

## BÀI TẬP THÊM BUỔI 9 (CHƯƠNG 4)

Học phần: Phát triển ứng dụng trên thiết bị di động\_TN216

Sinh viên thực hiện: **Họ và tên – MSSV**

### Lưu ý

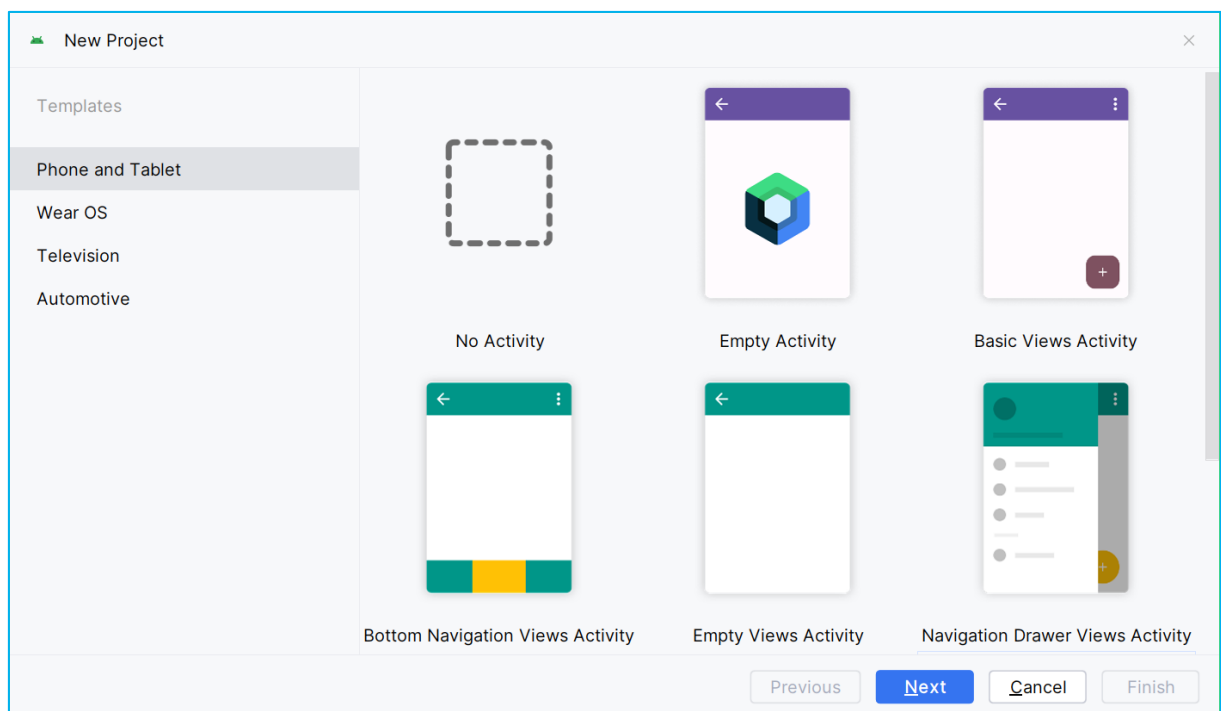
- Sử dụng file mẫu trên Hệ thống học trực tuyến của Trường để làm bài.
- Hiểu rõ mục tiêu, ý nghĩa, và mã lệnh của các bài tập.
- Sử dụng Snipping Tool để chụp hình kết quả thực hiện các yêu cầu.

### Mục tiêu

- Thiết kế giao diện với các Layout và điều khiển thông dụng.
- Lập trình xử lý sự kiện, sử dụng Intent không tường minh (Implicit Intents).
- Tạo ứng dụng theo mẫu Navigation Drawer.

