Collabits Journal
Vol. 2 No. 2 May 2025 : 52-60
E-ISSN : 1979-5254, P-ISSN : 3062-8601
https://publikasi.mercubuana.ac.id/index.php/collabits

Countryinfo App Development to Facilitate Access to Android-Based Country Information

Thanel Richard Angwarmasse[1*], Mariano Antonino Mingga Juluk[2], Muchamad Rachel Rafliola[3], Misni[4]

[1,2,3,4] Informatics Engineering, Universitas Mercu Buana, Indonesia

*Coressponden Author: misni.muhamad@mercubuana.ac.id

Abstract - In the era of globalization and technological advancement, the need for fast, accurate, and easily accessible information has significantly increased. The CountryInfo App was developed as an innovative solution to simplify access to real-time country information. This application offers two main features: searching for country details by name and a randomizer feature for exploring country data interactively. Utilizing technologies such as Kotlin for user interface development, Flask for backend management, and public APIs for retrieving real-time data in JSON format, the application is designed to provide users with an intuitive and interactive experience. Testing results demonstrate a responsive performance, with an average access time of less than 2 seconds. This study aims to provide practical benefits to society in education, research, and personal needs while serving as a case study for API technology implementation in mobile application development.

Keywords:
Country Information;
Mobile Application;
API;
JSON;
Kotlin;
Flask;

Article History:
Received: 13-03-2025
Revised: 16-04-2025
Accepted: 22-05-2025

Article DOI : 10.22441/collabits.2025.v2i2.31258

1. INTRODUCTION

In the current era of globalization and the development of digital technology, access to information has become a very important need for society. One of the most sought-after Information is data about countries in the world. This information plays a significant role in various fields, such as education, tourism, international business, and academic research. For example, a student may need information about a particular country to complete a school assignment, or a tourist may want to know the details of the country they are going on vacation to, including the currency, official language, and geographic location. On the other hand, international business people often need economic or social data of a country to support strategic decision-making.

However, the challenge that is often faced is the difficulty of finding a reliable and integrated source of information. Most users have to access various different platforms to get the data they need, which is not only time consuming but also risks getting inconsistent information. With the high need for fast, accurate, and easily accessible information, an innovative solution is needed to overcome this problem.

Technological advances in mobile-based application development provide great opportunities to answer these needs. One effective approach is integration with the Application Programming Interface (API), which allows applications to retrieve data in real time from trusted sources. APIs also allow for the presentation of data that is always up-to-date, thus increasing the accuracy and relevance of the information provided to users.

The Country Info App is designed as an answer to this problem. This application offers users the convenience of accessing country information through two main features: a search feature by country name and a randomizer feature that allows for random exploration of new knowledge. With an intuitive interface and reliable API integration, this application not only provides detailed information such as population, official languages, and currencies, but also becomes an efficient educational and reference tool for users from various backgrounds.

Through the development of this application, it is hoped that the public can more easily access state data practically, both for personal, educational, and professional needs. In addition, this application is also an example of the implementation of API technology in the development of mobile-based applications, providing benefits both technically and practically.

2. LITERATURE REVIEW

In this study, the author will implement the Search Algorithm in an application system that will be created to search for the most relevant and frequently searched country information. This application will allow users to search for complete information about the country they want.

Some of the information that will be displayed includes:

- Country name
- Country code
- The continent of the country
- Capital
- Population
- Time zone
- Flag image
- Currency details (Code, Name, Symbol)

By using an effective search algorithm, this application can provide accurate and relevant results according to user searches. This application will help users in finding country information quickly and efficiently, as well as providing a better experience in accessing country-related data that is often searched for or used as a reference.

In developing a country information search application using Android Studio, JSON, API, and Android Studio play an important role to ensure the application runs efficiently and provides a good user experience.

1. **JSON**is a data format used to transfer information between applications and servers. In a country search application, JSON is used to store and transmit country data such as name, capital, population, currency, language, and more. The lightweight and easy-to-process JSON format makes it ideal for exchanging data between Android applications and servers [1].
2. **Android Studio**as the main development environment allows you to create user interfaces (UI) and connect applications to servers using APIs. Users can enter country names, and the application will display relevant data in a responsive and interactive manner [6].
3. **API** serves as a liaison between the application and the server. The application will send a request to the API to get data in JSON format. The API provides data that is then processed by the application to display information about the country searched by the user [2].

2.1 JSON

Data Storage and Retrieval: JSON can store structured data, such as country information, in a format that is easy to parse and create. This makes it ideal for storing data such as country names, populations, capitals, languages, etc. JSON is commonly used as a response format. This means that our application can easily request data about countries from a web service and parse the JSON response to display relevant information [6].

