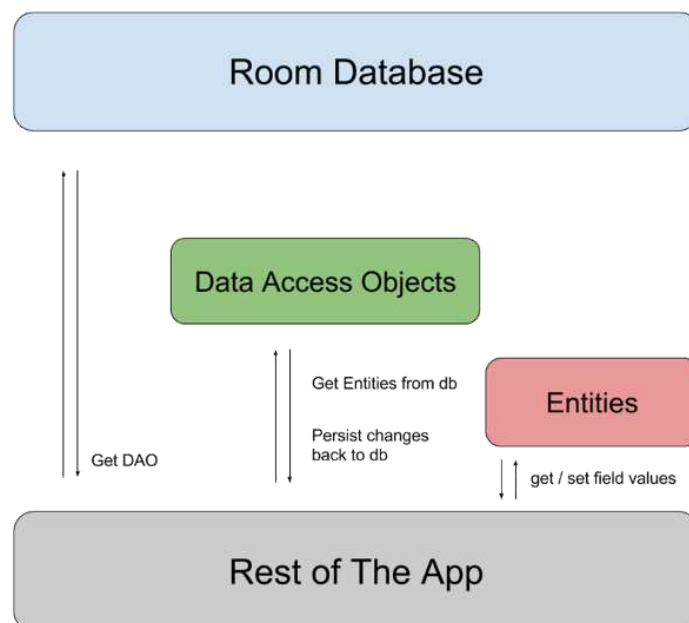


HƯỚNG DẪN TẠO DỰ ÁN VỚI ROOM

Lưu dữ liệu trong cơ sở dữ liệu cụ bộ

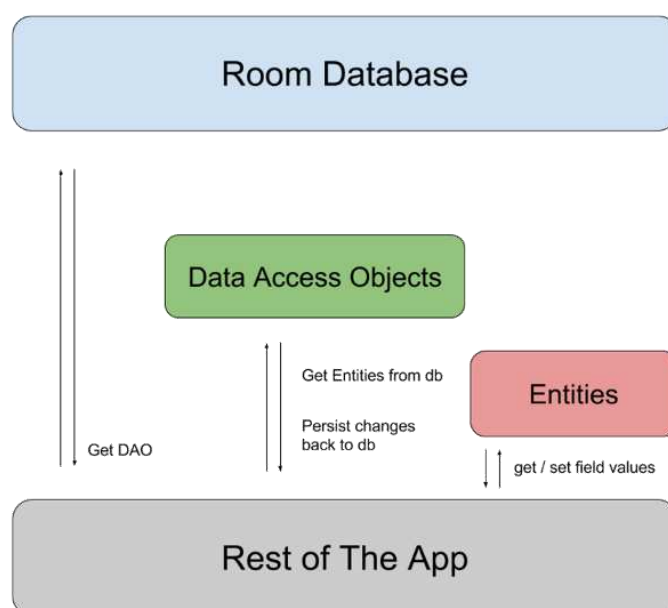


Mục lục

1. Giới thiệu.....	3
2. Các bước thao tác với Room.....	3
3. Ví dụ	4
4. Tự làm	15

1. Giới thiệu

- Room là một ORM (Object Relational Mapping) cho phép thao tác với cơ sở dữ liệu cục bộ
- Room cho phép lập trình viên làm việc với SQLite một cách dễ dàng và nhanh chóng
- Các thành phần của Room
 - o Entity: Đại diện cho một bảng trong cơ sở dữ liệu
 - o DAO (Data Access Object): Chứa các phương thức được sử dụng để truy cập cơ sở dữ liệu
 - o Database: Liệt kê các entity có trong cơ sở dữ liệu và các DAO truy cập chúng



2. Các bước thao tác với Room

- a) Thêm các phụ thuộc Gradle
- b) Tạo Entity
- c) Tạo DAO
- d) Tạo Database
- e) CRUD

3. Ví dụ

a) Thêm các phụ thuộc

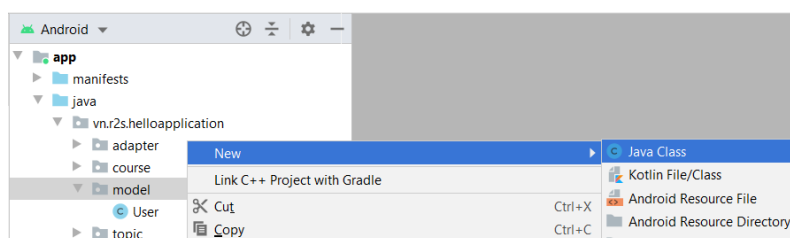
- Mở build.gradle -> Bổ sung code như ghi chú (2)