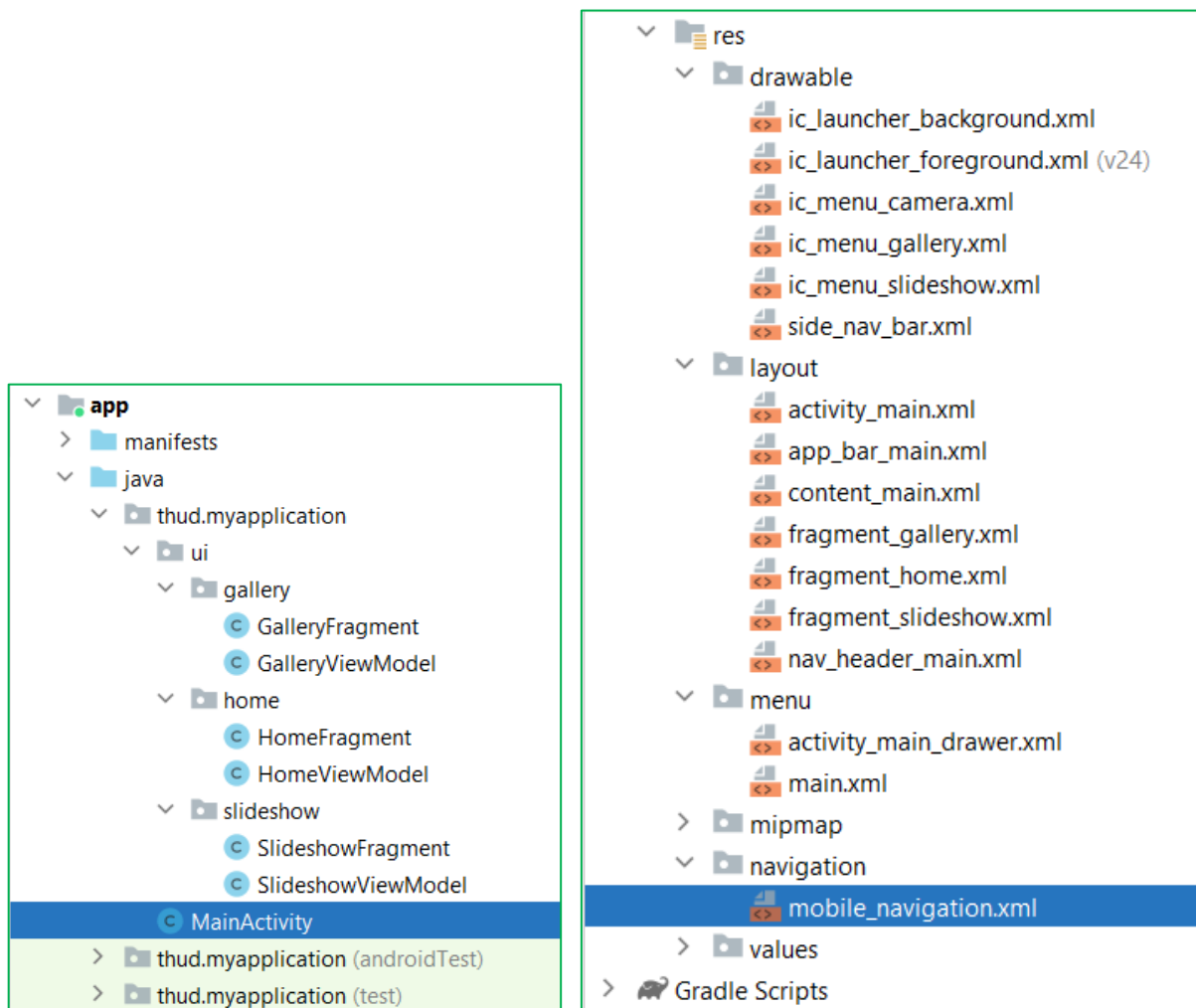
### Kết quả thực hiện

- **Bước 1: tạo ứng dụng mới, tìm hiểu cấu trúc và các thành phần trong ứng dụng Navigation Drawer**
  - + Tạo dự án mới, chọn mẫu Navigation Drawer Views Activity như Hình 1, đặt tên cho dự án là DuLichDaLat với Package Name là thud.dulichdalat.



**Hình 1. Tạo dự án mới theo mẫu Navigation Drawer Activity**

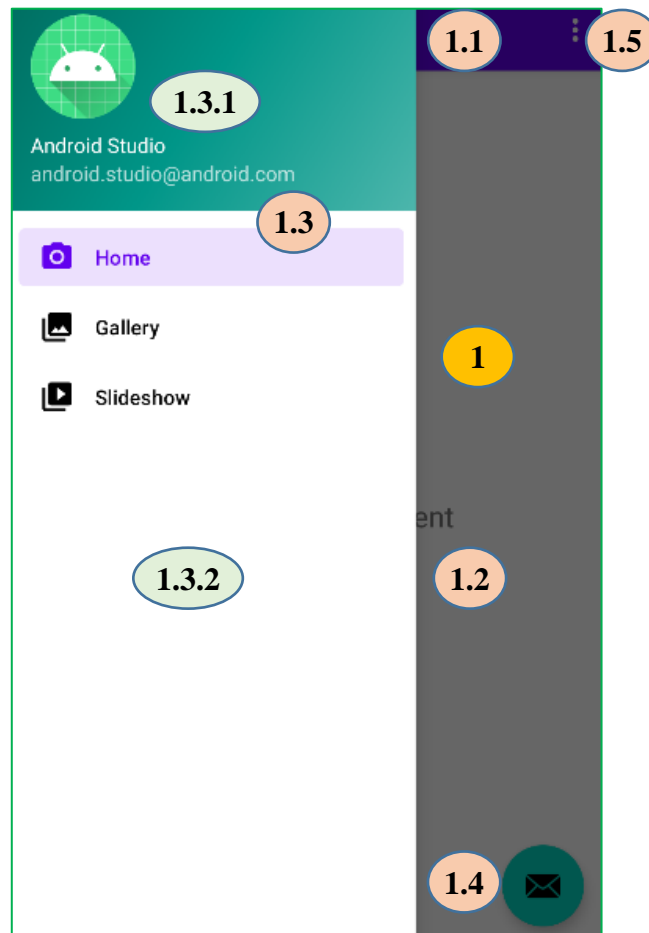
- + Ứng dụng được tạo sẵn với các thành phần như Hình 2, kết quả thực thi như Hình 3. Cụ thể gồm:



**Hình 2. Cấu trúc và các thành phần trong ứng dụng Navigation Drawer**

- ✓ Package **ui** có 3 Package con lưu các Fragment: gallery, home, và slideshow. Mỗi Package con có 2 lớp kế thừa từ Fragment và ViewModel.
- ✓ Các tập tin giao diện (trong thư mục res/layout) được tự động tạo tương ứng vị trí các thành phần như Hình 2 và Hình 3 gồm:
  - activity\_main.xml: giao diện Activity chủ (Vùng 1)
  - app\_bar\_main.xml: giao diện cho ActionBar (Vùng 1.1), vùng nội dung và nút lệnh nổi FloatingActionButton (vị trí 1.4)
  - content\_main.xml: giao diện chính thể hiện nội dung của Fragment (Vùng 1.2)
  - nav\_header\_main.xml: giao diện thể hiện tiêu đề của thanh điều hướng (Vùng 1.3.1)
  - Các tập tin layout được tạo sẵn cho các Fragment: fragment\_home, fragment\_gallery, và fragment\_slideshow

- ✓ Các tập tin thực đơn (trong thư mục res/menu) được tự động tạo gồm:
  - activity\_main\_drawer.xml: thực đơn chức năng (Vùng 1.3.2)
  - main.xml: thực đơn tùy chọn (Vị trí 1.5).
- + Tập tin giao diện điều hướng (res/navigation/mobile\_navigation.xml) được tự động tạo: khai báo các Fragment trong ứng dụng.