2.2 Android Studio

Android Studio is the core tool for Android app development, offering a full suite of features for designing, developing, and testing apps. In the context of a country search app, Android Studio allows you to build a responsive app, fetch data from external APIs using Retrofit or Volley, and display the country information searched by the user in an easy-to-understand format [6].

2.3 API

The API is used to connect the application to a server that provides country data. The application will send a request to the API and receive a response in the form of data in JSON format which is then processed and displayed in the application UI [8].

3. METHODOLOGY

3.1 Software Development Methods

This Country Info application is developed using Android Studio as a software development platform, because it provides a structured and systematic approach. By using this model, each stage of application development is carried out sequentially and measurably. This application is designed to provide complete information about countries, either by searching for country names directly or by selecting countries randomly. The main goal of this application is to provide an easy-to-use and efficient user experience, allowing users to obtain information quickly and practically.

3.2 Analysis Stage

In the analysis phase of Country Info application development, the main focus is to identify the needs and objectives of the application, as well as analyzing the various components that will support its functionality. Here are some steps taken in the analysis phase:

3.2.1 Running System Analysis

At this stage We analyze the conditions or systems that are running before the application development begins. In this case, the Country Info application aims to replace the manual process of accessing country information that previously might have required manual searches in various sources or websites. From the analysis of the running system, we found several problems such as:

- **Inefficient data search process**: Users have to search for country information one by one manually, which can be time consuming.
- **Limitations in accessibility**: If the required information is only available on one device or in a particular format (e.g. a large document or file), users may have difficulty accessing it.
- **Limited data processing**: Without a classification or filter system, users cannot get information quickly and accurately.

3.2.2 Interview Results Analysis

In this stage, we conduct interviews with potential users or

stakeholders to find out their problems and needs related to the application to be developed. From the interview results, some of the problems that emerged were:

- Time to search for information: Users want a faster way to efficiently obtain country data, for example with a search feature or random selection.
- **No easy accessibility**: Users want to be able to access this application on multiple devices, especially with a user-friendly interface.
- **Limitations in data management**: Currently, there is no application or system that allows users to easily search and view country-related information.

3.2.3 Functional Requirements Analysis

Functional requirements refer to the features or capabilities that an application must have to meet user needs. Here are some functional requirements for the Country Info application:

- **User Admin**:
  a. Enter country data manually or set up access to external data sources (e.g. APIs).
  b. Manage the country search feature by name or other categories.
  c. Presenting country information in an easy-to-read format.
- **General User/User**:
  a. Search for country by name.
  b. View detailed country information such as population, language, flag, and geographic location.
  c. Randomly select a country to find information quickly.

3.2.4 Non-Functional Requirements Analysis

Non-functional requirements focus on the technical specifications for the system, both in terms of hardware and software needed to support the application.

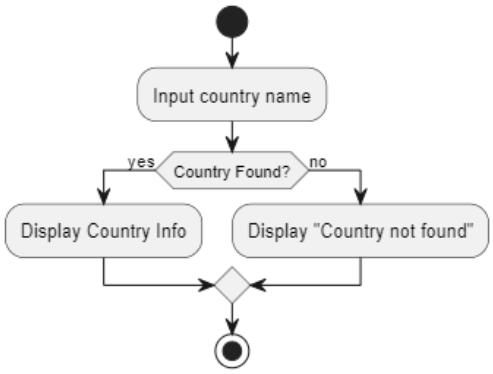
- **Hardware**:
  - Ensure that the application can run smoothly on Android devices with minimum specifications, for example:
    - Qualcomm Snapdragon processor or equivalent.
    - Minimum 2 GB RAM to ensure good performance.
    - 16 GB or more internal storage for installing apps and storing user data.
- **Software**:
  - **Android Studio**as the main development platform.
  - **Kotlin**or Java for programming languages.
  - **Retrofit**for communication with API
  - **Material Design**to ensure the application has a responsive and attractive user interface.

