

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO TIỂU LUẬN
LẬP TRÌNH THIẾT BỊ ĐI ĐỘNG

ĐỀ TÀI

**TÌM HIỂU VỀ CƠ CHẾ XỬ LÝ SỰ KIỆN TRONG ỨNG DỤNG
ANDROID, NGUYÊN TẮC SỬ DỤNG TRONG TƯỜNG TRƯỜNG HỢP
CỤ THỂ? THỰC HIỆN ỨNG DỤNG DEMO MINH HỌA CHI TIẾT CHO
TỪNG CƠ CHẾ XỬ LÝ.**

Giảng viên: ThS. NGUYỄN QUANG PHÚC

Sinh viên thực hiện: LÊ NGUYỄN THÁI KHANG 5951071043

NGUYỄN ĐÌNH GIA KHẢI 5951071042

NGUYỄN HOÀNG NHẬT 5951071069

Lớp: CÔNG NGHỆ THÔNG TIN K59

Tp.Hồ Chí Minh, năm 2021

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO TIỂU LUẬN
LẬP TRÌNH THIẾT BỊ ĐI ĐỘNG

ĐỀ TÀI
TÌM HIỂU VỀ CƠ CHẾ XỬ LÝ SỰ KIỆN TRONG ỨNG DỤNG
ANDROID, NGUYÊN TẮC SỬ DỤNG TRONG TƯỜNG TRƯỜNG HỢP
CỤ THỂ? THỰC HIỆN ỨNG DỤNG DEMO MINH HỌA CHI TIẾT CHO
TỪNG CƠ CHẾ XỬ LÝ.

Giảng viên: ThS. NGUYỄN QUANG PHÚC

Sinh viên thực hiện: LÊ NGUYỄN THÁI KHANG 5951071043

NGUYỄN ĐÌNH GIA KHẢI 5951071042

NGUYỄN HOÀNG NHẬT 5951071069

Lớp: CÔNG NGHỆ THÔNG TIN K59

TP. Hồ Chí Minh, năm 2021

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do - Hạnh phúc

THIẾT KẾ TỔNG QUAN ĐỀ TÀI

-----*-----

1. Tên đề tài

Tìm hiểu về cơ chế xử lý sự kiện trong ứng dụng Android, nguyên tắc sử dụng trong từng trường hợp cụ thể? Thực hiện ứng dụng demo minh họa chi tiết cho từng cơ chế xử lý.

2. Nhiệm vụ thực hiện

- Tìm hiểu về ngôn ngữ lập trình java
- Tìm hiểu về lập trình Android
- Hoàn thành ứng dụng minh họa.

4. Công cụ ,công nghệ và ngôn ngữ lập trình

- Công cụ : Android Studio
- Ngôn ngữ lập trình : Java

5. Ngày bắt đầu thực hiện: Ngày 27/06/2021

6. Thành viên và hân công công việc.

STT	Họ tên	MSV	Công việc
1	Lê Nguyễn Thái Khang	5951071043	Lên kế hoạch, thiết kế , thực hiện demo trên android studio, tổng hợp nội dung, làm báo cáo
2	Nguyễn Đình Gia Khải	5951071042	Nghiên cứu về Event Listeners & Event Handlers và đưa ra ý tưởng làm demo.
3	Nguyễn Hoàng Nhật	5951071069	Nghiên cứu về Trình xử lý sự kiện và đưa ra ý tưởng làm demo.

LỜI CẢM ƠN

Để hoàn thành đề tài này trước hết chúng em xin gửi đến quý thầy, cô Bộ môn Công nghệ thông tin – Phân hiệu Trường Đại học Giao thông Vận tải tại Thành phố Hồ Chí Minh lời cảm ơn chân thành vì đã truyền đạt cho chúng em những kiến thức không chỉ từ sách vở, mà còn những kinh nghiệm quý giá từ cuộc sống trong khoảng thời gian học tập tại trường. Đặc biệt chúng em xin gửi đến thầy Nguyễn Quang Phúc lời cảm ơn sâu sắc nhất, thầy đã truyền đạt những kiến thức quan trọng giúp chúng em hoàn thành được đề tài. Tuy đề tài không được lớn lao nhưng nếu không được sự hướng dẫn chỉ bảo tận tình của thầy thì đề tài này khó có thể hoàn thành được.

Vì thời gian làm đồ án có hạn cũng như hiểu biết cá nhân còn hạn chế, chúng em cũng đã nỗ lực hết sức để hoàn thành bài báo cáo đồ án một cách tốt nhất, nhưng chắc chắn vẫn sẽ có những thiếu sót không thể tránh khỏi. Em kính mong nhận được sự thông cảm và những ý kiến đóng góp chân thành từ quý thầy cô.

Sau cùng, em xin kính chúc Quý Thầy Cô luôn mạnh khỏe, hạnh phúc và thành công hơn nữa trong công việc cũng như trong cuộc sống.

Chúng em xin chân thành cảm ơn!

NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp. Hồ Chí Minh, ngày tháng năm 2021
Giảng viên

Nguyễn Quang Phúc

MỤC LỤC

TỔNG QUAN	7
1. Tổng quan về đề tài.....	7
2. Mục tiêu nghiên cứu	7
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT	8
1. Tổng quan về lập trình Android.....	8
1.1 Giới thiệu về lập trình android.....	8
1.2 Công cụ phát triển ứng dụng Android	8
2. Xử lý sự kiện trong Android.....	9
2.1 Sự kiện trong lập trình android.	9
2.2 Event Listeners & Event Handlers	9
2.3 Đăng ký Event Listener trong Android	10
CHƯƠNG 2: cài đặt	11
1. Cài đặt Event Listeners & Event Handlers	11
1.1 onClick().	11
1.2 onLongClick().	11
1.3 onFocusChange()	11
1.4 onKey().....	12
1.5 onTouch().....	12
1.6 onCreateContextMenu()	13
1.7 onOptionsItemSelected()	13
2. Trình xử lý sự kiện (event handler)	14
2.1 Onclick XML.....	14
2.2 Anonymous Listener.....	14
2.3 Variable as Listener.....	15
2.4 Activity as Listener	15

