

# **ĐIỀU HƯỚNG NAVIGATION TRONG THIẾT KẾ GIAO DIỆN LẬP TRÌNH DI ĐỘNG ANDROID**

**Lê Minh Phát<sup>1,\*</sup>, Lê Tâm Như<sup>1</sup>, Ngô Thị Kim Ngân<sup>1</sup>, Nguyễn Thị Kim Tuyền<sup>1</sup>,  
Nguyễn Thị Bích Ngân<sup>1</sup>**

<sup>1</sup>*Trường Đại học Công nghiệp Thực phẩm Thành phố Hồ Chí Minh*

*\*Email: [lephat508@gmail.com](mailto:lephat508@gmail.com) ; [letamnhu6602@gmail.com](mailto:letamnhu6602@gmail.com) ; [ngothikimngan0509@gmail.com](mailto:ngothikimngan0509@gmail.com) ;  
[tuyenfb78@gmail.com](mailto:tuyenfb78@gmail.com) ; [nganntb@hufi.edu.vn](mailto:nganntb@hufi.edu.vn)*

Ngày nhận bài..... ; Ngày chấp nhận đăng: , 2023

## **TÓM TẮT**

Trong bài báo này, chúng tôi giới thiệu việc thiết kế và xử lý các điều hướng (navigation) trong quá trình thiết kế giao diện lập trình di động android. Bài báo nhấn mạnh tầm quan trọng cùng các lợi ích của navigation đối với trải nghiệm người dùng. Đồng thời, chúng tôi cung cấp một quy trình chi tiết cho việc xây dựng hai loại navigation phổ biến là bottom navigation và navigation drawer. Ngoài ra, chúng tôi thực hiện thực nghiệm thông qua việc dùng hai loại navigation nói trên vào trong thiết kế giao diện của quá trình phát triển ứng dụng di động android “Quảng bá địa điểm du lịch tại Việt Nam”. Kết quả cho thấy ứng dụng có một giao diện đẹp mắt, tiện lợi và mang đến cho người dùng trải nghiệm thú vị thông qua hiệu quả của các navigation mang lại.

*Từ khóa:* Navigation, bottom navigation, navigation drawer.

## **1. MỞ ĐẦU**

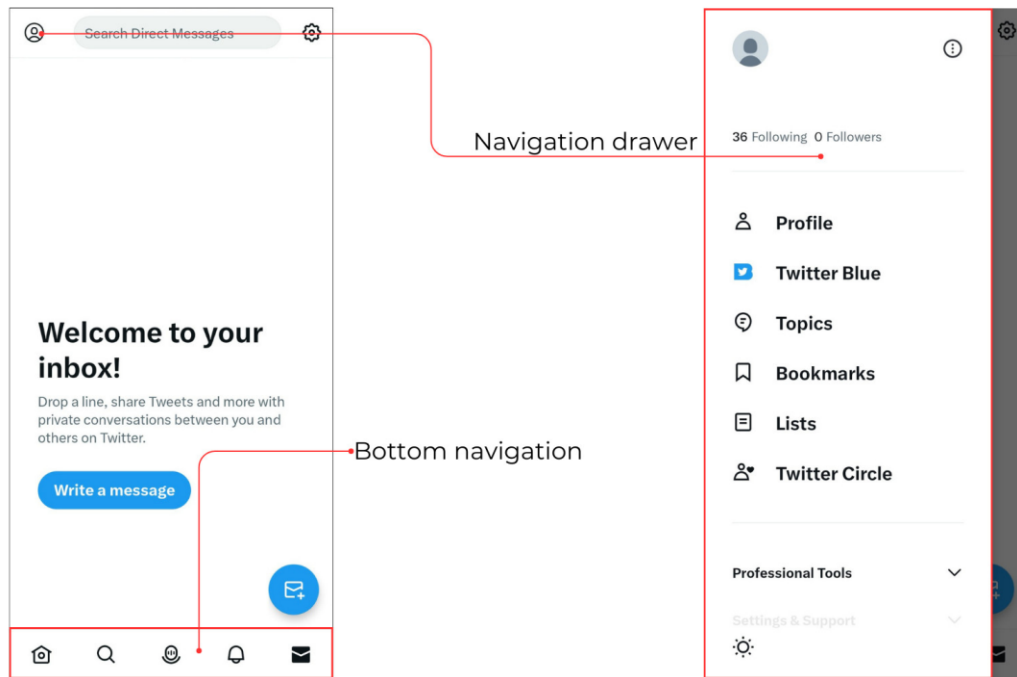
Navigation là những công cụ hỗ trợ quản lý và điều hướng màn hình trong lập trình ứng dụng di động nói chung, riêng trong phạm vi bài báo này chúng tôi tập trung vào lập trình di động trên hệ điều hành android. Navigation trở thành một trong những phương pháp được sử dụng phổ biến. Với việc dùng navigation, người dùng có thể chuyển đổi giữa các màn hình và hoạt động trong ứng dụng một cách hiệu quả, và “mượt mà”. Sử dụng navigation đã trở thành một xu hướng thiết kế phổ biến trong các ứng dụng di động hiện nay. Tính đồng nhất giữa các màn hình trong ứng dụng được cải thiện và trải nghiệm người dùng được tối ưu hóa nhờ các thành phần điều hướng này [1].

Hiện nay, việc sử dụng navigation giúp giảm thời gian và chi phí phát triển ứng dụng. Nhà phát triển không cần phải viết mã cho từng màn hình và hoạt động riêng lẻ. Đồng thời, navigation giúp tăng tính linh hoạt và sự dễ dàng trong quá trình bảo trì và cập nhật ứng dụng. Vì vậy, navigation được xem là một bước tiến quan trọng trong việc phát triển ứng dụng di động trên nền tảng android. Với tầm quan trọng của việc quản lý điều hướng trong ứng dụng, navigation đã trở thành một công cụ hỗ trợ quan trọng đối với nhà phát triển [1].

