран профілю, де можна поміняти, або зареєструвати номер телефону (якщо не був зареєстрований до цього), змінити аватар та вийти або видалити акаунт.

У даному випадку усі операції збереження даних будуть проводитись на стороні Firebase Storage.

Натиснувши на кнопку редагування профілю, відкривається наступний вид редагування екрану.



а)



б)

Рисунок 4.11 – а) Профіль в режимі редагування; б) Вибір фото з накопичувача

Натиснувши на карнитку профілю відкривається меню вибору нового фото. Після вибору та підтвердження, фото в профілі змінюється.

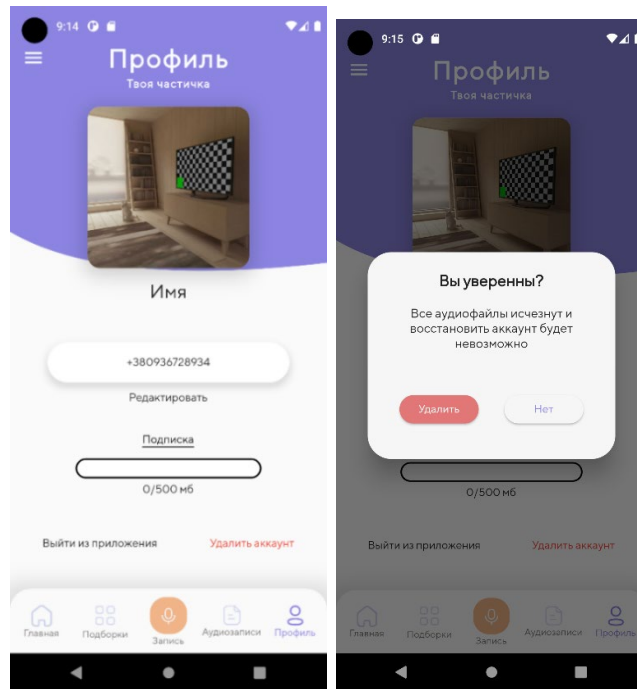


Рисунок 4.12 – а) Профіль зі зміненим фото; б) AletrDialog при видаленні

Також можна видалити акаунт, буде визвано AlertDialog з питанням про підтвердженні цієї дії.

У додатку є бокова панель для зручної навігації.

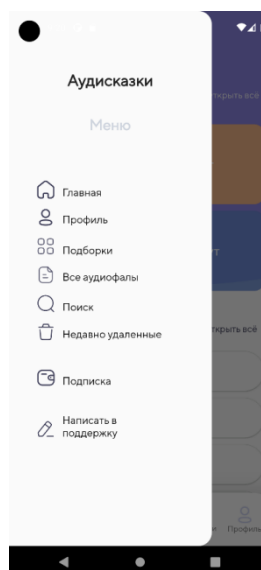


Рисунок 4.13 – Бокова панель застосунку

4.2 Firebase сховище

Головна сторінка Firebase вітає нас переліком проєктів.