**Hình 3. Kết quả thực thi ứng dụng Navigation Drawer**

- **Bước 2: kiểm tra (và điều chỉnh nếu chưa đúng) tập tin giao diện activity\_main**
  - + Hỗ trợ thực đơn điều hướng thực hiện đúng chức năng được chọn,
  - + Tập tin giao diện res/layout/activity\_main.xml như bên dưới,
  - + Thực thi kiểm tra kết quả.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
```

```
        android:layout_height="match_parent"
        android:fitsSystemWindows="true"
        tools:openDrawer="start">

        <include
            android:id="@+id/app_bar_main"
            layout="@layout/app_bar_main"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />

        <com.google.android.material.navigation.NavigationView
            android:id="@+id/nav_view"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_gravity="start"
            android:fitsSystemWindows="true"
            app:headerLayout="@layout/nav_header_main"
            app:menu="@menu/activity_main_drawer" />

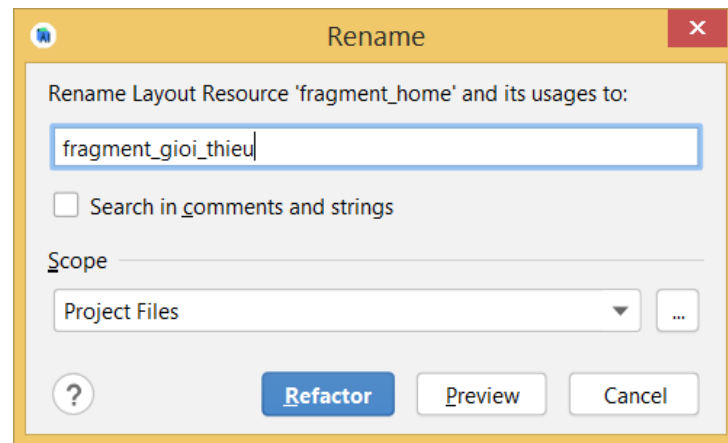
    </androidx.drawerlayout.widget.DrawerLayout>
```

– **Bước 3: thêm các đối tượng hỗ trợ**

- + Trong res/values, thêm các Styles cần thiết: LabelStyle, TextStyle, và ButtonStyle
- + Trong res/drawable, thêm các tập tin hình ảnh làm icon hiển thị trong thực đơn chức năng (tập tin ic\_menu\_contacts.xml, ic\_menu\_send.xml, ic\_menu\_share.xml)
- + Trong res/mipmap, thêm các tập tin hình ảnh
  - ✓ Hình làm Icon cho ứng dụng (tập tin globelogo.png, globelogo\_small.png),
  - ✓ Hình hiển thị trong Fragment HìnhAnh (tập tin dalat01.png, dalat02.png, dalat03.png, dalat04.png).

– **Bước 4: đổi tên các đối tượng và điều chỉnh mã lệnh cho phù hợp với các chức năng: home → giới thiệu, gallery → hình ảnh, slideshow → liên hệ**

- + Trong res/layout, thực hiện đổi tên các tập tin giao diện: R\_Click lên tập tin → chọn Refactor → chọn Rename → Nhập tên mới → Refactor
  - fragment\_home.xml           →   fragment\_gioi\_thieu.xml
  - fragment\_gallery.xml       →   fragment\_hinh\_anh.xml
  - fragment\_slideshow.xml   →   fragment\_lien\_he.xml



Hình 4. Đổi tên đối tượng

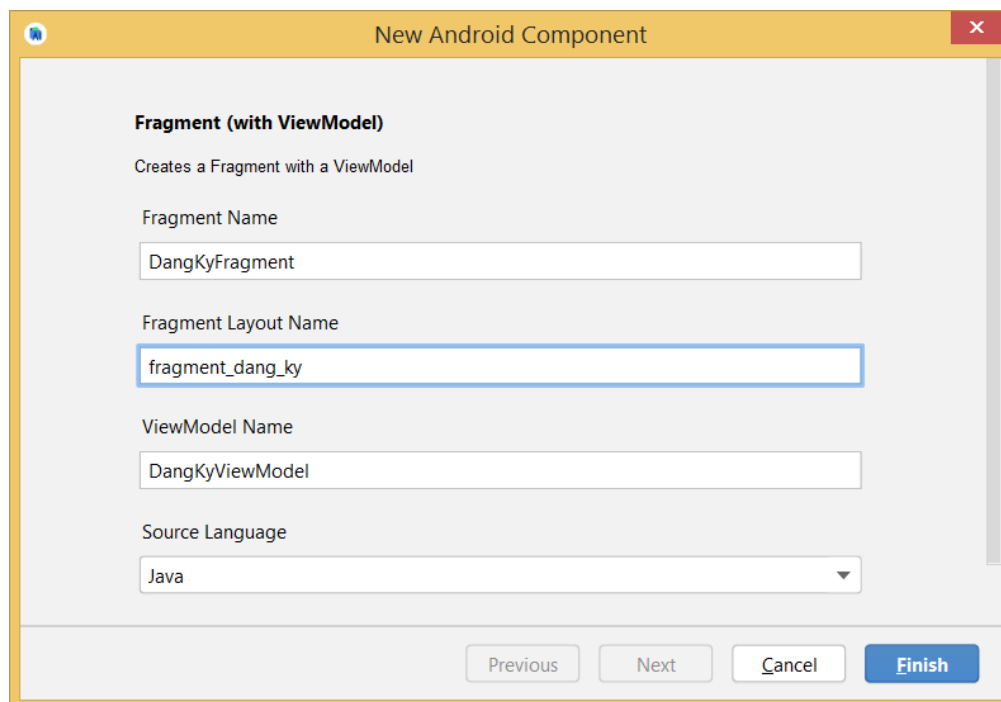
**Lưu ý:** Android dựa trên tên các tập tin giao diện để đặt tên các đối tượng Binding nên phải sử dụng theo đúng quy ước. Ví dụ: **fragment\_gioi\_thieu.xml** → **FragmentGioiThieuBinding**.

- + Trong **ui**, thực hiện đổi tên (Refactor → Rename) các đối tượng theo từng chức năng
  - ✓ home → gioithieu
  - ✓ HomeViewModel → GioiThieuViewModel
  - ✓ HomeFragment → GioiThieuFragment. Mã lệnh Java của GioiThieuFragment được điều chỉnh lại như sau (chú ý các đối tượng tô màu):

