

УДК 004.4

РОЗРОБКА СТРУКТУРИ МОДЕЛІ ДАНИХ НАВЧАЛЬНОГО МОБІЛЬНОГО ДОДАТКУ НА ОПЕРАЦІЙНІЙ СИСТЕМІ ANDROID ВІДПОВІДНО ДО КРИТЕРІЇВ ЯКОСТІ GOOGLE

Дюков Євген

Науковий керівник: канд. фіз.-мат. наук Пузікова А.В.

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

Стаття присвячена питанням забезпечення виконання загальних критеріїв компанії Google до додатків при розробці навчальних додатків на операційній системі Android. Проаналізовано відповідність цим критеріям поширених навчальних додатків для навчання програмування на операційній системі Android, реалізованих на цій же системі. Реалізовано проектування моделей бази даних у Firebase RealTime Database та їх оптимізацію для продуктивної та швидкої роботи власного додатку. Проведено тестування швидкості виконання запитів для створеного мобільного додатку, яке підтвердило ефективність виконаної оптимізації коду.

Ключові слова: навчальний додаток, Android, Firebase RealTime Database, JSON.

ANDROID TRAINING APPLICATION DATA MODEL STRUCTURE DEVELOPMENT ACCORDING TO GOOGLE QUALITY CRITERIA

Diukov Yevhen

Scientific adviser: Candidate of Physical and Mathematical Sciences

Puzikova A.V.

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

The article is devoted to ensuring compliance with Google's general criteria for applications when developing training applications on the Android operating system. The conformity of the widespread educational applications for learning programming on the Android operating system, implemented on the same system, to these criteria is analyzed. The design of database models in Firebase RealTime Database and their optimization for productive and fast operation of own application is realized. Testing of the speed of execution of requests for the created mobile application was carried out, which confirmed the effectiveness of the performed code optimization.

Keywords: training application, Android, Firebase RealTime Database, JSON.

Постановка проблеми. Велика розповсюдженість мобільних пристроїв на операційній системі (ОС) Android зумовила великий попит на

розробників відповідної платформи. В той же час сама ОС Android постійно розвивається: випускаються нові версії, додається новий функціонал, з'являються нові технології, але через велику кількість тем та інформації у процесі навчання основам розробки можуть виникати багато труднощів. Для систематизації та узагальнення знань існують спеціальні навчальні мобільні додатки, які допомагають вивчити особливості розробки на цій платформі та перевірити знання на практиці за допомогою тестів або вправ.

Користувачі ОС Android очікують високої якості від встановлених додатків. Якість додатків безпосередньо впливає на затребуваність додатку: кількість завантажень, рейтинг та відгуки користувачів, популярність і тривалість використання. У зв'язку з цим компанією Google було прописано і затверджено основні критерії якості додатків [1], яких мають дотримуватися розробники для створення якісного продукту.

Метою статті є аналіз відповідності додатків для навчання програмування на операційній системі Android критеріям компанії Google до мобільних додатків, а також реалізація правильної організації структури моделі даних при проектуванні власного додатку з метою забезпечення відповідності цим критеріям.

Аналіз готових додатків для навчання програмування на операційній системі Android. За вибілковими критеріями з документу [1] та з точки зору користувача був проведений аналіз 6 популярних додатків: Learn Android App Development [3], Android Tutorials [4], Learn Android Studio [5], Learn AStudio [6], Android Tutorial [7] та Smart Android Course [8]. Результати аналізу наведені у таблиці 1.

Як можна бачити, жоден з розглянутих додатків не задовольняє усім наведеним в табл. 1 вимогам. Таким чином, розробка мобільного додатку з елементами інтерактивності для навчання програмування на операційній системі Android, який би відповідав зазначеним вище вимогам, надавав актуальну інформацію та не містив реклами і платного контенту, є актуальною задачею.

Таблиця 1.