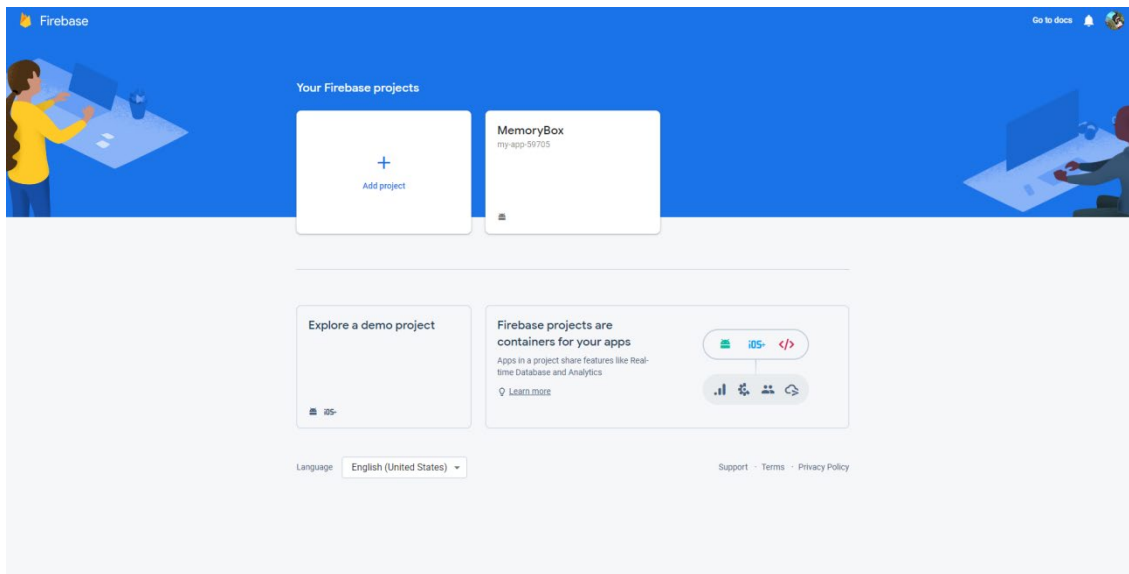


Рисунок 4.14 – Головна сторінка Firebase

Необхідно вибрати проєкт MemoryBox.

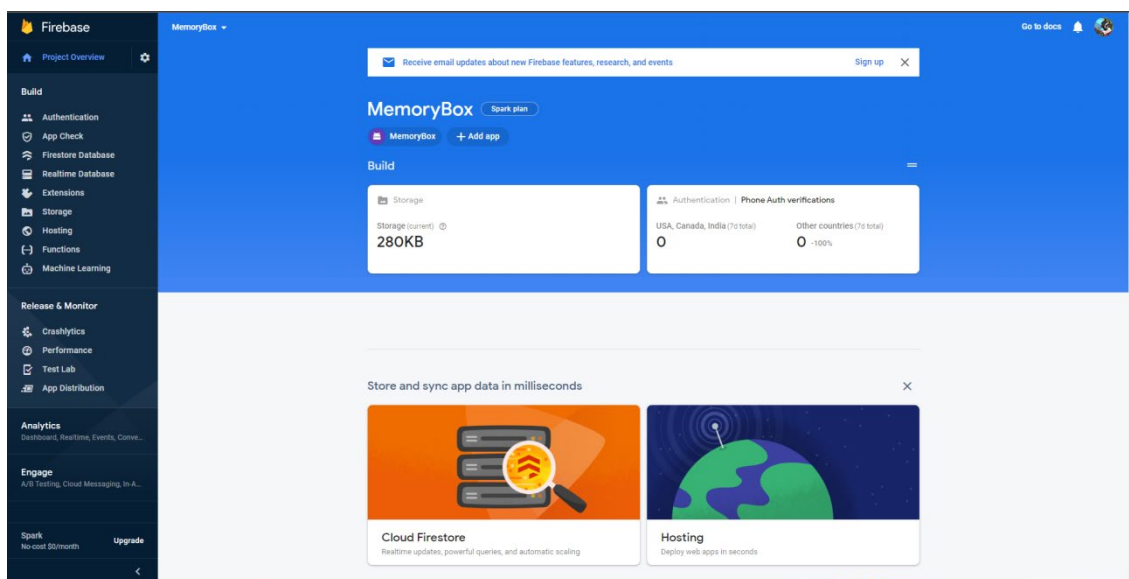
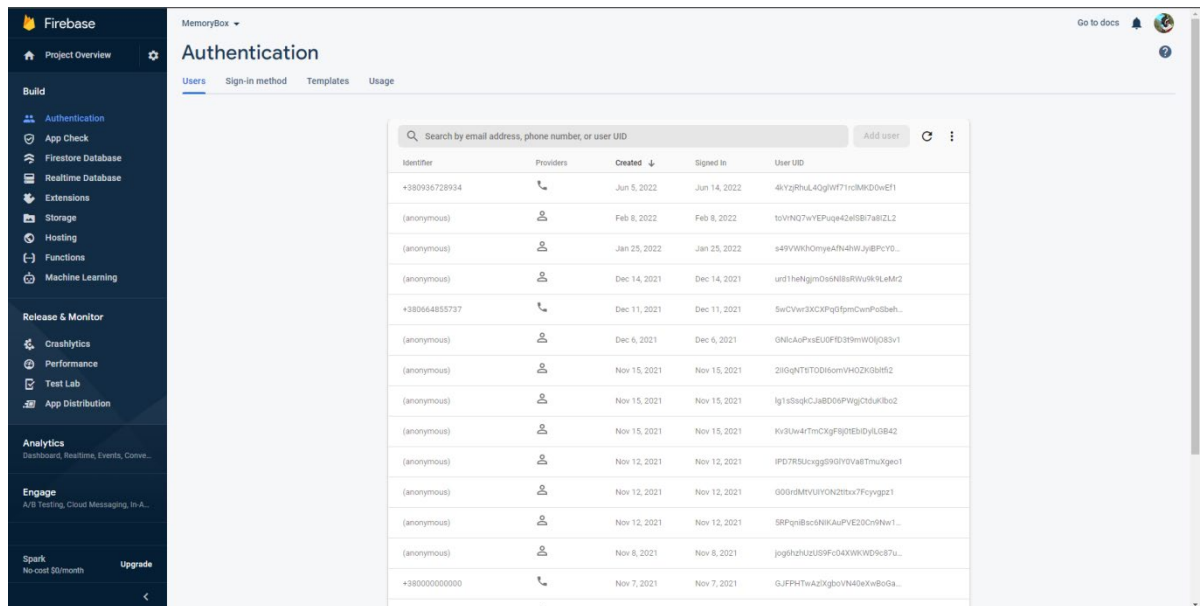