3.3 Experimental Phase

1. Determining the Theme
   The first step is to choose a relevant and interesting theme for the application to be developed. The Country Info theme was chosen because it aims to provide complete and accurate information about countries in the world, as well as provide convenience for users to access the information efficiently.
2. Identification of problems
   At this stage We identified existing problems in presenting country information that is still done manually, such as time-consuming data searches, lack of data accessibility across devices, and difficulty in managing information efficiently. The Country Info application is designed to address these issues by providing a faster and easier way to access country information through search or random selection.
3. Collecting Data
   At this stage, the data required for the application will be collected. We use external APIs such as RESTful APIs that provide country-related data, such as data on population, language, geographic location, and country flags. Collecting this data is important to ensure that the application presents accurate and up-to-date information.
4. Literature Study
   Literature study is conducted to understand the concepts relevant to application development, including the concept of using APIs to retrieve data, how to present information in an easy-to-read format, and the implementation of responsive and intuitive interface design. References from journals, articles, and tutorials can be used to gain insight into the best algorithms and methodologies that can be applied in application development.
5. System Design
   After gathering the necessary information, the design phase is carried out to describe the flow and structure of the application. We create UML diagrams (Use Case Diagram, Activity Diagram, Class Diagram) to design the application system. This design also includes creating a user interface (UI) prototype using Figma to ensure the application is easy to use and provides a good user experience.
6. Application Design
   At this stage, the system design that has been created is implemented into a real application using Android Studio. The use of Kotlin and Java programming languages is chosen to develop Android-based applications, while for communication with the API, we use Retrofit. This process involves coding application functionality, such as country search, random country selection, and data retrieval and presentation from the API.
7. Testing
   At this stage, we perform application testing to ensure that the application runs according to expectations.

3.4 Design Stage

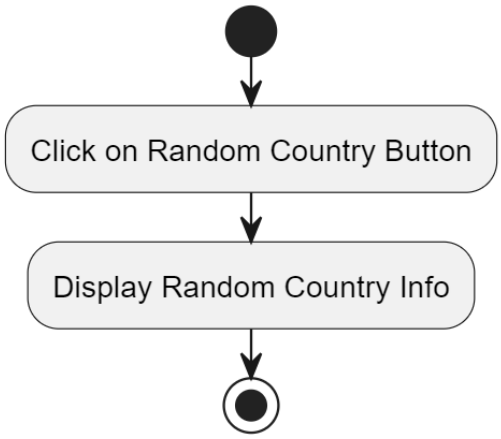
3.4.1 Flowchart For Country Search



The explanation:

- The user enters the name of the country he wants to search for.
- The system checks whether the country is found.
- If the country is found, the country information will be displayed.
- If the country is not found, a message will appear that the country is not found.

3.4.2 Flowchart to display Random Countries

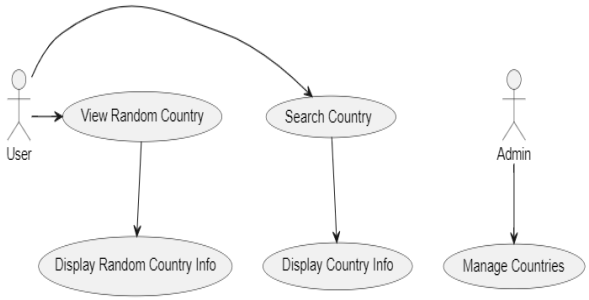


The explanation:

- The user clicks a button to display a random country.
- The system displays random country information without requiring input from the user.

3.4.3 Use Case Diagram

Shows the interaction between actors (User and Admin) with the system. There are two main features for users: "Search Country" and "View Random Country". Admin can manage country data.



Explanation of the Use Case Diagram above as follows:

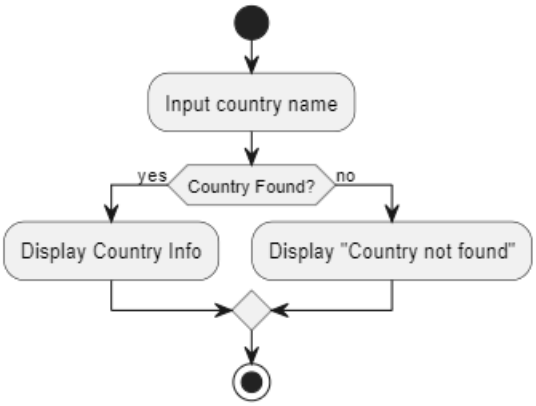
- Search Country: Users can search for countries by name.
- **View Random Country**: Users can view countries randomly.

3.4.4 Activity Diagram

Activity Diagram describes the flow or sequence of activities in a system to achieve a certain goal. This diagram focuses on how the application workflow takes place.

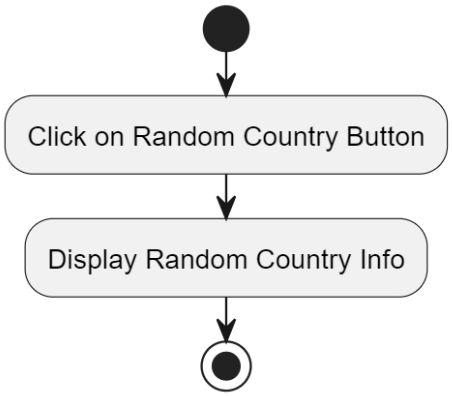
- Activity for Country search

  The explanation:
  - Enter the country name.
  - Check whether the country is found or not.
  - Displays country information or a message that the country was not found.



