Міністерство освіти і науки України

Національний університет «Одеська політехніка»

Інститут комп’ютерних систем

Кафедра інформаційних систем

ДИСЦИПЛІНА

Програмування мобільних пристроїв

КУРСОВА РОБОТА

Спеціальність:

122 Комп’ютерні науки

Освітня програма: Комп’ютерні науки

Виконав:

студент групи АІ-214  
Цветков Олександр Андрійович

Керівник:

Годовиченко Микола Анатолійович, кандидат технічних наук, доцент

Одеса – 2024

ЗМІСТ

Анотація 3

Вступ 4

1. [Огляд систем-аналогів та технологій їх розробки 4](#_bookmark0)
   1. [Особливості використання Android для розробки застосунків](#_bookmark1) 4
   2. [Аналіз конкурентів на ринку](#_bookmark2) 6
   3. [Формування вимог до основних функцій нашого Android застосунку](#_bookmark3) 9
   4. [Технології](#_bookmark4) 11
      1. Room 12
      2. Retrofit 13
2. [Проектування Android застосунку BuddyBook](#_bookmark6) 14
   1. [Мета та задачі Android застосунку](#_bookmark7) 14
   2. [Формування користувацьких історій Android застосунку](#_bookmark8) 15
   3. [Визначення нефункціональних вимог до Android застосунку](#_bookmark10) 16
3. Програмна реалізація Android застосунку BuddyBook 17
   1. [Інструкція користувача Android застосунку](#_bookmark11) 17

[ВИСНОВКИ](#_bookmark12) 34

[ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ……………………………………………………………… ……..34](#_bookmark13)

**Анотація**

Курсова робота присвячена розробці Android додатка «BuddyBook». Цей Android додаток - цифрова книга дружби, де ви можете зберігати всі свої нотатки, дні народження, адреси та багато іншого про своїх друзів. Записуйте, що подобається чи не подобається вашому другові, які у нього хобі або які ідеї щодо різдвяного подарунка ви маєте. Або дозвольте вашому другові заповнити форму самостійно - все залежить від вас. Під час розроблення будуть використані сучасні технології та практики розроблення Android додатків, зокрема мова програмування Kotlin, архітектура MVVM, взаємодія з базою даних через Room та мережеві запити через Retrofit.

**Вступ**

У сучасному світі, де мобільні додатки набувають все більшої популярності, зростає потреба у зручних та функціональних інструментах для організації особистої інформації. Одним із таких інструментів є додатки для зберігання інформації про друзів та контактів.

Застосунок «BuddyBook» являє собою застосунок для створення та керування нотатками на платформі Android. Мета проєкту - розробити зручний і функціональний застосунок для повсякденного використання, який дасть змогу користувачам мати під рукою всю необхідну інформацію про своїх друзів, що робить управління контактами значно зручнішим та ефективнішим.

**Аналітична частина**

На ринку існує безліч додатків для нотаток, як-от Google Keep, Microsoft OneNote і Evernote. Однак, більшість із них або складні у використанні, або вимагають підписки на додаткові функції. BuddyBook спрямовано на надання простого та інтуїтивно зрозумілого інтерфейсу без зайвих функцій.

**1 ОГЛЯД СИСТЕМ-АНАЛОГІВ ТА ТЕХНОЛОГІЙ ЇХ РОЗРОБКИ**

**1.1 Особливості використання Android для розробки застосунків**

Розробка застосунків для платформи Android має свої унікальні особливості та переваги, що робить її популярною серед розробників по всьому світу. У цьому розділі розглянемо ключові аспекти розробки для Android, включаючи інструменти, технології, архітектурні підходи, особливості середовища виконання та основні принципи розробки.

**Популярність та масштабованість Android**

* **Відкритий вихідний код**: Android є платформою з відкритим вихідним кодом, що дозволяє розробникам вільно використовувати та модифікувати його код. Це сприяє розвитку інновацій та швидкому впровадженню нових технологій.
* **Широкий ринок пристроїв**: Android встановлений на мільйонах пристроїв різних виробників, включаючи смартфони, планшети, смарт-годинники, телевізори та автомобільні системи. Це дозволяє розробникам досягати великої аудиторії.

**Інструменти для розробки**

* **Android Studio**: Офіційна середовище розробки (IDE) для Android, надає потужні інструменти для розробки, налагодження та тестування застосунків. Включає в себе редактор коду, емулятори пристроїв, інструменти для аналізу продуктивності та багато іншого.
* **Gradle**: Система збірки, яка автоматизує процеси компіляції, тестування та розгортання застосунків. Gradle дозволяє налаштовувати залежності та створювати складні сценарії збірки.

**Мова програмування**

* **Java**: Традиційно основна мова для розробки Android-застосунків. Завдяки широкому використанню та багатій екосистемі бібліотек, Java залишається популярною мовою для розробки.
* **Kotlin**: Нова офіційна мова для розробки Android-застосунків, яка пропонує більш сучасний і безпечний синтаксис. Kotlin сумісний з Java, що дозволяє поступово переходити на нову мову без необхідності переписування всього коду.

