

УДК 004.03

## СТВОРЕННЯ ТУРИСТИЧНИХ МОБІЛЬНИХ ДОДАТКІВ ЗА ДОПОМОГОЮ ПЛАТФОРМИ FIREBASE

Лисоконь Дмитро

Науковий керівник: канд. фіз.-мат. наук Гуртовий Ю.В.

*Центральноукраїнський державний педагогічний університет імені  
Володимира Винниченка, м. Кропивницький, Україна*

*Стаття присвячена загальним принципам створення туристичних мобільних Android-додатків на мові Java за допомогою хмарних технологій. Проаналізовані основні можливості платформи Firebase, зокрема, особливості роботи з базою даних. Показано можливість створення туристичного мобільного додатку з універсальними функціями.*

*Ключові слова: Android, хмарні технології, платформа Firebase.*

## CREATION OF MOBILE TRAVEL ADDITIONS BASED ON THE FIREBASE PLATFORM

Lisokon Dmitro

Scientific supervisor: Yu.V. Hurtovyi PhD

*Volodymyr Vynnychenko Central Ukrainian State Pedagogical University,  
Kropyvnytskyi, Ukraine*

*The article is about the general principles of creating mobile travel Android applications in Java using cloud technologies. The basic features of the Firebase platform are analyzed, in particular, features of work with the database. Possibility of creation of tourist mobile application with universal functions is shown.*

*Keywords: Android, cloud technology, Firebase platform..*

**Постановка проблеми.** Активний туризм – найбільш масовий вид туризму. Його основна особливість в тому, що він доступний практично кожному. В середньому більшість людей подорожує 2-3 рази на рік, подорож може бути як і короткою – всього пару діб, так і довгою – більше тижня. При підготовці до поїздки, люди дуже багато часу витрачають на її планування, що є проблемою, оскільки час для сучасної людини є дуже важливим ресурсом.

Подорожуючи з тих чи інших причин, люди часто хочуть відвідати цікаві місця – це можуть бути кінотеатри, парки, кафе, музеї і т.д., причому все частіше

цікавлять саме ті локації, які є не дуже відомими. Дуже зручно знаходити інформацію такого роду в звичайному смартфоні, тому туристичні мобільні додатки є досить бажаними на ринку.

### **Аналіз попередніх розробок**

Веб-сервіс Google Maps, хоча і не позиціонує себе як додаток для подорожей, але їхні карти ймовірно найчастіше використовуються для знаходження різних туристичних локацій. Для пошуку якогось місця доведеться перейти на їхні карти і витратити багато часу для того щоб знайти серед безлічі об'єктів бажане місце, без можливості нормальної фільтрації. Важливим недоліком є слабкий зв'язок з іншими користувачами.

У сегменті мобільних додатків для туристів на базі операційної системи Android вже існують рішення зі своїми перевагами і недоліками. Аналіз літератури виявляє те, як повинен працювати і виглядати цей додаток, щоб бути максимально зручним для користувача [2,3]. Найбільша кількість додатків розроблено для великих та відомих міст, наприклад, для Києва чи Львова. Часто воно надають в різному вигляді одну і ту ж саму інформацію, і мають наступні недоліки:

- малий діапазон охоплених місць;
- відсутність актуальної інформації про локації;
- базу з інформацією наповнює мала кількість людей;
- відсутність можливості спілкування з іншими користувачами.

**Метою статті** є дослідження можливостей створення мобільного додатку для мобільної платформи Android, що використовує хмарні технології та є результатом спільних дій користувачів програми.

### **Характеристика мобільних Android-додатків та платформи Firebase**

Кожна програма для мобільної операційної системи - це пакет з розширенням файлу .apk. Даний пакет формується утилітою aapt і складається з усього компільованої Java-коду[1] і ресурсів, які використовуються в процесі роботи програми. Саме такий пакет користувач завантажує та встановлює з усім відомого сервісу Google Play.

Кожному з додатків в ОС Android присвоюється свій окремий процес. Сам процес створюється, коли це необхідно, а по завершенню виконання необхідних дій операційна система автоматично завершує процес, щоб звільнити оперативну пам'ять для виконання інших процесів. Кожна програма, крім усього іншого, має унікальний Linux User ID і виконується в своїй віртуальній машині. Природно, що кожен додаток має власні права, які присвоєні даному User ID. За замовчуванням програма має доступ тільки до власних ресурсів, але можливі шляхи експорту ресурсів іншим додаткам. Кілька додатків також можуть мати один User ID - в такому випадку вони будуть мати і загальні ресурси [3-5].

Для досягнення поставленої мети було вирішено використати широкі можливості платформи Firebase. Платформа Firebase для розробки мобільних та веб-додатків. розвивається з 2011 року компанією Firebase Inc., яка була придбана Google у 2014.