ID критерію та його значення	Learn Android App Development	Android Tutorials	Learn Android Studio	Learn AStudio	Android Tutorial	Smart Android Course
PS-V1: додаток відображає графіку, тексти, зображення і різні елементи інтерфейсу без помітних спотворень, змазувань або ефектів пікселізації.	-	-	-	-	-	-
PS-V2: тексти і текстові блоки відображаються зручними для читання	-	-	-	-	-	-
PS-S1: не відбувається аварійного або вимушеного закриття додатку, зависань або інших аномалій в його роботі на будь-яких підтримуваних пристроях	+	-	+	+	-	-
PS-P1: додаток завантажується швидко, а також пропонує користувачеві відповідну інформацію на екрані (індикатор виконання і т. п.), якщо для завантаження потрібно більше двох секунд	+	-	-	+	+	-
UX-B1: додатки відповідають рекомендаціям по дизайну Android і використовують загальноприйняті значки та шаблони інтерфейсу користувача.	+	+	-	+	+	+
UX-N1: у додатку підтримується стандартна системна навігація за допомогою кнопки "Назад", для якої не використовуються будь-які адаптовані екранні підказки	+	+	+	-	+	+
FN-P1: додаток повинен вимагати лише абсолютний мінімум прав доступу, які необхідні	+	-	+	+	+	-

йому для підтримки основних функціональних можливостей.						
FN-P2: додаток не запитує прав доступу до найбільш важливих даних (наприклад, до контактів або системного журналу), а також до платних сервісів (наприклад, до служб дозвону або відправки СМС), якщо це не пов'язано з основною функціональністю цього додатку.	+	-	+	+	+	-
PS-M1: у процесі завантаження і нормальної роботи з додатком звукові та відеодані відтворюються плавно, без клацань, затримок і інших артефактів	+	+	+	+	+	-

Виклад основного матеріалу дослідження. Початковим етапом розробки будь-якого додатку є проектування бази даних. Одним з основних аспектів високої продуктивності додатків є правильна структура бази даних. У якості синхронізації інформації з додатком було використано Firebase Realtime Database, тому інформація зберігається у форматі JSON і саме правильна організація та оформлення JSON сприятиме швидкій та оптимізованій роботі додатку [2].

Базова структура бази даних складається з моделей User, Lesson та Task. Модель User містить основну інформацію про користувача, його рівень та прогрес. Lesson містить дані про урок та список завдань. Task містить самі завдання та теорію до них. Таким чином, User матиме свою модель Lesson та Task, у якому буде міститися прогрес та інші дані. При великій кількості інформації така модель може бути занадто великою і складною. Нижче наведено приклад базової структури моделі JSON.

```
{
  "users": [
    {
      "id": 1,
      "email": "dyuksev@gmail.com",
```

```

    "username": "dyuks",
    "level": 1,
    "points": 0,
    "currentLessonId": 0,
    "lessons": [
      {
        "description": "Your first lesson! Its will help you learn the basics
of Android development",
        "done": false,
        "id": 1,
        "name": "Основи",
        "points": 65,
        "tasks": [
          {
            "id": 1,
            "theoryId": 1,
            "isPrimary": true,
            "type": 1,
            "data": "some data"
          },
          ...
        ]
      },
      ...
    ]
  },
  ...
]
}

```

Як можна бачити, наведена вище модель не є оптимізованою, тому що кількість вкладень може бути великою, і при завантаженні даних також будуть завантажуватися усі дочірні вузли моделі. Тому на практиці краще зберігати структуру даних як можна більш плоскою, що матиме вплив на продуктивність роботи додатку. Оптимізація коду полягає в розділенні структури, що покращує процес вибірки, але також призводить до надлишковості інформації. Але саме ця надлишковість дозволяє швидко та ефективно отримувати інформацію під час роботи додатку, навіть тоді, коли кількість даних у базі даних налічуватиметься мільйонами.