**Архітектурні підходи**

* **MVVM (Model-View-ViewModel)**: Поширена архітектура, яка сприяє розділенню логіки представлення, бізнес-логіки та даних. Використовується для забезпечення більшої керованості та тестованості коду.
* **MVP (Model-View-Presenter)**: Інший архітектурний підхід, що розділяє додаток на три компоненти: модель, представлення та презентер. Презентер діє як посередник між моделлю та представленням.

**Особливості середовища виконання**

* **Dalvik/ART**: Android використовує віртуальну машину Dalvik (DVM) або Android Runtime (ART) для виконання застосунків. ART є новішим та більш ефективним, забезпечуючи покращену продуктивність та швидше завантаження.
* **Розподілена архітектура**: Android використовує компоненти, такі як активності, сервіси, контент-провайдери та широкомовні приймачі для організації логіки застосунків. Це дозволяє краще організувати код та забезпечити повторне використання компонентів.

**Основні принципи розробки**

* **Реактивне програмування**: Використання бібліотек, таких як RxJava або LiveData, для створення асинхронних і подієвих застосунків. Це дозволяє більш ефективно обробляти потоки даних та події.
* **Матеріальний дизайн**: Набір рекомендацій щодо дизайну від Google, який забезпечує єдиний користувацький досвід на всіх платформах і пристроях. Використання компонентів матеріального дизайну робить інтерфейси більш інтуїтивними та привабливими.

**Тестування та забезпечення якості**

* **Unit-тести**: Перевірка окремих модулів коду на коректність виконання їх функцій. Для цього використовуються такі інструменти, як JUnit і Mockito.
* **Інструментальні тести**: Тестування застосунків на реальних або емуляторних пристроях. Використовуються для перевірки інтеграції різних компонентів та забезпечення їх коректної роботи в реальних умовах.

**Використання зовнішніх бібліотек та сервісів**

* **Retrofit**: Бібліотека для взаємодії з REST API, що спрощує процес виконання мережевих запитів та обробки відповідей.
* **Room**: Бібліотека для роботи з базами даних SQLite, яка забезпечує більш зручний інтерфейс та автоматичну перевірку запитів під час компіляції.
* **Firebase**: Набір хмарних сервісів від Google, включаючи аналітику, аутентифікацію, базу даних реального часу, хмарне зберігання та багато іншого.

Розробка застосунків для платформи Android є захоплюючим процесом, що вимагає знань у багатьох областях, включаючи програмування, проектування інтерфейсів, архітектурні підходи та тестування. Використання сучасних інструментів та бібліотек значно спрощує цей процес та дозволяє створювати високоякісні та продуктивні застосунки.

**1.2 Аналіз конкурентів на ринку**

**Застусунки зі схожим функціоналом: Google Keep**, **Microsoft OneNote** та **Evernote**. Кожен з них має свої унікальні функції та особливості, що робить їх корисними для різних типів користувачів. Я провів порівняльний аналіз, визначивши плюси та мінуси кожного з цих застосунків.

**Google Keep**

**Плюси:**

1. **Інтеграція з Google Services**:
   * Безшовна інтеграція з іншими сервісами Google, такими як Google Drive, Google Calendar та Google Docs.
   * Можливість створення нагадувань, які автоматично синхронізуються з календарем.
2. **Простота у використанні**:
   * Мінімалістичний дизайн та інтуїтивно зрозумілий інтерфейс.
   * Легко створювати та організовувати нотатки за допомогою кольорових міток і тегів.
3. **Кросплатформеність**:
   * Доступний на різних платформах, включаючи Android, iOS та веб-версію.
   * Автоматична синхронізація між усіма пристроями.

**Мінуси:**

1. **Обмежені функціональні можливості**:
   * Відсутність розширених функцій для організації та структурування нотаток, таких як ієрархічні структури або вбудовані таблиці.
2. **Обмежені можливості налаштування**:
   * Обмежені можливості налаштування інтерфейсу та функцій під потреби користувача.
3. **Обмеження на спільний доступ**:
   * Обмежені можливості для спільного редагування та управління доступом до нотаток.

**Microsoft OneNote**

**Плюси:**

1. **Розширені можливості організації**:
   * Підтримка ієрархічної структури організації нотаток (нотатники, розділи, сторінки).
   * Можливість створення складних нотаток з використанням таблиць, списків, тегів і малюнків.
2. **Інтеграція з Microsoft Office**:
   * Інтеграція з іншими продуктами Microsoft Office, такими як Word, Excel і PowerPoint.
   * Можливість вставки та редагування документів Office безпосередньо в нотатках.
3. **Кросплатформеність**:
   * Доступний на різних платформах, включаючи Windows, macOS, iOS, Android та веб-версію.
   * Автоматична синхронізація між усіма пристроями.

**Мінуси:**

1. **Складність інтерфейсу**:
   * Інтерфейс може здатися занадто складним для нових користувачів через велику кількість функцій і можливостей.
2. **Вимоги до ресурсів**:
   * OneNote може бути вимогливим до ресурсів пристрою, що може призводити до зниження продуктивності на старіших або менш потужних пристроях.
3. **Вартість**:
   * Деякі функції доступні лише при наявності підписки на Microsoft 365.