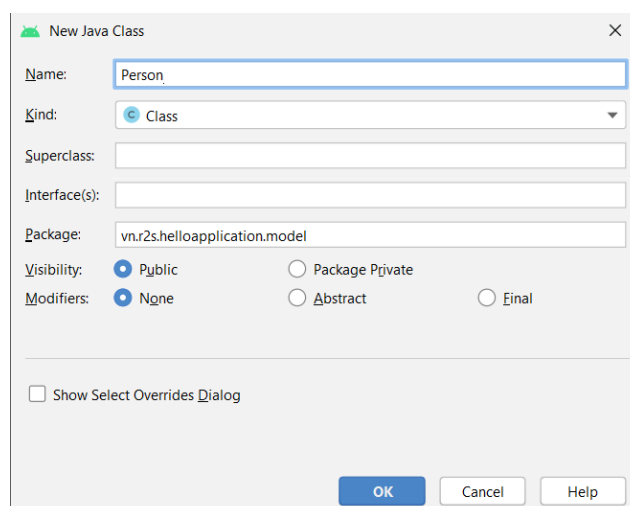
- Chọn Sync Now

b) Tạo Entity

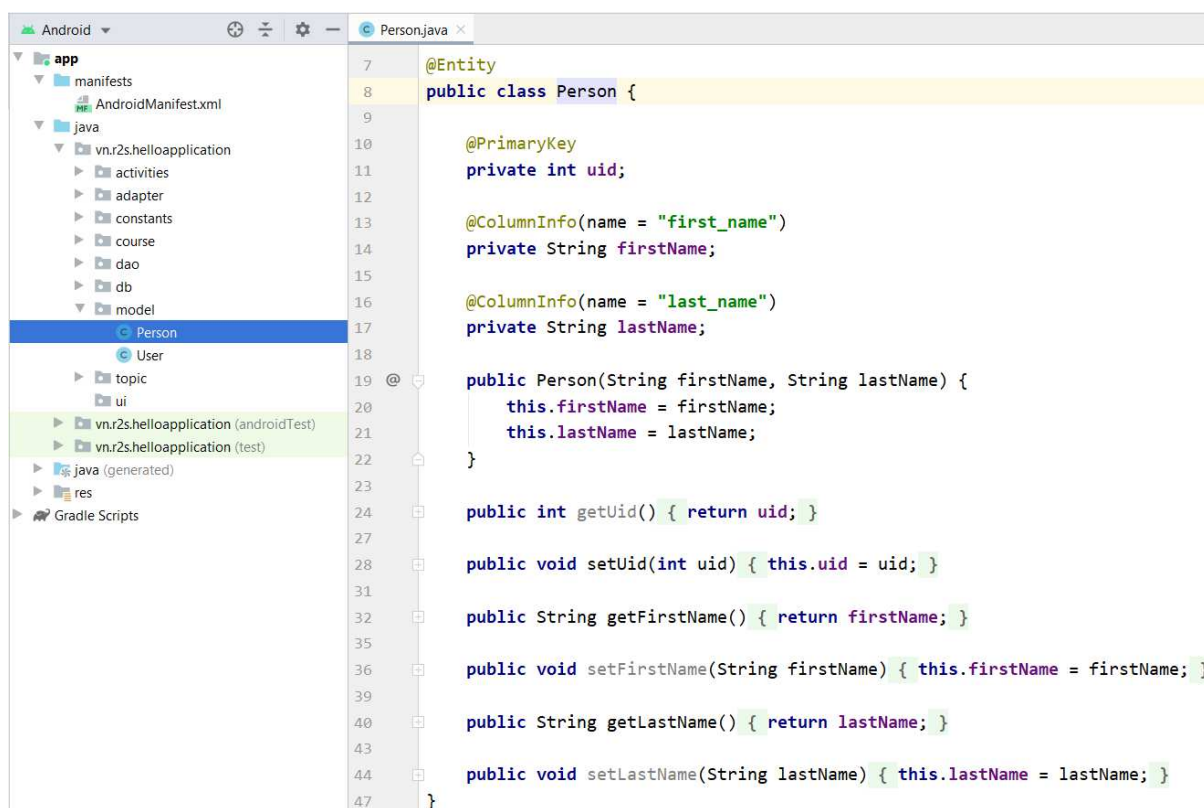
- Tạo Java Class
 - o Chuột phải package -> chọn New -> chọn Java Class