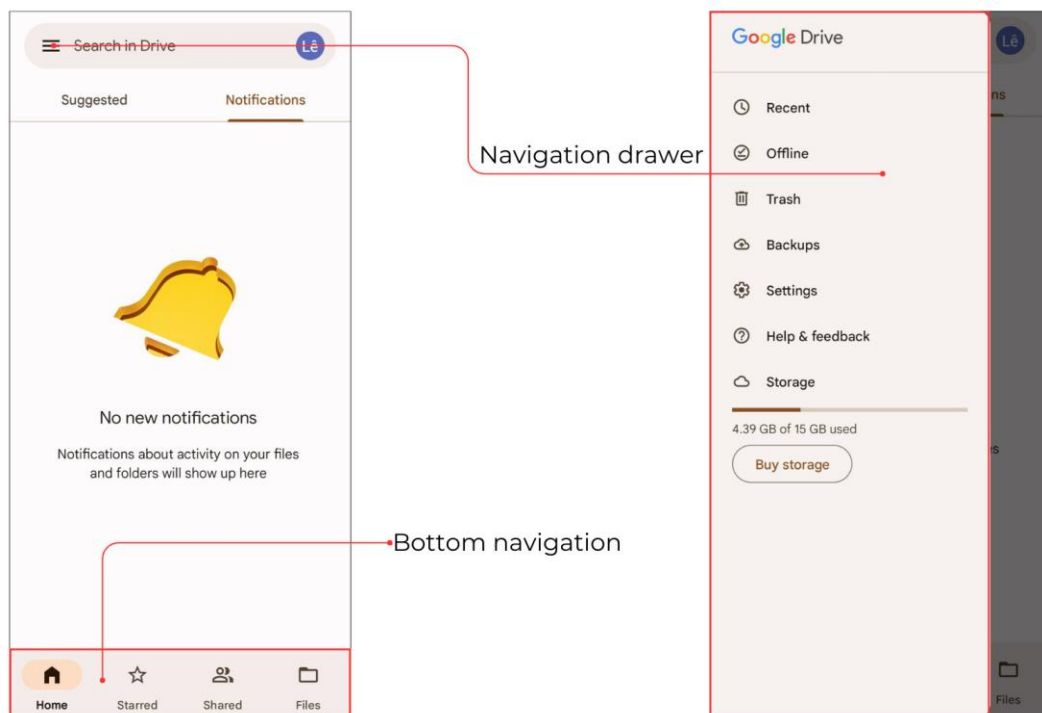
### **Navigation là gì?**

Navigation là công cụ hỗ trợ quản lý và điều hướng màn hình trong ứng dụng di động. Tính năng điều hướng là các hành động tương tác cho phép người dùng di chuyển qua lại, vào và ra khỏi các phần nội dung khác nhau trong ứng dụng [1]. Điều hướng giữa các màn hình là một trong những kỹ thuật cốt lõi trong kiến trúc ứng dụng. Cách tiếp cận và xử lý việc điều hướng giữa các màn hình sẽ ảnh hưởng đáng kể tới khả năng bảo trì ứng dụng sau này.

Trong phạm vi của bài báo này, chúng tôi tập trung giới thiệu hai loại navigation phổ biến là bottom navigation và navigation drawer. Các loại navigation này được sử dụng trong rất nhiều các ứng dụng di động phổ biến với người dùng, chẳng hạn: Twitter (hình 1), Gmail (hình 2), Lazada, Shopee, Facebook, Zalo, Google Map.



Hình 1. Bottom navigation và navigation drawer được dùng trong ứng dụng Twitter



Hình 2. Bottom navigation và navigation drawer được dùng trong ứng dụng Gmail

### Lợi ích:

Sử dụng navigation trong xây dựng ứng dụng android mang lại nhiều lợi ích cho nhà phát triển, giúp tăng tính năng dễ sử dụng và cải thiện trải nghiệm người dùng. Cụ thể, các lợi ích của sử dụng navigation bao gồm [3]:

*Giảm thời gian và công sức phát triển:* navigation giúp thiết kế và xây dựng ứng dụng nhanh chóng hơn, giảm thời gian và công sức của nhà phát triển.

*Tính linh hoạt và dễ quản lý:* navigation giúp quản lý màn hình dễ dàng hơn bằng cách kết nối chúng với nhau nhờ navigation graphs, cung cấp tính linh hoạt và dễ quản lý hơn.

*Trải nghiệm người dùng tốt hơn:* navigation tạo ra trải nghiệm người dùng tốt hơn, giúp người dùng tìm kiếm và đưa ra quyết định một cách nhanh chóng và dễ dàng.

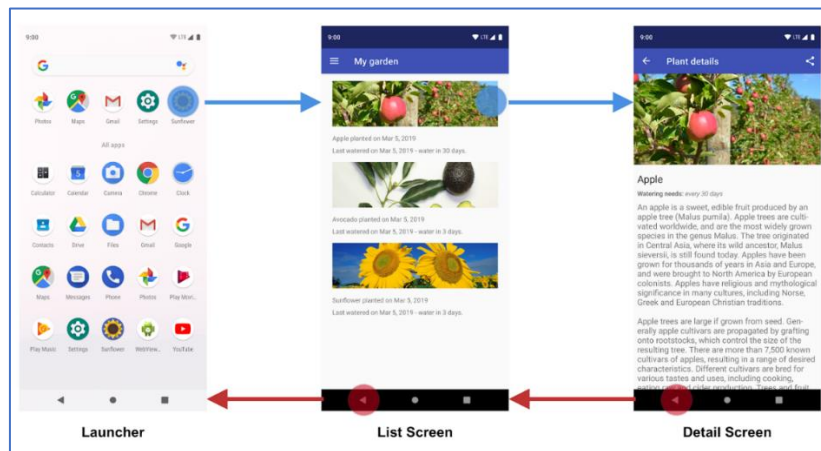
*Hiệu suất tốt hơn:* navigation sử dụng các kỹ thuật tối ưu để cải thiện hiệu suất ứng dụng, đảm bảo tiêu thụ ít tài nguyên hệ thống và tốc độ chuyển đổi màn hình nhanh chóng.

*Phù hợp với nhiều kiểu điều hướng:* navigation hỗ trợ nhiều kiểu điều hướng khác nhau như điều hướng bằng button, drawer và bottom navigation bar.

*Hỗ trợ cho việc mở rộng ứng dụng:* navigation giúp việc mở rộng ứng dụng dễ dàng hơn bằng cách giúp nhà phát triển thêm màn hình mới một cách nhanh chóng và dễ dàng.

### **Nguyên tắc điều hướng quan trọng:**

Theo trang tài liệu chính thức của Google về lập trình Android [2], việc điều hướng giữa các màn hình và ứng dụng là một phần cốt lõi trong trải nghiệm người dùng. Do đó, các nguyên tắc về điều hướng phải được tuân thủ để đảm bảo tính nhất quán và trực quan trong các ứng dụng. Thành phần navigation được thiết kế để triển khai theo các nguyên tắc mặc định, giúp đảm bảo rằng người dùng sẽ áp dụng được mô hình tương tự khi di chuyển giữa các ứng dụng.



Hình 3. List Screen (màn hình danh sách) là điểm bắt đầu của ứng dụng được trích xuất từ website Android Developers

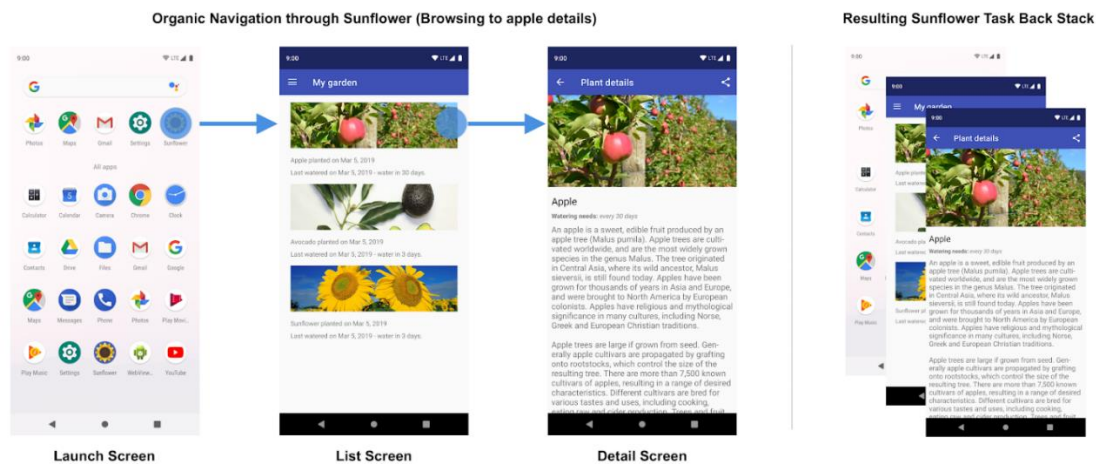
Việc triển khai navigation với Màn hình danh sách (List Screen) làm màn hình đầu tiên được hiển thị cho người dùng khi khởi chạy ứng dụng và cũng là màn hình cuối cùng được hiển thị trước khi người dùng thoát khỏi ứng dụng đáp ứng nguyên tắc Điểm bắt đầu cố định (Fixed start destination) [3].