```
public class GioiThieuFragment extends Fragment {  
  
    private FragmentGioiThieuBinding binding;  
  
    public View onCreateView(@NonNull LayoutInflater inflater,  
                             ViewGroup container, Bundle savedInstanceState) {  
        GioiThieuViewModel gioiThieuViewModel;  
        gioiThieuViewModel = new ViewModelProvider(this).  
                                get(GioiThieuViewModel.class);  
        binding = FragmentGioiThieuBinding.inflate(inflater,  
                                                    container, false);  
        View root = binding.getRoot();  
        final TextView textView = binding.textHome;  
        gioiThieuViewModel.getText().observe(  
            getViewLifecycleOwner(), textView::setText);  
        return root;  
    }  
}
```

```
@Override
public void onDestroyView() {
    super.onDestroyView();
    binding = null;
}
}
```

- ✓ gallery → hinhanh
  - ✓ GalleryViewModel → HinhAnhViewModel
  - ✓ GalleryFragment → HinhAnhFragment. Điều chỉnh mã lệnh Java của HinhAnhFragment tương tự GioiThieuFragment.
  - ✓ slideshow → lienhe
  - ✓ SlideShowViewModel → LienHeViewModel
  - ✓ SlideShowFragment → LienHeFragment. Điều chỉnh mã lệnh Java của LienHeFragment tương tự GioiThieuFragment.
  - ✓ Thực thi kiểm tra kết quả.
- **Bước 5: tạo các thành phần cho chức năng đăng ký**
- + Trong **ui**, tạo Package mới với tên dangky
  - + Trong ui/dangky, tạo Fragment mới, chọn mẫu Fragment (with ViewModel). Đặt tên Fragment là DangKyFragment, Layout là fragment\_dang\_ky, ViewModel là DangKyViewModel như Hình 5.



**Hình 5. Tạo mới Fragment (with ViewModel)**

– **Bước 5: điều chỉnh các tập tin XML cho thực đơn điều hướng**

- + Tập tin thực đơn điều hướng res/layout/nav\_header\_main.xml: điều chỉnh hình ảnh và nội dung hiển thị (các vị trí tô màu) như sau:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="@dimen/nav_header_height"
    android:background="@drawable/side_nav_bar"
    android:gravity="bottom"
    android:orientation="vertical"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:theme="@style/ThemeOverlay.AppCompat.Dark">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:contentDescription="@string/nav_header_desc"
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        app:srcCompat="@mipmap/globelogo_small" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        android:text="Du lịch Đà Lạt"
        android:textAppearance="@style/TextAppearance.AppCompat.Body1"/>

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="dalattourist@gmail.com" />
</LinearLayout>
```

- + Điều chỉnh tập tin thực đơn res/menu/activity\_main\_drawer.xml: thêm thực đơn cho chức năng “Đăng ký”; điều chỉnh tên, tiêu đề, và icon cho các thực đơn điều hướng như sau:

**Lưu ý:** thực thi kiểm tra thực đơn chức năng sẽ không thực hiện đúng do liên quan đến tập tin điều hướng res/navigation/mobile\_navigation.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:tools="http://schemas.android.com/tools"
      tools:showIn="navigation_view">

    <group android:checkableBehavior="single">
        <item
            android:id="@+id/nav_gioithieu"
            android:icon="@drawable/ic_menu_share"
            android:title="Giới thiệu" />
        <item
            android:id="@+id/nav_hinhanh"
            android:icon="@drawable/ic_menu_gallery"
            android:title="Hình ảnh" />
        <item
            android:id="@+id/nav_dangky"
            android:icon="@drawable/ic_menu_send"
            android:title="Đăng ký" />
        <item
            android:id="@+id/nav_lienhe"
            android:icon="@drawable/ic_menu_contacts"
            android:title="Liên hệ" />
    </group>
</menu>
```

- + Điều chỉnh tập tin điều hướng res/navigation/mobile\_navigation như sau:

```
<?xml version="1.0" encoding="utf-8"?>
<navigation
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/mobile_navigation"
    app:startDestination="@+id/nav_gioithieu">
```



```
<fragment
    android:id="@+id/nav_gioithieu"
    android:name="thud.dulichdalat.ui.gioithieu.GioiThieuFragment"
    android:label="Giới thiệu"
    tools:layout="@layout/fragment_gioi_thieu" />

<fragment
    android:id="@+id/nav_hinhanh"
    android:name="thud.dulichdalat.ui.hinhanh.HinhAnhFragment"
    android:label="Hình ảnh"
    tools:layout="@layout/fragment_hinh_anh" />

<fragment
    android:id="@+id/nav_dangky"
    android:name="thud.dulichdalat.ui.dangky.DangKyFragment"
    android:label="Đăng ký"
    tools:layout="@layout/fragment_dang_ky" />

<fragment
    android:id="@+id/nav_lienhe"
    android:name="thud.dulichdalat.ui.lienhe.LienHeFragment"
    android:label="Liên hệ"
    tools:layout="@layout/fragment_lien_he" />
</navigation>
```

– **Bước 6: điều chỉnh giao diện và mã lệnh Java cho MainActivity**

+ **Điều chỉnh các tập tin giao diện của MainActivity**

- ✓ Tập tin giao diện res/layout/activity\_main.xml đã điều chỉnh ở Bước 2,
- ✓ Tập tin giao diện res/layout/content\_main.xml không điều chỉnh,
- ✓ Xóa bỏ tập tin thực đơn res/menu/main.xml (không sử dụng),
- ✓ Điều chỉnh tập tin giao diện res/layout/app\_bar\_main.xml (xóa bỏ phần khai báo FloatingActionButton) như sau:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

```
<com.google.android.material.appbar.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/Theme.DuLichDaLat.AppBarOverlay">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        app:popupTheme="@style/Theme.DuLichDaLat.PopupOverlay" />
    </com.google.android.material.appbar.AppBarLayout>

    <include layout="@layout/content_main" />
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