- o Nhập Name, ví dụ Person -> chọn OK

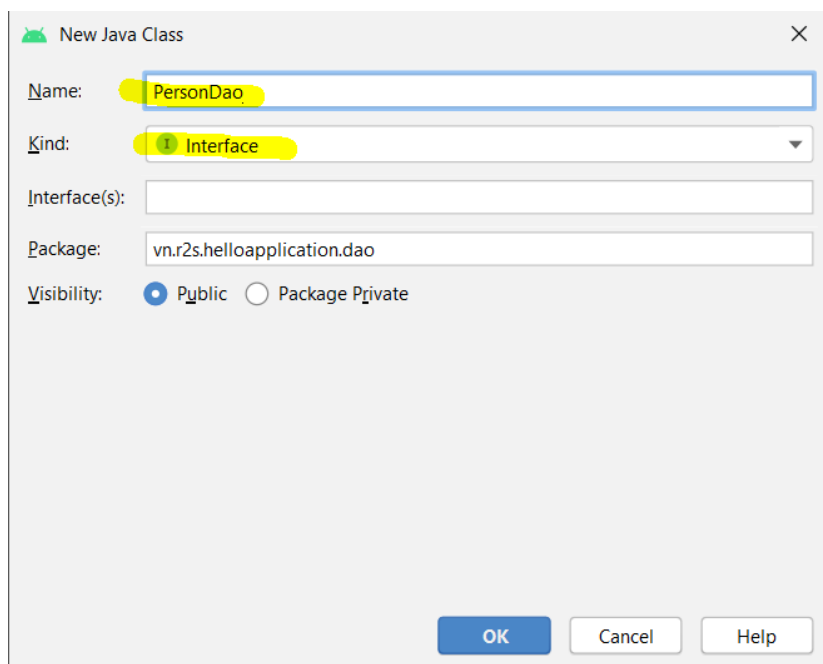


- Nhập code

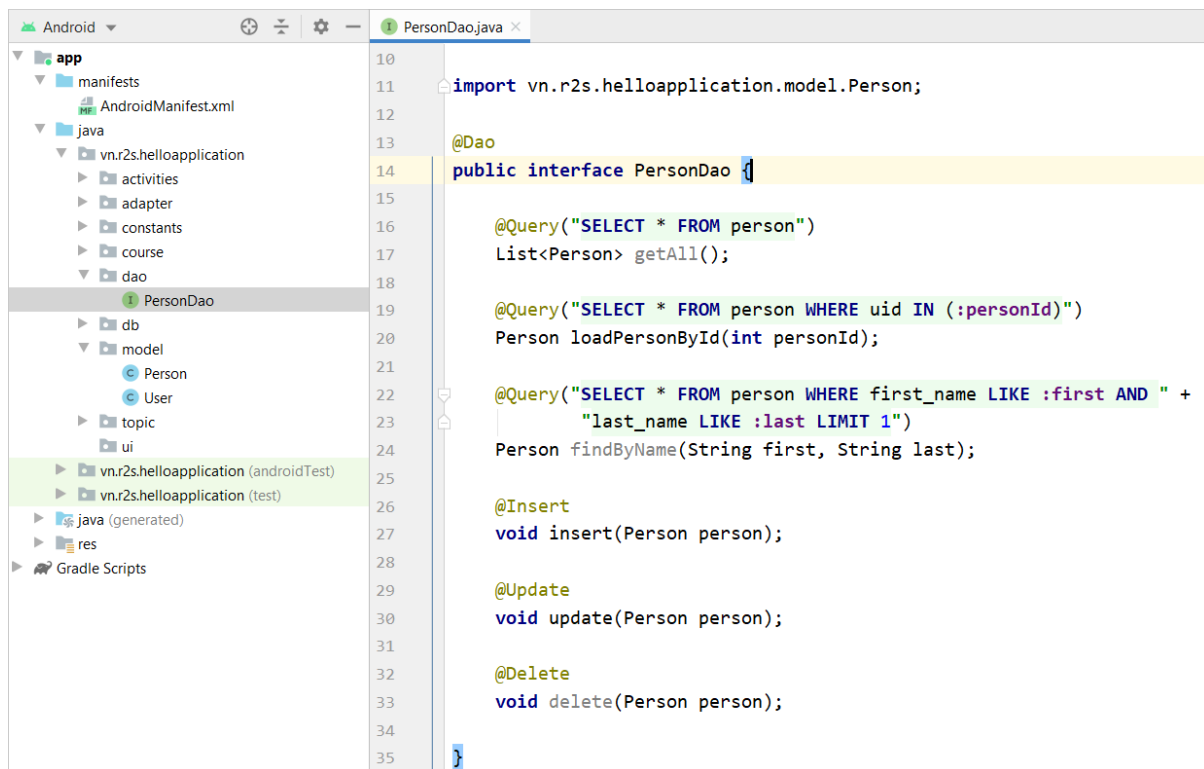


c) Tạo DAO (interface)

- Chuột phải package-> chọn New -> Java Class
- Nhập Name, ví dụ PersonDao và tại Kind chọn Interface -> chọn OK