- Activity for Random Country

  The explanation:
  - Click the Random Country button
  - Country name will be displayed



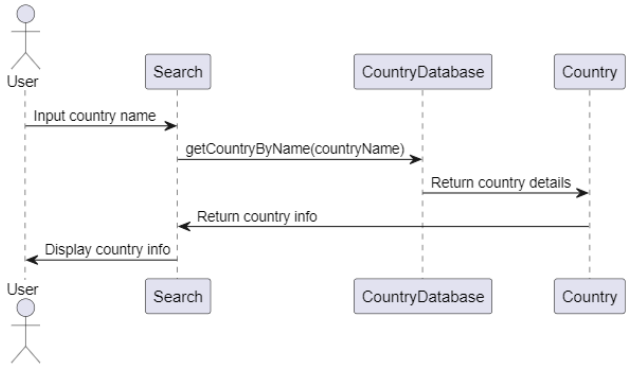
3.4.5 Sequence Diagram

Sequence Diagrams describe the interactions between objects (classes or entities) in a system based on time sequence. This diagram shows how objects communicate with each other to achieve a certain goal.

- Sequence for Country search

The explanation:

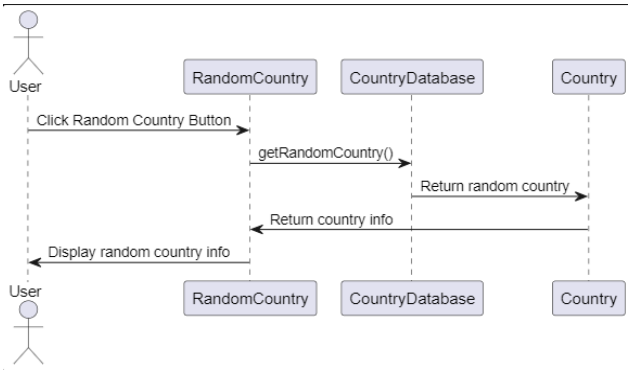
User enters country name → System (Search) requests country data from CountryDatabase → CountryDatabase returns country data



- Sequence for Random Countries

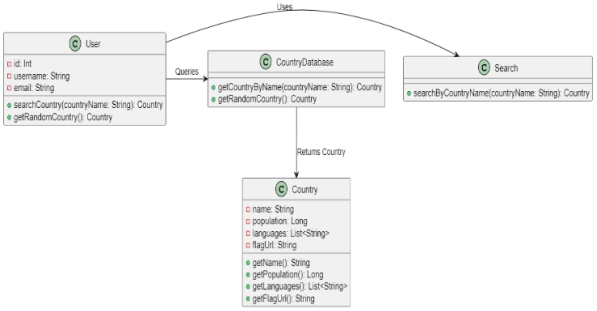
  The explanation:

  - User clicks Random Country → System (Search) requests country data from CountryDatabase → CountryDatabase returns country data.



3.4.1 Class Diagram

Class Diagram describes the static structure of a system by showing the classes in the application, attributes, methods, and relationships between classes.



4. RESULTS AND DISCUSSION

The CountryInfo App application system is designed as a mobile-based application that functions to provide practical and interactive country information. The development process of this application is divided into two main parts, namely Backend and Frontend. On the backend side, this application uses an API that is deployed on the PythonAnywhere platform, with the Python programming language and the Flask framework as a server-side logic processor. Meanwhile, the frontend or user interface is developed using Android Studio with the Kotlin programming language. The application interface is designed to be responsive and intuitive by utilizing design elements from Jetpack Compose. The data used in the application is taken in real-time from an external API that provides detailed information about the country.

In this writing, several important parts of the application source code will be included, including the source code for the main features such as the country search page, detailed country information display, and the randomizer feature that displays country data randomly.

