**Mô tả hệ thống**

**Mô tả nghiệp vụ:**

**miêu tả các chức năng đã lập trình và cách hoạt động: Giải thích có funtione nào làm công việc gì, công nghệ sử dụng**

gồm 8 Activity:

MainActivity:

1. Sau khi đã tạo ra các styles.xml thì chỉ cần viết lại phương thức loadNightModeState đã làm ở ChangeTheme là có thể load được theme đã chọn. (Tương tự với các activity khác)
2. Chức năng swipeToRefresh có thể tạo ra icon refresh và set up màu sắc cho icon refresh qua phương thức setColorSchemeColors từ file colors.xml. Có thể tạo độ trễ cho chức năng swipeToRefresh thông qua delayMiillis.
3. Tạo ra 1 thanh search trong hàm void search, sử dụng 1 Arraylist để lưu các giá trị đã tìm ra được. Có thể search kí tự hoa và kí tự thường đều được . Tìm kiếm cả tiêu đề ghi chú lẫn nội dung ghi chú thông qua phương thức getTitledoes và getDoesdoes trong MyDoesApp. Mỗi lần tìm thấy thì add vào Arraylist và hiển thị lên qua DoesAdapter.
4. Tạo ra các Menu cho các chức năng phụ như đổi ngôn ngữ, đổi chủ đề, refresh lại app, thông tin liên lạc qua phương thức onCreateOptionMenu và đối tượng MenuInflater, chọn các chức năng phụ thông qua phương thức onOptionsItemSelected.
5. Chức năng thay đổi ngôn ngữ cho phép đổi tự chọn ngôn ngữ 1 cách chủ động mà không cần thông qua việc phải thay đổi ngôn ngữ của thiết bị android một cách bị động. Tạo ra 2 file strings.xml trong thư mục strings, một file strings.xml chứa ngôn ngữ anh và 1 file strings.xml(vi) chứa ngôn ngữ việt. Sau đó tạo 1 hàm setLocale chứa các object như locale và configuration để thiệt lập lại cấu hình. Tạo ra hàm loadLocale sử dụng Shared Preferences để lưu lại ngôn ngữ đã thay đổi cho lần mở app sau. Tạo hàm showChangeLanguageDialog để hiện thị một Dialog cho việc chọn ngôn ngữ và đưa hàm showChangeLanguageDialog vào sự kiện onOptionsItemSelected.

AboutUs:

1. Cho phép liên hệ với đội ngũ kỹ thuật qua Facebook, Youtube, Email, CH Play, Website và hiển thị các thông tin về app.
2. Sử dụng một thư viện đã có sẵn trong jCenter và import thư viện đó vào activity. Không cần phải thiết kế layout.
3. Gọi các đối tượng đã có sẵn như Element và aboutPage và thêm các thông tin chi tiết vào đối tượng đó.
4. Sử dụng các phương thức như addFacebook, add Youtube, … và truyền đường dẫn vào để thêm được các thông tin liên lạc.

ChangeTheme:

1. Tạo ra các color được lưu trong phần styles.xml để sử dụng cho lightmode và darkmode
2. Tạo file attrs.xml để lưu tên các màu đã tạo trong styles.xml
3. Gán các attrs đã tạo cho các object trong các layout
4. Tạo 1 activity mới, sử dụng nút switch để chuyển qua lại giữa các theme và Shared Preferences để lưu lại các giá trị đã thay đổi khi chuyển theme

DeleteButton☹( phương thức này để xóa các Item bạn check và xóa)

Tạo ra một layout với các Item trong List thay đổi có thêm checkbox. Gồm:

1. Sau khi đã tạo ra các styles.xml thì chỉ cần viết lại phương thức loadNightModeState đã làm ở ChangeTheme là có thể load được theme đã chọn. (Tương tự với các activity khác)
2. Chức năng swipeToRefresh có thể tạo ra icon refresh và set up màu sắc cho icon refresh qua phương thức setColorSchemeColors từ file colors.xml. Có thể tạo độ trễ cho chức năng swipeToRefresh thông qua delayMiillis.
3. Tạo ra 1 thanh search trong hàm void search, sử dụng 1 Arraylist để lưu các giá trị đã tìm ra được. Có thể search kí tự hoa và kí tự thường đều được . Tìm kiếm cả tiêu đề ghi chú lẫn nội dung ghi chú thông qua phương thức getTitledoes và getDoesdoes trong MyDoesApp. Mỗi lần tìm thấy thì add vào Arraylist và hiển thị lên qua DoesAdapter.
4. Chức năng thay đổi ngôn ngữ cho phép đổi tự chọn ngôn ngữ 1 cách chủ động mà không cần thông qua việc phải thay đổi ngôn ngữ của thiết bị android một cách bị động. Tạo ra 2 file strings.xml trong thư mục strings, một file strings.xml chứa ngôn ngữ anh và 1 file strings.xml(vi) chứa ngôn ngữ việt. Sau đó tạo 1 hàm setLocale chứa các object như locale và configuration để thiệt lập lại cấu hình. Tạo ra hàm loadLocale sử dụng Shared Preferences để lưu lại ngôn ngữ đã thay đổi cho lần mở app sau. Tạo hàm showChangeLanguageDialog để hiện thị một Dialog cho việc chọn ngôn ngữ và đưa hàm showChangeLanguageDialog vào sự kiện onOptionsItemSelected.

DoesAdapter: kế thừa phương thức RecycleviewView để lấy các phương thức các Item load các danh sách Note đã lưu

DoesAdapter\_Delete:

1.kế thừa phương thức RecycleviewView để lấy các phương thức các Item load các danh sách Note đã lưu

2.Override phương thức onBinViewHolder và getItemcount để set các phương thức các Item.

EditTaskDesk:

Sau khi các Item được click thì phương thức này sẽ chạy để thay đổi thông tin các Item .

Phương thức getDataChange để nhận lại các thông tin đã được Edit().

Với chức năng chọn ngày thì người dùng sẽ không được chọn ngày ở quá khứ với phương thức setsetMinDate(System.currentTimeMillis().

Hàm Switch case để Load lại menu như ở các Action trước.

MyDoesApp:

Định nghĩa đối Note gồm Title ,nội dung note , set Ngày và Giờ với kiểu String .

Checkbox để chọn các đối tượng.

NewTaskAct:

Để tạo mới một Note với các thuộc tính của một MyDoesAp.

Các Phương thức gởi tạo và truy xuất các thuộc tính.

NotitifierAlarm:

Set âm thanh và nội dung của một Note để thông báo lên thanh trạng thái khi đến ngày giờ đã hẹn .

Set âm thanh dùng Đối tượng Uri, RingtoneManager.getDefaultUri(RingtoneManager.TYPE\_NOTIFICATION);

Set trạng thái thông báo với phương thức NotificationManager notificationManager =(NotificationManager) context.getSystemService(Context.NOTIFICATION\_SERVICE);

SharedPref:

1. Viết ra các setNightModeState và loadNightModeState để lưu và truyền các giá trị cho ChangeTheme Activity.
2. setNightModeState sẽ gán giá trị boolean cho việc thay đổi chủ đề và loadNightModeState sẽ lấy về giá trị boolean đó để load lên chủ đề đã thay đổi. Sau đó gọi hàm loadNightModeState trong các acitivity để thiết lập theme đã chọn

Utils:☹

Layout:

Activity\_about\_us:

Activity\_change\_theme:

1. Sử dụng LinearLayout
2. Sử dụng CardView chứa các textview và nút swicth

Activity\_chose\_delete:(Chưa xử lí được,không áp dụng trong app)

Load lại layout menu với các item có checkbox được hiển thị.