**Evernote**

**Плюси:**

1. **Розширені можливості організації та пошуку**:
   * Підтримка створення нотаток з використанням тегів, папок та підпапок.
   * Потужні інструменти пошуку, включаючи пошук за текстом всередині зображень і документів PDF.
2. **Інтеграція з багатьма сервісами**:
   * Підтримка інтеграцій з різними сторонніми сервісами та додатками, такими як Google Drive, Slack, Outlook та інші.
3. **Багатофункціональність**:
   * Можливість створення різних типів нотаток, включаючи текстові нотатки, аудіозаписи, веб-сторінки та сканування документів.

**Мінуси:**

1. **Вартість**:
   * Безкоштовна версія має суттєві обмеження, такі як обмежений обсяг пам'яті та кількість пристроїв для синхронізації.
   * Преміум-функції доступні лише при наявності платної підписки.
2. **Складність інтерфейсу**:
   * Інтерфейс може здатися перевантаженим для нових користувачів через велику кількість функцій і налаштувань.
3. **Вимоги до ресурсів**:
   * Evernote може бути вимогливим до ресурсів пристрою, що може призводити до зниження продуктивності на старіших або менш потужних пристроях.

Розглянуті застосунки мають свої унікальні переваги та недоліки.

**Google Keep** відрізняється простотою та інтеграцією з сервісами Google, але має обмежені можливості для організації нотаток.

**Microsoft OneNote** пропонує потужні інструменти для організації та інтеграції з іншими продуктами Microsoft, але може бути складним для нових користувачів і вимогливим до ресурсів.

**Evernote** надає розширені можливості організації та інтеграції з багатьма сервісами, але його безкоштовна версія має суттєві обмеження, а інтерфейс може здаватися перевантаженим.

Програма **BuddyBook** може виділитися на фоні цих застосунків завдяки своїй спеціалізації на управлінні інформацією про друзів та інтерактивними функціями, такими як прямий дзвінок, перегляд адреси в "Картах" та збереження днів народження в календарі. Це може забезпечити унікальну цінність для користувачів, які шукають саме таку функціональність.

**1.3 Формування вимог до основних функцій мого Android застосунку**

1. **Створення та управління профілями друзів**
   * Користувачі повинні мати можливість створювати профілі для своїх друзів, включаючи такі поля:
     + Фотографія профілю
     + Ім'я
     + День народження
     + Адреса
     + Номер телефону
     + Рід занять
     + Хоббі
     + Що йому подобається
     + Що він не любить
     + Особисті нотатки та коментарі
   * Профілі повинні бути збережені в локальній базі даних з використанням Room.
2. **Редагування та видалення профілів друзів**
   * Користувачі повинні мати можливість редагувати інформацію в профілі друга .
   * Користувачі повинні мати можливість видаляти профілі друзів з бази даних.
3. **Список друзів та пошук**
   * Застосунок повинен відображати список друзів з можливістю пошуку за ім'ям.
   * Список друзів повинен бути відсортований за алфавітом.
4. **Взаємодія з профілем друга**
   * Можливість зателефонувати другові, натиснувши біля його номеру телефона.
   * Можливість переглянути адресу друга в застосунку «Карти», натиснувши біля його адреси.
   * Можливість додати день народження друга до календаря, натиснувши на відповідну іконку.

**Нефункціональні вимоги**

1. **Інтерфейс користувача**
   * Застосунок повинен мати сучасний та інтуїтивно зрозумілий інтерфейс, розроблений згідно з принципами матеріального дизайну.
   * Забезпечення підтримки різних розмірів екранів та орієнтацій пристроїв.
2. **Продуктивність**
   * Час завантаження застосунку не повинен перевищувати 3 секунд на сучасних пристроях.
   * Операції створення, редагування та видалення профілів повинні виконуватися негайно.
3. **Сумісність**
   * Застосунок повинен підтримувати всі сучасні версії Android, починаючи з Android 6.0 (Marshmallow) і вище.

**Технічні вимоги**

1. **Мова програмування**
   * Застосунок повинен бути розроблений на мові програмування Kotlin.
2. **База даних**
   * Для зберігання даних використовується бібліотека Room, яка забезпечує роботу з локальною базою даних SQLite.
3. **Інтеграція з сторонніми сервісами**
   * Інтеграція з Google Maps API для відображення адрес друзів на карті.
   * Інтеграція з Google Calendar API для додавання днів народжень до календаря.

Застосунок **BuddyBook** повинен забезпечити зручний і ефективний спосіб управління інформацією про друзів, надаючи користувачам інструменти для створення, редагування, видалення та пошуку профілів. Використання сучасних технологій та бібліотек, таких як Room, Retrofit та Gson, забезпечить високу продуктивність і надійність застосунку.

**1.4 Технології**

Проєкт використовує такі технології та бібліотеки:

- Kotlin: Основна мова програмування.

- Android SDK: Набір інструментів для розробки Android-додатків.

- Gradle: Інструмент для автоматичного складання проєктів.

- AndroidX: Бібліотека, що надає підтримку сучасних компонент і API.

- ViewModel і LiveData: Компоненти архітектури MVVM для управління даними та їх представлення.

- Room: Бібліотека для роботи з базою даних SQLite.

- Retrofit: Бібліотека для роботи з мережевими запитами.

- Material Design: Набір рекомендацій і компонентів для створення сучасних інтерфейсів.