+ **Điều chỉnh tập tin mã lệnh Java của MainActivity**

- ✓ Xóa bỏ mã lệnh liên quan FloatingActionButton đã xóa bỏ ở trên,
- ✓ Điều chỉnh mã lệnh hỗ trợ thực đơn điều hướng các chức năng: chuyển khai báo NavController lên trên và sử dụng **phương thức getGraph() để nhận thực đơn thay cho khai báo chi tiết**,
- ✓ Xóa bỏ phương thức ghi đè (onCreateOptionsMenu) không sử dụng trong ứng dụng.

```
public class MainActivity extends AppCompatActivity {

    private AppBarConfiguration mAppBarConfiguration;
    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        setSupportActionBar(binding.appBarMain.toolbar);
        DrawerLayout drawer = binding.drawerLayout;
        NavigationView navigationView = binding.navView;

        NavController navController = Navigation.findNavController(
            this, R.id.nav_host_fragment_content_main);
```

```
mAppBarConfiguration = new AppBarConfiguration.Builder(  
    navController.getGraph()  
    .setOpenableLayout(drawer)  
    .build();  
  
NavigationUI.setupActionBarWithNavController(this,  
    navController, mAppBarConfiguration);  
NavigationUI.setupWithNavController(navigationView, navController);  
}  
  
@Override  
public boolean onSupportNavigateUp() {  
    NavController navController = Navigation.findNavController(  
        this, R.id.nav_host_fragment_content_main);  
    return NavigationUI.navigateUp(navController, mAppBarConfiguration)  
        || super.onSupportNavigateUp();  
}  
}
```

– **Bước 6: điều chỉnh giao diện và mã lệnh Java cho chức năng Giới thiệu**

- + Tập tin giao diện res/layout/fragment\_gioi\_thieu.xml điều chỉnh như sau:

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
    tools:context=".ui.gioithieu.GioiThieuFragment">  
  
    <TextView  
        android:id="@+id/txt_gioithieu"  
        style="@style/LabelStyle"  
        android:layout_margin="3dp" />  
</LinearLayout>
```

- + Điều chỉnh mã lệnh Java cho GioiThieuViewModel: khai báo và gán nội dung giới thiệu Đà Lạt như sau:

```
public class GioiThieuViewModel extends ViewModel {

    private MutableLiveData<String> mText;

    public GioiThieuViewModel() {
        mText = new MutableLiveData<>();
        String strGioiThieu ="Đà Lạt từ lâu đã trở thành một trong
những thành phố du lịch nổi tiếng nhất của Việt Nam, với nhiều cảnh
quan thiên nhiên đẹp và thơ mộng. Có ai đó đã từng ví Đà Lạt cũng
giống một thành phố trong rừng, nơi khí trời se lạnh, thiên nhiên và
con người gần gũi như luôn hoà quyện, đan xen vào nhau.";

        strGioiThieu += "\n\nAi đến Đà Lạt đều phải thốt lên rằng:
Ôi! Sao Đà Lạt đẹp đến thế! Đẹp mê hồn người lữ khách, đẹp cả trong
tim người chưa đến Đà Lạt bao giờ. Phải chăng người ta yêu Đà Lạt
bởi những dòng thác cuộn trào, ảo ảnh sương khói của hương hoa, của
nước Hồ Xuân Hương dịu mát.";

        strGioiThieu += "\n\nNúi Lang Biang nằm trên địa bàn huyện
Lạc Dương, cách trung tâm thành phố Đà Lạt 12km về phía bắc. Núi
Lang Biang còn được gọi là Núi Mẹ, gồm 2 ngọn, có độ cao 2.167m.";

        mText.setValue(strGioiThieu);
    }

    public LiveData<String> getText() {
        return mText;
    }
}
```

- + Điều chỉnh mã lệnh Java cho GioiThieuFragment như bên dưới,
- + Thực thi kiểm tra kết quả.

```
public class GioiThieuFragment extends Fragment {

    private FragmentGioiThieuBinding binding;

    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState) {
        GioiThieuViewModel gioiThieuViewModel = new
        ViewModelProvider(this).get(GioiThieuViewModel.class);
        binding = FragmentGioiThieuBinding.inflate(inflater,
            container, false);
    }
}
```

```

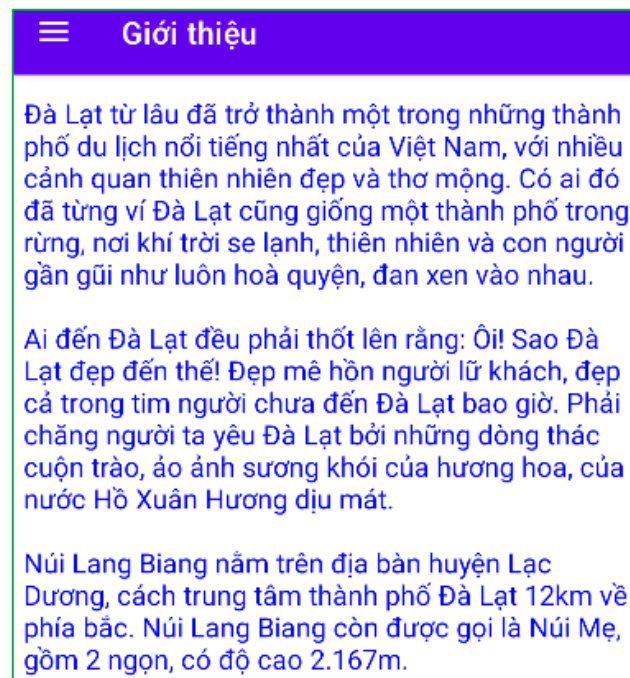
View root = binding.getRoot();

final TextView textView = binding.txtGioithieu;
gioiThieuViewModel.getText().observe(
    getViewLifecycleOwner(), textView::setText);

return root;
}

@Override
public void onDestroyView() {
    super.onDestroyView();
    binding = null;
}
}

```



**Hình 6. Kiểm thử chức năng Giới thiệu**

**Bước 6: điều chỉnh giao diện và mã lệnh Java cho chức năng Liên hệ**

- + Tập tin giao diện res/layout/fragment\_lien\_he.xml điều chỉnh gồm 1 TextView và 1 Button như sau:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"