Một nguyên tắc điều hướng khác cần đảm bảo đó là Trạng thái điều hướng được thể hiện dưới dạng một ngăn xếp các đích đến [3], nơi mà màn hình bắt đầu của tác vụ được hiển thị làm cơ sở cho trạng thái điều hướng của ứng dụng. Khi chạy ứng dụng lần đầu, một tác vụ mới được tạo ra và điểm bắt đầu của tác vụ này sẽ được hiển thị. Ngăn xếp lui được sử dụng để lưu trữ lịch sử các đích đến mà người dùng đã truy cập, với đích đến mới nhất được đặt ở đầu ngăn xếp. Người dùng có thể thực hiện các thao tác như đẩy đích đến mới lên đầu ngăn xếp hoặc xóa đích đến trên cùng để thay đổi ngăn xếp lui. Khi người dùng chọn một đích đến nào đó, thì đích đến đó sẽ trở thành đích đến mới nhất và được đặt ở đầu ngăn xếp.

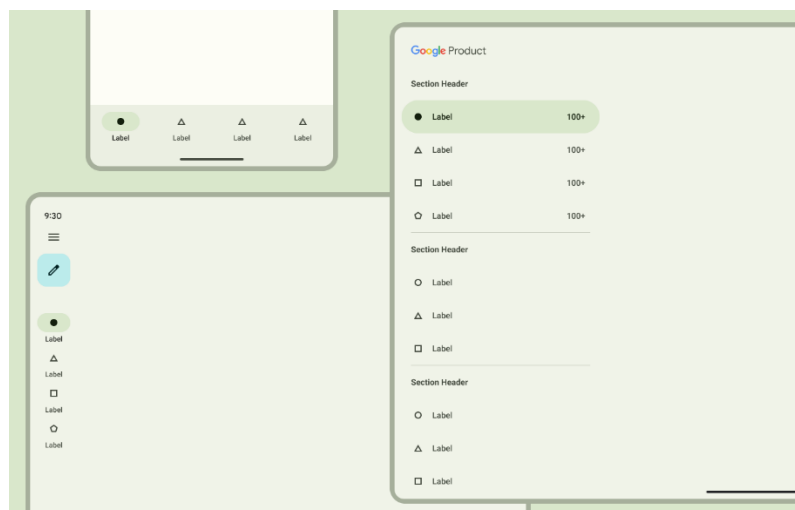
### **Xử lý các thay đổi về cấu hình**

Nếu nguyên tắc điều hướng của Android giúp nhóm phát triển xây dựng việc điều hướng trong ứng dụng một cách nhất quán và trực quan, thì giao diện người dùng thích ứng cung cấp đích đến nội dung thích ứng và thường bao gồm nhiều loại phần tử điều hướng để đáp ứng các thay đổi về kích thước hiển thị (hình 4). Ví dụ: thanh điều hướng ở cuối màn hình nhỏ, dải điều hướng trên màn hình trung bình hoặc ngăn điều hướng

cố định trên màn hình lớn, nhưng giao diện người dùng thích ứng vẫn phải tuân thủ các nguyên tắc điều hướng.



Hình 4. Điều hướng người dùng và kết quả là chuyển đến ngăn xếp lui được trích từ website Android Developers



Hình 5. Màn hình mở rộng, trung bình và nhỏ gọn có ngăn điều hướng, dải điều hướng và thanh dưới cùng màn hình trích từ website Material Design

Trong thiết kế theo nguyên tắc Material Design [4], phần điều hướng có kích thước ổn định nằm ở phía trên cùng và có thể di chuyển xuống phía dưới khi ứng dụng thu nhỏ. Trong giao diện người dùng thích ứng, phần điều hướng thường bao gồm nhiều loại phần tử để đáp ứng các thay đổi về kích thước hiển thị. Các lớp kích thước cửa sổ có thể giúp xác định và đặt phần điều hướng ở vị trí dễ tiếp cận nhất, và việc lựa chọn loại phần điều hướng phụ thuộc vào kích thước cửa sổ ứng dụng và số lượng mục cần chứa trong phần điều hướng. Trong thiết kế theo nguyên tắc Material Design, phần điều hướng có kích thước ổn định nằm ở phía trên cùng và có thể di chuyển xuống phía dưới khi ứng dụng thu nhỏ.

Việc sử dụng navigation không chỉ đáp ứng yêu cầu cho mở rộng phần hiển thị nội dung mà giao diện ứng dụng cũng phải mang đến trải nghiệm người dùng lý tưởng cho các kích thước cửa sổ nhất định, ngay cả khi sử dụng bộ cục và tiện ích hoàn toàn khác nhau.

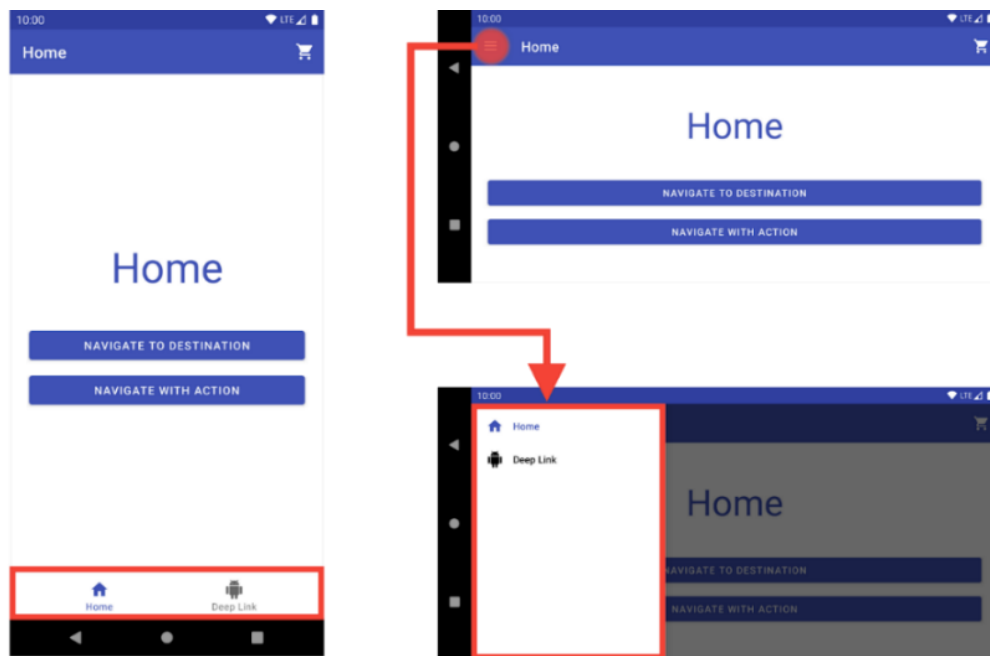
Trong phạm vi bài báo này, chúng tôi đề xuất triển khai điều hướng chung trong giao diện người dùng thích ứng.

