



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Các nội dung mở rộng

Mục lục

1. **Cảm biến (sensor) trong thiết bị di động**
2. Các vấn đề về bảo mật cho ứng dụng
3. Chính sách về quyền riêng tư

Cảm biến (sensor) trong thiết bị di động

- Sensor: chip cảm ứng nằm trong thiết bị, cung cấp dữ liệu mà nó đo đạc được cho hệ điều hành
- Sử dụng cảm biến có thể tạo ra những ứng dụng hữu ích
- Các loại cảm biến thường gặp:

Loại cảm biến	Mô tả
Cảm biến chuyển động (Motion Sensors)	Những cảm biến này rất hữu ích để đo lực gia tốc và lực quay dọc theo ba trục. Bao gồm gia tốc kế, cảm biến trọng lực, con quay hồi chuyển và cảm biến vector quay.
Cảm biến môi trường (Environmental Sensors)	Những cảm biến này rất hữu ích để đo các thông số môi trường khác nhau, chẳng hạn như nhiệt độ và áp suất không khí xung quanh, độ chiếu sáng và độ ẩm. Bao gồm áp kế, quang kế và nhiệt kế.
Cảm biến vị trí (Position Sensors)	Những cảm biến này rất hữu ích để đo vị trí vật lý của thiết bị. Bao gồm cảm biến định hướng và từ kế.

3

Cảm biến (sensor) trong thiết bị di động

- Mỗi loại sensor có những đặc điểm vật lý khác nhau, muốn hiểu chính xác các chi tiết các sensor cần đọc tài liệu hướng dẫn (cần có kiến thức nhất định về vật lý và xử lý số liệu)
- Một số sensor là loại virtual (ảo), tức là kết quả được tính toán hoặc nội suy từ nguồn khác

Cảm biến (sensor) trong thiết bị di động

- Các lưu ý khi làm việc với sensor:
 - Nhất thiết phải giải phóng sensor khi không cần thiết, nếu không ứng dụng sẽ rất hao pin
 - Hệ thống không tự động tắt sensor kể cả khi tắt màn hình
 - Chú ý khi làm việc với các thông số 3D: các chiều có thể bị hoán đổi vị trí khi người sử dụng đặt thiết bị theo chiều âm (ví dụ: máy bị lật úp)
 - Nên kiểm thử trên thiết bị thật và hiệu chỉnh độ nhạy dần dần
 - Kết hợp nhiều sensor để thiết bị “nhạy cảm” hơn

Sensor trong Android

- Android SDK không có các class định sẵn cho từng loại sensor mà chỉ có TYPE của sensor, dữ liệu do sensor trả về là float (trường hợp cảm biến 1 đầu ra – chẳng hạn đo ánh sáng) hoặc float[] (trường hợp cảm biến nhiều đầu ra)
- