App covid-19

# Đặt vấn đề

App được viết với mục đích là hiển thị kết quả đo của người dùng lên điện thoại

# Mục tiêu

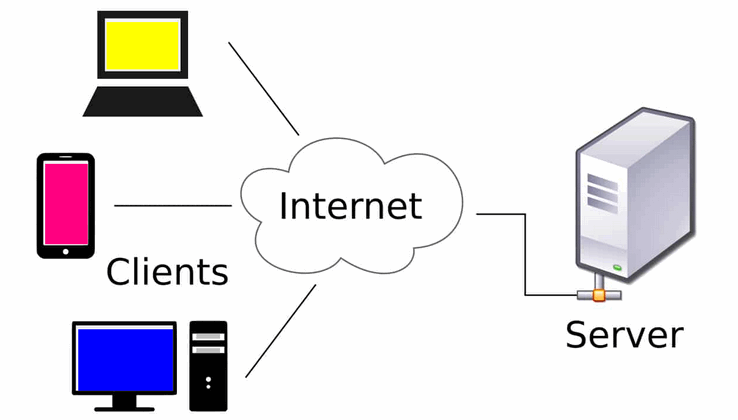
Lấy được thông tin và hiển thị dữ liệu lên app

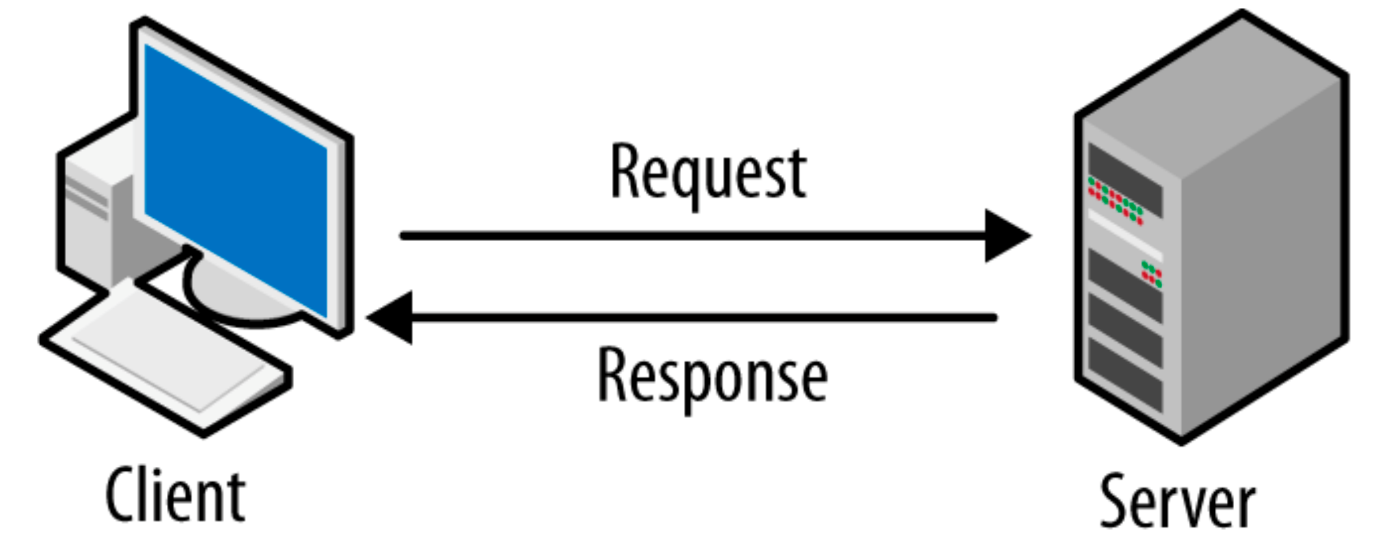
Giao diện dễ nhìn, dễ sử dụng

…

# Thiết kế:

**Mô hình ứng dụng: Client – Server**

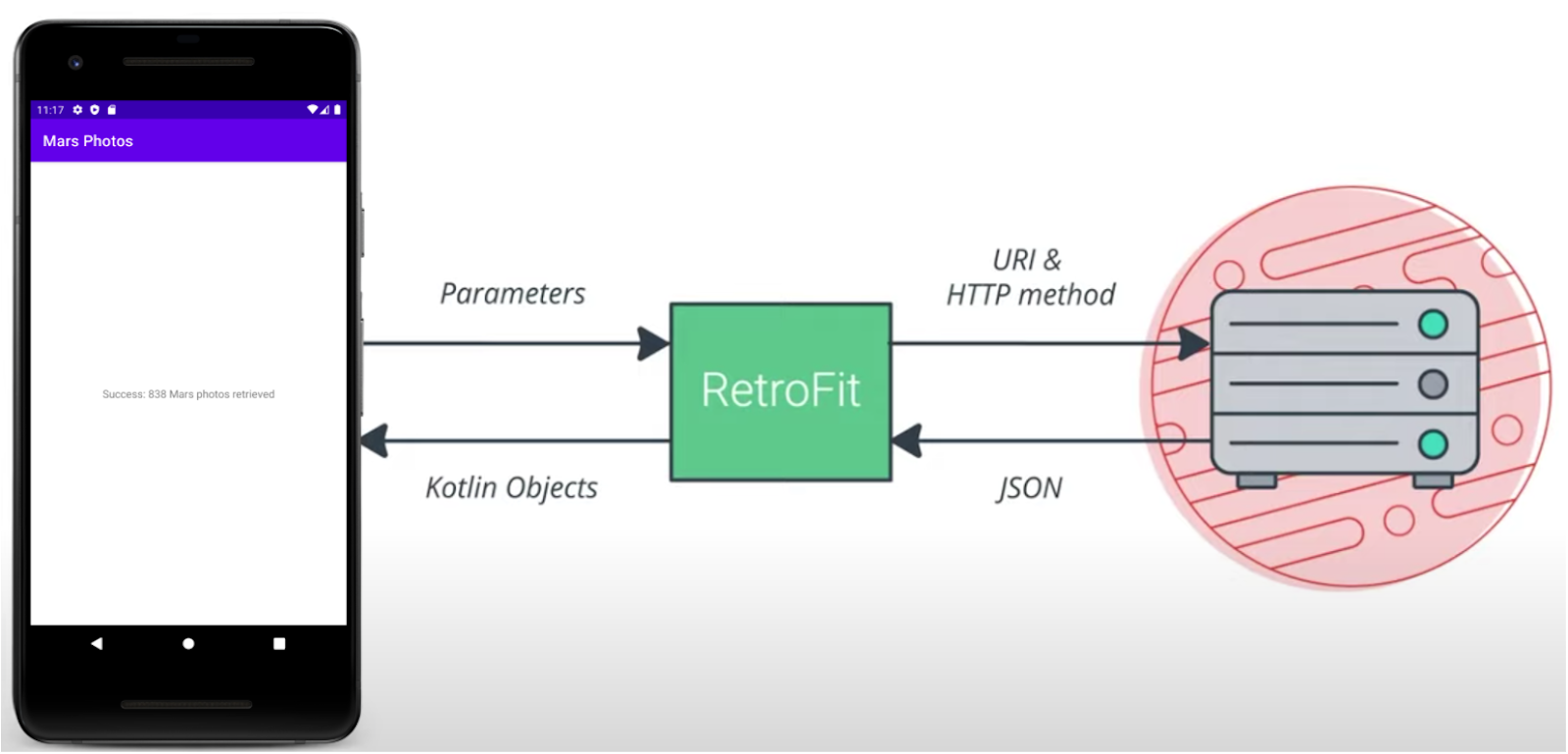