Рисунок 4.15 – Сторінка проєкту

На цій сторінці відображається скільки проект використовує пам'яті у сховищі, та бокову панель, де можна користуватись функціями “Authentication”, “Firestore Database”, “Storage” та ін. У вкладці Authentication відображаються усі зареєстровані і незареєстровані користувачі.



Identifier	Providers	Created	Signed In	User UID
+380936728934	Phone	Jun 5, 2022	Jun 14, 2022	4kYzRnL4QgWf71cIMK0d4E1
(anonymous)	Anonymous	Feb 8, 2022	Feb 8, 2022	toWNO7wEPuq42MIS87aBZL2
(anonymous)	Anonymous	Jan 25, 2022	Jan 25, 2022	s4RvWkOmYsA4N4WJyBPCYO...
(anonymous)	Anonymous	Dec 14, 2021	Dec 14, 2021	urd1heNgmDs6N8sRW9k9LeM2
+380664855737	Phone	Dec 11, 2021	Dec 11, 2021	SwCVv3XCKPg0fpmCwvPssBeh...
(anonymous)	Anonymous	Dec 6, 2021	Dec 6, 2021	GNcAaPxsEUGFD39mW0jG83v1
(anonymous)	Anonymous	Nov 15, 2021	Nov 15, 2021	2iGqN75TOD66omVHOZK98H52
(anonymous)	Anonymous	Nov 15, 2021	Nov 15, 2021	Ig1s3a9jCJa8D6iPWjgC5ukBo2
(anonymous)	Anonymous	Nov 15, 2021	Nov 15, 2021	Kv3Uw47mCXgF8j0E3bDyLGB42
(anonymous)	Anonymous	Nov 12, 2021	Nov 12, 2021	IPD785Jcgg9G0Y0V8TmUXgeo1
(anonymous)	Anonymous	Nov 12, 2021	Nov 12, 2021	00GdMvY3YON2btux7Fcyggz1
(anonymous)	Anonymous	Nov 12, 2021	Nov 12, 2021	SRPqB8oc8NKAuPVE20CnMw1...
(anonymous)	Anonymous	Nov 8, 2021	Nov 8, 2021	jgg6hzHuZuS9F04XWkWD5c87u...
+380000000000	Phone	Nov 7, 2021	Nov 7, 2021	GJFFHtwA2XgboVn40xw8oSa...