Bảng 1. Các lớp kích thước và vị trí của thành phần điều hướng

Lớp kích thước cửa sổ	Một vài mục	Nhiều mục
Chiều rộng thu gọn	Thanh điều hướng ở dưới cùng	Ngăn điều hướng (cạnh trên hoặc cuối)
Chiều rộng trung bình	Ray điều hướng	Ngăn điều hướng (cạnh trên)
Chiều rộng được mở rộng	Dải điều hướng	Ngăn điều hướng cố định (cạnh trên)

Khi triển khai điều hướng chung trong giao diện người dùng thích ứng tập trung vào việc điều chỉnh các yếu tố của giao diện để phù hợp với các thay đổi về môi trường. Việc điều chỉnh chiều rộng, chiều cao, hướng và ngôn ngữ của giao diện đều có ảnh hưởng lớn đến trải nghiệm người dùng. Các thuộc tính môi trường này được gọi là cấu hình của thiết bị, và đối với việc triển khai navigation, các nhà phát triển cần chú ý đến cách bố trí các thành phần điều hướng để chúng phù hợp với cấu hình thiết bị [4].

Các kích thước và hướng thiết bị khác nhau sẽ được linh động chuyển đổi giữa navigation drawer hoặc bottom navigation tùy vào nhu cầu sử dụng. Điều này giúp đảm bảo rằng các thành phần điều hướng luôn được đặt ở vị trí dễ tiếp cận nhất cho người dùng, bất kể họ đang sử dụng thiết bị có kích thước và hướng khác nhau. Tuy nhiên, việc triển khai navigation cần phải thuận tiện và dễ sử dụng cho người dùng, bao gồm cả việc đảm bảo tính tương thích với các kích thước và hướng thiết bị khác nhau [6].



Hình 6. Thành phần điều hướng sử dụng bottom navigation và navigation drawer để hiển thị trình đơn điều hướng trên các thiết bị trích từ website Android Developers

Phần còn lại của bài báo được tổ chức như sau: Trong mục 2, chúng tôi sẽ mô tả chi tiết thành phần điều hướng navigation drawer và bottom navigation, mô tả quy trình xây dựng ứng dụng sử dụng thành phần điều hướng navigation drawer và bottom navigation. Cuối cùng, ở mục 3, chúng tôi sẽ trình bày kết luận các vấn đề đã đề cập và đề xuất hướng phát triển.

## 2. MỘT SỐ DẠNG NAVIGATION THƯỜNG DÙNG VÀ QUI TRÌNH KHỞI TẠO

### 2.1. Bottom navigation

### 2.1.1. Bottom navigation là gì ?

Bottom navigation là thanh điều hướng nằm ở phía dưới màn hình giúp dễ dàng chuyển đổi giữa các top - level views trong ứng dụng. bottom navigation được sử dụng khi ứng dụng có từ 3 đến 5 top-level views. Hình 7 thể hiện một ví dụ về bottom navigation [8][9].

**Lưu ý:** Bottom navigation sẽ không hỗ trợ từ 6 top-level views trở lên.

### 2.1.2. Quy trình khởi tạo

#### Bước 1:

Thêm bottomnavigation View widget vào file *activity\_main.xml*. Sau đó thêm *FrameLayout* để load Fragments khi navigation item click được chọn.

Code file *activity\_main.xml*:

```
<FrameLayout
    android:id="@+id/frame_container"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</FrameLayout>
<com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/navigation"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom"
    android:background="?android:attr/windowBackground"
    app:itemBackground="@color/bgBottomNavigation"
    android:foreground="?attr/selectableItemBackground"
    app:itemIconTint="@android:color/white"
    app:itemTextColor="@android:color/white"
    app:menu="@menu/navigation" />
```



Hình 7. Bottom navigation

#### Bước 2:

Bottom navigation items được rendered vào bằng việc sử dụng một file menu, hãy tạo một file xml mới đặt tên là *navigation.xml* trong thư mục *res/menu*.

Code file *navigation.xml*:

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/navigation_shop"
        android:icon="@drawable/ic_store_white_24dp"
        android:title="@string/title_shop" />
    <item
        android:id="@+id/navigation_gifts"
        android:icon="@drawable/ic_card_giftcard_white_24dp"
        android:title="@string/title_gifts" />
    <item
        android:id="@+id/navigation_cart"
        android:icon="@drawable/ic_shopping_cart_white_24dp"
```

```
        android:title="@string/title_cart" />
    <item
        android:id="@+id/navigation_profile"
        android:icon="@drawable/ic_person_white_24dp"
        android:title="@string/title_profile" />
</menu>
```

**Bước 3:** Tạo fragment mới: res/layout/New/Fragment/Fragment (Blank) đặt tên cho Fragment chọn Finish. Tương tự như vậy tạo thêm ba mảnh khác nhau và viết code xử lý cho từng Fragment.

**Bước 4:** Mở file MainActivity.java và tùy chỉnh theo mong muốn của nhà phát triển ứng dụng.

Code file *MainActivity.java*:

```
public class MainActivity extends AppCompatActivity {
    private ActionBar toolbar;
    Fragment f;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        toolbar = getSupportActionBar();
        toolbar.setBackgroundDrawable(new
ColorDrawable(getColor(R.color.bgBottomNavigation)));
        BottomNavigationView navigation = (BottomNavigationView)
findViewById(R.id.navigation);
        navigation.setOnNavigationItemSelectedListener
(mOnNavigationItemSelectedListener);
        toolbar.setTitle("Shop");
        f= new FirstFragment();
        loadFragment(f);