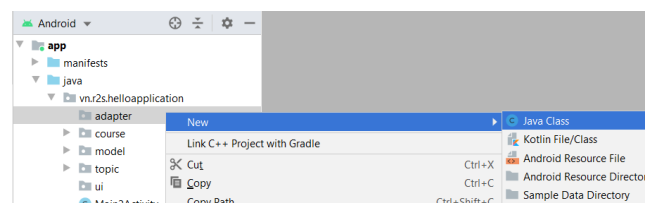


- Nhập code

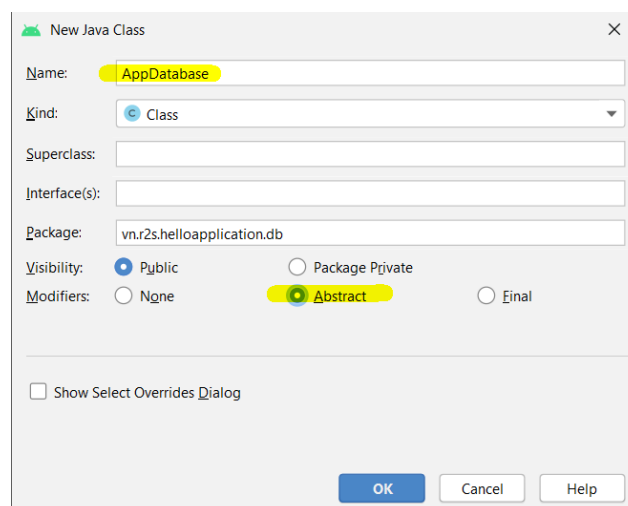


- d) Tạo database (Tạo một lớp trừu tượng (Abstract class))

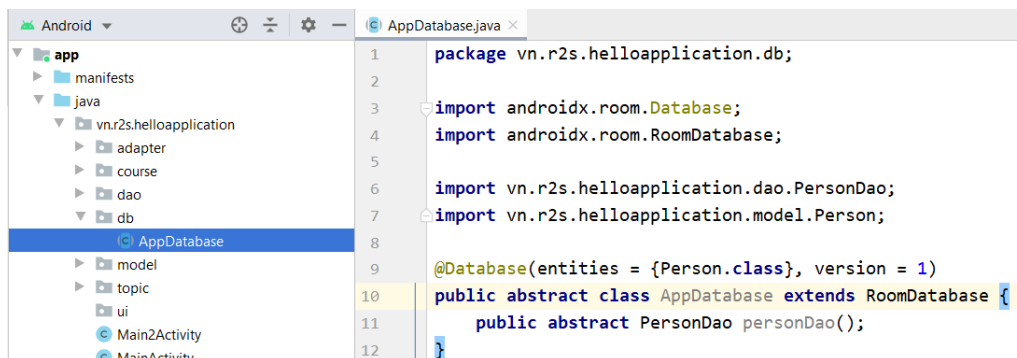
- Chuột phải package -> chọn New -> chọn Java Class



- Nhập Name, ví dụ AppDatabase -> chọn OK



- Nhập code



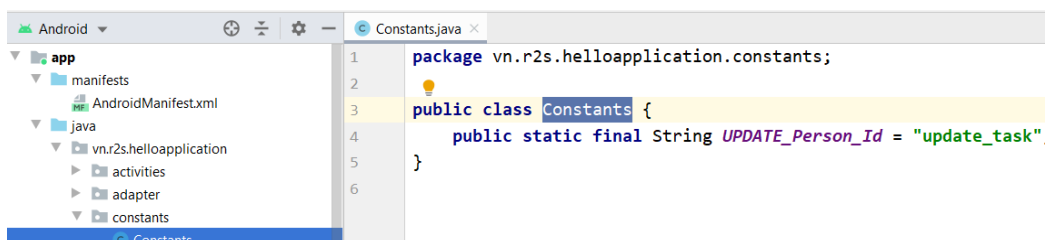
```

1 package vn.r2s.helloapplication.db;
2
3 import androidx.room.Database;
4 import androidx.room.RoomDatabase;
5
6 import vn.r2s.helloapplication.dao.PersonDao;
7 import vn.r2s.helloapplication.model.Person;
8
9 @Database(entities = {Person.class}, version = 1)
10 public abstract class AppDatabase extends RoomDatabase {
11     public abstract PersonDao personDao();
12 }

```

e) CRUD

- Tạo một Java Class và đặt tên **Constants**



```

1 package vn.r2s.helloapplication.constants;
2
3 public class Constants {
4     public static final String UPDATE_PERSON_ID = "update_task";
5 }
6