**1.4.1 Room**

Room - це бібліотека для роботи з базами даних SQLite в Android-додатках. Вона надає зручний і простий спосіб взаємодії з базою даних через високорівневий API. Ось деякі переваги та недоліки Room:

**Переваги:**

1. Зручний ORM (об'єктно-реляційне відображення): Room дозволяє вам взаємодіяти з базою даних, використовуючи об'єкти Java або Kotlin, замість написання складних SQL-запитів. Це полегшує розробку та підтримку коду.
2. Compile-time проверка: Room використовує анотації для визначення структури бази даних, що дозволяє виявляти помилки в час компіляції, такі як помилки в іменах стовпців або таблиць.
3. Підтримка LiveData та RxJava: Room інтегрується з архітектурним компонентом LiveData або з RxJava, що дозволяє створювати реактивні додатки, які автоматично оновлюються при зміні даних в базі даних.
4. Міграція бази даних: Room дозволяє вам здійснювати міграцію бази даних при зміні схеми бази даних, що дозволяє зберігати існуючі дані при оновленні додатка.

**Недоліки:**

1. Обмеженість функціональності: Room спрощує багато задач, пов'язаних з роботою з SQLite, але в деяких випадках вам все ще може знадобитися писати власні SQL-запити або використовувати додаткові бібліотеки.
2. Потребує вивчення: Хоча Room і спрощує взаємодію з базою даних, вам все ще доведеться вивчити документацію і освоїти його API, щоб ефективно використовувати його в своєму додатку.

Room є потужним і зручним інструментом для роботи з SQLite в Android-додатках, який забезпечує високий рівень абстракції і полегшує розробку, підтримку та оновлення баз даних.

**1.4.2 Retrofit**

Retrofit - це бібліотека для Android і Java, розроблена компанією Square, що спрощує взаємодію з HTTP-запитами. Вона надає простий і ефективний спосіб виконання веб-запитів, використовуючи анотації для опису запитів та підтримуючи конвертацію даних у різні формати, такі як JSON або XML.

**Переваги:**

1. Анотації: Використання анотацій для опису HTTP-запитів робить код зрозумілим і чистим.
2. Автоматична конвертація даних: Підтримує автоматичну конвертацію JSON, XML та інших форматів у Java-об'єкти і навпаки.
3. Підтримка різних типів запитів: Підтримує основні типи HTTP-запитів, такі як GET, POST, PUT, DELETE, PATCH та інші.
4. Можливість налаштування: Легко налаштовується для роботи з різними API, підтримує обробку різних форматів даних та дозволяє додавати кастомні конвертери та адаптери.
5. Обробка асинхронних запитів: Забезпечує просту реалізацію асинхронних HTTP-запитів, уникнення блокування основного потоку програми.
6. Інтеграція з OkHttp: Використовує OkHttp для здійснення HTTP-запитів, що забезпечує високу продуктивність та додаткові можливості, такі як кешування та аутентифікація.
7. Реактивне програмування: Інтегрується з RxJava, дозволяючи використовувати реактивні підходи для обробки запитів і відповідей.
8. Підтримка корутин: Також підтримує Kotlin Coroutines, спрощуючи написання асинхронного коду в Kotlin.

**Недоліки :**

1. Нові концепції: Вимагає часу для освоєння нових концепцій, таких як анотації та інтеграція з конвертерами.
2. Проблеми з великими об'ємами даних: Може бути повільним або вимагати додаткових налаштувань для оптимізації продуктивності при обробці великих об'ємів даних.
3. Залежність від інших бібліотек: Покладається на інші бібліотеки, що збільшує кількість залежностей у проекті.
4. Стандартна обробка помилок: Стандартні механізми обробки помилок можуть бути недостатніми для складних сценаріїв, що вимагають додаткової роботи для налаштування.

Retrofit є потужним і гнучким інструментом для взаємодії з веб-сервісами в Android та Java, який значно спрощує роботу з HTTP-запитами завдяки використанню анотацій та автоматичної конвертації даних. Проте, як і будь-яка інша бібліотека, Retrofit має свої недоліки, включаючи необхідність навчання та потенційні проблеми з продуктивністю при обробці великих об'ємів даних.

**2 Проектування Android застосунку BuddyBook**

**2.1 Мета та задачі Android застосунку**

Мета застосунку **"BuddyBook"** полягає в створенні зручного та функціонального інструменту для зберігання та управління інформацією про друзів. Головною метою є забезпечення користувачам простого та ефективного способу організації нотаток про друзів, включаючи їхні дні народження, адреси, номери телефонів, хобі та інші персональні дані. Це дозволить користувачам завжди мати під рукою інформацію про своїх друзів та легко керувати цією інформацією з мобільного пристрою.

**Задачі застосунку " BuddyBook ":**

**1)Створення та управління профілями друзів**

Забезпечити можливість створення профілів для друзів з такими полями як ім'я, фотографія профілю, день народження, адреса, номер телефону, рід занять та особисті нотатки.

Реалізувати функції редагування та видалення профілів друзів.

**2)Відображення списку друзів та пошук**

Реалізувати відображення списку друзів з можливістю сортування за алфавітом.

Забезпечити функцію пошуку друзів за ім'ям для швидкого доступу до потрібного профілю.