2.5 Explicit class Listener	16
2.6 View Subclassing.....	16
CHƯƠNG 3 : ỨNG DỤNG DEMO	17
1. Demo Event Listeners & Event Handlers	17
1.1 Demo onClick()	17
1.2 Demo onLongClick()	17
1.3 Demo onFocusChange().....	18
1.4 Demo onKeyDown().....	19
1.5 Demo onTouch().....	20
1.6 Demo onCreateContextMenu()	21
1.7 Demo onOptionsItemSelected()	22
2. Demo trình xử lý sự kiện (event handler).....	23
2.1 Demo Onclick XML.....	23
2.2 Demo Anonymous Listener.....	23
2.3 Demo Variable as Listener.....	24
2.4 Demo Activity as Listener	24
2.5 Demo Explicit class Listener.....	25
2.6 Demo View Subclassing.....	25
KẾT LUẬN.....	26
TÀI LIỆU THAM KHẢO	26

MỤC LỤC HÌNH ẢNH

Hình 1 Code cài đặt <code>onClick()</code>	11
Hình 2 Code cài đặt <code>onLongClick()</code>	11
Hình 3 Code cài đặt <code>onFocusChange()</code>	11
Hình 4 Code cài đặt <code>onKey()</code>	12
Hình 5 Code cài đặt <code>onTouch()</code>	12
Hình 6 Code cài đặt <code>onCreateContextMenu()</code>	13
Hình 7 Code cài đặt <code>onMenuItemClick()</code>	13
Hình 8 <code>android:onclick</code> trong file <code>xml</code>	14
Hình 9 tạo sự kiện trong file <code>java</code> từ file <code>xml</code>	14
Hình 10 cách sử dụng <code>Anonymous Listener</code>	14
Hình 11 Cách sử dụng <code>Variable as Listener</code>	15
Hình 12 Cách sử dụng <code>Activity as Listener</code>	15
Hình 13 Cách sử dụng <code>Explicit class Listener</code>	16
Hình 14 Cách sử dụng <code>View Subclassing</code>	16
Hình 15 Demo <code>onClick()</code>	17
Hình 16 Demo <code>onLongClick()</code>	18
Hình 17 Demo <code>onFocusChange()</code>	19
Hình 18 Demo <code>onKey()</code>	20
Hình 19 Demo <code>onTouch()</code>	21
Hình 20 Demo <code>onCreateContextMenu()</code>	22
Hình 21 Demo <code>onMenuItemClick()</code>	22
Hình 22 Demo <code>OnClick XML</code>	23
Hình 23 Demo <code>Anonymous Listener</code>	23
Hình 24 Demo <code>Variable as Listener</code>	24
Hình 25 Demo <code>Activity as Listener</code>	24
Hình 26 Demo <code>Explicit class Listener</code>	25
Hình 27 Demo <code>View Subclassing</code>	25

TỔNG QUAN

1. Tổng quan về đề tài

Trên Android, có nhiều cách để chặn (intercept) các sự kiện từ một action tương tác của người dùng với ứng dụng của bạn. Khi xem xét các sự kiện trong giao diện người dùng, bạn nên áp dụng phương pháp bắt các sự kiện từ đối tượng View cụ thể mà người dùng tương tác cùng. Lớp View cung cấp phương tiện để thực hiện điều này.

Trong những lớp View khác nhau dùng để tạo layout, hãy lưu ý rằng có vài phương thức callback dạng public hữu ích cho các sự kiện giao diện người dùng. Những phương thức này được framework Android gọi khi action tương ứng xảy ra trên đối tượng đó. Ví dụ, khi nhấn vào một View (chẳng hạn như một Button), phương thức `onTouchEvent()` sẽ được gọi trên đối tượng đó. Tuy nhiên, để ngăn chặn điều này, bạn phải mở rộng lớp và ghi đè lên phương thức đó. Nhưng việc mở rộng mọi đối tượng View để xử lý sự kiện như vậy sẽ không thiết thực. Đó là lý do tại sao lớp View còn chứa một tập các giao diện lồng nhau với những callback mà bạn có thể dễ định nghĩa hơn nhiều. Các giao diện như vậy gọi là trình lắng nghe sự kiện (event listener), đó là chìa khóa để bắt những tương tác của người dùng với giao diện.

Trong khi bạn sử dụng trình lắng nghe sự kiện để lắng nghe tương tác của người dùng, có thể sẽ đến lúc nào đó bạn muốn mở rộng một lớp View để xây dựng một thành phần tùy chỉnh. Có thể bạn muốn mở rộng lớp Button để tạo ra cái gì đó hay hơn chẳng hạn. Trong trường hợp này, bạn có thể định nghĩa các hành vi sự kiện mặc định cho lớp bằng cách sử dụng trình xử lý sự kiện (event handler) cho lớp.

2. Mục tiêu nghiên cứu

- Tìm hiểu cơ chế xử lý sự kiện trong ứng dụng Android.
- Tìm hiểu về những trường hợp trong xử lý sự kiện và nguyên tắc sử dụng chúng.
- Tìm hiểu về trình lắng nghe sự kiện (event listener)
- Tìm hiểu về trình xử lý sự kiện (event handler)
- Thực hiện ứng dụng demo.

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1. Tổng quan về lập trình Android

1.1 Giới thiệu về lập trình android.

Tính đến đầu năm 2016, có hơn một tỷ người tích cực sử dụng Android, một hệ điều hành di động giống Unix chạy trên điện thoại, máy tính bảng, TV thông minh, đồng hồ thông minh và hàng loạt thiết bị khác. Mặc dù lập trình Android được Google phát triển và bảo trì, nhưng đây là phần mềm miễn phí và nguồn mở có thể được sử dụng trong lập trình di động và tùy chỉnh bởi bất kỳ ai. Do đó, các thiết bị Android được sản xuất bởi một số nhà sản xuất thiết bị lớn, như Samsung, LG, Sony và HTC.