```

- Tạo một Java Class và đặt tên **AppExecutors**



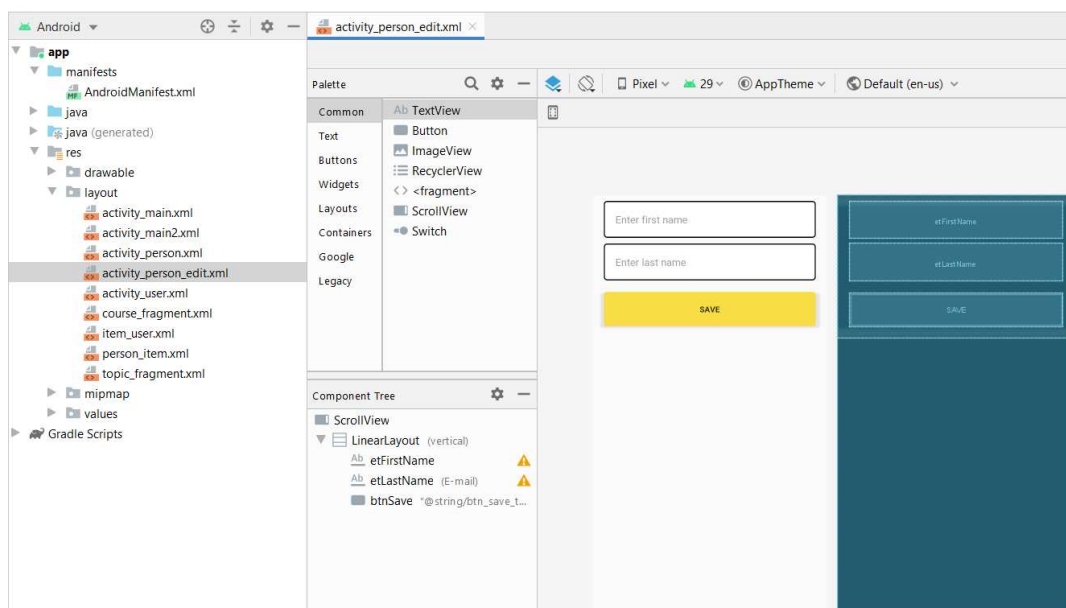
```

33 public class AppExecutors {
34
35     // For Singleton instantiation
36     private static final Object LOCK = new Object();
37     private static AppExecutors sInstance;
38     private final Executor diskIO;
39     private final Executor mainThread;
40     private final Executor networkIO;
41
42     private AppExecutors(Executor diskIO, Executor networkIO, Executor mainThread) {
43         this.diskIO = diskIO;
44         this.networkIO = networkIO;
45         this.mainThread = mainThread;
46     }
47
48     public static AppExecutors getInstance() {
49         if (sInstance == null) {
50             synchronized (LOCK) {
51                 sInstance = new AppExecutors(Executors.newSingleThreadExecutor(),
52                     Executors.newFixedThreadPool(nThreads: 3),
53                     new MainThreadExecutor());
54             }
55         }
56         return sInstance;
57     }
58 }

```

```
58
59     public Executor diskIO() {
60         return diskIO;
61     }
62
63     public Executor mainThread() {
64         return mainThread;
65     }
66
67     public Executor networkIO() {
68         return networkIO;
69     }
70
71     private static class MainThreadExecutor implements Executor {
72         private Handler mainThreadHandler = new Handler(Looper.getMainLooper());
73
74         @Override
75         public void execute(@NonNull Runnable command) {
76             mainThreadHandler.post(command);
77         }
78     }
79 }
```

- Tạo **EditPersonActivity** gồm 2 EditText và 1 Button
 - Chuột phải package -> chọn New -> chọn Activity -> chọn Empty Activity
 - Thiết kế layout



- Nhập code


```
19 public class EditPersonActivity extends AppCompatActivity {
20
21     private EditText etFistName;
22     private EditText etLastName;
23
24     private Button btnSave;
25     private int mPersonId;
26
27     private Intent intent;
28     private AppDatabase mDb;
29
30     @Override
31     protected void onCreate(final Bundle savedInstanceState) {
32         super.onCreate(savedInstanceState);
33
34         setContentView(R.layout.activity_person_edit);
35         getSupportActionBar().setDisplayHomeAsUpEnabled(true);
36
37         initView();
38         mDb = Room.databaseBuilder(getApplicationContext(),
39             AppDatabase.class, name: "app-database").build();
40
41         intent = getIntent();
42         if (intent != null && intent.hasExtra(Constants.UPDATE_Person_Id)) {
43             btnSave.setText("Update");
44
45             mPersonId = intent.getIntExtra(Constants.UPDATE_Person_Id, defaultValue: -1);
46
47             AppExecutors.getInstance().diskIO().execute(new Runnable() {
48                 @Override
49                 public void run() {
50                     Person person = mDb.personDao().loadPersonById(mPersonId);
51                     populateUI(person);
52                 }
53             });
54
55         }
56     }
57
58     private void populateUI(Person person) {
59
60         if (person == null) {
61             return;
62         }
63
64         etFistName.setText(person.getFirstName());
65         etLastName.setText(person.getLastName());
66     }
```

```
70     private void initView() {
71         etFirstName = findViewById(R.id.etFirstName);
72         etLastName = findViewById(R.id.etLastName);
73
74         btnSave = findViewById(R.id.btnSave);
75         btnSave.setOnClickListener(new View.OnClickListener() {
76             @Override
77             public void onClick(View v) {
78                 onSaveButtonClicked();
79             }
80         });
81     }
82
83     public void onSaveButtonClicked() {
84         final Person person = new Person(
85             etFirstName.getText().toString(),
86             etLastName.getText().toString());
87
88         AppExecutors.getInstance().diskIO().execute(new Runnable() {
89             @Override
90             public void run() {
91                 if (!intent.hasExtra(Constants.UPDATE_PERSON_ID)) {
92                     mDb.personDao().insert(person);
93                 } else {
94                     person.setUid(mPersonId);
95                     mDb.personDao().update(person);
96                 }
97                 finish();
98             }
99         });
100     }
101
102     @Override
103     public boolean onOptionsItemSelected(MenuItem item) {
104         switch (item.getItemId()) {
105             // Respond to the action bar's Up/Home button
106             case android.R.id.home:
107                 onBackPressed();
108                 return true;
109         }
110         return super.onOptionsItemSelected(item);
111     }
```