**3)Взаємодія з профілем друга**

Реалізувати можливість зателефонувати другові безпосередньо з профілю, натиснувши на кнопку біля номеру телефона.

Забезпечити функцію перегляду адреси друга в застосунку «Карти».

Реалізувати можливість додавання днів народжень друзів до календаря.

**4)Інтеграція з Google Maps та Google Calendar API**

Використовувати Google Maps API для відображення адрес друзів на карті.

Інтегрувати Google Calendar API для додавання днів народжень до календаря користувача.

Застосунок **BuddyBook** має стати надійним інструментом для управління інформацією про друзів, забезпечуючи зручність, простоту використання та високу продуктивність. Використання сучасних технологій та інтеграція з популярними сервісами, такими як Google Maps та Google Calendar, дозволить користувачам легко взаємодіяти з інформацією про своїх друзів та зберігати важливі дані в одному місці.

**2.2Формування користувацьких історій Android застосунку**

Як користувач, я хочу мати можливість створювати профілі своїх друзів, щоб зберігати всю важливу інформацію про них в одному місці.

Як користувач, я хочу мати можливість додавати фотографії до профілів друзів, щоб легше ідентифікувати їх.

Як користувач, я хочу мати можливість редагувати та видаляти профілі друзів, щоб завжди мати актуальну інформацію.

Як користувач, я хочу мати можливість шукати друзів за ім'ям у списку контактів, щоб швидко знаходити потрібну інформацію.

Як користувач, я хочу мати можливість телефонувати другу безпосередньо з його профілю, щоб швидко зв'язатися з ним.

Як користувач, я хочу мати можливість переглядати адресу друга на карті, щоб знати, де він живе.

Як користувач, я хочу мати можливість додавати дні народження друзів до календаря, щоб не забувати про важливі дати.

Як користувач, я хочу мати інтуїтивний та зрозумілий інтерфейс для управління профілями друзів, щоб легко користуватися застосунком без додаткового

**2.3Визначення нефункціональних вимог до Android застосунку**

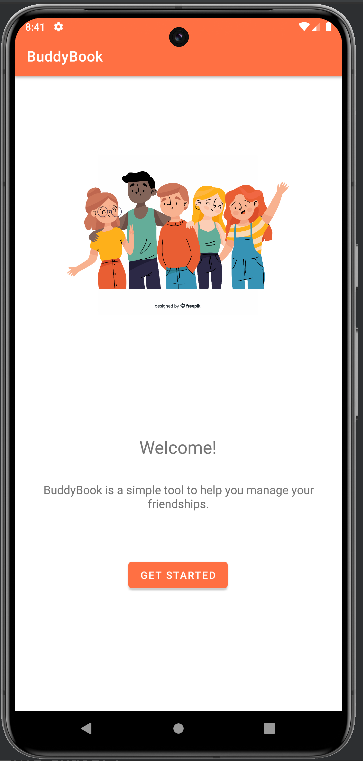
Після визначення функціональних вимог, необхідно сформувати нефункціональні вимоги, які висуваються до Android застосунку.

NFR1 Пристрій – будь-який, але з не зовсім старим Android.

**3 Програмна Реалізація Android застосунку**

**3.1** Інструкція користувача Android застосунку  
Для забезпечення успішного користувацького досвіду використання розробленого Android застосунку необхідно скласти інструкцію користувача, яка включає в себе знімки екранів та пояснювальний текст до кожного знімка. Це допоможе користувачам легко зорієнтуватись у функціоналі Android застосунку та ефективно використовувати його.

1) Перше, що ми бачимо під час переходу в Android застосунку - це його початковий екран . Ми можемо побачити тут вітальне вікно застосунку «BuddyBook».



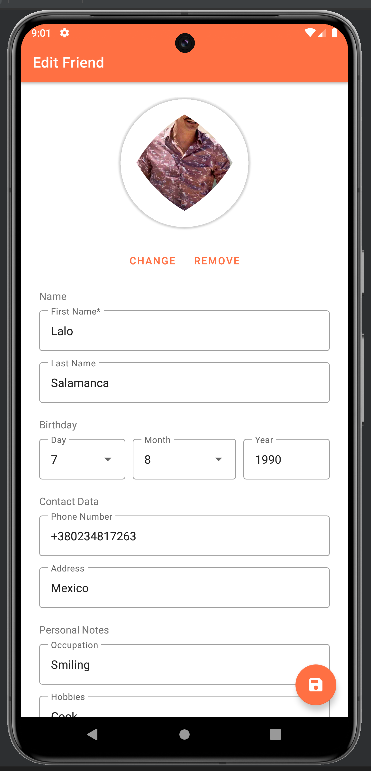
2)Нажимаємо на Кнопку “Get Started” і переходимо до наступного екрану.

Бачимо, пустий список і знизу помаранжеву круглу кнопку “+”.

3)Якщо користувач хоче додати першого друга, наскаємо на кнопку “+”.



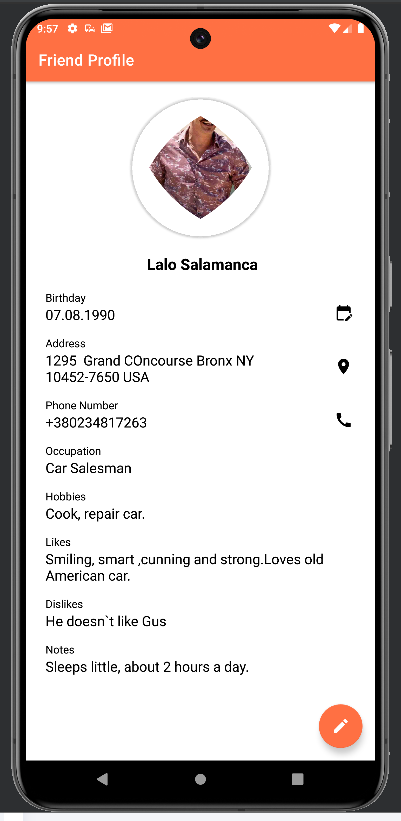
Далі заповняємо профіль друга.



Вводимо ім’я, дату народження, ставимо його фото, місто де він живе,де він працює, хоббі, що він любить, що він не любить і іншу інформацію.

Коли закінчите вводи інформацію, натискніть знизу на помаранжеву кнопку збереженя.

Відкриваэться нове вікно, з всією указаною вами інформацією.



4)Якщо користувач десь допустили помилку, або якась інформація змінилась (переїзд в інший дім, захоче змінити фото друга), він в любий комфортний час може відкрити профіль друга, натискнути на помаранжеву кнопку з олівцем і змінити інформацію.

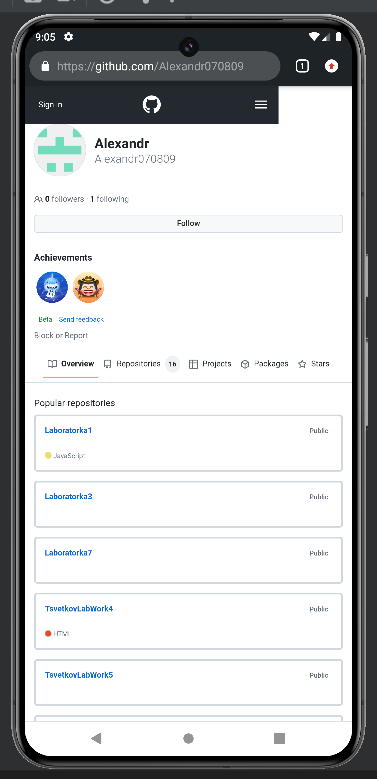
5)Якщо користувач хоче подивитися інформацію про застосунок, хто його створив, переходимо на головний екран. Знаходячись на головному екрані застосунку, натискаємо зверху на 3 точки. Потім відктивається маленьке вікно “About”



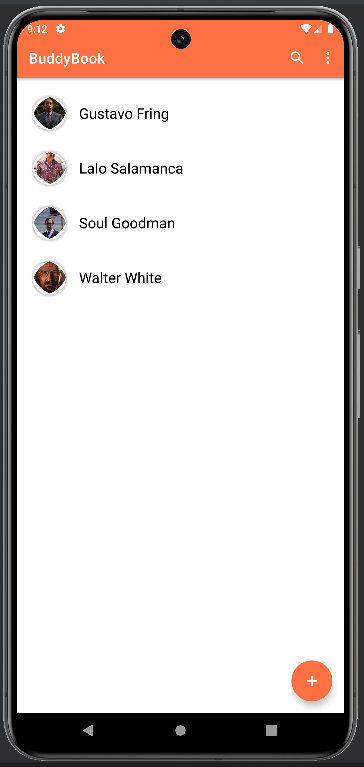
Натискаємо на нього. Відкривається новий екран з версією застосунку, розробником і ліцензією.



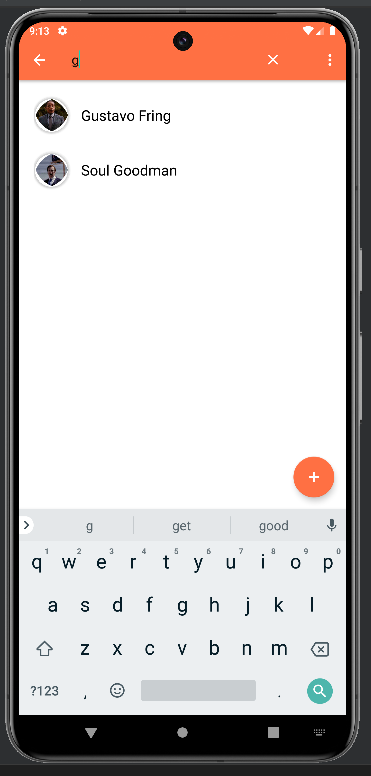
Натискаємо на кнопку “Develooer” і автоматично відкривається браузер з моїм GitHub профілем.