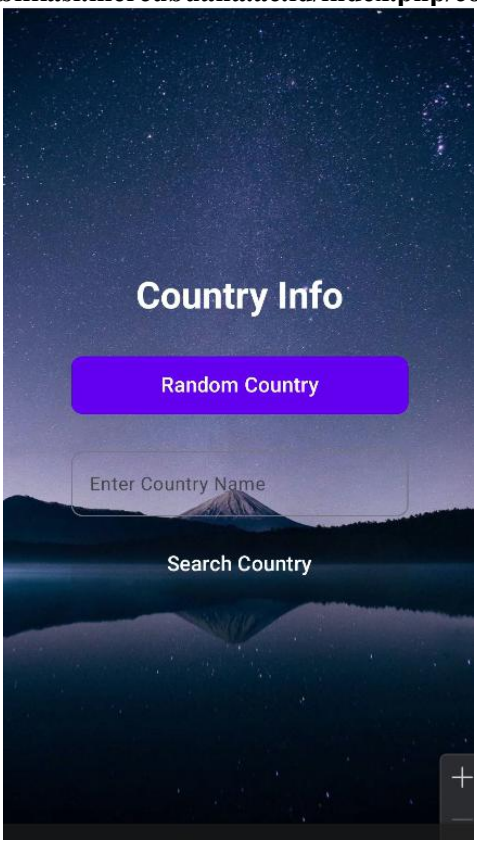


Figure 1. Country info

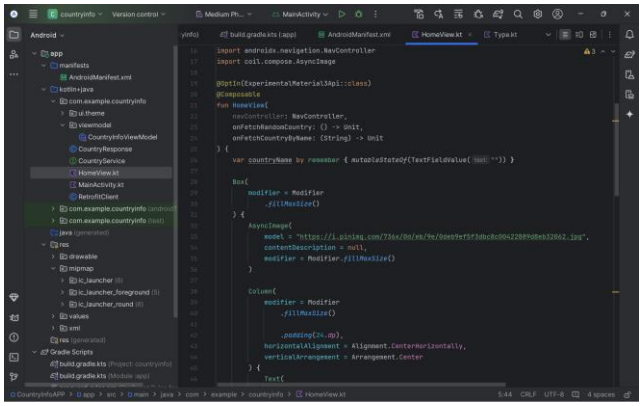


Figure 1. Source code

This picture is a screenshot of the HomeView frontend section, which includes the UI/UX logic and styling as well as the user input process.

The frontend of this application was developed using Android Studio with the Kotlin programming language. This technology was chosen because it supports the development of native applications for Android with optimal performance and good integration with device features.

The user interface (UI) is designed using Jetpack Compose, a modern framework for building declarative and responsive UIs. Jetpack Compose enables developers to create dynamic, scalable, and efficient designs that fit the needs of the application.

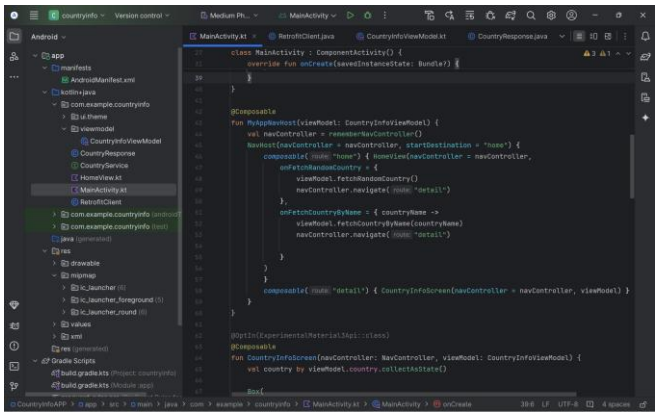


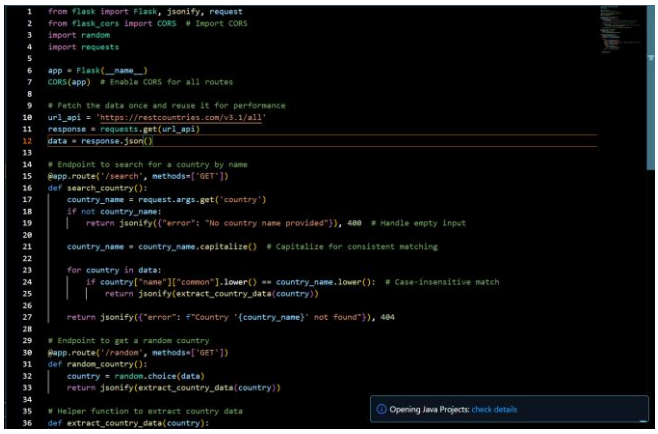
Figure 3. Source code

This picture is the frontend part of MainActivity where the main logic and settings around the appearance of the application run.