Рисунок 4.16 – Список користувачів додатку

Вкладка Firestore Database надає данні у форматі вкладеної колекції, що являється NoSQL сховищем, та дозволяє зберігати та оновлювати данні у онлайн режимі.

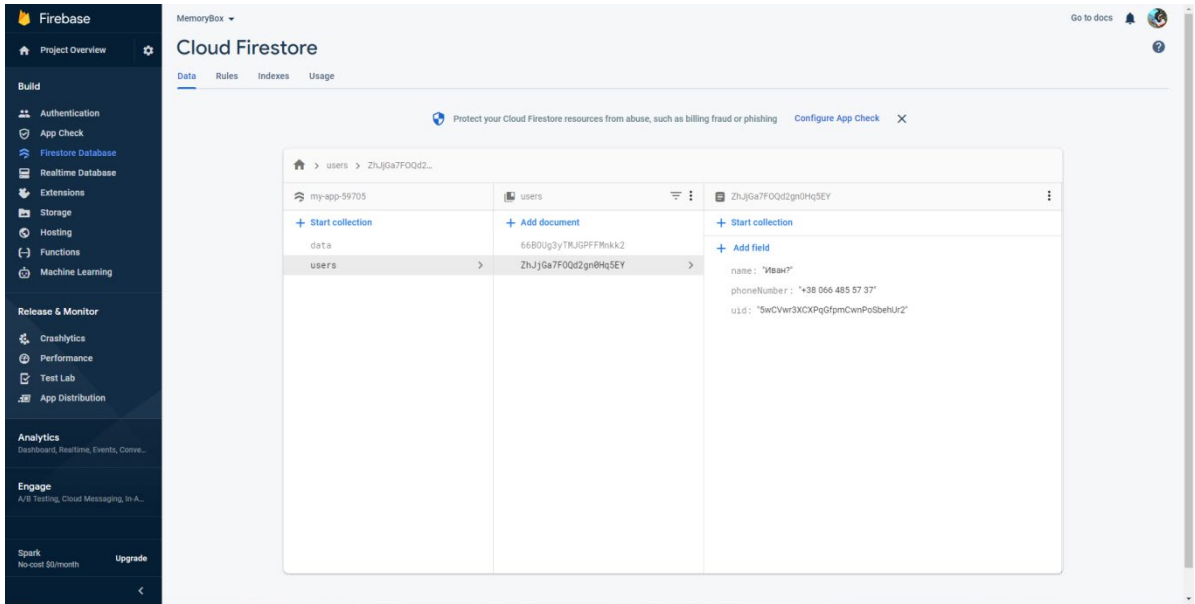


Рисунок 4.17 – Сховище даних Firebase

Storage відповідає за зберігання файлів у Firebase, ви можете бачити початкову фотографію для користувачів, та папку з записами усіх користувачів.