Activity\_edit\_task\_desk:

1. Load Layout khi Edit các Item :
2. Sử dụng LinearLayout
3. Sử dụng các textview để hiển thị tên các đối tương ,edittext để nhập nội dung các đối tượng

Activity\_main:

1. Load Layout khi Edit các Item :
2. Sử dụng LinearLayout
3. Sử dụng các textview để hiển thị tên các đối tương ,edittext để nhập nội dung các đối tượng

Activity\_new\_task:

1. Load layout hiển thị nội dung khi sửa các đối tượng Item bao gồm: LinearLayout, edittext, textview

Item\_delete:

1. Sử dụng LinearLayout
2. Sử dụng CardView chứa các textview và nút swicth

Item\_does:

Set lại các Item có checkBox sẽ xóa khi chọn

Delete\_menu:Set lại menu mới

Mymenu:

Gồm 4 items: Change Theme, Change Language, Refresh, About Us

**Các quan hệ trong hệ thống:(Layout nào. Activity nào làm cái gì liên quan tới nhau)**

1. Các Activity đều sử dụng mymenu.xml
2. MainActivity sử dụng About us, ChangeTheme
3. ChangeTheme sử dụng SharedPref
4. Các thuộc tính trong các layout đều sử dụng màu sắc trong styles.xml

**Công nghệ sử dụng trong đó: (các item công nghệ để viết nên chức năng đó)**

Các công nghệ sử dụng trong project:

Firebase

Write:

Để đưa data lên Firebase ta sẽ cần 1 reference. Ta khai báo như sau:

1. Import Database reference:

**import** com.google.firebase.database.DatabaseReference;

**import** com.google.firebase.database.FirebaseDatabase;

**import** com.google.firebase.database.DataSnapshot;

1. Khai báo reference:

DatabaseReference reference;

1. Sử dụng:

**reference**= FirebaseDatabase.*getInstance*().getReference().child()

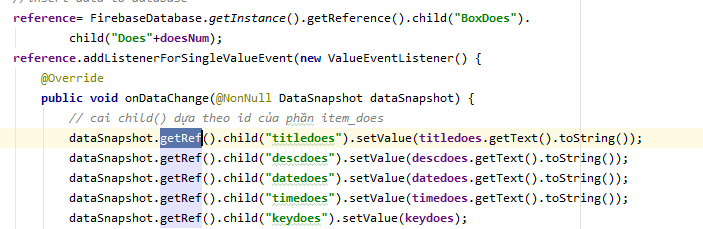
Child chính là object trong database.Do data được lưu theo dạng json nên ta cũng đưa data lên theo 1 object tự đặt tên.Trong object bao gồm các key và value. Ta có thể hiểu key là thuộc tính trong 1 bảng dữ liệu còn value là các giá trị tương ứng với các thuộc tính đó.

1. Sau đó ta dùng phương thức :

**reference**.addListenerForSingleValueEvent().

Overrider lại hàm onDataChange() và dùng dataSnapshot (là data được Database trả về dưới dạng là 1 dataSnapshot) thông qua getRef().child() để tạo ra các key trong object mà ta nói ở trên.Và dùng setValue để thêm vào giá trị cho key đó.

Ví dụ như hình dưới đây:



Read:

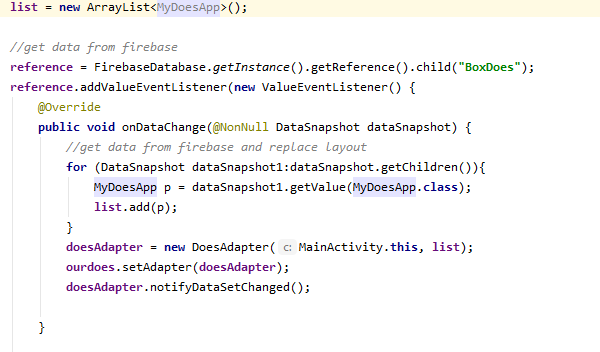
Để đọc lại data từ Firebase ta cũng sẽ cần reference như write.

Khi đọc ta cũng sẽ gán lại tên của object muốn lại dữ liệu mà bên write đã tạo ra. Write đặt tên object là gì thì read sẽ đọc lại tên object đó.

Ví dụ trong project khi write ta tạo ra object tên BoxDoes thì khi đọc cũng đọc từ object BoxDoes đó.