        ...
    }
    private void loadFragment(Fragment k)
    {
        FragmentTransaction tra =
getSupportFragmentManager().beginTransaction();
        tra.replace(R.id.frame_container, k);
        tra.addToBackStack(null);
        tra.commit();
    }
}
```

Trong ví dụ này *OnNavigationItemSelectedListener* sẽ được gọi khi bottom navigation item được chọn sẽ thay đổi tiêu đề toolbar và hiển thị fragment tương ứng vào trong *FrameLayout*.

## 2.2. Navigation drawer

### 2.2.1. Navigation drawer là gì?

Navigation drawer là một menu hiển thị như là một ngăn kéo ở cạnh bên của màn hình. Nó được ẩn đi khi không sử dụng nhưng ngăn này sẽ xuất hiện khi người dùng nhấn vào biểu tượng Hamburger button trong thanh ứng dụng hoặc khi người dùng vuốt từ mép trái màn hình. Hình 8 thể hiện một ví dụ về navigation drawer [7].

### 2.2.2. Quy trình khởi tạo

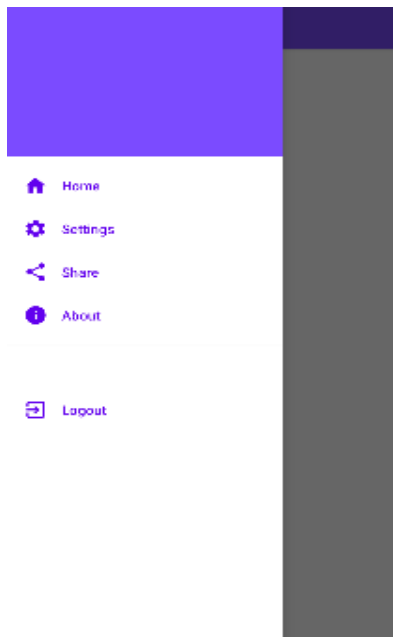


---

### Bước 1: Thêm drawer vào Layout

Để thêm được navigation drawer, cần khai báo DrawerLayout như một view gốc. Bên trong DrawerLayout, thêm view có chứa nội dung chính của layout (vd: FrameLayout,...) và một view khác chứa nội dung của navigation drawer.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
    android:id="@+id/drawer_layout"
    android:fitsSystemWindows="true"
    tools:openDrawer="start"
    tools:context=".MainActivity">
    <com.google.android.material.navigation.NavigationView
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:id="@+id/nav_view"
        android:layout_gravity="start"
        app:itemIconTint="@color/purple_500"
        app:itemTextColor="@color/purple_500"/>
</androidx.drawerlayout.widget.DrawerLayout>
```



Hình 8. Navigation drawer

### Lưu ý:

Khi sử dụng FrameLayout thì thiết lập match\_parent với cả chiều rộng và chiều cao. Vì nó sẽ hiển thị giao diện khi navigation drawer ẩn.

Về phần navigation View thì phải chỉnh thành ngang với thuộc tính android:layout\_gravity. Thiết lập giá trị với "start" để navigation drawer xuất hiện từ bên trái ra.

navigation View nên được thiết lập thuộc tính android:fitsSystemWindows bằng "true" để đảm bảo nội dung của navigation drawer không được phủ lên thanh trạng thái và các hệ thống khác.

### Bước 2: Khai báo các item menu cho navigation drawer

Để có thể cấu hình danh sách Menu Item trong navigation drawer thì sử dụng thuộc tính app:menu, Giống như code ở bên dưới:



```
<com.google.android.material.navigation.NavigationView
...
    app:menu="@menu/nav_menu"/>
```

Sau đó tạo menu tương ứng với tên: nav\_menu.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_home"
            android:icon="@drawable/ic_home"
            android:title="Home" />
        <item
            android:id="@+id/nav_settings"
            android:icon="@drawable/ic_settings"
            android:title="Settings" />
        </group>

        <item
            android:title="">
            <menu>
                <item
                    android:id="@+id/nav_exit"
                    android:icon="@drawable/ic_exit"
                    android:title="Logout"></item>
            </menu>
        </item>
    </menu>
```

Ngoài ra, có thể tạo nhóm với các item riêng biệt bằng cách sử dụng thuộc tính `android:checkableBehavior="single"` được thiết lập ở nhóm. Nó cho phép hiển thị danh sách các item có trong menu.

### **Bước 3:** Thêm Header cho navigation drawer

Có thể thêm Header tại phía trên của drawer bằng cách sử dụng thuộc tính `app:headerLayout` như code phía bên dưới:

```
<com.google.android.material.navigation.NavigationView
...
    app:headerLayout="@layout/nav_header"/>
```

Và sau tạo layout Header với tên: nav\_header.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="176dp"
    android:background="@color/colorPrimary"
    android:gravity="bottom"
```

```

android:padding="16dp"
android:theme="@style/Theme.AppCompat.Light.NoActionBar">

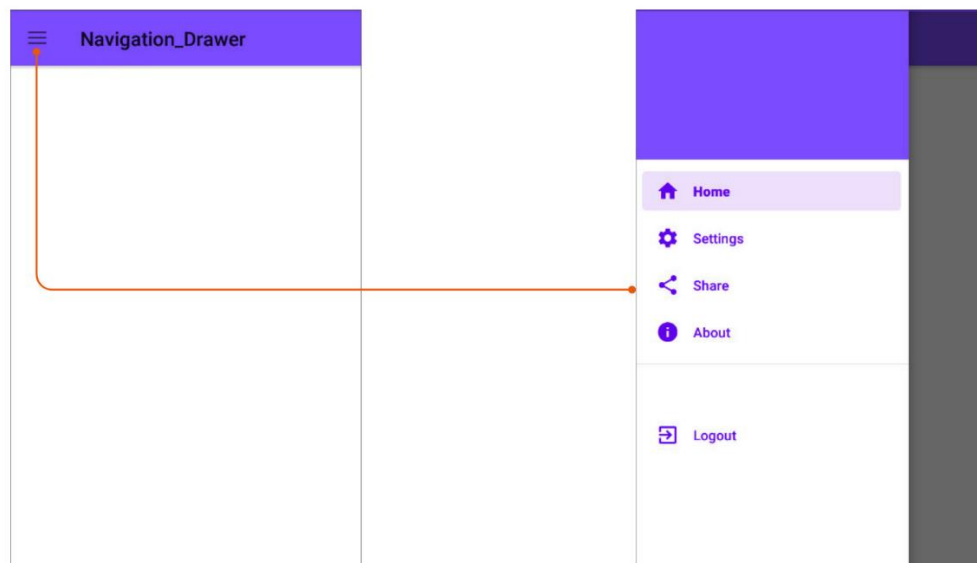
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"
    android:textColor="@color/white"
    android:textSize="14sp" />

</LinearLayout>

```

#### Bước 4: Thêm Toolbar

DrawerLayout đã cung cấp cho người dùng mở và đóng navigation drawer chỉ với việc vuốt bên mép của màn hình. Nhưng nếu giao diện bao gồm có cả ActionBar, nên cho phép người dùng đóng mở Drawer bằng cách nhấn vào Icon Drawer ở góc phía trên bên trái của ActionBar.



