1. Mordern Art UI
2. Graphical user interface, application, chat or text message

   Description automatically generatedMultiNotes

Một cái note sẽ gồm 4 thuộc tính:

ID: để đại diện cho vị trí của note trên danh sách

content: ngày chỉnh sửa lần cuối của note

content: nội dung của note

title: tựa đề của note

Graphical user interface, application

Description automatically generatedTrong phần chỉnh sửa có:

title: tựa đề của note

content: nội dung của note

**RecyclerView là gì?**

RecyclerView là một ViewGroup, cho phép chứa một danh sách trên một bộ sưu tập dữ liệu được cung cấp với sự hỗ trợ của ViewHolder và đưa chúng lên màn hình hiển thị của người dùng.

Các thành phần chính của RecyclerView bao gồm:

* **Adapter**
* **ViewHolder**
* **LayoutManager**

**Adapter**

Là một class kế thừa từ class RecyclerView.Adapter. Nó nhận vào một tập dữ liệu để hiển thị lên màn hình người dùng thông qua RecyclerView. Nó giống như một lớp chịu trách nhiệm chính để liên kết dữ liệu lên views và hiển thị chúng. Hầu hết các tác vụ xảy ra bên trong lớp Adapter của RecyclerView.

**ViewHolder**

ViewHolder là một lớp kiểu helper class, giúp chúng ta vẽ UI cho từng item mà chúng ta muốn lên trên màn hình. Tất cả các tác vụ binding views của từng item diễn ra trong class này. Nó là một subclass của lớp of RecyclerView.ViewHolder.

**LayoutManager**

LayoutManager trong RecyclerView giúp chúng ta chỉ rõ làm cách nào để chúng ta hiển thị danh sách item lên trên màn hình. Nó có thể là dạng Linear (tuyến tính) hoặc Grid (lưới). Theo mặc định, RecyclerView cung cấp một số triển khai LayoutManager sẵn có. Nó giống như bộ phận quản lý của RecyclerView sẽ thông báo cho adapter của RecyclerView khi nào nên tạo view mới.

Graphical user interface, application

Description automatically generated

Data Set ở đây là dữ liệu trong note của mình, nó sẽ được kéo vào ViewHolder, RycyclerViewAdapter bọc nó lại rồi bỏ qua cho LayoutManager quản lý rồi để RecyclerView để hiển thị lên.

* Khi tạo 1 note mới:
  + - * MainActivity tạo 1 intent và 1 note mới
      * Intent vừa tạo này sẽ mang dữ liệu của note vừa tạo đi để mở EditActivity bằng hàm startActivityForResult(), cho phép nhập tiêu đề và nội dung note với requestCode là 1, tức là cái requestCode đại diện cho việc chỉnh sửa, để mở Activity chỉnh sửa note (chỉnh sửa trước hoặc sau khi thêm mới cũng dùng chung 1 activity này).
      * Vì note đã kế thừa lớp Serializable (lớp cho phép di chuyển dữ liệu là đối tượng giữa các activity), nên có thể lấy được các thuộc tính của note, lấy id của nó để so sánh với với số lượng note có trong danh sách, nếu lớn hơn hoặc bằng số lượng của danh sách, tức là note chưa hề tồn tại trong danh sách, thì sẽ thêm note vào vị trí thứ 0 (đầu tiên của danh sách)
      * Cập nhật list
* Khi chỉnh sửa 1 note:
  + Tạo 1 biến pos để lưu vị trí của note cần chỉnh sửa trên RecyclerView
  + Tạo một note và gán dữ liệu giống như note có vị trí đã chọn
  + Tạo intent, dùng nó để mở EditActivity, truyền cái note đó vào
  + Dùng hàm startActivityForResult() để thực hiện intent đó với requestCode là Edit (có giải thích ở trên)
  + Lấy id của nó để so sánh với với số lượng note có trong danh sách, nếu lớn hơn hoặc bằng số lượng của danh sách, tức là note đã tồn tại trong danh sách, tạo note tạm mới, duyệt danh sách, gán note tạm đó bằng mỗi note duyệt qua, so sánh note đó với note cần chỉnh bằng id, nếu đúng thì break, không thì duyệt tiếp đến khi gặp
  + Xóa note đó ra khỏi danh sách note
  + Add note đó vào lại danh sách vào vị trí đầu tiên, chứ không ghi đè cái cũ.
* Ghi file:
  + Sử dụng file .json, FileOutputStream và JsonWriter để thực hiện
  + Mở file database.json lên bằng FileOutputStream
  + Thực hiện việc ghi trên file vừa được mở bằng FileOutputStream
  + Dùng hàm beginArray() để bắt đầu mảng trong file database.json (là dấu [)
  + Duyệt mảng này,

1. RSSReader