Firebase має такі характеристики:

1. Firebase є хмарною базою даних, яка працює в режимі реального часу. Дані зберігаються як JSON і синхронізуються в реальному часу з кожним підключеним клієнтом. Коли користувач створює кросплатформені додатки на базі iOS, Android і JavaScript SDK, всі клієнти отримують ідентичний екземпляр актуальної бази даних і автоматично отримують оновлення.
2. Робота в автономному режимі (Offline) - додатки Firebase працюють також в автономному режимі, оскільки SDK Firebase Realtime Database кешує поточні дані на пристрої. Після відновлення підключення до мережі Інтернет клієнтський пристрій отримує пропущені зміни, синхронізуючи їх з поточним станом сервера.
3. Доступ з клієнтських пристроїв (Accessible from Client Devices) - доступ до Firebase Realtime Database можна отримати безпосередньо з мобільного пристрою або веб-браузера. Тому немає необхідності в написанні сервера для додатків. Безпека і перевірка даних доступні через

Firebase Realtime Database Security Rules, гнучка мова правил на основі виразів, які виконуються при читанні або запису даних.

4. **Firebase Storage** - хмарне сховище, створене для розробників додатків, яким необхідно зберігати і обслуговувати завантажуються користувачами будь-якого бачені, наприклад, зображення або відеофайли. **Cloud Storage for Firebase** - це масштабна, легко освоювана служба збереження медіа даних. Що включає в безкоштовному тарифі до 10 гб. вільного простору в хмарі. Програміст може використовувати SDK для читання і збереження аудіофайлів, зображень, відеофайлів або будь-якого іншого формату дані, які можуть знадобитися для відображення контенту. Завантажені файли, можна переглянути в **Firebase Console**.
5. **Firebase Analytics** - це новий безкоштовний інструмент для аналізу мобільних додатків. Частково він успадкував дещо від **Google Analytics**. Він надає достовірну інформацію про активність користувачів всередині програми, що дозволяє реалізувати нову функцію - Аудиторії. Аудиторії дозволяють виявити різні групи користувачів зі схожим властивостями. Згодом можна буде отримувати роздільну інформацію для різних аудиторій усередині єдиного середовища **Firebase**.
6. **Автентифікація Firebase**. Багато додатків повинні ідентифікувати користувача. Ідентифікація дозволяє додатку безпечно зберігати і структурувати призначені для користувача дані в хмарі, забезпечуючи синхронізацію на різних операційних системах пристроїв користувача. Також ідентифікація дозволяє відображати релевантні дані для користувача. **Firebase Authentication** надає базові служби, прості в використанні SDK і готові бібліотеки призначеного для користувача інтерфейсу для аутентифікації користувачів у вашому додатку. **Firebase Authentication** підтримує аутентифікацію з використанням:
  - пароля і електронної пошти;
  - телефонного номера;
  - соціальної мережі google; facebook; twitter; gitHub.

Також Firebase підтримує анонімний вхід. Firebase Authentication безпосередньо інтегрується з іншими сервісами Firebase, використовуючи галузеві стандарти, такі як OAuth 2.0 і OpenID Connect, що дозволяє легко інтегрувати з будь-яким сервером.

Таким чином, є широкі можливості платформи Firebase, які дозволяють створити реалізувати проект туристичного мобільного додатку, який надає :

- можливість отримання інформації про цікаві місця з підготовленої бази даних;
- можливість створення інформації про місця користувачами системи;
- систему авторизації;
- реалізацію інтерактивної карти з зазначеним на ній місцем;
- реалізацію можливості налаштувань виведення місць;
- отримання та надання інформації про погоду для місць;
- реалізацію додавання відгуків про місце, користувачами системи;
- надання інформації про місцезнаходження та можливість зв'язатись з іншими користувачами додатку;
- можливість зберігати відвідані чи вподобані місця
- систему винагородження.

### **Початок роботи з платформою Firebase**

Для початку в консолі розробника firebase, що знаходиться за посиланням <https://console.firebase.google.com/u/1/?pli=1> потрібно створити новий проект. Для цього потрібно казати ім'я проекту, для прикладу дамо назву “example-kspu”, та погодитися з умовами використання firebase.