```

```
        android:layout_height="match_parent"
        android:orientation="vertical"
        tools:context=".ui.lienhe.LienHeFragment">

        <TextView
            android:id="@+id/txt_lienhe"
            style="@style/LabelStyle"
            android:layout_margin="3dp" />

        <Button
            android:id="@+id/btn_trove"
            style="@style/ButtonStyle"
            android:layout_gravity="center"
            android:text="Trở về" />
    </LinearLayout>
```

- + Điều chỉnh mã lệnh Java cho LienHeViewModel như sau:

```
public class LienHeViewModel extends ViewModel {

    private MutableLiveData<String> mText;

    public LienHeViewModel() {
        mText = new MutableLiveData<>();
        String strLienHe = "Du lịch Đà Lạt";
        strLienHe += "\n\nEmail: dalattourist@gmail.com";
        mText.setValue(strLienHe);
    }

    public LiveData<String> getText() {
        return mText;
    }
}
```

- + Điều chỉnh mã lệnh Java cho LienHeFragment: điều chỉnh đúng tên các đối tượng và thêm mã lệnh xử lý nút Trở về như sau:

```
public class LienHeFragment extends Fragment {

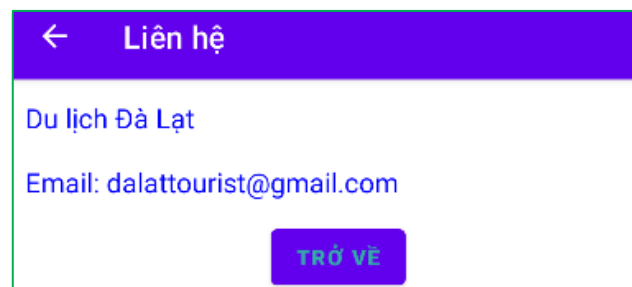
    private FragmentLienHeBinding binding;

    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState) {
        LienHeViewModel lienHeViewModel = new
            ViewModelProvider(this).get(LienHeViewModel.class);
    }
}
```

```
        binding = FragmentLienHeBinding.inflate(inflater,
            container, false);
        View root = binding.getRoot();
        final TextView textView = binding.txtLienhe;
        lienHeViewModel.getText().observe(
            getViewLifecycleOwner(), textView::setText);
        final Button button = binding.btnTroVe;
        button.setOnClickListener(new TroVe());
        return root;
    }

    private class TroVe implements View.OnClickListener {
        @Override
        public void onClick(View v) {
            getActivity().getOnBackPressedDispatcher().onBackPressed();
        }
    }

    @Override
    public void onDestroyView () {
        super.onDestroyView();
        binding = null;
    }
}
```



Hình 7. Kiểm thử chức năng Liên hệ

– **Bước 6: điều chỉnh giao diện và mã lệnh Java cho chức năng Hình ảnh**

- + Tập tin giao diện res/layout/fragment\_hinh\_anh.xml điều chỉnh gồm 1 GridView như sau:

```
<?xml version="1.0" encoding="utf-8"?>
<GridView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
```

```
android:id="@+id/gdv_hinhanh"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:numColumns="2"
android:verticalSpacing="10dp"
android:horizontalSpacing="10dp"
android:stretchMode="columnWidth"
android:gravity="center"
tools:context=".ui.hinhanh.HinhAnhFragment" />
```

- + Trong thud.dulichdalat, tạo thêm Package bus (Business Logic)
- + Trong bus, tạo lớp ImageAdapter kế thừa BaseAdapter cho phép hiển thị hình ảnh trong GridView như sau:

```
public class ImageAdapter extends BaseAdapter {

    private Context mContext;
    private Integer[] Images;

    public ImageAdapter(Context mContext, Integer[] images) {
        this.mContext = mContext;
        Images = images;
    }

    public void setImages(Integer[] images) {
        Images = images;
        notifyDataSetChanged();
    }

    @Override
    public int getCount() {
        return Images.length;
    }

    @Override
    public Object getItem(int position) {
        return Images[position];
    }

    @Override
    public long getItemId(int position) {
        return position;
    }

    @Override
```



```
public View getView(int i, View convertView, ViewGroup parent) {
    ImageView imageView = new ImageView(mContext);
    imageView.setImageResource(Images[i]);
    imageView.setScaleType(ImageView.ScaleType.FIT_CENTER);
    imageView.setLayoutParams(new GridView.LayoutParams(360, 360));
    imageView.setPadding(10, 30, 10, 30);
    return imageView;
}
}
```

- + Điều chỉnh mã lệnh Java cho HìnhAnhViewModel: xóa bỏ lệnh ban đầu như sau:

```
public class HìnhAnhViewModel extends ViewModel {

}
```

- + Điều chỉnh mã lệnh Java cho HìnhAnhFragment: khai báo mảng hình ảnh, nhận và gán ảnh hiển thị trong GridView như sau:

```
public class HìnhAnhFragment extends Fragment {

    private FragmentHìnhAnhBinding binding;
    private Integer[] Images = { R.mipmap.dalat01, R.mipmap.dalat02,
        R.mipmap.dalat03, R.mipmap.dalat04 };

    public View onCreateView(@NonNull LayoutInflater inflater,
        ViewGroup container, Bundle savedInstanceState) {
        HìnhAnhViewModel hìnhAnhViewModel = new
            ViewModelProvider(this).get(HìnhAnhViewModel.class);
        binding = FragmentHìnhAnhBinding.inflate(inflater,
            container, false);
        View root = binding.getRoot();
        final GridView gridView = binding.gdvHinhanh;
        gridView.setAdapter(new ImageAdapter(getActivity(), Images));
        return root;
    }

    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
}
```



**Hình 8. Kiểm thử chức năng Hình ảnh**

- **Bước 6: điều chỉnh giao diện và mã lệnh Java cho chức năng Đăng ký**
  - + Tập tin giao diện res/layout/fragment\_dang\_ky.xml điều chỉnh như bên dưới,
  - + Thực thi kiểm tra kết quả.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".ui.dangky.DangKyFragment">