Một trong những yếu tố chính thúc đẩy thành công của lập trình Android là Google Play, một nền tảng phân phối ứng dụng Android mà người dùng có thể sử dụng để tải xuống và cài đặt ứng dụng trên thiết bị của họ. Với hơn một triệu ứng dụng và hàng tỷ lượt tải xuống, Google Play lớn hơn và có lưu lượng truy cập nhiều hơn so với các nền tảng tương tự cho iOS và Windows 10.

1.2 Công cụ phát triển ứng dụng Android

Các ứng dụng Android Native được phát triển bằng ngôn ngữ lập trình Android như Java và trên các thiết bị chạy Android 5.0 trở lên, sử dụng thời gian chạy được quản lý có tên là Android Runtime (ART) theo mặc định. Để tạo các ứng dụng như vậy, hầu hết các nhà phát triển chọn sử dụng Android Studio, IDE Android chính thức do Google phát triển. Android Studio miễn phí và chạy trên Windows, OS X và hầu hết các bản phân phối Linux.

Android Studio 1.5.1, phiên bản ổn định hiện tại, có một số lượng lớn các tính năng để đơn giản hóa tất cả các khía cạnh phát triển ứng dụng Android, chẳng hạn như trình chỉnh sửa code mạnh mẽ, bố cục trực quan và trình chỉnh sửa chủ đề, studio asset với hình ảnh và vector, v.v. Android Studio 2.0, vẫn chỉ có sẵn dưới dạng bản phát hành xem trước, có các tính năng nâng cao hơn như chạy ngay lập tức và trình cấu hình GPU để cải thiện hơn nữa trải nghiệm phát triển ứng dụng.

Android Studio cũng đi kèm với trình giả lập thực tế mà bạn có thể sử dụng lúc lập trình Android nếu bạn không sở hữu thiết bị Android. Vì hình ảnh hệ thống cho

nó có sẵn miễn phí, bạn có thể tải xuống và chạy bất kỳ phiên bản Android nào trên đó.

2. Xử lý sự kiện trong Android.

2.1 Sự kiện trong lập trình android.

Các sự kiện (Event) là một cách hữu ích để thu thập dữ liệu về tương tác của người dùng với các thành phần tương tác của ứng dụng. Ví dụ như sự kiện nhấn vào nút hoặc chạm vào màn hình, ... Android Framework duy trì một dãy sự kiện dựa trên nền tảng là First-In, First-out (FIFO). Bạn có thể nắm bắt các sự kiện này trong chương trình và thực hiện hành động thích hợp tùy theo yêu cầu.

Liên quan tới quản lý sự kiện ta cần hiểu 3 khái niệm sau:

2.1.1 Event Listeners

Một Event Listener là một Interface trong lớp View mà chứa một phương thức callback. Các phương thức này sẽ có thể được gọi bởi Android Framework khi Listener đã đăng ký được kích hoạt bởi tương tác của người dùng với item trong giao diện UI.

2.1.2 Event Listeners Registration

Event Registration là tiến trình mà bởi đó một Event Handler được đăng ký với một Event Listener để mà Handler này được gọi khi Event Listener kích hoạt sự kiện.

2.1.3 Event Handlers

Khi một Event xảy ra, và chúng ta đã đăng ký một Event Listener cho sự kiện, thì Event Listener gọi Event Handler, là phương thức mà thực sự xử lý sự kiện đó.

2.2 Event Listeners & Event Handlers

Event Handler	Event Listener & Miêu tả
onClick()	OnClickListener() Được gọi khi người dùng hoặc nhấn hoặc chạm hoặc tập trung vào bất cứ widget nào như button, text, image ... Bạn sẽ sử dụng event handler là onClick() để xử lý sự kiện như thế

onLongClick()	OnLongClickListener() Được gọi khi người dùng hoặc nhấn hoặc chạm vào bất cứ widget nào như button, text, image ... trong một hoặc vài giây. Bạn sẽ sử dụng event handler là onLongClick() để xử lý sự kiện như thế
onFocusChange()	OnFocusChangeListener() Được gọi khi widget mất tiêu điểm của nó, ví dụ như người dùng rời khỏi item trên view. Bạn sẽ sử dụng event handler là onFocusChange() để xử lý các sự kiện như thế
onKey()	OnKeyListener() Được gọi khi người dùng tập trung vào item và nhấn hoặc thả ra một phím cứng trên thiết bị. Bạn sẽ sử dụng event handler là onKey() để xử lý sự kiện như thế
onTouch()	OnTouchListener() Được gọi khi người dùng nhấn key, nhả phím hoặc bất cứ động tác nào trên màn hình. Bạn sẽ sử dụng event handler là onTouch() để xử lý sự kiện như thế
onMenuItemClick()	OnMenuItemClickListener() Được gọi khi người dùng lựa chọn một item trong menu. Bạn sẽ sử dụng event handler là onMenuItemClick() để xử lý sự kiện như thế
onCreateContextMenu()	onCreateContextMenuListener() Được gọi khi context menu đang được xây dựng (là kết quả của một click liên tục trong thời gian dài)

2.3 Đăng ký Event Listener trong Android

Đăng ký Event là tiến trình mà bởi đó một Event Handler được đăng ký với một Event Listener để mà handler được gọi khi Event Listener kích hoạt sự kiện. Dưới đây là một số cách để đăng ký Event Handler, chúng ta sẽ demo từng cách một trong các trang tiếp theo:

- Xử lý sự kiện trong Layout (Handle event in Layout)
- Xử lý sự kiện bằng lớp nặc danh (Inline anonymous listener)
- Kế thừa Interface OnClickListener (Implements OnClickListener Interface)
- Sử dụng biến (Event Listener using Variable)
- Bắt sự kiện thông qua lớp lắng nghe (Listener Class)
- PerformClick method

CHƯƠNG 2: CÀI ĐẶT

1. Cài đặt Event Listeners & Event Handlers

1.1 onClick().

```
private void addevent() {
    btn1.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent( packageContext: MainActivity.this, OnclickXML.class);
            startActivity(intent);
        }
    });
}
```

Hình 1 Code cài đặt onClick()

1.2 onLongClick().

```
private void addEventlongclick() {
    edt1.setOnLongClickListener(new View.OnLongClickListener() {
        @Override
        public boolean onLongClick(View v) {
            Toast.makeText(v.getContext(), text: "Nhập họ và tên của bạn", Toast.LENGTH_SHORT).show();
            return true;
        }
    });
}
```

Hình 2 Code cài đặt onLongClick().