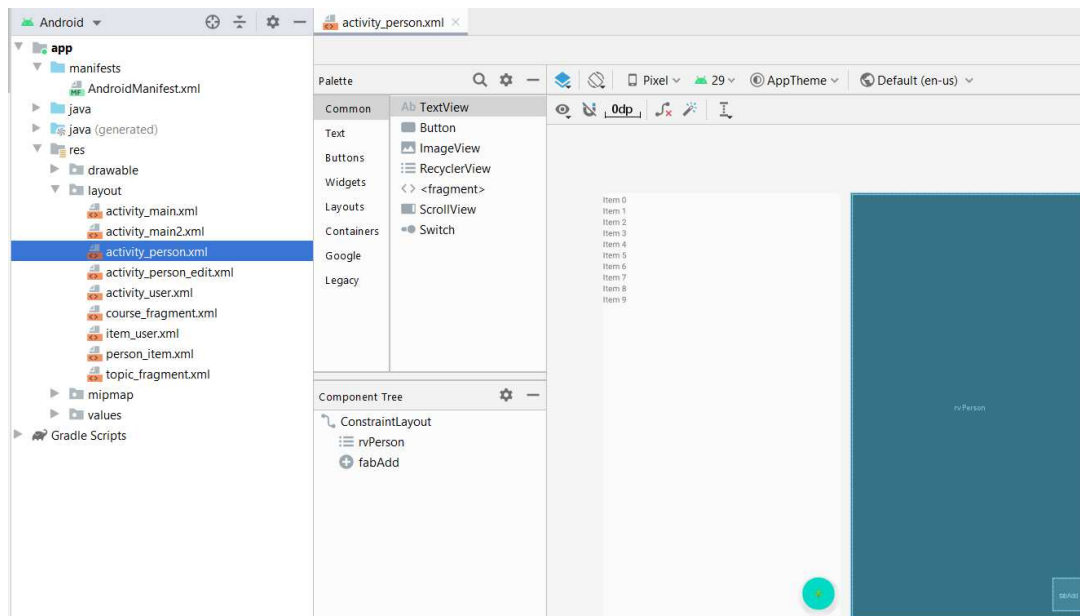
- Tạo PersonAdapter
 - o Tạo một Java Class
 - o Nhập code

```

22 public class PersonAdapter extends RecyclerView.Adapter<PersonAdapter.MyViewHolder> {
23     private Context context;
24     private List<Person> mPersonList;
25
26     public PersonAdapter(Context context) {
27         this.context = context;
28     }
29
30     @NonNull
31     @Override
32     public MyViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {
33         View view = LayoutInflater.from(context).inflate(R.layout.person_item,
34             viewGroup, attachToRoot: false);
35         return new MyViewHolder(view);
36     }
37
38     @Override
39     public void onBindViewHolder(@NonNull PersonAdapter.MyViewHolder myViewHolder, int i) {
40         myViewHolder.name.setText(mPersonList.get(i).getFirstName());
41         myViewHolder.email.setText(mPersonList.get(i).getLastName());
42     }
43
44     @Override
45     public int getItemCount() {
46         if (mPersonList == null) {
47             return 0;
48         }
49
50         return mPersonList.size();
51     }
52
53     public void setTasks(List<Person> personList) {
54         mPersonList = personList;
55         notifyDataSetChanged();
56     }
57
58     public List<Person> getTasks() {
59         return mPersonList;
60     }
61
62     class MyViewHolder extends RecyclerView.ViewHolder {
63         TextView name, email;
64         ImageView editImage;
65
66         MyViewHolder(@NonNull final View itemView) {
67             super(itemView);
68
69             name = itemView.findViewById(R.id.tvFistName);
70             email = itemView.findViewById(R.id.tvLastName);
71
72             editImage = itemView.findViewById(R.id.ivEdit);
73             editImage.setOnClickListener(new View.OnClickListener() {
74                 @Override
75                 public void onClick(View v) {
76                     int elementId = mPersonList.get(getAdapterPosition()).getUid();
77                     Intent i = new Intent(context, EditPersonActivity.class);
78                     i.putExtra(Constants.UPDATE_Person_Id, elementId);
79                     context.startActivity(i);
80                 }
81             });
82         }
83     }
84 }

```

- Tạo PersonActivity gồm RecyclerView và FloatingActionButton
 - Chuột phải package -> chọn New -> chọn Activity -> chọn Empty Activity
 - Thiết kế layout