4.1 Source Code

4.1.1 Frontend

4.1.2 Backend Process



```python
from flask import Flask, jsonify, request
from flask_cors import CORS  # Import CORS
import random
import requests

app = Flask(__name__)
CORS(app)  # Enable CORS for all routes

# Fetch the data once and reuse it for performance
url_api = 'https://restcountries.com/v3.1/all'
response = requests.get(url_api)
data = response.json()

# Endpoint to search for a country by name
@app.route('/search', methods=['GET'])
def search_country():
    country_name = request.args.get('country')
    if not country_name:
        return jsonify({"error": "No country name provided"}), 400  # Handle empty input

    country_name = country_name.capitalize()  # Capitalize for consistent matching

    for country in data:
        if country["name"]["common"].lower() == country_name.lower():  # Case-insensitive match
            return jsonify(extract_country_data(country))

    return jsonify({"error": f"Country '{country_name}' not found"}), 404

# Endpoint to get a random country
@app.route('/random', methods=['GET'])
def random_country():
    country = random.choice(data)
    return jsonify(extract_country_data(country))

# Helper function to extract country data
def extract_country_data(country):
```

Figure 5. Source code

This picture is the result of the capture layer from the main backend which is the managed API logic system. This code is then deployed using python anywhere so that it can be accessed publicly and managed without using localhost
The backend side of this application is tasked with providing state data accessed by the frontend. The backend is developed using the Python programming language, which was chosen for its simplicity and strong capabilities in data management. In addition, the Flask framework is used, a lightweight web framework that allows for fast and efficient development of RESTful APIs. Flask was chosen for its flexibility and support for various libraries that support optimal data processing.

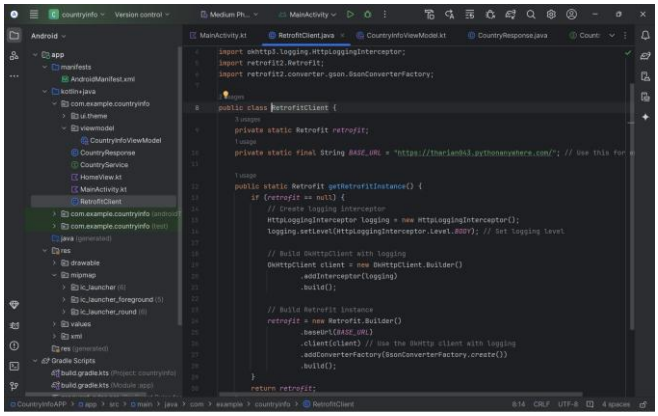


Figure 6. Source code
This picture is the main backend in creating applications in Android Studio, which takes data and systems from the API that has been created

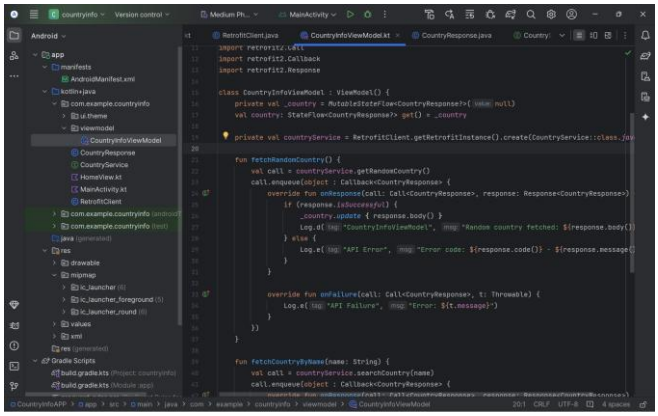


Figure 7. Source code

This picture is the logical part where data modeling techniques and systems are taken from the API and provided to the application.

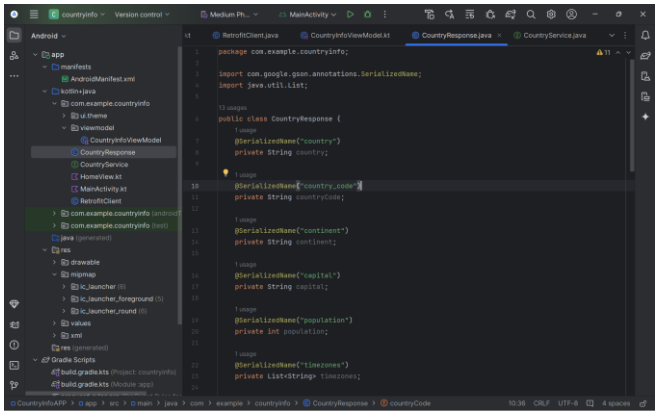


Figure 8. Source code

The following is an image of CountryResponse which functions to manage and label the desired data.