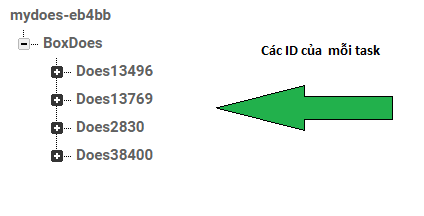
**reference** = FirebaseDatabase.*getInstance*().getReference().child(**"BoxDoes"**);

ta sử dụng phương thức addValueEventListener() và cũng Override lại hàm onDataChange().Trong hàm onDataChange, ta nhận được data từ firebase thông qua dataSnapshot và từ dataSnapshot ta có thể lấy dữ liệu ra tùy ý. Nhưng vì data trả về sẽ bao gồm nhiều child (có thể hiểu là nhiều object con) nên ta có thể dùng vòng lặp for để lấy dữ liệu ra.Như trong project thì ta lấy data ra, add vào trong 1 list.Và từ list này ta có thể đem data ra để hiển thị cho người dùng theo adapter mà ta tự định nghĩa trước.



**Xóa:**

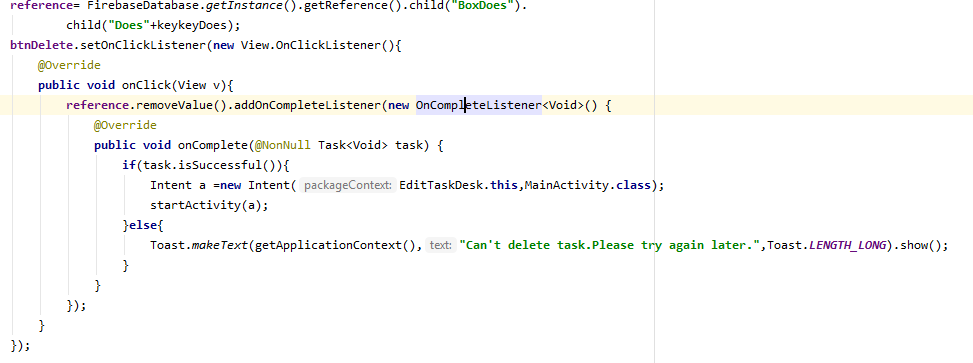
Để xóa 1 task trong firebase ta sẽ xóa thêm tên của object ta đã đặt khi tạo mới 1 task.Trong project mỗi 1 task (object) có tên là 1 cái id riêng biệt. Do đó ta có thể xóa task dựa theo số ID này.



Để xóa ta cũng sẽ cần reference rồi truyền vào cái ID của task muốn xóa theo như hình sau:



keykeyDoes chính là id của task muốn xóa.Sau đó dùng hàm remove() để xóa đi object đó trong firebase.Trong project ta làm như sau:

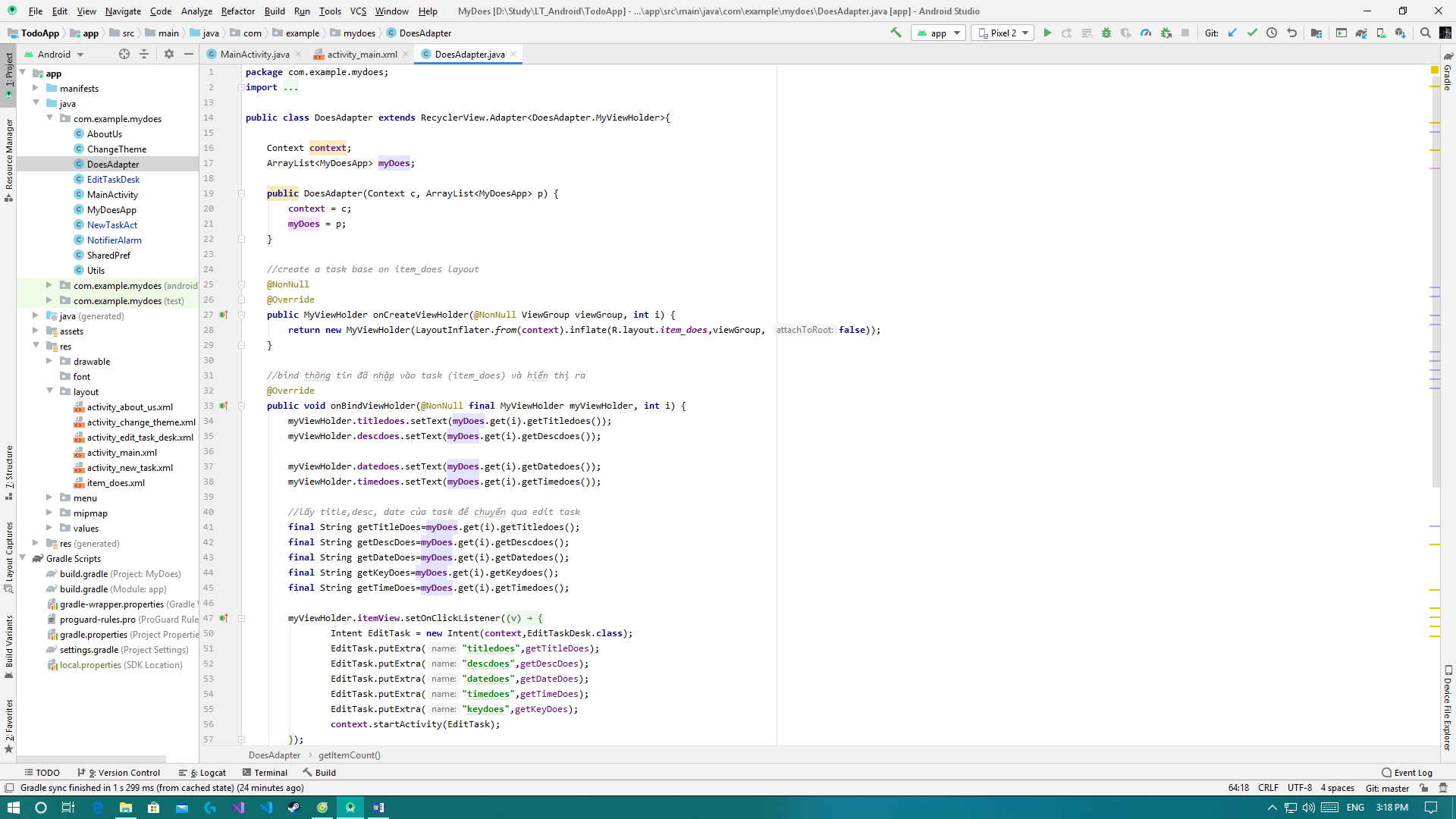


Có thể Override lại hàm onComplete để xử lý sau khi xóa thành công hoặc khi xóa thất bại.

**Recyclerview**

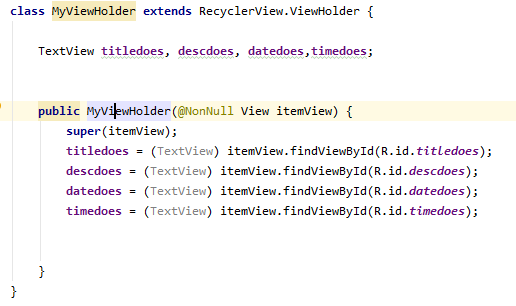
Việc thiết kế layout bằng RecyclerView được làm như sau :



Để RecyclerView có thể hoạt động ta cần 1 adapter.Như trong project ta có DoesAdapter:

Sau khi RecyclerView được kết nối với Adapter, Adapter sẽ tạo ra đối tượng của các hàng (ViewHolder object)cho đến khi lấp đầy kích thước của RecyclerView.RecyclerView sẽ hiển thị các item trong ViewHolder theo layout tùy ý ta muốn qua Layoutmanager.

ViewHolder trong project:



Việc kết nối adapter và RecyclerView được xử lý ở MainActivity như sau:

Data lấy đc từ Firebase sẽ đc thêm vào 1 list ,sau đó truyền list vào trong adapter để làm thành các item, chuẩn bị được đem lên RecyclerView để hiển thị thành các task riêng biệt trên giao diện.



Luồng code chạy sẽ là :

Ở MainActivity bấm nút  để thêm mới 1 task > NewTaskAct sẽ tạo mới 1 task, các dữ liệu nhập vào sẽ được đưa lên firebase.Sau khi quay lại MainActivity, RecyclerView sẽ hiển thị ra các task dựa theo DoesAdapter với các dữ liệu lấy được thông qua dataSnapshot ở MainActivity được biến thành 1 list đưa qua cho ViewHolder trong DoesAdapter