1.3 onFocusChange()

```
private void addEventforcus() {
    edt1.setOnFocusChangeListener(new View.OnFocusChangeListener() {
        @Override
        public void onFocusChange(View v, boolean hasFocus) {
            if (!hasFocus && edt1.getText() != null) {
                Toast.makeText(getApplicationContext(), text: "Bạn đã nhập: " + edt1.getText().toString(), Toast.LENGTH_SHORT).show();
            }
        }
    });
}
```

Hình 3 Code cài đặt onFocusChange().

1.4 onKeyDown()

```

public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode==event.KEYCODE_BACK){
        AlertDialog.Builder builder = new AlertDialog.Builder( context: demochinh.this);
        builder.setMessage("Bạn có muốn xóa hết thông tin và quay lại trang trước không?");
        builder.setTitle("Xác nhận");
        builder.setPositiveButton( text: "Yes", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                finish();
            }
        });
        builder.setNegativeButton( text: "No", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.dismiss();
            }
        });
        AlertDialog dialog= builder.create();
        dialog.show();
        return true;
    }
    return super.onKeyDown(keyCode, event);
}

```

Hình 4 Code cài đặt onKeyDown().

1.5 onTouch()

```

private void addeventOnTouch() {
    lol.setOnTouchListener(new View.OnTouchListener() {
        int i=0;
        @Override
        public boolean onTouch(View v, MotionEvent event) {
            x = event.getX();
            y = event.getY();
            if (event.getAction()==MotionEvent.ACTION_MOVE){
                btn2.setX(x);
                btn2.setY(y);
            }
            return true;
        }
    });
}

```

Hình 5 Code cài đặt onTouch()

1.6 onCreateContextMenu()

```
private void addeventcontextmenu() {
    registerForContextMenu(10);
}

@Override
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    getMenuInflater().inflate(R.menu.menu2, menu);
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    switch (item.getItemId()) {
        case R.id.ex:
            finish();
            System.exit( status: 0);

        case R.id.b:
            Intent intent1 = new Intent( packageContext: demochinh.this, MainActivity.class);
            startActivity(intent1);
    }
    return super.onOptionsItemSelected(item);
}
```

Hình 6 Code cài đặt onCreateContextMenu()

1.7 onOptionsItemSelected()

```
private void addeventonmenu() {
    btn3.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            PopupMenu popupMenu = new PopupMenu( context: demochinh.this,v);
            popupMenu.getMenuInflater().inflate(R.menu.menu1, popupMenu.getMenu());
            popupMenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
                @Override
                public boolean onMenuItemClick(MenuItem item) {
                    switch (item.getItemId()) {
                        case R.id.au:
                            Toast.makeText(getApplicationContext(), ("LÊ NGUYỄN THÁI KHANG 5951071043\n " +
                                "NGUYỄN HOÀNG NHẬT 5951071069\n " +
                                "NGUYỄN ĐÌNH GIA KHẢI 5951071042 ")
                                .toString(), Toast.LENGTH_LONG).show();

                            return true;
                        case R.id.ws:
                            String url = "http://dept.utc2.edu.vn/bomoncntt/";
                            Intent i = new Intent(Intent.ACTION_VIEW);
                            i.setData(Uri.parse(url));
                            startActivity(i);
                            return true;
                        default:
                            return false;
                    }
                }
            });
            popupMenu.show();
        }
    });
}
```

Hình 7 Code cài đặt onOptionsItemSelected()

2. Trình xử lý sự kiện (event handler)

2.1 Onclick XML

Sử dụng android:onclick trong file xml .

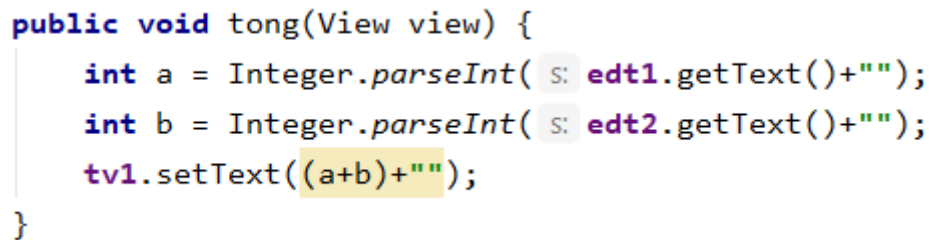


```

        android:layout_margin="30dp">
        <Button
            android:onClick="tong"
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:text="+"
            android:textStyle="bold"
            android:textSize="23dp"
            android:layout_margin="10dp"
            android:id="@+id/btn1"
            tools:ignore="OnClick" />
        </Button>
        android:onClick="hieu"
    
```

Hình 8 android:onclick trong file xml

Sau đó tạo sự kiện trong file java:

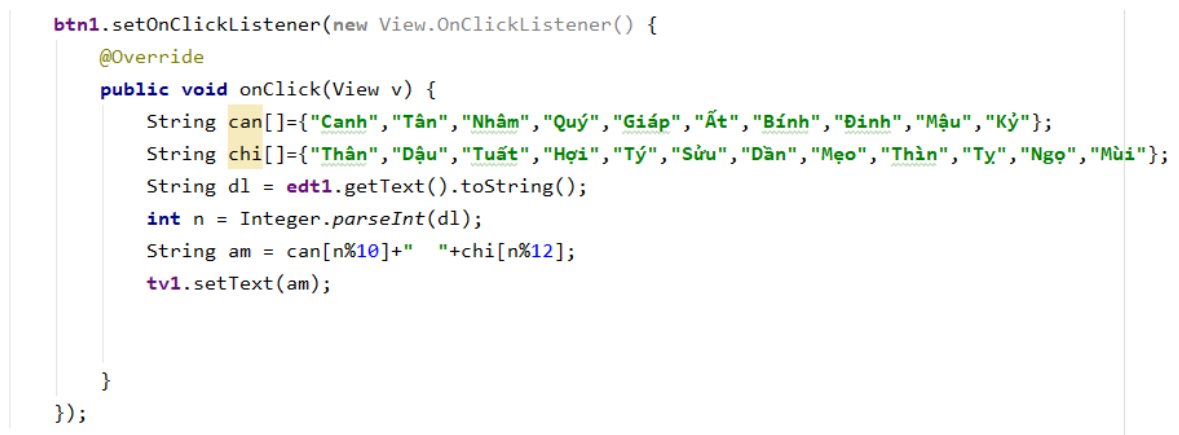


```

public void tong(View view) {
    int a = Integer.parseInt(edt1.getText().toString());
    int b = Integer.parseInt(edt2.getText().toString());
    tv1.setText((a+b).toString());
}
    
```

Hình 9 tạo sự kiện trong file java từ file xml

2.2 Anonymous Listener



```

btn1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String can[]={"Canh","Tân","Nhâm","Quý","Giáp","Ất","Bính","Đinh","Mậu","Kỷ"};
        String chi[]={"Thân","Dậu","Tuất","Hợi","Tý","Sửu","Dần","Mẹo","Thìn","Tỵ","Ngọ","Mùi"};
        String dl = edt1.getText().toString();
        int n = Integer.parseInt(dl);
        String am = can[n%10]+" "+chi[n%12];
        tv1.setText(am);
    }
});
    
```

Hình 10 cách sử dụng Anonymous Listener

2.3 Variable as Listener

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_variable_as_listener);
    addView();
    on.setOnClickListener(myclick);
    off.setOnClickListener(myclick);
}

private View.OnClickListener myclick = new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (v.getId()==R.id.on){
            img.setImageResource(R.drawable.actions_help_hint_icon);
        }
        else if(v.getId()==R.id.off){
            img.setImageResource(R.drawable.actions_help_hint_icon1);
        }
    }
};
```

Hình 11 Cách sử dụng Variable as Listener

2.4 Activity as Listener

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_as_listener);
    linkView();
    btn1.setOnClickListener(this);
}

@Override
public void onClick(View v) {

    double H = Double.parseDouble( s: edt1.getText()+"");
    double W = Double.parseDouble( s: edt2.getText()+"");
    double BMI = W/Math.pow(H,2);
    String chuandoan="";
    if (BMI<18){
        chuandoan="bạn gầy";
    }
    else if (BMI<=24.9){
        chuandoan="bạn bình thường ";
    }
    else if (BMI<=29.9){
        chuandoan="bạn béo phì độ 1";
    }
    else if (BMI<=34.9){
        chuandoan="bạn béo phì độ 2";
    }
    else {
        chuandoan="bạn béo phì độ 3";
    }
    DecimalFormat df = new DecimalFormat( pattern: "#.0");
    t3.setText(df.format(BMI));
    t4.setText(chuandoan);
}
```

Hình 12 Cách sử dụng Activity as Listener

2.5 Explicit class Listener

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_explicit_class_listener);
    linkview();
    btn1.setOnClickListener(new myevent1());
    btn2.setOnClickListener(new myevent2());
    btn3.setOnClickListener(new myevent3());
}

private class myevent1 implements View.OnClickListener {
    @Override
    public void onClick(View v) {
        int a = Integer.parseInt(edt1.getText().toString());
        edt2.setText(((a-32)*(9/5)).toString());
    }
}
```

Hình 13 Cách sử dụng Explicit class Listener

2.6 View Subclassing

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_view_subclassing);
    view();
    Button btn1 = new androidx.appcompat.widget.AppCompatButton(context, this) {
        @Override
        public boolean performClick() {
            Intent intent = new Intent(getApplicationContext(), MainActivity.class);
            startActivity(intent);
            // Tao su kien ket thuc app
            Intent startMain = new Intent(Intent.ACTION_MAIN);
            startMain.addCategory(Intent.CATEGORY_HOME);
            startActivity(startMain);
            finish();
            System.exit(status: 1);
            return super.performClick();
        }
    };
    btn1.setText("close");
    LinearLayout layout= (LinearLayout)findViewById(R.id.iii);
    layout.addView(btn1);
}
```