Hình 9. Click vào icon drawer trên thanh toolbar sẽ mở navigation drawer

Navigation drawer nên được xuất hiện ở phía trước ActionBar. Để có thể tạo được nó, thì cần phải sử dụng đến Toolbar. Để có thể thêm Toolbar vào layout chúng ta làm như code phía bên dưới:

```

<androidx.drawerlayout.widget.DrawerLayout
    ... >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <androidx.appcompat.widget.Toolbar
            android:layout_width="match_parent"
            android:layout_height="56dp"
            android:id="@+id/toolbar"
            android:elevation="4dp"
            android:background="@color/colorPrimary"

```

```
        android:theme="@style/Theme.AppCompat.Light.NoActionBar"
        android:popupTheme="@style/ThemeOverlay.AppCompat.Light" />

        <FrameLayout
            .../>

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

Sau đó vào folder themes thiết lập như sau:

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <style name="..."
        parent="Theme.MaterialComponents.DayNight.NoActionBar">
        ...
    </style>
</resources>
```

### **Thiết lập Toolbar trong MainActivity.java:**

Lúc này, Toolbar đã xuất hiện trong layout nhưng không có chức năng như ActionBar. Để có thể áp dụng Toolbar như ActionBar, đầu tiên phải chắc chắn rằng Activity đã được thừa kế từ AppCompatActivity. Sau đó gọi `setSupportActionBar()`; và truyền toolbar vào:

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
    }
}
```

Tuy nhiên, đến đây Icon Drawer vẫn chưa thể xuất hiện vì sẽ sử dụng toggle button trên Toolbar, thế nên sẽ tiếp tục thực hiện bước kế tiếp:

### **Bước 5: Xử lý Click sự kiện cho navigation drawer**

Để nhận sự kiện khi người sử dụng chạm vào của danh sách item trong DrawerLayout, thực thi câu lệnh `OnNavigationItemSelectedListener` và gắn nó tới `NavigationView` bằng cách gọi `setNavigationItemSelectedListener()`, như sau:

```
public class MainActivity extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener {
    private DrawerLayout drawerLayout;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        ...
        drawerLayout = findViewById(R.id.drawer_layout);
        NavigationView navigationView = findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);
    }
    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        switch (item.getItemId()) {
            case R.id.nav_home:
```

```

        //
        break;
    case R.id.nav_settings:
        //
        break;
    case R.id.nav_exit:
        //
        break;
    }
    drawerLayout.closeDrawer(GravityCompat.START);
    return true;
}
}

```

Khi mà item được chạm, đoạn code này sẽ thiết lập lựa chọn item và DrawerLayout cũng được đóng bởi hàm `closeDrawers()`.

Sử dụng `ActionBarDrawerToggle` để đồng bộ hóa trạng thái của `DrawerLayout` tương ứng với trạng thái của `ActionBar`, cung cấp giao diện điều khiển `DrawerLayout` để sử dụng và người dùng dễ dàng hiểu được chức năng của nó.

Khi khởi tạo `ActionBarDrawerToggle` yêu cầu truyền vào 5 tham số: `ActionBarDrawerToggle` (`this`, `drawerLayout`, `toolbar`, `R.string.open_Navigation`, `R.string.close_Navigation`):

- Activity hiện tại (`this`).
- `DrawerLayout` sẽ được hiển thị thông qua component navigation drawer (`drawerLayout` được khởi tạo trước đó).
- `Toolbar`: tổng quan trên giao diện người dùng cho Activity của bạn chứa các nút điều hướng, tiêu đề và các element khác.
- Tham số 4 và 5 chỉ định văn bản sẽ được hiển thị khi drawer được mở ra hoặc đóng lại.

Tiếp đến sẽ sử dụng `drawerLayout.addDrawerListener(toggle)` để đăng ký một `DrawerListener` với `DrawerLayout`, khi `DrawerLayout` thay đổi, `DrawerListener` sẽ được cập nhật và thực hiện một số xử lý. Và `DrawerListener` được dùng để lắng nghe sự kiện đóng hay mở `DrawerLayout` và thay đổi trạng thái.

```

public class MainActivity extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener {

    private DrawerLayout drawerLayout;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        ...
        drawerLayout = findViewById(R.id.drawer_layout);
        NavigationView navigationView = findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(this,
drawerLayout, toolbar, R.string.open_Navigation,
R.string.close_Navigation);
        drawerLayout.addDrawerListener(toggle);
        toggle.syncState();
    }
    ...
}