- Nhập code

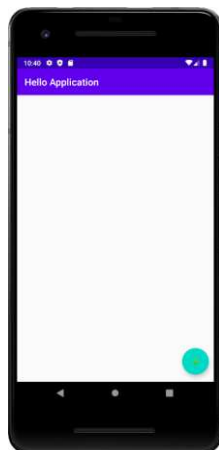
```

23 public class PersonActivity extends AppCompatActivity {
24
25     private FloatingActionButton fabAdd;
26     private RecyclerView mRecyclerView;
27     private PersonAdapter mAdapter;
28     private AppDatabase mDb;
29
30     @Override
31     protected void onCreate(Bundle savedInstanceState) {
32         super.onCreate(savedInstanceState);
33         setContentView(R.layout.activity_person);
34
35         fabAdd = findViewById(R.id.fabAdd);
36
37         fabAdd.setOnClickListener(new View.OnClickListener() {
38             @Override
39             public void onClick(View v) {
40                 startActivity(new Intent( packageContext, PersonActivity.this
41                                     , EditPersonActivity.class));
42             }
43         });
44
45         mRecyclerView = findViewById(R.id.rvPerson);
46
47         mRecyclerView.setLayoutManager(new LinearLayoutManager( context, this));
48
49         // Initialize the adapter and attach it to the RecyclerView
50         mAdapter = new PersonAdapter( context, this);
51         mRecyclerView.setAdapter(mAdapter);

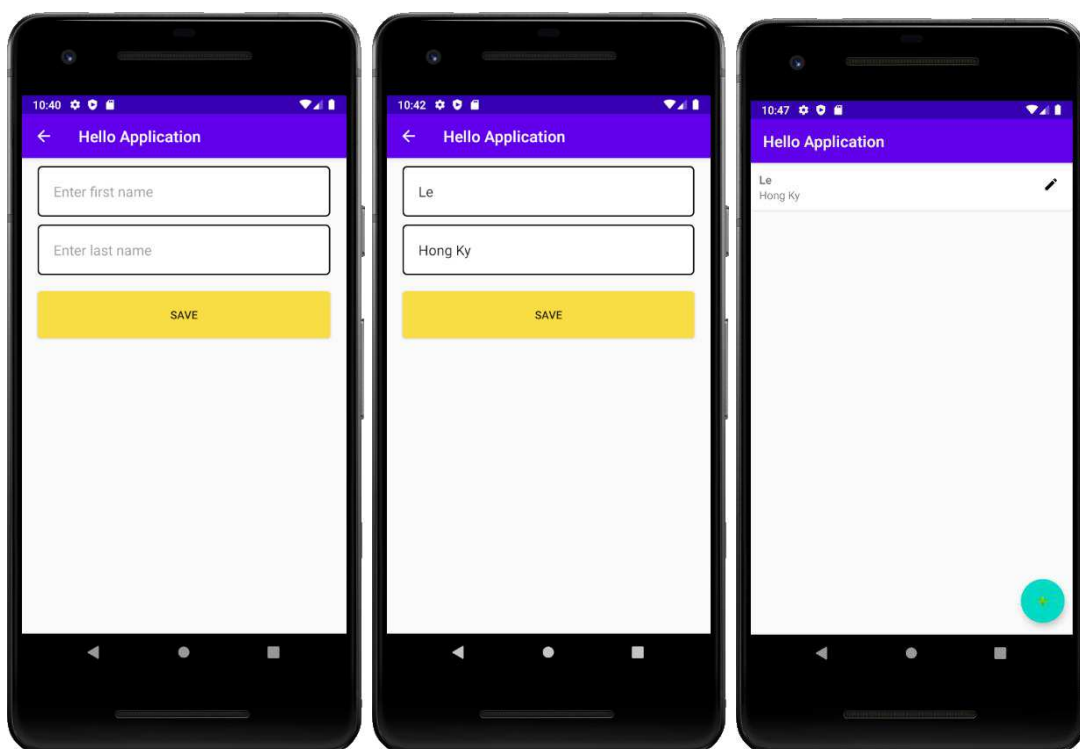
```

```
52
53     mDb = Room.databaseBuilder(getApplicationContext(),
54         AppDatabase.class, name: "app-database").build();
55
56     new ItemTouchHelper(new ItemTouchHelper.SimpleCallback( dragDirs: 0,
57         swipeDirs: ItemTouchHelper.LEFT | ItemTouchHelper.RIGHT) {
58         @Override
59         public boolean onMove(RecyclerView recyclerView, RecyclerView.ViewHolder viewHolder
60             , RecyclerView.ViewHolder target) {
61             return false;
62         }
63
64         // Called when a user swipes Left or right on a ViewHolder
65         @Override
66         public void onSwiped(final RecyclerView.ViewHolder viewHolder, int swipeDir) {
67             // Here is where you'll implement swipe to delete
68             AppExecutors.getInstance().diskIO().execute(new Runnable() {
69                 @Override
70                 public void run() {
71                     int position = viewHolder.getAdapterPosition();
72                     List<Person> tasks = mAdapter.getTasks();
73                     mDb.personDao().delete(tasks.get(position));
74                     retrieveTasks();
75                 }
76             });
77         }
78     }).attachToRecyclerView(mRecyclerView);
79
80
81     @Override
82     protected void onResume() {
83         super.onResume();
84         retrieveTasks();
85     }
86
87     private void retrieveTasks() {
88         AppExecutors.getInstance().diskIO().execute(new Runnable() {
89             @Override
90             public void run() {
91                 final List<Person> persons = mDb.personDao().getAll();
92                 runOnUiThread(new Runnable() {
93                     @Override
94                     public void run() {
95                         mAdapter.setTasks(persons);
96                     }
97                 });
98             }
99         });
100
101 }
```