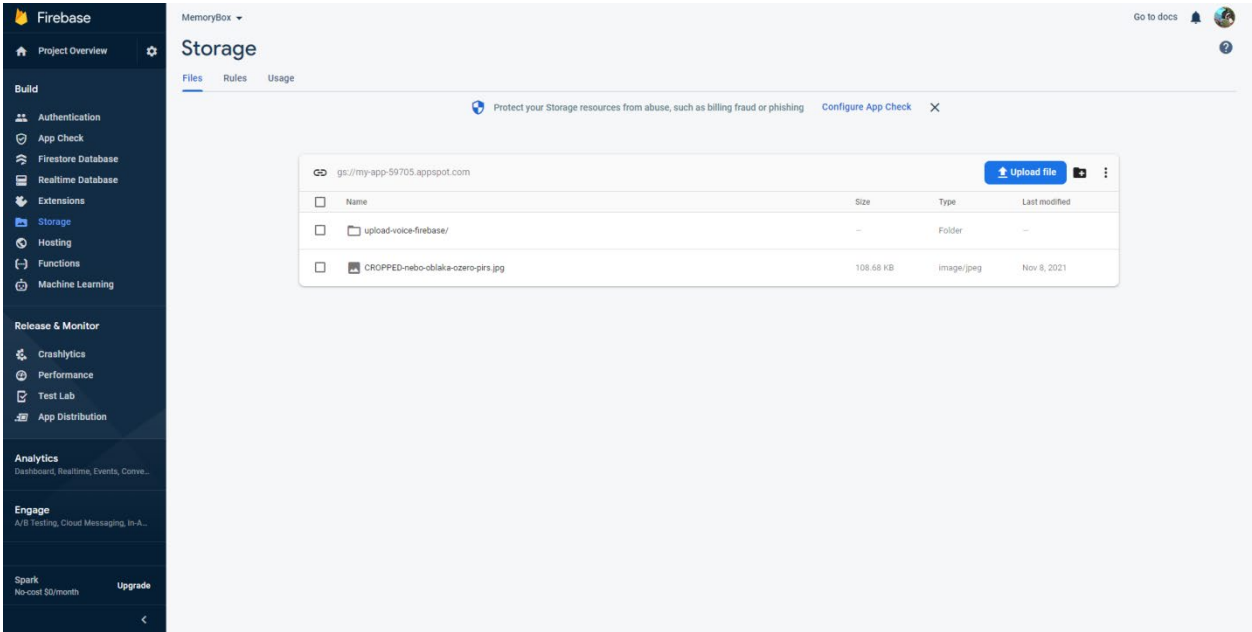


Рисунок 4.18 – Сховище файлів користувачів

4.3 Ринок Flutter розробників

При роботі з будь-яким фреймворком, або технологією, необхідним являється аналіз попиту для розуміння релевантності даної навички.

Вакансії надані djinni.co, який вважається одним з найкращих українських порталів для пошуку роботи в ІТ.

The screenshot shows the djinni.co website interface. At the top, there are navigation tabs: "Пропозиції", "Вакансії", and "Зарплати". The user profile in the top right shows the name "Nikita Z." and email "nik.shadkov@gm...".

The main heading is "Вакансії Flutter 108". Below it are filter buttons: "Всі", "За моїм профілем", "Мої підписки", and "Обране".

There are four job listings visible:

- Senior Flutter Developer \$4000**: Recruited by Peter Marets at Qualified Consult. Location: Україна. Product: Product. Full Remote. 5 years experience. Intermediate level.
- Flutter Developer**: Recruited by Anna Krasovskaya at Shark Software. Location: Україна. Full Remote. 1 year experience. Intermediate level.
- Middle\Senior Mobile Developer (IOS\Android) \$2500-4000**: Recruited by Vitaliy Basiuk at EvaCodes. Location: Україна (Київ). Product: Product. Full Remote. 3 years experience. Advanced/Fluent level.
- Flutter developer**: Recruited by Oksana Boyko at Powercode. Location: Україна (Запоріжжя, Київ). Office/Remote. 1 year experience. Intermediate level.

On the right side, there is a filter sidebar:

- Search for position: "Наприклад: Front-end engineer" with a search button.
- Expanded search: Розширений пошук
- Specialization: "Розробка" (Development)
- Skills: C# / .NET, Android, C / C++ / Embedded, Flutter (selected), Golang, iOS, Java, JavaScript / Front-End, Node.js, PHP, Python, Ruby, Scala.
- Other technical skills: Business Analyst, Data Science, Data Analyst, Data Engineer, DevOps, Design / UI/UX, 2D/3D Artist / Illustrator, Gamedev / Unity, Product Manager, Project Manager, Scrum Master / Agile Coach, QA Automation, QA Manual, SQL / DBA, Security, Sysadmin, Lead / Architect / CTO, SEO.
- Non-technical skills: Marketing, HR / Recruiters, Sales.

Рисунок 4.19 – Вакансії Flutter розробника на djinni.co

З 108 вакансій, треба відокремити вакансії для початківців фільтром, для того щоб мати розуміння попиту саме на новачків даної технології. 26 вакансій для позиції Junior Flutter Developer.

Вакансії Flutter 26

Всі За моім профілем Мої підписки Обране

Flutter Developer вчора

We are the team of developers focused on delivering the high-quality solution to enhance business. Our company provides outsource and outstaff services to reliable constant customers and startups. We prefer to use unique solutions for the best expectations of product's implementation.

Anna Krasovskaya, Recruiter в Shark Software
📍 Україна 🌐 Full Remote · 1 рік досвіду · Intermediate

Flutter developer вчора

Powercode company is currently looking for crazy-ambitious and result oriented person to join our team!

Oksana Boyko, IT Recruiter в Powercode
📍 Україна (Запоріжжя, Київ) 🌐 Office/Remote на ваш вибір · 1 рік досвіду · Intermediate

Flutter Developer вчора

Компания DODOTAP — работает в сфере mobile разработки. Мы создаем интересные проекты. Ищем Flutter Developer

Елизавета Воробченко, HR в Dodotap
📍 Україна, Польща 🌐 Full Remote · 1 рік досвіду

Flutter Engineer 10 червня

It is a platform for searching of extraordinary services for travelers, where you can book art/boutique hotels, find new "non-typical-touristic" places and co-working spaces, get useful acquaintances. Selina provides guests with beautiful places to stay, travel, and work abroad indefinitely.

Елена Тимошенко, Researcher в Cventy
📍 Україна 🏆 Product 🌐 Office/Remote на ваш вибір · 1 рік досвіду · Intermediate

Junior / Middle Flutter Developer 10 червня

Шукати за посадою
Наприклад: Front-end engineer
 Розширений пошук

Спеціалізація

Розробка

C# / .NET Android C / C++ / Embedded

Flutter Golang iOS Java

JavaScript / Front-End Node.js PHP

Python Ruby Scala

Ще технічні

Business Analyst Data Science

Data Analyst Data Engineer DevOps

Design / UI/UX 2D/3D Artist / Illustrator

GameDev / Unity Product Manager

Project Manager Scrum Master / Agile Coach

QA Automation QA Manual SQL / DBA

Security Sysadmin Lead / Architect / CTO

SEO

Нетехнічні

Marketing HR / Recruiters Sales

Customer/Technical Support

Рисунок 4.20 – Вакансії Junior Flutter розробника на djinni.co

Є така позиція, як Trainee Developer, що уявляє з себе позицію на якій розробника будуть вчити, це хороший варіант для початкових розробників.

Вакансії Flutter 3

Flutter Intern до \$200 6 червня

CHI Software is looking for a highly motivated Intern/Trainee Flutter, who wants to develop with us. Let's start working together :)

Elena Lyalyaeva, Recruiter в CHI Software
📍 Україна (Львів) 🌐 Office/Remote на ваш вибір · Без досвіду · Intermediate

Trainee Flutter Engineer 3 червня

Solid Software, a company that specializes in Dart/Flutter, is looking for a talented Trainee Flutter Engineer to join our team.

Yana Mandziuk, Talent Acquisition Specialist в Solid Software
📍 Україна 🌐 Full Remote · Без досвіду · Intermediate

Dart/Flutter Developer \$2290-2300 23 травня

Разработка, тестирование и поддержка проекта на Flutter удаленно. Ставка от 2300 USD. Переезд за границу

Роман Килелов, Рекрутер в IT Solutions Management International Pte. Ltd.
📍 Релкейт, Коста-Рика 🏆 Product 🌐 Full Remote · Без досвіду

Шукати за посадою
Наприклад: Front-end engineer
 Розширений пошук

Спеціалізація

Розробка

C# / .NET Android C / C++ / Embedded

Flutter Golang iOS Java

JavaScript / Front-End Node.js PHP

Python Ruby Scala

Ще технічні

Business Analyst Data Science

Data Analyst Data Engineer DevOps

Рисунок 4.21 – Вакансії Trainee Flutter розробника на djinni.co

На WORK.ua було нараховано 15 позицій.