Після створення потрібно перейти в налаштування проекту та натиснути на вкладку «сервісні акаунти» та згенерувати закритий ключ, який потрібно помістити в корінь нашого проекту. Щоб можна було використовувати всі можливості firebase, потрібно підключити в maven підтримку цієї бібліотеки кодом нижче.

```
<dependencies>  
  <dependency>
```

```
        <groupId>com.google.firebase</groupId>
        <artifactId>firebase-admin</artifactId>
        <version>6.8.0</version>
    </dependency>
</dependencies>
```

Далі потрібно додати **Realtime Database** в консолі розробника **Firebase**.

```
try {
    FileInputStream serviceAccount =
FileInputStream("serviceAccountKey.json");
    FirebaseOptions options = new FirebaseOptions.Builder()

.setCredentials(GoogleCredentials.fromStream(serviceAccount))
        .setDatabaseUrl("https://example-
kspu.firebaseio.com/")
        .build();
```

Для зчитування даних з файлу призначений клас `FileInputStream`, який є спадкоємцем класу `InputStream` і тому реалізує всі його методи. Для створення об'єкта `FileInputStream` ми можемо використовувати конструктор, який приймає вхідним параметром шлях до файлу. Тому ми вказуємо шлях до файлу з ключом.

В налаштуванні для підключення до бази вказуємо ключ та шлях до бази, який можна знайти в консолі розробника на вкладці `Database`. Після ініціалізації налаштувань, можна використовувати базу через `DatabaseReference`. Функція `child()` приймає в себе значення ключа, якому будуть присвоєні якісь значення, але якщо ключа не існує, то `firebase` його створить.

Щоб зчитати данні з бази використовується одна з трьох функцій: `addValueEventListener`, `addListenerForSingleValueEvent`, `addChildEventListener`.

Ці функції приймає вхідним параметром інтерфейс, який реалізовує два методи: `onDataChange` та `onCancelled`. Перший метод заповниться об'єктами із бази даних в `dataSnapshot`, який ми можемо використовувати. Отримавши всі дочірні об'єкти з `dataSnapshot`, за допомогою циклу ми можемо вивести їх

ключі та значення. Якщо дані більше не потрібні, їх можна видалити функцією `removeValueAsync()`.

## Робота з базою даних **Firestore**

Будь які операції з базою даних відбуваються через створені класи `FirestoreTraveler` та `FirestoreUsers`. Основні дії в роботі з базою даних є:

- зчитування країн, регіонів та їх населених пунктів. При створенні об'єкту класу `FirestoreTraveler` конструктор бере посилання на `firebase`, що представляє собою певне місце в базі даних і може бути використане для читання або запису до цього місця. `DatabaseReference` – клас, який є відповідною точкою для всіх операцій з базою даних. Після ініціалізації його з URL-адресою ви можете використовувати його для читання даних, запису даних та створення нових `DatabaseReferences`. Щоб отримати доступ для цих операцій використовується такий код: 

```
FirestoreDatabase database = FirestoreDatabase.getInstance();  
DatabaseReference referenceCountries = database.getReference("countries");
```

 За його допомогою з'явився доступ до даних країн. Кожна країна вміщає в себе регіони, вони в свою чергу населені пункти в яких і знаходяться створені локації. Тому маючи доступ до бази країн, можна перейти далі по дереву.
- створення нового місця відбувається по шляху вищевказаного дерева. За допомогою функції `push()` створюються новий випадковий, неповторюваний ключ для нової локації. В ньому генеруються такі значення, як: ім'я, опис, тип, координати, нульова популярність, посилання на фото, час створення місця та автора.
- створення нового місця, коли генерується новий тип локації, якщо такого ще не існує. Він вноситься в іншу гілку бази даних. Це необхідно для фільтрації місць при пошуку користувачем.
- пошук людей поблизу відбувається також через базу даних. Всі користувачі, які увімкнули видимість в налаштуваннях, створюють новий потік, який обновляє координати в базі даних. Коли хтось користується

пошуком людей поблизу, то наступна функція витягує всіх дані активних користувачів, та порівнює їх координати та стать, якщо вона була вибрана, щоб знайти тих, які знаходяться на такій максимальній відстані, що вибрав користувач.

### Практичний результат дослідження

В результаті проведеної роботи було розроблено Android-додаток (рис.1).

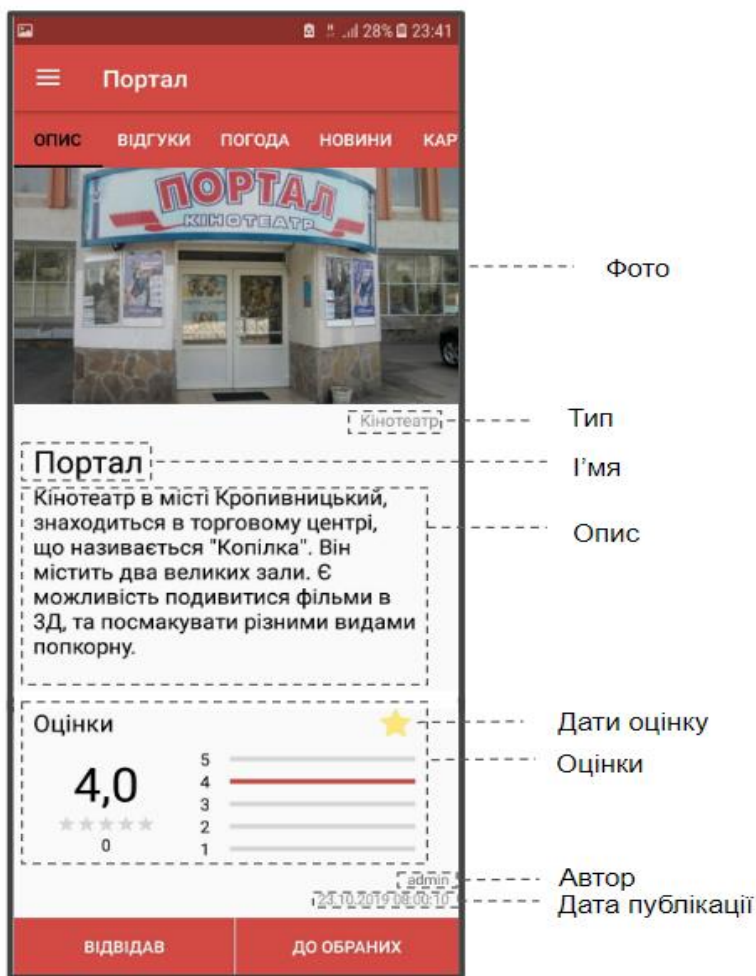


Рисунок 1. Інтерфейс додатку «Traveler»

Створений продукт надає інформацію про локації, погоду та дає можливість створювати відгуки. Кожен користувач має можливість створювати нові локації, зв'язуватись з іншими користувачами, які знаходяться поруч, отримувати винагородження за дії в додатку.



## **Висновки та перспективи подальших пошуків у напрямі дослідження**

Отже, дослідження можливостей використання платформи Firebase значно спрощує роботу і дозволяє розробникам зосередитися на розробці клієнтської частини програми, що позначається на якості програмного забезпечення. Важливою перевагою є те, що розробнику не потрібно керувати серверами і створювати код для API. Крім того Firebase – надає розробнику сховище для даних, причому є можливість модифікації відповідно до більшості потреб. Це дало змогу зберігати всі дані системи (опис місць, фото, відгуки, повідомлення і т.д.), створювати облікові записи користувачів, для авторизації і обліку. Крім того, в рамках даного дослідження були вивчені інструменти для інтеграції карт в мобільний додаток.

Підсумком став невеликий і зручний додаток, який підійде всім, хто любить подорожувати чи відвідувати цікаві місця, та знаходити там нові знайомства. При цьому є можливість додавання нового функціоналу для додатку, наприклад: створення нових типів винагород за активність користувачів; додавання особистих сторінок користувачів; оптимізація завантаження картинок; покращення графічного інтерфейсу; створення редактора завантажених фото

## **Список використаної літератури**

1. Эккель Б. Философия Java. 4-е полное изд. – СПб.: Питер, 2018 – 1168 с.: ил. – (Серия «Классика computer science»)
2. Efficient Android Threading: Asynchronous Processing Techniques for Android Applications. By Anders Goransson. O'Reilly Media; 1 edition June 13, 2014; 280 pages;
3. Android Programming: The Big Nerd Ranch Guide. By Bill Phillips, Chris Stewart & Kristin Marsicano. Big Nerd Ranch Guides; 3 edition February 9, 2017; 624 pages;
4. Mastering Android Development with Kotlin: Deep dive into the world of Android to create robust applications with Kotlin By Milos Vasic. Packt Publishing — ebooks Account, November 8, 2017; 378 pages;
5. Дэвид Гриффитс, Дон Гриффитс «Head First. Программирование для Android» Питер, 2018 год, 912 стр, O'Reilly, 2-е изд.,
6. Java 8. Полное руководство. 9-е издание 89480К, 1377 с.

7. Android Studio [електронний ресурс] // Режим доступу:  
[https://en.wikipedia.org/wiki/Android\\_Studio](https://en.wikipedia.org/wiki/Android_Studio)
8. Gartner [електронний ресурс] // Режим доступу:  
<https://www.gartner.com/en/newsroom/press-releases/2018-02-22-gartner-says-worldwide-sales-of-smartphones-recorded-first-ever-decline-during-the-fourth-quarter-of-2017>
9. Android [електронний ресурс]  
// Режим доступу: <https://uk.wikipedia.org/wiki/Android>
10. Firebase Cookbook: Over 70 recipes to help you create real-time web and mobile applications with Firebase (English Edition).
11. Поняття про додатки для Android і їх компоненти [електронний ресурс] // Режим доступу: <https://fans-android.com/ponyatiya-o-prilozheniyax-dlya-android-i-ix-komponenty/>
12. Firebase [електронний ресурс] // Режим доступу: <https://uk.wikipedia.org/wiki/Firebase>