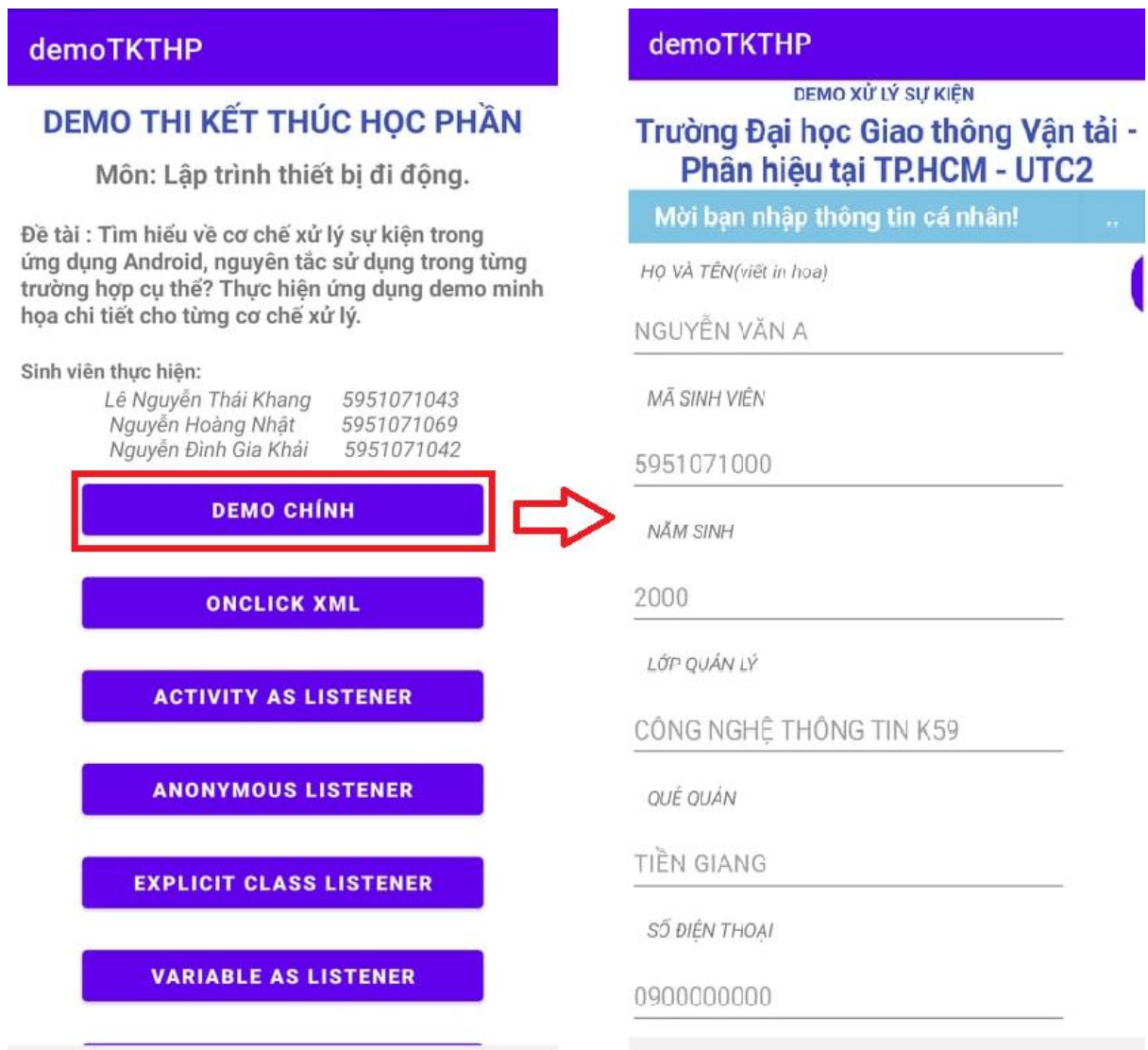
Hình 14 Cách sử dụng View Subclassing

CHƯƠNG 3 : ỨNG DỤNG DEMO

1. Demo Event Listeners & Event Handlers

1.1 Demo onClick()

Khi người dùng hoặc nhấn hoặc chạm vào bất cứ các widget như button, text, image,... bạn sẽ được sử dụng onClickListener để xử lý sự kiện mà bạn mong muốn. Ví dụ như hình bên dưới, khi nhấn vào Button thì chương trình sẽ chuyển sang trang khác theo như mình đã cài đặt và viết code.



Hình 15 Demo onClick()

1.2 Demo onLongClick()

Phương thức này được sử dụng khi người dùng hoặc nhấn hoặc chạm vào bất cứ widget nào như button, text, image,... trong vòng một giây hoặc nhiều

hơn. Ví dụ như hình bên dưới, khi ta giữ lâu vào phần edit text họ và tên , một dòng thông báo sẽ xuất hiện để giải thích cho chúng ta biết về cái mà chúng ta sẽ phải điền vào đó.

demoTKTHP

DEMO XỬ LÝ SỰ KIỆN

**Trường Đại học Giao thông Vận tải -
Phân hiệu tại TP.HCM - UTC2**

Mời bạn nhập thông tin cá nhân! ..

HỌ VÀ TÊN(viết in hoa)

NGUYỄN VĂN A

MÃ SINH VIÊN

5951071000

NĂM SINH

2000

LỚP QUẢN LÝ

CÔNG NGHỆ THÔNG TIN K59

QUÊ QUÁN

TIỀN GIANG

SỐ ĐIỆN THOẠI

0900000000

Nhập họ và tên của bạn

Hình 16 Demo onLongClick()

1.3 Demo onForcusChange()

Phương thức này được gọi khi widget làm mất tiêu điểm của nó . Ví dụ như trong hình bên dưới khi ta đã điền xong thông tin vào edit text , sau đó ta rời khỏi phần đó để chuyển sang điền thông tin vào mục kế tiếp thì ngay lập tức sẽ có một dòng thông báo