The screenshot shows the WORK.ua website interface. At the top, there is a search bar with 'flutter' entered, a location dropdown set to 'Вся Україна', and buttons for 'Знайти' (Find) and 'Розширений пошук' (Advanced search). Below the search bar, it indicates '15 актуальних вакансій Flutter за 30 днів'. The main content area displays two job listings:

- Flutter developer**: Posted by 'Левел99' in Kyiv, 1.4 km from the center. Description: 'Повна зайнятість. Досвід роботи від 1 року. Flutter Developer МИ НА ТЕБЕ ЧЕКАЄМО По-перше, СЛАВА УКРАЇНІ! По-друге, СМЕРТЬ ВОРОГАМ! Привіт, Айтїшнік!...'. Includes a 'Зберегти' (Save) button.
- Flutter-разработчик**: Salary range '50 000 – 85 000 грн'.

On the right side, there is a section 'Ці вакансії за містами' (These vacancies by city) with a list: Київ, Дніпро, Львів, Одеса, Харків, and Ще. Below this is an advertisement for a t-shirt with the text '200 грн з кожної футболки'.

Рисунок 4.22 – Вакансії Flutter розробнику на WORK.ua

Robota.ua пропонує 9 вакансій

The screenshot shows the Robota.ua website interface. At the top, there is a yellow banner with the text 'Є вакансія, яка допоможе Україні? Опублікуємо безкоштовно' and buttons for 'Написати листа' (Write letter) and 'Заповнити форму' (Fill form). Below the banner, there is a navigation bar with 'roboata.ua', 'Рекомендовані', 'Створити резюме', 'Увійти', and 'Сайт для роботодавця'. The search bar contains 'flutter' and the location is set to 'Вся Україна'. The main content area displays 9 vacancies for Flutter in Ukraine:

- Junior Flutter Developer**: Salary '14 000 – 42 000 ₴', company 'UBRAINIANS', location 'Київ'. Includes buttons for 'Відгукнутись', a star, and a speech bubble. Posted '1 тиждень тому'.
- Trainee Flutter Engineer**: Company 'Solid Software', location 'Київ'. Includes buttons for 'Відгукнутись', a star, and a speech bubble. Posted '1 тиждень тому'. Features tags: 'Корпоративні заходи', 'Регулярний перегляд зарплат', 'Компенсація навчання', 'Компенсація спортзалу', and 'Безкоштовне навчання'.



Рисунок 4.23 – Вакансії Flutter розробнику на Robota.ua

На сайті з найбільшою спільнотою розробників в Україні dou.ua відображається 26 пропозицій [8].

Швидкий перехід: [Design](#), [початківцям](#), [віддалена робота](#), [робота за кордоном](#), [Poland](#).



26 вакансій в категорії Flutter RSS

13 червня 2022

Junior Software Developer with Flutter в  **Erbis**  Дніпро



WHO WE ARE: We are Erbis, a full-cycle development team building innovative solutions to help entrepreneurs and enterprises change the world for the better. We are looking for a Junior Developer to join our team. REQUIREMENTS: Internet : HTTP/HTTPS, DNS.

10 червня 2022

Flutter Developer (1+ year of experience) в  **Axon** \$1000–2000  Київ, віддалено


About the project Axon expects fruitful cooperation with high-skilled, open-minded and goal-oriented Engineers, who are able to expand the Flutter department. You will develop the podcast solution for the Axon team's education needs.

8 червня 2022

Lead Flutter Developer в  **Intetics**  Київ, Харків, Львів, Одеса, Дюссельдорф (Німеччина), Краків (Польща), віддалено



Intetics, a leading global technology company providing custom software application development, distributed professional teams, software product quality assessment, and "all-things-digital" solutions, is looking for Lead Flutter Engineer to enrich its team with a skilled professionalL...

7 червня 2022

Junior Flutter developer в **UBRAINIANS** до \$1500  Київ

Програмування на Flutter/Dart.

6 червня 2022

Senior Flutter Software Engineer (ID 81095) в  **SoftServe**  Львів, Івано-Франківськ, Ужгород

WE ARE SoftServe isn't just about technology – It's about open-minded people who love technology and have everything they need to pursue their passion to create innovation. We are about people who create bold things, who make a difference, who have fun, and who love their work.

