

Algorithms and Data Structures (if applicable)

(as in Report #2, revised as needed)

System Architecture and System Design

1. Architectural Styles

Describe the software architecture style/pattern used in your design.

Kiến trúc MVP (Model – View – Presenter):

MVP là một mô hình kiến trúc hướng giao diện người dùng, được thiết kế để tạo thuận lợi cho việc kiểm thử đơn vị (unit testing) và tăng tính tách biệt giữa tầng dữ liệu và tầng hiển thị dữ liệu trong mô hình MVC.

Mô hình MVP cho phép tách tầng trình diễn (Presenter) ra khỏi tầng dữ liệu (Model), vì vậy tương tác với giao diện được tách biệt với cách chúng ta biểu diễn nó trên màn hình (View), hay nói cách khác, tất cả logic khi người dùng tương tác được tách ra và đưa vào tầng trình diễn. Thiết kế lý tưởng nhất là với cùng một logic được áp dụng cho nhiều View khác nhau và hoán đổi được cho nhau.

Trong đó:

Model là một interface xác định cách mà dữ liệu được hiển thị trong giao diện người dùng.

View là một giao diện người dùng thụ động hiển thị dữ liệu (Model) và tiếp nhận tương tác người dùng và truyền tới để Presenter xử lý tương tác.

Presenter được ví như middle-man. Khi người dùng tương tác với View, Presenter tiếp nhận tương tác người dùng và update Model. Khi Model được update hay có thay đổi, Presenter lấy dữ liệu từ Model, định dạng và đưa tới View để hiển thị.

Kiến trúc Client-Server

- Là mô hình mà trong đó các máy chủ (server) cung cấp và quản lý các nguồn lực và dịch vụ cho máy khách (client)
- Kiểu kiến trúc này có một hoặc nhiều máy khách kết nối với máy chủ trung tâm thông qua mạng hoặc Internet.
- Kiến trúc Client-Server cũng có thể được gọi là mạng mô hình tính toán vì tất cả các yêu cầu và dịch vụ là được phân phối qua mạng.

2. Identifying Subsystems

Draw and describe UML package diagram of subsystems in your system.

3. Mapping Subsystems to Hardware

Does your system need to run on multiple computers? For example, you may have a client (web browser) and a server (web server) subsystems, running on different machines.

If NO, skip to the next item;

If YES, what subsystems run on which machine?

4. Persistent Data Storage

- Không áp dụng.

5. Network Protocol

- Không áp dụng.

6. Global Control Flow

- Quá trình lấy dữ liệu và hiển thị lên view (Activity, Fragment) (Mô hình MVP):

Trong Model chạy 1 tiến trình (gọi là tiến trình A) chạy song song với tiến trình chính, đặt 1 callback trong tiến trình A. Khi dữ liệu được lấy hoàn tất sẽ kích hoạt callback gửi dữ liệu cho Activity hiển thị. Khi đó tiến trình A sẽ kết thúc.

- Quá trình hiển thị và lưu hình ảnh vào bộ nhớ:

Trong hàm onBindViewHolder chạy 1 tiến trình (gọi là tiến trình B) để download ảnh trên server về khi quá trình lấy dữ liệu hoàn tất ta sẽ hiển thị lên view. Sau đó tạo một tiến trình để lưu hình ảnh vào bộ nhớ (gọi là tiến trình C) (ngay khi tạo xong tiến trình C, tiến trình B kết thúc) và khi lưu ảnh xong tiến trình C sẽ kết thúc.

- Quá trình lưu dữ liệu trên server và vào bộ nhớ:

Khi người dùng tiến hành lưu dữ liệu (nhấn button) thì sẽ tạo 2 tiến trình để lưu dữ liệu lên server và vào bộ nhớ. Khi lưu xong tiến trình lưu dữ liệu vào bộ sẽ tự kết thúc, tiến trình lưu dữ liệu trên server sẽ báo thành công hoặc thất bại về và ta sẽ hiển thị lên view cho người dùng biết và sau đó kết thúc.

7. Hardware Requirements

- Phiên bản android ≥ 5.0 (Lollipop (API 21)).
- Màn hình ≥ 5 inch.
- Cần bộ nhớ để chạy và lưu trữ dữ liệu. Sau 1 tuần dùng app bộ nhớ mà app sử dụng trong máy chỉ 8MB rất nhỏ so với bộ nhớ của điện thoại hiện nay. Vì vậy app chỉ cần bộ nhớ máy trống ~ 50 MB là app hoạt động ổn định (Trong điều kiện chỉ có 1 app JobIT hoạt động).
- Cần cấp quyền truy cập bộ nhớ cho app để truy cập, lưu trữ dữ liệu trong bộ nhớ trong.