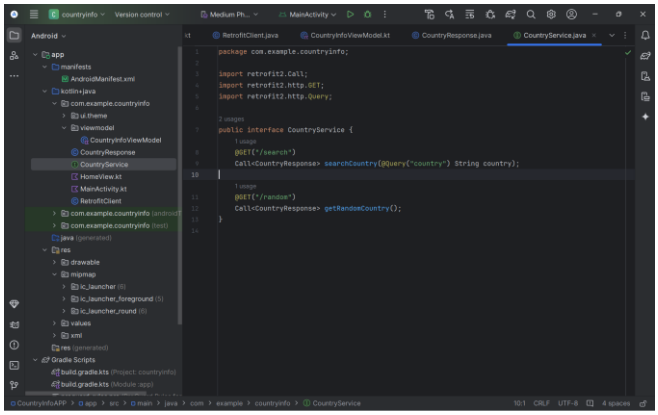


Figure 9. Source code

This picture is the main logic part for requesting data from the application to the managed API.

4.2 Application Compilation Stage

The detailed source code of this application, both frontend and backend, can be taken from the following github:https://github.com/Richard043/CountryApi. On

Github, it is clearly stated how to run the application, its configuration, and its usage system.

4.3 Results of All Experiments

This section presents the results of testing the CountryInfo App application, both through positive test and negative test scenarios. The testing process is carried out by installing the application on an Android device. Testing is carried out using the blackbox testing method to evaluate whether all functions in the application can run according to user needs.

Testing was conducted on several main features of the application, including the country search feature by name, the randomizer feature to display country data randomly, and the detailed display of country information. Test results include the speed of the application's response to data requests, the accuracy of the data displayed, and the application's ability to handle valid and invalid input.

This application is designed to have one type of user, namely the end-user, who can use all the features without requiring an admin role or other special user.

No.	Features Tested	Testing Scenario	Expected results	Test Results	Conclusion
Positive Test					
1.	Random Button	Click random button	Country data comes out randomly	Country data comes out randomly	Passed
2.	Search Button	Enter the country name and click the search button	Output country data according to the country searched	Output country data according to the country searched	Passed
Negative Test					
1.	Search Button	Input the origin and click the search button	Output that country not found	Output that country not found	Passed

Analysis Results and Test Results

After the testing stage using the blackbox testing method for all functions in the CountryInfo App application, it can be concluded:

1. The functions in the application run well, such as the request and response process between the frontend and backend, displaying country data accurately through searching by country name or the randomizer feature, and displaying detailed country information, including population, currency, official language, and geographic area.

2. The application requires a stable internet connection to get real-time data from external APIs, so it cannot be used in offline mode.

3. Test results on the search feature show that the application can handle valid input by displaying appropriate country information, while invalid or empty input produces an informative error message.

4. The randomizer feature works well, displaying random country information every time the button is activated without any data duplication or errors.

5. The application's response to requests is quite fast, with an average time of less than 2 seconds to fetch and display country data, demonstrating the efficiency of the integration between the frontend and backend.

With these test results, it can be concluded that the CountryInfo App has fulfilled the designed functionality and can be used by users effectively.

5. CONCLUSION

5. 1 Conclusion

The CountryInfo App is a modern solution to facilitate access to country information quickly and practically via Android devices. This application is designed with two main features, namely searching by country name to get specific information, such as population, currency, language, and geographic location, and a randomizer feature that allows users to explore country data randomly. By utilizing technologies such as Kotlin for the frontend, Flask and Python for the backend, and integrating public APIs in JSON format, this application is able to present data in real-time with a responsive and intuitive interface, using Jetpack Compose as a design framework.

Application testing using the blackbox testing method showed satisfactory results, with all features running as needed, fast application response (less than 2 seconds), and informative error messages for invalid input. Although it requires an internet connection and is only available for the Android platform, this application provides significant benefits for users, both in the context of education, research, and other personal needs. In addition, the development of this application is an example of the application of modern technology in solving information problems, while also showing the potential for API integration and real-time data processing in mobile-based software development.

5. 2 Suggestion

To improve the quality and functionality of the CountryInfo App, several suggestions can be considered.

First, adding an offline mode feature to store frequently searched country data so that it can still be accessed without an internet connection. This will provide flexibility for users in areas with limited internet connectivity. Second, expanding the application platform to iOS so that more users can enjoy the benefits of this application, while increasing the audience reach. In addition, developing additional features such as global statistics or integration with weather data can increase the appeal and usability of the application for users from various backgrounds.

In addition to features, user interface (UI/UX) optimization can also be done to ensure a more comfortable user experience. Implementing the User-Centered Design (UCD) method can help understand user needs better. Finally, the data monitoring system displayed from the API needs to be strengthened to detect any discrepancies or delays in data updates, so that the application can remain a reliable source of information. By implementing these suggestions, the CountryInfo App will not only be more practical but also relevant to the needs of its.