Досвід

Без досвіду

< 1 року

1...3 роки

3...5 років

5+ років

Місто

Київ 11

Львів 5

Дніпро 4

Харків 2

Івано-Франківськ 2

Запоріжжя 1

Ужгород 1

Одеса 1

віддалена робота 19


 за кордоном 2

Рисунок 4.24 – Вакансії Flutter розробнику на dou.ua

Огляд ринку проводився у воєнний час в Україні, що дуже негативно впливає на ІТ сектор України, з оглядом на це, та на те, що у мирний час було, і буде набагато більше пропозицій, було нараховано 158 вакансій в українській спільноті.

ВИСНОВКИ

У ході роботи над кваліфікаційною роботою бакалавра було розроблено функціональний застосунок для iOS та Android пристроїв з сучасним дизайном. Досліджено, та відтворено процес розробки, розуміння, та виконання ТЗ. Проаналізовано сучасні рішення кросплатформеної розробки та їх відмінність. Було проаналізовано та досліджено сервіс Firebase, як він використовується і для яких цілей. Проведено практичну роботу зі створення моделі об'єктів. Розроблено систему реєстрації по мобільному телефону, що є невід'ємною частиною сучасних додатків.

У ході цієї роботи було розглянуто специфіку верстання елементів завдяки фреймворку Flutter, проаналізована архітектура Flutter та його відбудова на різних платформах завдяки архітектурі ARM.

Були застосовані базові знання з сортування масивів при створенні анімованого відтворення вхідних звуків у мікрофон.

Проведена робота з обробки вхідних даних з мікрофону та їх збереження у різних форматах, таких як .acc та .mp4.

У ході роботи з вхідним аудіо було досліджено subscription на події, їх обробка та використання, була отримана інформація про те, як зберігати записані аудіо та конвертувати їх. Проведено аналіз про різні форми збереження даних, таких як локальне збереження та збереження у хмарі, як відбувається хмарне сховище, та що з себе уявляє.

Проаналізовано рішення для керування стану додатку та застосовано GetX технологію, проаналізовано та реалізовано створення плеєра, відтворення записаних аудіо-файли, а також сортування їх по підбіркам.

Застосовано технологію хмарного сховища, що зберігає та відновлює інформацію у Real-Time з можливістю вводити зміни.

Найголовніше, що було виконано у кваліфікаційній роботі, це – повноцінний додаток, з повним циклом роботи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Builtin. Визначення репозиторію. URL: <https://builtin.com/software-engineering-perspectives/repository> (дата звернення 22.05.2023).
2. Визначення Firebase – AvadaMedia. URL: <https://avada-media.ua/ua/services/firebase/> (дата звернення 27.05.2023).
3. Пакети для проекту. URL: <https://pub.dev/> (дата звернення 27.05.2023).
4. Android Studio URL: <https://qagroup.com.ua/publications/android-studio-perevagy-ta-osoblyvosti/> (дата звернення 27.05.2023).
5. Визначення мови програмування Dart – Emetrius. URL: <https://emeritus.org/blog/coding-dart-programming-language/> (дата звернення 30.05.2023).
6. Вимоги для встановлення фреймворку Flutter. URL: <https://docs.flutter.dev/get-started/install/windows> (дата звернення 17.05.2023).
7. Сімоне Алессандрія. Flutter Cookbook: 100+ step-by-step recipes for building cross-platform, professional-grade apps with Flutter and Dart. 2021. 102 - 105 с.
8. Репозиторій з кодом даного проекту. URL: <https://github.com/Nikzed/MemoryBox> (дата звернення 14.06.2023).
9. Robert C. Martin. Clean Architecture A Craftman’s Guide to Software Structure and Design. 2017. 161 с.
10. Порівняння Native/Flutter. URL: <https://proandroiddev.com/why-flutter-will-change-mobile-development-for-the-best-c249f71fa63c>. (дата звернення 17.05.2023).
11. Michael Katz. Flutter Apprentice. 2022. Розділ 2 “Hello Flutter”. ISBN 978-1-95032-574-0.