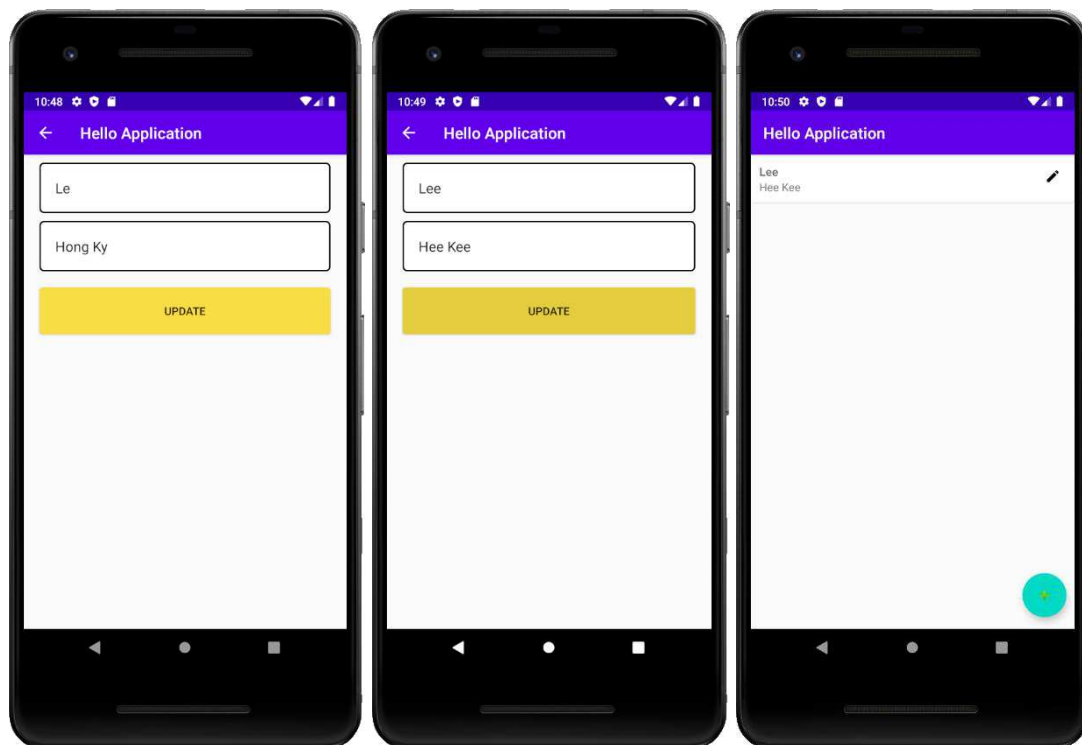
- Chạy ứng dụng



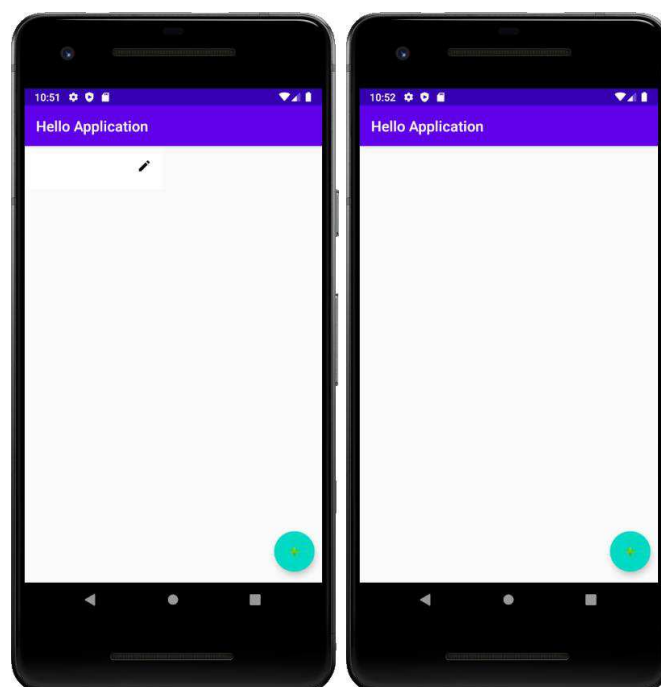
- Chọn (+) -> nhập first name, last name và chọn 'Save'



- Chọn 'Edit' (Biểu tượng cây bút) -> Thay đổi First name và Last name -> chọn UPDATE



- Chức năng xóa được thực hiện bằng cách chọn giữ và vuốt sang trái hoặc sang phải



4. Tự làm

- Sử dụng bài tự làm của tài liệu RecyclerView và bổ sung tính năng

- Cho phép nhập thông tin Module gồm Title, Description, Category (bỏ icon)
- Lưu vào local database
- Ứng dụng hỗ trợ CRUD