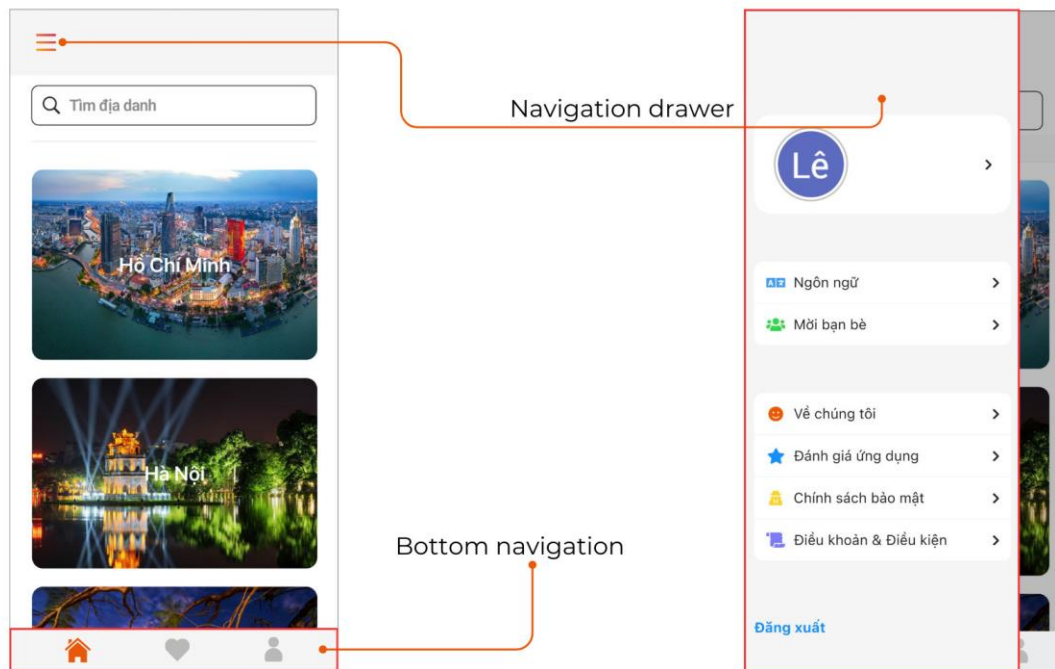
```

`ActionBarDrawerToggle` được sử dụng để đặt một toggle button trên `Toolbar` để bật/tắt `DrawerLayout`; và đến đây thì Icon Drawer cũng sẽ xuất hiện. Khi `DrawerLayout` được mở ra, Icon của toggle button trên `Toolbar` sẽ được thay đổi từ biểu tượng của tiêu đề Activity sang biểu tượng của `DrawerLayout`. Khi `DrawerLayout` đã được đóng, Icon của toggle button sẽ được đặt lại về trạng thái như mới khởi đầu [5].

### 3. KẾT QUẢ THỰC NGHIỆM

Chúng tôi đã dùng navigation vào trong thiết kế của dự án “Quảng bá địa điểm du lịch tại Việt Nam”. Ứng dụng “Quảng bá địa điểm du lịch tại Việt Nam” là một ứng dụng di động trên nền tảng Android, được thiết kế để cung cấp thông tin toàn diện về các địa điểm du lịch tại Việt Nam cho người dùng. Ứng dụng còn có các chức năng chính như là cung cấp thông tin chi tiết của từng địa điểm du lịch, bao gồm hình ảnh, vị trí, đánh giá của khách hàng và nhiều thông tin hữu ích khác. Ngoài ra, ứng dụng còn cho phép người dùng đăng nhập và tạo tài khoản cá nhân để theo dõi các địa điểm yêu thích của mình, lưu trữ với các địa điểm mà mình mong muốn ghé thăm trong tương lai và trao đổi với những người dùng khác.

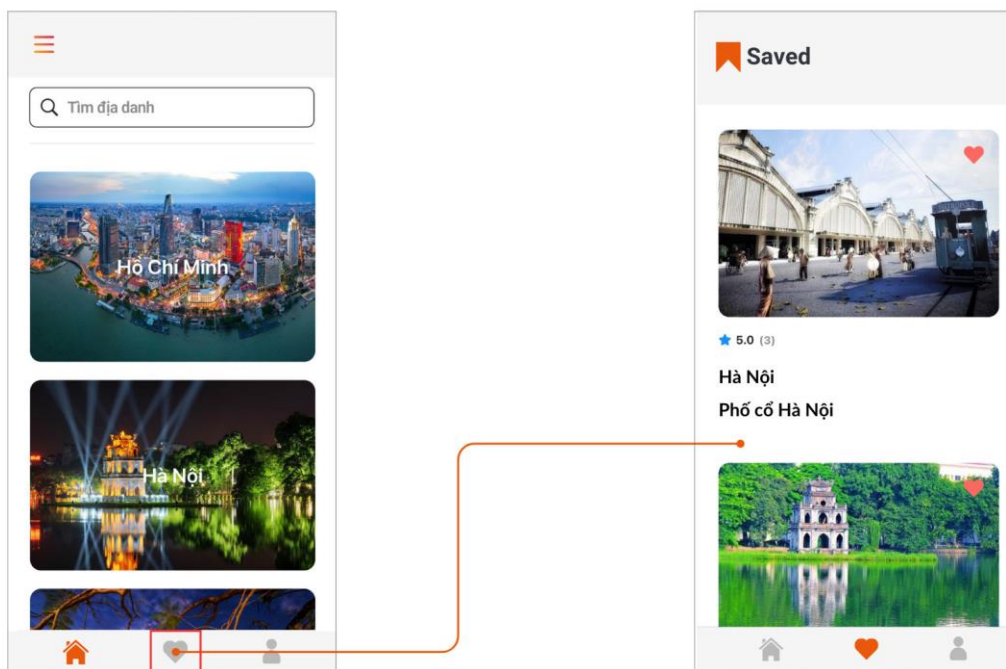
Trong ứng dụng “Quảng bá địa điểm du lịch tại Việt Nam”, chúng tôi sử dụng cả hai bottom navigation và navigation drawer để tăng hiệu quả trải nghiệm cho người dùng, giúp người dùng dễ dàng sử dụng ứng dụng trong việc tìm kiếm thông tin và điều hướng các màn hình, như hình 10:



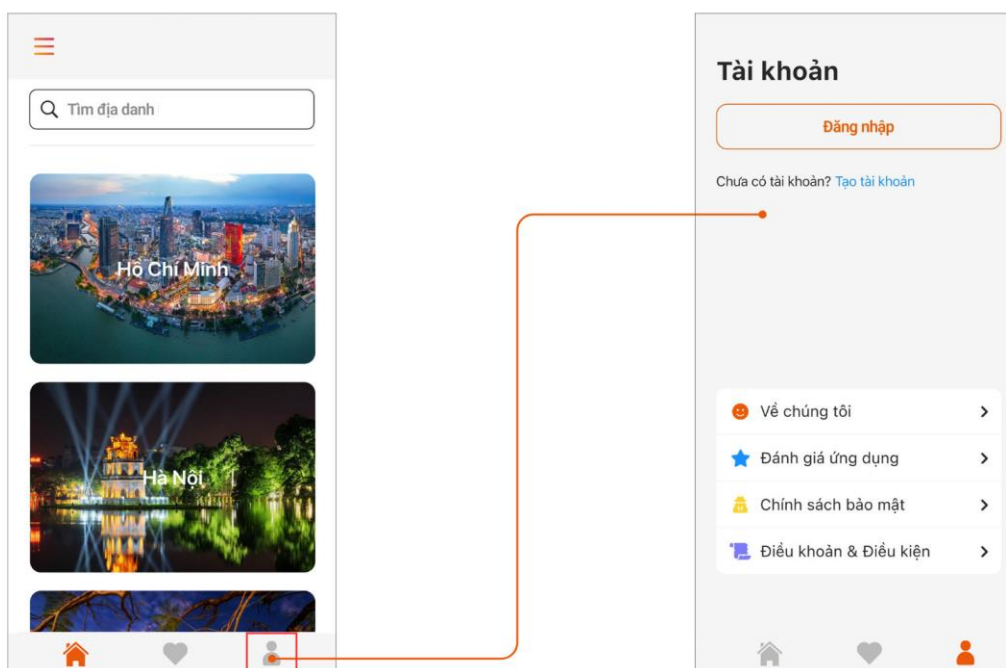
Hình 10. Dùng bottom navigation và navigation drawer trong thiết kế giao diện ứng dụng "Quảng bá địa điểm du lịch tại Việt Nam"

Về phần bottom navigation, bằng cách đặt các chức năng chính của ứng dụng ở phía dưới màn hình, người dùng có thể truy cập nhanh và thuận tiện đến các tính năng chính của ứng dụng. Trong ứng dụng "Quảng bá địa điểm du lịch tại Việt Nam", bottom navigation cung cấp các chức năng như trang chủ, địa điểm yêu thích và tài khoản (hình 11 và hình 12).

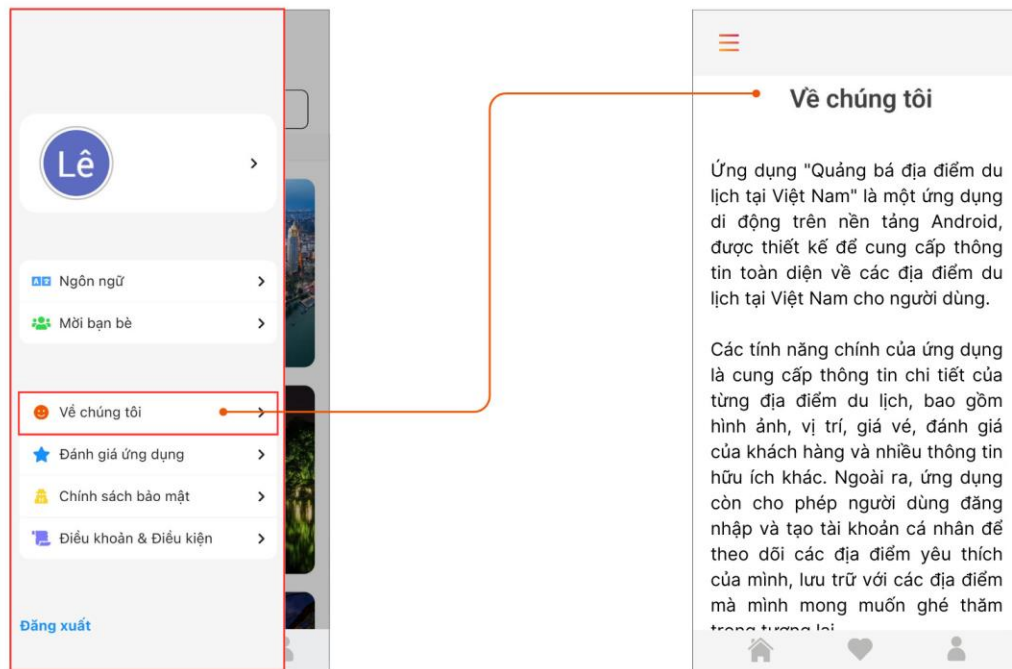
Về phần navigation drawer, việc thêm menu trượt ở cạnh bên trái màn hình cho phép người dùng dễ dàng tiếp cận các tính năng tùy chọn bao gồm các tính năng như: điều khoản – điều kiện, thông tin ứng dụng, đăng xuất. Menu được hiển thị khi người dùng thực hiện hành động chạm kéo từ cạnh bên trái của màn hình.



Hình 11. Dùng bottom navigation điều hướng đến chức năng địa điểm yêu thích



Hình 12. Dùng bottom navigation điều hướng đến chức năng tài khoản



Hình 13. Dùng navigation drawer điều hướng đến chức năng thông tin ứng dụng

Sự kết hợp giữa bottom navigation và navigation drawer đã giúp người dùng dễ dàng điều hướng và truy cập các tính năng chính một cách thuận tiện nhất. Bên cạnh đó, cả hai kiểu navigation này được thiết kế đơn giản và thân thiện để dễ dàng sử dụng cho người dùng, đồng thời tạo nên sự tương tác trực quan và hấp dẫn cho người dùng sử dụng ứng dụng "Quảng bá địa điểm du lịch tại Việt Nam".

Tóm lại, Sau khi tiến hành thực nghiệm bằng cách tích hợp navigation vào trong thiết kế của ứng dụng lập trình di động "Quảng bá địa điểm du lịch tại Việt Nam", chúng tôi nhận thấy rằng việc sử dụng bottom navigation và navigation drawer đã tăng hiệu quả trải nghiệm cho người dùng. Việc sử dụng navigation trong thiết kế giao diện ứng dụng lập trình di động Android đã cải thiện trải nghiệm người dùng một cách đáng kể và giúp tăng cường tính năng của ứng dụng, đặc biệt trong việc điều hướng các màn hình của ứng dụng.

#### 4. KẾT LUẬN

Trong bài báo này, chúng tôi trình bày tổng quát về một số dạng navigation dựa trên trang web cung cấp tài liệu chính thức của Google về lập trình Android [2], nêu chi tiết các bước thực hiện cơ bản để tạo các navigation vào một ứng dụng android. Đồng thời, chúng tôi tiến hành phát triển ứng dụng "Quảng bá địa điểm du lịch tại Việt Nam" có dùng bottom navigation và navigation drawer. Qua đó cho thấy những ưu điểm và hạn chế của từng loại công cụ này trong việc giúp người dùng điều hướng trên các ứng dụng di động. Navigation drawer với khả năng hiển thị nhiều tùy chọn điều hướng khác nhau trong một menu ẩn giúp tiết kiệm không gian màn hình vì vậy phù hợp với các ứng dụng có nhiều tùy chọn điều hướng. Tuy nhiên navigation drawer có hạn chế là gây khó khăn cho người mới sử dụng ứng dụng và khởi động chậm hơn. Bottom navigation giúp tăng tính trực quan, giúp người dùng thực hiện điều hướng dễ dàng và chuyển đổi giữa các tùy chọn một cách nhanh chóng, tuy nhiên nó có giới hạn về số lượng tùy chọn điều hướng được hiển thị.

Tùy thuộc vào nhu cầu và yêu cầu của ứng dụng, sự lựa chọn giữa navigation drawer và bottom navigation là quan trọng để đảm bảo tính tiện dụng và hiệu quả cho người dùng khi sử dụng ứng dụng, hoặc chúng ta có thể kết hợp cả hai loại này nhưng cần được xác định phù hợp với tính chất và mục đích của ứng dụng.



---

## TÀI LIỆU THAM KHẢO

1. Android Developers. *Navigation*. Truy cập ngày 25/4/2023, từ <https://developer.android.com/guide/navigation>
2. Google. *Android Developers*. Truy cập ngày 25/4/2023, từ <https://developer.android.com>
3. Android Developers. *Principles of navigation*. Truy cập ngày 25/4/2023, từ [https://developer.android.com/guide/navigation/navigation-principles#fixed\\_start\\_destination](https://developer.android.com/guide/navigation/navigation-principles#fixed_start_destination)
4. Google. *Material Design*. Truy cập ngày 25/4/2023, từ <https://m3.material.io/foundations/layout/understanding-layout/parts-of-layout>
5. Android Developers. *Handling configuration changes*. Truy cập ngày 25/4/2023, từ <https://developer.android.com/guide/navigation/navigation-config-changes>
6. Trần Minh Hải. (2018). *Cách tạo Navigation Drawer*. Truy cập ngày 25/4/2023, từ <https://viblo.asia/p/cach-cao-navigation-drawer-ByEZk0NElQ0>
7. Trần Minh Hải. (2018). *Hướng dẫn Bottom Navigation Bar trong Android (Phần 1)*. Truy cập ngày 25/4/2023, từ <https://viblo.asia/p/huong-dan-bottom-navigation-bar-trong-android-phan-1-RQqKLvLpl7z>
8. Nguyễn Minh Quân. (2018). *Làm việc với Bottom Navigation trong Android*. Truy cập ngày 25/4/2023, từ <https://viblo.asia/p/lam-viec-voi-bottom-navigation-trong-android-gGJ59j6pKX2>
9. Material Design. *Bottom navigation*. Truy cập ngày 25/4/2023, từ <https://m2.material.io/components/bottom-navigation/android>

## ABSTRACT

In this paper, we introduce the design and handling of navigation in developing the user interface of Android mobile applications. The paper emphasizes the importance and benefits of navigation for user experience. Additionally, we provide a detailed process for building two popular types of navigation, bottom navigation and navigation drawer. Furthermore, we conducted experiments by incorporating these two types of navigation into the design of the user interface during the development of the Android mobile application “Promoting tourist destinations in Vietnam”. The results showed that the application has a beautiful and user-friendly interface, providing an enjoyable experience for users through the effectiveness of the implemented navigations.

## USING NAVIGATION IN MOBILE ANDROID PROGRAMMING INTERFACE DESIGN

Tên tác giả: Lê Minh Phát (2001200130, 11DHTH3), Lê Tâm Như (2001200582, 11DHTH3), Ngô Thị Kim Ngân (2001200110, 11DHTH3), Nguyễn Thị Kim Tuyền (2001200178, 11DHTH4), TS. Nguyễn Thị Bích Ngân

Cơ quan, địa chỉ (tiếng Anh): Faculty of Information Technology, Ho Chi Minh City University of Food Industry, Vietnam.

Email: [lephat508@gmail.com](mailto:lephat508@gmail.com)

*Tôi xin cam kết bài báo này chưa được đăng và gửi đăng ở bất kỳ tạp chí nào khác.*

*Tác giả:*

*Họ và tên: Lê Minh Phát*

*Số điện thoại liên hệ: 0907 399 926*