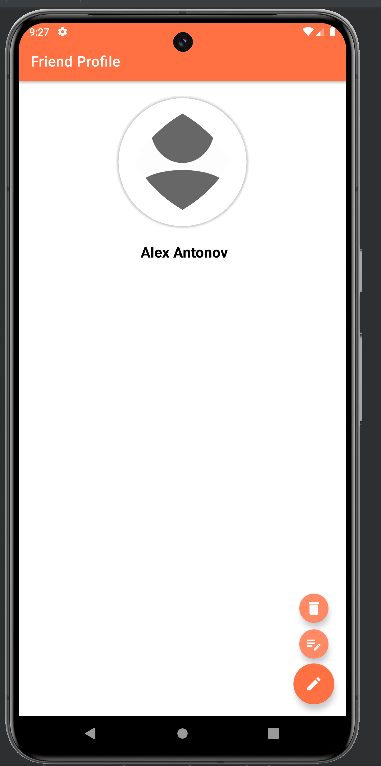
6)Створивши ще декілька профілей друзів, можемо побачити, що їх імена розташовані в алфавітному порядку.



7)Якщо користувачу потрібно знайти друга по імені, вверху екрану можете побачити лупу. Натискаємо її і вводимо перші літери , але любі літери імені друга.

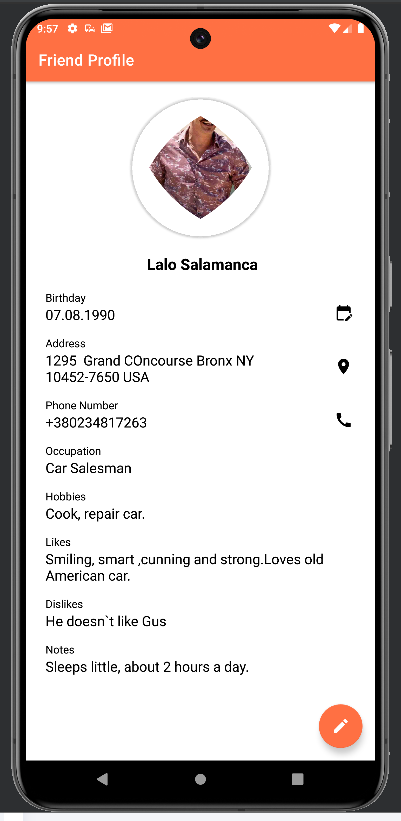


8)Якщо користувачу потрібо відалити профіль друга. Натискаємо на профіль друга, натискаємо на помаранжеву кнопку з олівцем і обераєму урночку.

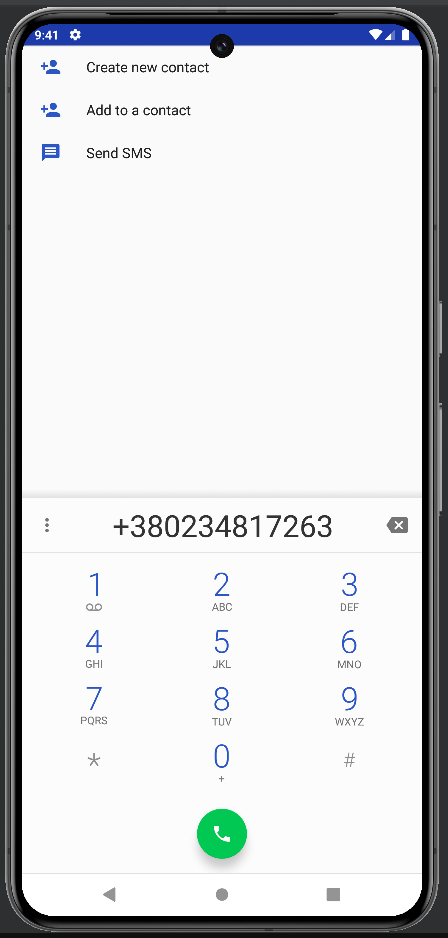


Застосунок перепитує: ”Точно ви хочете видалити профіль друга?” Якщо ми дійсно хочимо видалити, натискаємо на кнопку “Delete” , або якщо ми передумали, натискаємо “Cancel”

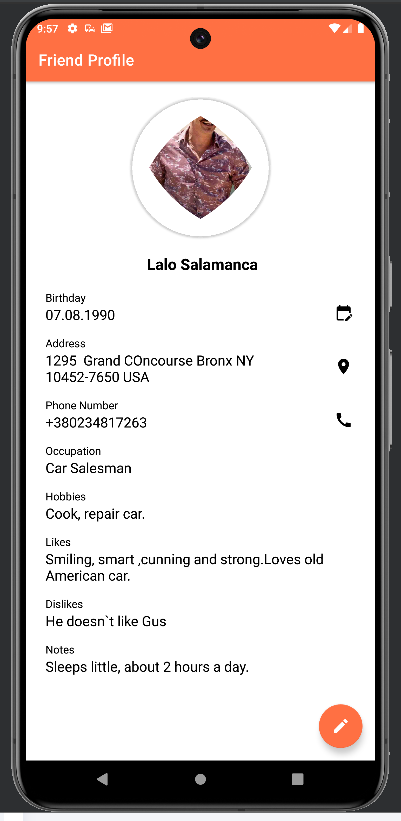
9)Якщо користувачу потрібно подзвонити другу, заходимо в профіль друга, і біля його номеру телефона є трубочка.