--------------------------------------------

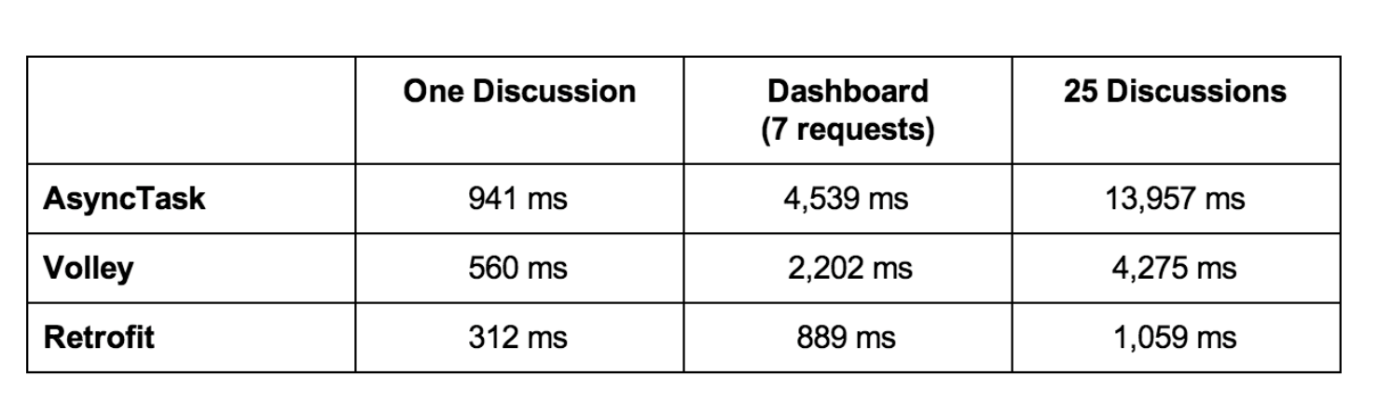
**App sử dụng thư viện retrofit**. Retrofit là một HTTP client type-safe cho Android và Java. Retrofit giúp dễ dàng kết nối đến một dịch vụ REST trên web bằng cách chyển đổi API thành Java Interface.