REFERENCE

[1] M. Z. Abdillah, D. A. Nawangnugraeni, Solikhin and T. W. Adi Putra, "JSON and MySQL Databases for Spatial Visualization of Polygon and Multipolygon Data in Geographic Information Systems: A Comparative Study," *cientific Journal of Informatics,* vol. 10, no. 4, pp. 435-444, 2023. May 2020, doi: 10.3390/ijerph17113789.

[2] M. Santoro, L. Vaccari, D. Mavridis, R. S. Smith, M. Posada and D. Gattwinkel, Web Application Programming Interfaces (APIs): general-purpose standards, terms and European Commission initiatives, Luxembourg: Publications Office of the European Union, 2019.

[3] M. B. Martinez and K. S. McAndrews, "The influence of mobile application design features on users' stickiness intentions as mediated by emotional response," *International Journal of Retail & Distribution Management,* 2021

[4] V. B. R. Bhimanapati, "UI/UX Design Principles for Mobile Health Applications," *International Journal for Research Publication and Seminar,* vol. 15, no. 3, pp. 216-231, 2024.

[5] N. Hiu and Y. Erlyana, "Redesigning User Interface of Datascripmall Mobile Apps Using User Centered Design Method," TEKNIKA, vol. 13, no. 2, pp. 283-292, 2024.

[6] A. Roihan, A. A. Wisanto, Y. Sulaeman, F. M. Nur, S. Wiliandi and W. Pribadi, "Implementasi Metode Realtime, Live Data Dan Parsing JSON Berbasis Mobile Dengan Menggunakan Android Studio Dan PHP Native," *Jurnal Teknologi Informasi,* vol. 5, no. 2, pp. 118-123, 2019.

[7] M. A. Alfaridzi, A. Premana and N. A. Ramdhan, "Aplikasi Berbasis WEB-Realtime Pemantauan Coronavirus Disease 2019 (COVID-19)," Jurnal Ilmiah Intech : Information Technology Journal of UMUS, vol. 3, no. 1, pp. 50-57, 2021.

[8] Hasanuddin, H. Asgar and B. Hartono, "RANCANG BANGUN REST API APLIKASI WESHARE SEBAGAI UPAYA MEMPERMUDAH PELAYANAN DONASI KEMANUSIAAN," *JINTEKS (Jurnal Informatika Teknologi dan Sains),* vol. 4, no. 1, pp. 8-14, 2022.

[9] "PERANCANGAN DAN IMPLEMENTASI APLIKASI ANDROID UNTUK LAYANAN INFORMASI PARIWISATA," *IT-EXPLORE Jurnal Penerapan Teknologi Informasi dan Komunikasi ,* vol. 01, no. 02, pp. 114-132, 2022.

[10] B. Setiawan and Triase, "IMPLEMENTASI DESAIN UI/UX APLIKASI OURTICLE KE DALAM APLIKASI BERBASIS ANDROID," *Sibatik Journal: Jurnal Ilmiah Bidang Sosial, Ekonomi Budaya, Teknologi, dan Pendidikan,* vol. 2, no. 3, pp. 805-818, 2023.

[11] G. Psaila and P. Fosci, "J-CO: A Platform-Independent Framework for Managing Geo-Referenced JSON Data Sets," *MDPI: electronics,* vol. 10, no. 621, pp. 1-35, 2021

[12] E. Chavarriaga, L. A. Roja, K. Sorbello, F. D. Rodríguez and F. Jurado, "Json-Based Domain-Specific Language: A Case Study Using Rhoarchitecture in Designing and Developing Api Restful," *SSRN,* pp. 1-35, 2024.

[13] M. Lnenicka, A. Nikiforova, D. Wang and F. Bernardini, "UX competitive analysis of smart city open data portals: from common features to best design practices and maturity model for a more sustainable smart city data ecosystem," *SSRN,* pp. 1-47, 2024

[14] B. Uyanık and A. Sayar, "Developing Web-Based Process Management with Automatic Code Generation," *MDPI: Applied Science,* vol. 13, no. 11737, pp. 1-29, 2023.

[15] Y. and S. , "PERANCANGAN SISTEM INFORMASI PEMESANAN MAKANAN BERBASIS ANDROID UNTUK MENINGKATKAN PENJUALAN BAGI UMKM," *Format : Jurnal Ilmiah Teknik Informatika,* vol. 13, no. 2, pp. 156-165, 2024.