    <com.google.android.material.textfield.TextInputLayout
        android:id="@+id/layout_hotenkh"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="7dp"
        app:endIconMode="clear_text" >

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/edt_hotenkh"
            style="@style/TextStyle"
            android:hint="Nhập họ tên khách hàng"
            android:background="@color/white"
            android:textColor="#0000FF" />
    </com.google.android.material.textfield.TextInputLayout>
```

```
<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/layout_emailkh"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="7dp"
    app:endIconMode="clear_text" >

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/edt_emailkh"
        style="@style/TextStyle"
        android:inputType="textEmailAddress"
        android:hint="Nhập địa chỉ Email"
        android:background="@color/white"
        android:textColor="#0000FF" />
</com.google.android.material.textfield.TextInputLayout>

<Spinner
    android:id="@+id/spn_loaitour"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="7dp" />

<EditText
    android:id="@+id/edt_dongia"
    style="@style/TextStyle"
    android:enabled="false"
    android:hint="Đơn giá"
    android:textColor="#0000FF" />

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/layout_sokhach"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="7dp"
    app:endIconMode="clear_text" >

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/edt_sokhach"
        style="@style/TextStyle"
        android:hint="Số khách"
        android:inputType="number"
        android:background="@color/white"
        android:textColor="#0000FF" />
</com.google.android.material.textfield.TextInputLayout>
```

```
<RadioGroup
    android:id="@+id/grp_thanhtoan"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <RadioButton
        android:id="@+id/rdo_tienmat"
        style="@style/LabelStyle"
        android:checked="true"
        android:text="Tiền mặt" />

    <RadioButton
        android:id="@+id/rdo_chuyenkhoan"
        style="@style/LabelStyle"
        android:checked="false"
        android:text="Chuyển khoản" />
</RadioGroup>

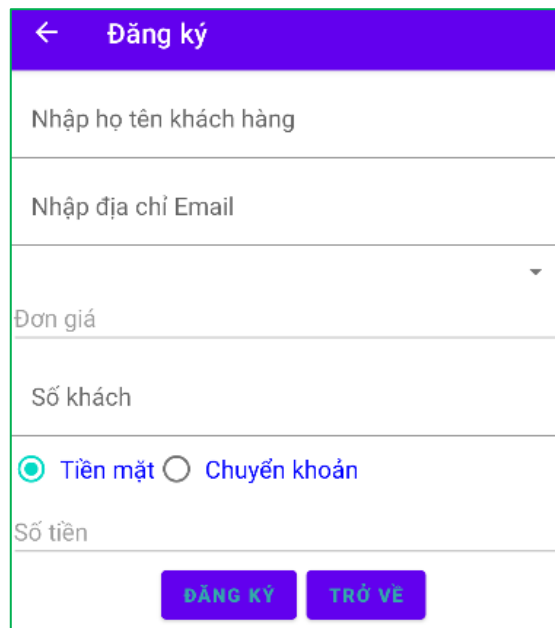
<EditText
    android:id="@+id/edt_sotien"
    style="@style/TextStyle"
    android:enabled="false"
    android:hint="Số tiền"
    android:textColor="#0000FF" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="horizontal" >

    <Button
        android:id="@+id/btn_dangky"
        style="@style/ButtonStyle"
        android:text="Đăng ký" />

    <Button
        android:id="@+id/btn_trove"
        style="@style/ButtonStyle"
        android:text="Trở về" />
</LinearLayout>

</LinearLayout>
```



**Hình 9. Giao diện Đăng ký khi chưa điều chỉnh mã lệnh Java**

- + Điều chỉnh mã lệnh Java cho DangKyViewModel như sau:

```
public class DangKyViewModel extends ViewModel {  
  
}
```

- + **Điều chỉnh mã lệnh Java cho DangKyFragment như sau:**
- ✓ Khai báo các biến hỗ trợ,
  - ✓ Trong sự kiện onCreate(), thực hiện:
    - Nhận các đối tượng,
    - Khai báo nguồn dữ liệu, Adapter, và đăng ký lắng nghe sự kiện chọn phần tử cho Spinner (lưu ý cách nhận trợ giúp khi gõ mã lệnh),
    - Đăng ký lắng nghe sự kiện xử lý cho nút lệnh Đăng ký,
    - Đăng ký lắng nghe sự kiện xử lý cho nút lệnh Trở về.
  - ✓ Thêm mã lệnh cho lớp cục bộ ChonLoaiTour xử lý sự kiện chọn phần tử cho Spinner,
  - ✓ Thêm phương thức GoiEmail: thực hiện gửi Email sử dụng Intent không tường minh (Implicit Intents).
  - ✓ Thêm mã lệnh cho lớp cục bộ DangKy xử lý sự kiện chọn Button “Đăng ký”,
  - ✓ Thêm mã lệnh cho lớp cục bộ QuayVe xử lý sự kiện chọn Button “Trở về”,
  - ✓ Thực thi kiểm tra kết quả.

```

public class DangKyFragment extends Fragment {

    private FragmentDangKyBinding binding;
    private View root;

    String arrLoaiTour[] = {"Chinh Phục đỉnh Langbiang", "Du lịch nhà vườn", "Thành phố tình yêu", "Giao lưu Văn hóa Cồng chiêng", "Nam Thiên đệ nhất thác"};
    int arrDonGia[] = {220, 280, 170, 150, 350};
    Spinner spnLoaiTour;
    String strKhachHang, strEmail, strLoaiTour, strThanhToan;
    int dongia, sokhach, sotien;
    TextInputLayout layoutKhachHang, layoutEmail, layoutSoKhach;
    TextInputEditText edtKhachHang, edtEmail, edtSoKhach;
    EditText edtDonGia, edtSoTien;