Retrofit rất mạnh mẽ giúp bạn dễ dàng xử lý dữ liệu JSON hoặc XML sau đó phân tích cú pháp thành Plain Old Java Objects (POJOs). Tất cả các yêu cầu GET, POST, PUT, PATCH, và DELETE đều có thể được thực thi.



**Tại sao chọn retrofit**

* Thư viện bên ngoài hoặc thư viện bên thứ ba giống như phần mở rộng cho các API Android cốt lõi. Các thư viện này chủ yếu là nguồn mở, do cộng đồng phát triển và duy trì được nhờ đóng góp tập thể của cộng đồng Android lớn mạnh trên toàn thế giới. Các thư viện này giúp các nhà phát triển Android như bạn xây dựng ứng dụng tốt hơn.
* Việc sử dụng các thư viện do cộng đồng xây dựng và duy trì có thể giúp tiết kiệm rất nhiều thời gian, nhưng quan trọng là bạn phải chọn thư viện một cách khôn ngoan vì ứng dụng của bạn mới là bên chịu trách nhiệm cuối cùng về hoạt động của mã trong các thư viện này.
* Thư viện Retrofit mà bạn sắp sử dụng trong lớp học lập trình này để trao đổi với dịch vụ web RESTful sao Hoả là một ví dụ điển hình về thư viện được hỗ trợ và duy trì tốt. Bạn có thể nhận thấy điều này bằng cách xem trang GitHub của thư viện, kiểm tra các vấn đề còn tồn đọng (một vài trong số đó là yêu cầu về tính năng) và các vấn đề đã đóng. Nếu các nhà phát triển đang giải quyết vấn đề và thường xuyên trả lời các yêu cầu về tính năng, thì tức là thư viện được duy trì tốt và là một ứng viên phù hợp để sử dụng trong ứng dụng. Họ cũng có trang tài liệu Retrofit.
* Ngoài ra Retrofit ngày càng được phổ biến hơn với nhiều ưu điểm mà nó đem lại, ví dụ như là hiệu năng, thời gian hoàn thành task nhanh hơn các thư viện phổ biến khác như AsyncTask và Volley.



Sử dụng cơ sở dữ liệu mysql để quản lý dữ liệu và lấy dữ liệu bằng api được viết bằng PHP

**Ứng dụng được phát triển dựa trên mô hình MVC (Model - View - Controller)**.

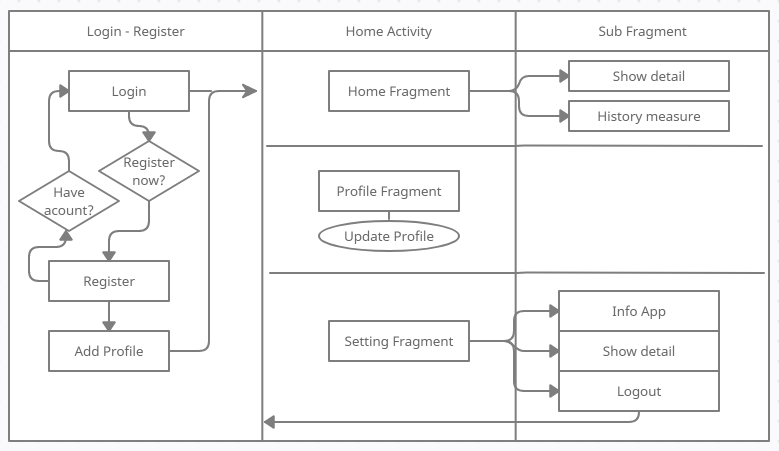
Model: Là bộ phận lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là cầu nối giữa 2 thành phần View và Controller.

View: Là phần giao diện dành cho người sử dụng. Nơi mà người dùng có thể lấy được các thông tin dữ liệu của MVC thông qua các thao tác truy vấn.

Controller: Bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua View, từ đó đưa ra dữ liệu phù hợp. Bên cạnh đó Controller còn có chức năng kết nối đến Model.

Khi một yêu cầu từ máy client gửi tới server, Controller sẽ thực hiện nhiệm vụ tiếp nhận và xử lý yêu cầu. Trong trường hợp cần thiết, nó có thể liên hệ Model - bộ phận làm việc với database để hỗ trợ. Khi xử lý xong yêu cầu, kết quả sẽ được trả về View

# Sơ đồ, luồng Flow giữa các trang



# Đánh giá, hướng phát triển

Dữ liệu chưa được update theo thời gian thực. Do tính chất của database

Thêm kết nối ble