У такому випадку об'єкти Lesson та Task будуть окремими об'єктами та не будуть залежати від конкретного екземпляру User. У свою чергу моделі

User, Lesson та Task будуть мати зв'язок між собою у вигляді масиву ідентифікаторів. Тобто у такому випадку JSON-файл буде містити окремо об'єкти User, Lesson і Task. Код оптимізованої моделі JSON наведено нижче:

```
{
  "users": {
    "S4wWCpLgVYSRN5RbcECWgK70Y322": {
      "email": "dyuksev@gmail.com",
      "level": 1,
      "points": 0,
      "currentLessonId": 0,
      "doneLessonsId": [
        1,
        ...
      ],
      "username": "dyuks"
    },
    ...
  },
  "lessons": [
    {
      "id": 1,
      "name": "Основи",
      "description": "Your first lesson! Its will help you learn the basics of
Android development",
      "points": 65,
      "theory": "some theory",
      "taskId": [
        1,
        ...
      ]
    },
    ...
  ],
  "tasks": [
    {
      "id": 1,
      "theoryId": 1,
      "isPrimary": true,
      "type": 1,
      "data": "some data"
    },
    ...
  ]
}
```

Таким чином нова модель дозволяє завантажувати Task окремо і тільки для конкретного Lesson, довантажувати їх за необхідності майже миттєво, а також швидко відображати їх у інтерфейсі.

Для дослідження швидкості виконання запитів було розроблено 5 тестів з різною кількістю моделей Lesson та Task і за допомогою вбудованого тестового середовища Android Studio підраховано час виконання запитів у мілісекундах від моменту, коли виконується запит на сервер, до моменту, коли дані завантажилися у локальних моделях на пристрої.

Результати виконання обчислення наведені у таблиці 2. Остання колонка показує у скільки разів відбувається пришвидшення виконання запитів в оптимізованій моделі.

Таблиця 2.

№	Стандартна модель, мс	Оптимізована модель, мс	Співвідношення швидкостей виконання запитів стандартної моделі до оптимізованої
1	2265	1675	1,35
2	2650	1895	1,4
3	3059	2168	1,41
4	3904	2772	1,41
5	4175	3048	1,37
Середнє значення			1,39

Тобто, використання оптимізованої моделі прискорює отримання даних у середньому в 1.39 рази, що позитивно впливає на продуктивність додатку. Також така модель дозволяє завантажувати усі дані не одночасно, а окремими викликами, тільки тоді, коли це необхідно, що значно прискорює роботу додатку в цілому.

Висновки. Наведені вище результати тестування дозволяють зробити висновок, що проведення оптимізації коду моделей JSON сприяє підвищенню продуктивності власного додатку, що, в свою чергу, є кроком до

забезпечення відповідності розробленого навчального додатку основним критеріям якості додатків, що висуваються компанією Google.

Список літератури

1. Core app quality [Електронний ресурс]. – Режим доступу: <https://developer.android.com/docs/quality-guidelines/core-app-quality>.
2. Structure Your Database [Електронний ресурс]. – Режим доступу: <https://firebase.google.com/docs/database/android/structure-data>.
3. Додаток Learn Android App Development [Електронний ресурс]. – Режим доступу: <https://play.google.com/store/apps/details?id=com.hussainlabs.learnandroidapp>.
4. Додаток Tutorial for Android: Quiz and Examples [Електронний ресурс]. – Режим доступу:
<https://play.google.com/store/apps/details?id=arjuntoshniwal.androidtutorials.advanced>.
5. Додаток Learn Android Studio [Електронний ресурс]. – Режим доступу: <https://play.google.com/store/apps/details?id=com.bhalla.gaurav.androidstudiotipstricks>.
6. Додаток Learn AStudio [Електронний ресурс]. – Режим доступу: <https://play.google.com/store/apps/details?id=com.devam03.learnandroidcoding>.
7. Додаток Android Tutorial: Complete App Development [Електронний ресурс]. – Режим доступу: <https://play.google.com/store/apps/details?id=com.helloworld.androidtutorial>.
8. Додаток Smart Android Course [Електронний ресурс]. – Режим доступу: <https://play.google.com/store/apps/details?id=colorsfx.smart.android.courses>.