    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
                             @Nullable ViewGroup container,
                             @Nullable Bundle savedInstanceState) {
        DangKyViewModel dangKyViewModel = new
            ViewModelProvider(this).get(DangKyViewModel.class);
        binding = FragmentDangKyBinding.inflate(inflater,
            container, false);
        root = binding.getRoot();
        edtKhachHang = binding.edtHotenkh;
        edtEmail = binding.edtEmailkh;
        edtDonGia = binding.edtDongia;
        edtSoKhach = binding.edtSokhach;
        edtSoTien = binding.edtSotien;
        layoutKhachHang = binding.layoutHotenkh;
        layoutEmail = binding.layoutEmailkh;
        layoutSoKhach = binding.layoutSokhach;
        spnLoaiTour = binding.spnLoaitour;
        ArrayAdapter<String> adapter = new ArrayAdapter<String>(
            getActivity(), android.R.layout.simple_spinner_item,
            arrLoaiTour);
        adapter.setDropDownViewResource(
            android.R.layout.simple_list_item_single_choice);
        spnLoaiTour.setAdapter(adapter);
        spnLoaiTour.setOnItemClickListener(new ChonLoaiTour());
        Button btnDangKy = binding.btnDangky;
        btnDangKy.setOnClickListener(new DangKyTour());
    }
}

```

```

        Button btnTroVe = binding.btnTroVe;
        btnTroVe.setOnClickListener(new QuayVe());
        return root;
    }

    private class ChonLoaiTour implements
        android.widget.AdapterView.OnItemClickListener {
        @Override
        public void onItemClick(AdapterView<?> parent, View view,
            int i, long id) {
            strLoaiTour = arrLoaiTour[i];
            dongia = arrDonGia[i];
            edtDonGia.setText("Đơn giá: " + dongia);
        }

        @Override
        public void onNothingSelected(AdapterView<?> parent) {
            strLoaiTour = "";
            dongia = 0;
        }
    }

    private void GoiEmail(String[] address, String subject,
        String body) {
        Intent intent = new Intent(Intent.ACTION_SENDTO);
        intent.setType("message/rfc822");
        intent.setData(Uri.parse("mailto:"));
        intent.putExtra(Intent.EXTRA_EMAIL, address);
        intent.putExtra(Intent.EXTRA_SUBJECT, subject);
        intent.putExtra(Intent.EXTRA_TEXT, body);
        if (intent.resolveActivity(
            getActivity().getPackageManager()) != null)
            startActivity(Intent.createChooser(intent,
                "Chọn ứng dụng gửi Email:"));
        else
            Toast.makeText(getActivity(), "Lỗi thực hiện gửi Email",
                Toast.LENGTH_LONG).show();
    }

    private class DangKyTour implements View.OnClickListener {
        @Override
        public void onClick(View v) {
            strKhachHang = edtKhachHang.getText().toString().trim();

```

```

        if(strKhachHang.length() == 0){
            layoutKhachHang.setError("Lỗi chưa nhập họ tên khách hàng");
            edtKhachHang.requestFocus();
            return;
        }
        else
            layoutKhachHang.setError(null);
        strEmail = edtEmail.getText().toString().trim();
        if(strEmail.length() == 0){
            layoutEmail.setError("Lỗi chưa nhập địa chỉ Email");
            edtEmail.requestFocus();
            return;
        }
        else
            layoutEmail.setError(null);
        String strSoKhach = edtSoKhach.getText().toString().trim();
        if(strSoKhach.length() == 0 ||
            Integer.parseInt(strSoKhach) == 0){
            layoutSoKhach.setError("Lỗi nhập số khách đặt Tour");
            edtSoKhach.requestFocus();
            return;
        }
        else {
            layoutSoKhach.setError(null);
            sokhach = Integer.parseInt(strSoKhach);
        }
        sotien = sokhach * dongia;
        edtSoTien.setText("Số tiền: " + sotien);
        RadioGroup grpThanhToan = binding.grpThanhToan;
        int id = grpThanhToan.getCheckedRadioButtonId();
        RadioButton rad = root.findViewById(id);
        strThanhToan = rad.getText().toString();
        String strChuDe = "Thông tin đăng ký Tour du lịch";
        String strThongTin = "Họ tên khách hàng: " + strKhachHang;
        strThongTin += "\nLoại Tour: " + strLoaiTour;
        strThongTin += "\nĐơn giá: " + dongia;
        strThongTin += "\nSố khách: " + sokhach;
        strThongTin += "\nHình thức thanh toán: " + strThanhToan;
        strThongTin += "\nSố tiền: " + sotien;
        strThongTin += "\n\nChúc quý khách vui vẻ!";
        GoiEmail(new String[] { strEmail }, strChuDe, strThongTin);
    }
}

```



```
private class QuayVe implements View.OnClickListener {  
    @Override  
    public void onClick(View v) {  
        getActivity().getOnBackPressedDispatcher().onBackPressed();  
    }  
}
```

**Lưu ý:** khi thực thi trên máy ảo

- + Phải khai báo tài khoản Email cho thiết bị trong lần thực hiện đầu tiên,
- + Nếu đoạn Code gọi thực hiện Intent gửi Email (phần đã tô màu) không thực hiện thì có thể sử dụng đoạn Code thay thế như bên dưới:

```
try {  
    startActivity(Intent.createChooser(intent,  
                                     "Chọn ứng dụng gửi Email:"));  
} catch (ActivityNotFoundException err){  
    Toast.makeText(getActivity(), "Lỗi thực hiện gửi Email",  
                   Toast.LENGTH_LONG).show();  
}
```

← Đăng ký

Nhập họ tên khách hàng  
Ho Van Tu

Nhập địa chỉ Email  
hvtu@ctu.edu.vn

Du lịch nhà vườn ▾  
Đơn giá: 280

Số khách  
9

☐ Tiền mặt ☒ Chuyển khoản

Số tiền: 2520

ĐĂNG KÝ TRỞ VỀ

**Thông tin đăng ký Tour du lịch**

**Ho Van Tu**  
to me ▾

Họ tên khách hàng: Ho Van Tu  
Loại Tour: Du lịch nhà vườn  
Đơn giá: 280  
Số khách: 9  
Hình thức thanh toán: Chuyển khoản  
Số tiền: 2520

Chúc quý khách vui vẻ!

**Hình 10. Kiểm thử chức năng Đăng ký**