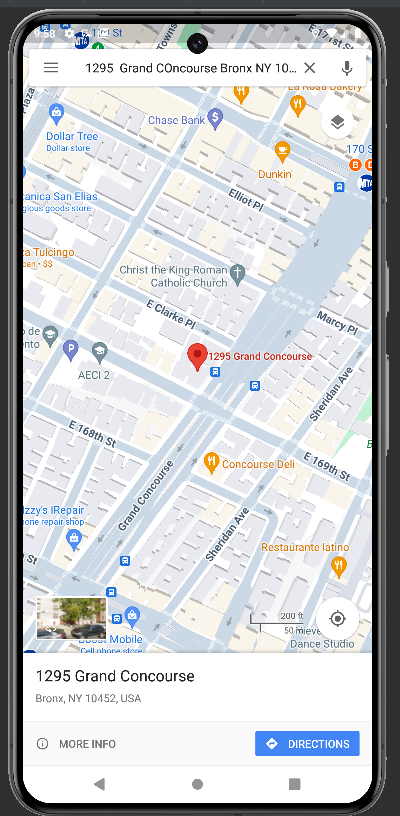
Натискаємо на неї і автоматично відкривається нове вікно з вже набраним номером телефону.



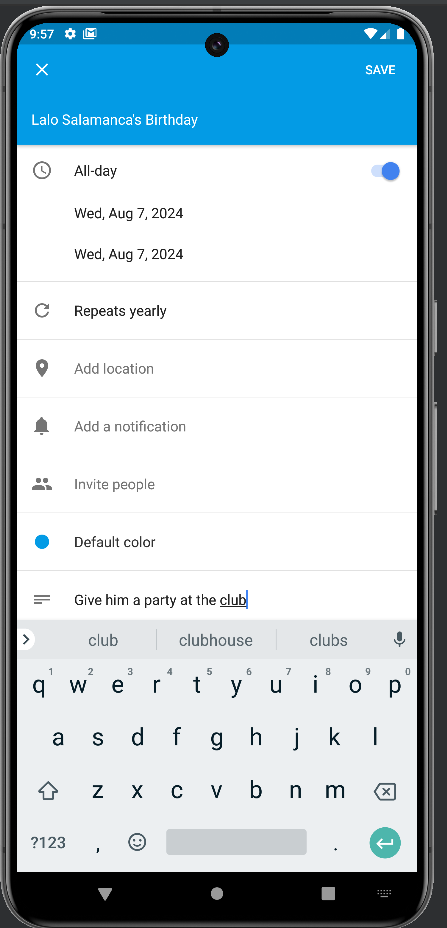
10)Якщо користувач хоче подивитися де живе його друг на карті, заходимо в профіль друга, і біля його адреси бачимо мітку.



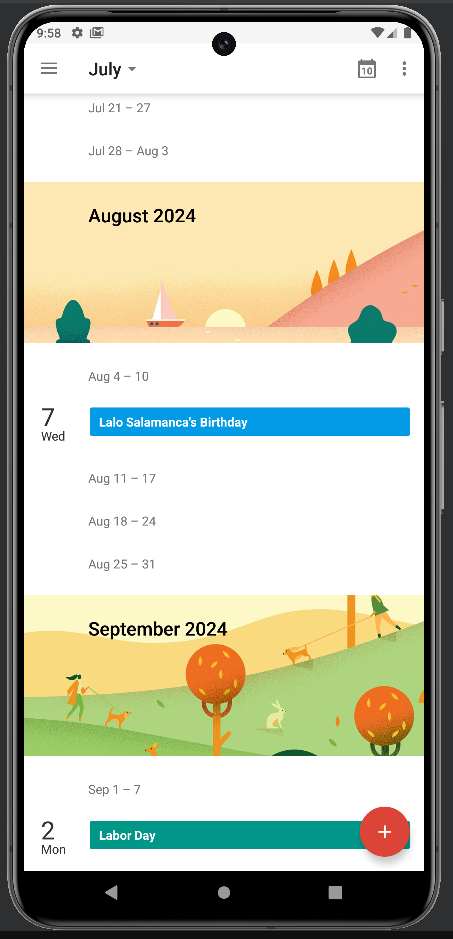
Натискаємо на неї і автоматично відкривається нове вікно з Google Картами і міткою з домом друга.



11)Якщо користувачу потрібно відмітити день народження друга у Google Календарі, заходимо в профіль друга, і біля його дати народження бачимо календарик. Натискаємо на нього і автоматично відкривається нове вікно з додаванням Пам’ятки.



Користувач може додати Сповіщення, вказати місце проведення свята,список людей, яких можна запросити на це свято, та додати коментарі які касаються свята. І далі бачимо, що в Google календарі з’явилася помітка про день народження друга.



**Висновок**

Під час роботи було розроблено Android-додаток «BuddyBook», який дає змогу користувачам керувати інформацією про своїх друзів. Додаток пройшов тестування і відповідає заявленим вимогам. У майбутньому планується додати функцію синхронізації даних із хмарою для забезпечення доступу з різних пристроїв.

**Список літератури**

1. Официальная документация Android: <https://developer.android.com/docs>
2. Kotlin для Android разработчиков: https://kotlinlang.org/docs/android-overview.html
3. Руководство по Material Design: https://material.io/design
4. Официальная документация Room: <https://developer.android.com/training/data-storage/room>
5. Официальная документация Retrofit: https://square.github.io/retrofit