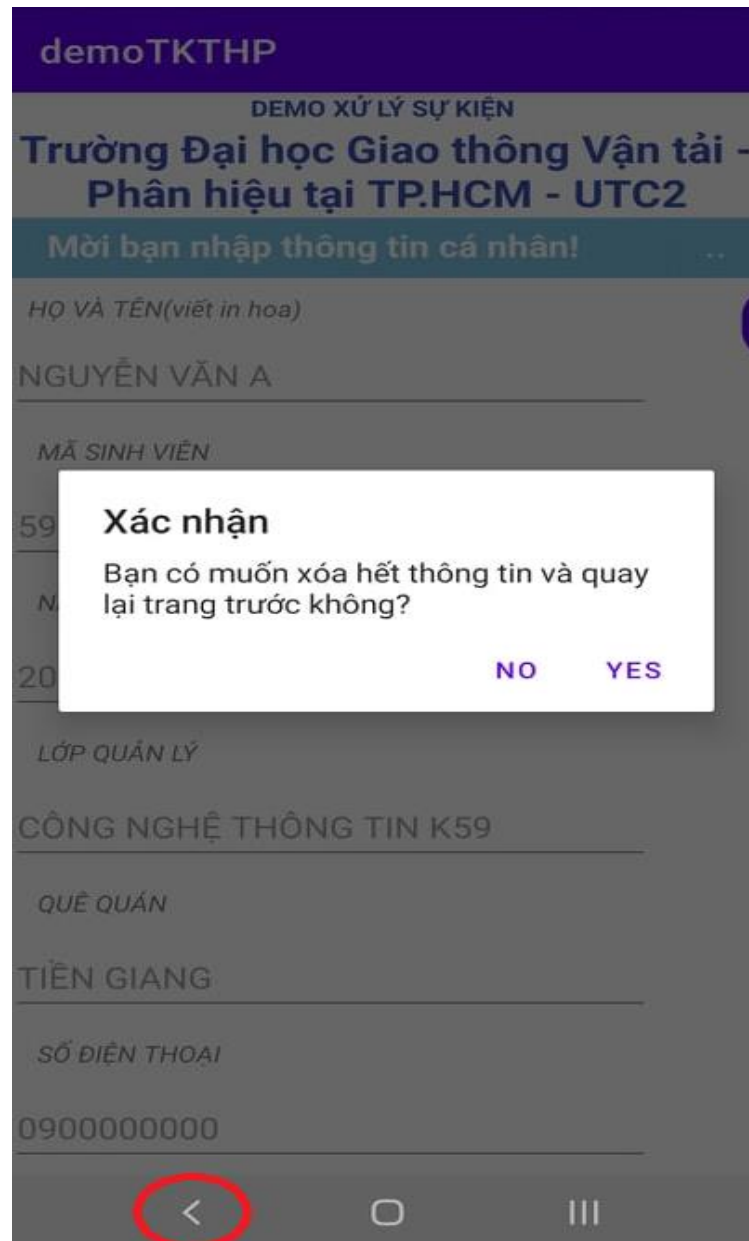
xuất hiện với mục đích để xác nhận cũng như thông báo với người điền thông tin rằng hệ thống đã nhận được thông tin đó .

The screenshot shows a mobile application interface for a demo. At the top, there's a purple header with the text "demoTKTHP". Below it, a subtitle reads "DEMO XỬ LÝ SỰ KIỆN". The main title is "Trường Đại học Giao thông Vận tải - Phân hiệu tại TP.HCM - UTC2". A light blue button says "Mời bạn nhập thông tin cá nhân!". Below this are three input fields: "HỌ VÀ TÊN(viết in hoa)" containing "LÊ NGUYỄN THÁI KHANG", "MÃ SINH VIÊN" containing "5951", and "NĂM SINH" which is empty. A grey confirmation bubble above the keyboard says "Bạn đã nhập: LÊ NGUYỄN THÁI KHANG". The bottom part of the image shows a standard Vietnamese QWERTY keyboard.

Hình 17 Demo onForcusChange()

1.4 Demo onKey()

Phương thức này được dùng tập trung vào item và nhấn hoặc thả ra một phím cứng trên thiết bị. Ví dụ như ở hình bên dưới, chúng ta muốn trở lại màn hình chính của app thì khi chúng ta nhấn phím Back thì 1 bản thông báo sẽ hiện ra để xác nhận hành động đối với người sử dụng.



Hình 18 Demo onKey()

1.5 Demo onTouch()

Phương thức này được gọi khi người dùng thực hiện một action, chẳng hạn như một sự kiện chạm (touch event), bao gồm nhấn, thả hay bất cứ động tác di chuyển nào trên màn hình (trong phạm vi của mục). Ví dụ như demo bên dưới, button ”!” có thể được kéo thả và đặt ở mọi vị trí trên màn hình mình đã cài đặt trong code.

demoTKTHP

DEMO XỬ LÝ SỰ KIỆN

**Trường Đại học Giao thông Vận tải -
Phân hiệu tại TP.HCM - UTC2**

Mời bạn nhập thông tin cá nhân! ..

HỌ VÀ TÊN(viết in hoa)

NGUYỄN VĂN A

MÃ SINH VIÊN

5951071000

NĂM SINH

2000

LỚP QUẢN LÝ

CÔNG NGHỆ THÔNG TIN K59

QUÊ QUÁN

TIỀN GIANG

SỐ ĐIỆN THOẠI

0900000000

Hình 19 Demo onTouch()

1.6 Demo onCreateContextMenu()

Phương thức này được gọi khi chúng ta giữ , nhấn lâu (long click) vào một điểm nào đó trên màn hình mà trong khi viết code chúng ta đã cài đặt , khi nó một context menu(một menu ngữ cảnh) sẽ hiện ra cùng với các menu item đã được xây dựng.

The screenshot shows a web form titled "demoTKTHP" with a subtitle "DEMO XỬ LÝ SỰ KIỆN". The main heading is "Trường Đại học Giao thông Vận tải - HCM - UTC2". Below this is a blue button labeled "Mời bạn nhập thông tin cá nhân!". The form contains several input fields with labels: "HỌ VÀ TÊN (viết in hoa)", "MÃ SINH VIÊN", "NĂM SINH", "LỚP QUẢN LÝ", "CÔNG NGHỆ THÔNG TIN K59", "QUÊ QUÁN", "TIỀN GIANG", and "SỐ ĐIỆN THOẠI". A red box highlights a context menu that appears over the "HỌ VÀ TÊN" field, containing the options "Home" and "Exit". The "TIỀN GIANG" field has a purple circle with an exclamation mark next to it.

Hình 20 Demo onContextMenu()

1.7 Demo onMenuItemClick()

Được gọi khi người dùng lựa chọn một item trong menu.

This screenshot shows the same web form as Figure 20, but with a different context menu. The red box now highlights a menu with the options "About us" and "Website". The "HỌ VÀ TÊN" field is filled with "NGUYỄN VĂN A". The "TIỀN GIANG" field still has the purple exclamation mark icon.

Hình 21 Demo onMenuItemClick()

2. Demo trình xử lý sự kiện (event handler)

2.1 Demo Onclick XML

Sử dụng android:onclick trong file xml .



Sau có tạo sự kiện trong file java:

```

public void tong(View view) {
    int a = Integer.parseInt(s: edt1.getText()+"");
    int b = Integer.parseInt(s: edt2.getText()+"");
    tv1.setText((a+b)+"");
}

```

Hình 22 Demo Onclick XML

2.2 Demo Anonymous Listener



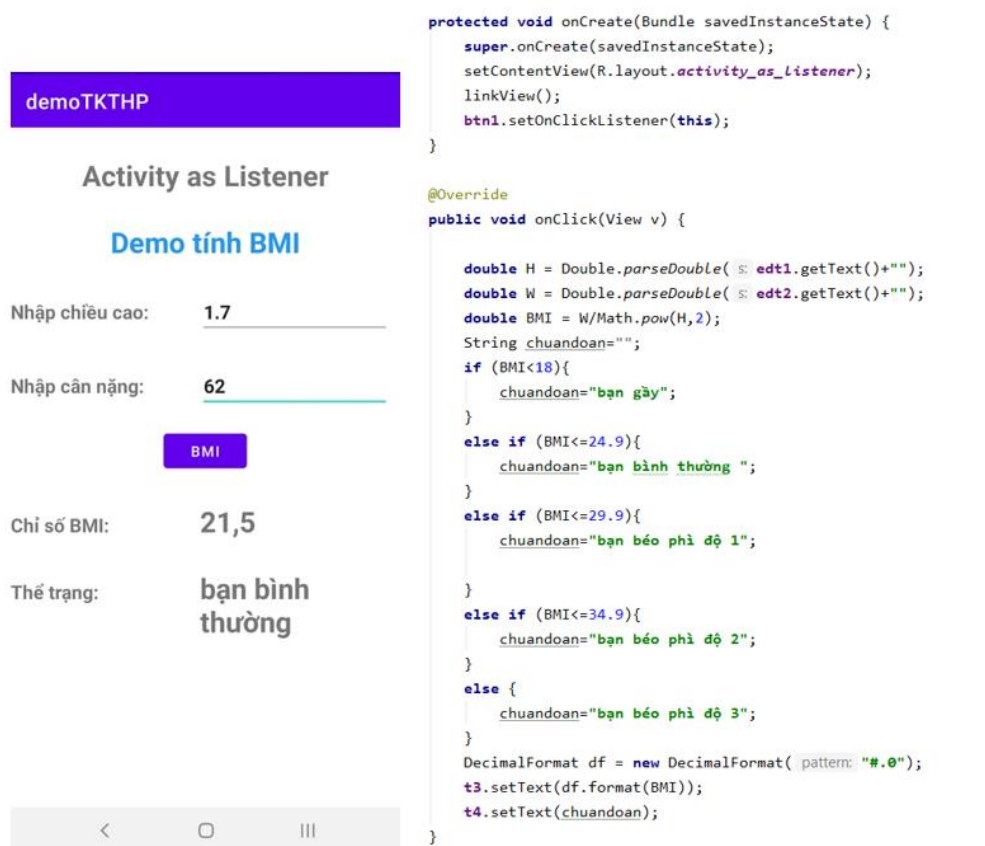
Hình 23 Demo Anonymous Listener

2.3 Demo Variable as Listener



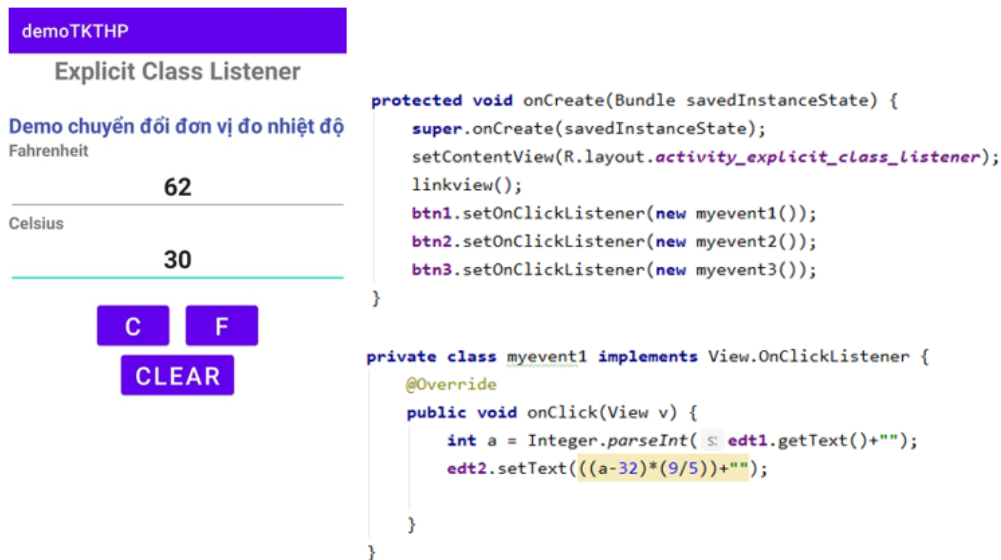
Hình 24 Demo Variable as Listener

2.4 Demo Activity as Listener



Hình 25 Demo Activity as Listener

2.5 Demo Explicit class Listener



Hình 26 Demo Explicit class Listener

2.6 Demo View Subclassing



Hình 27 Demo View Subclassing

KẾT LUẬN

- Tìm hiểu được cơ chế xử lý sự kiện trong ứng dụng Android, nguyên tắc sử dụng trong từng trường hợp cụ thể
- Thực hiện ứng dụng demo minh hoặc chi tiết cơ bản

TÀI LIỆU THAM KHẢO

- [1] [Documentation | Android Developers](#)
- [2] [Xử lý sự kiện \(Event Handling\) trong Android | 100 bài tự học lập trình Android hay nhất \(vietjack.com\)](#)
- [3] [Bài 4: Các kiểu lập trình sự kiện trong Android - www.AndroidCoBan.Com](#)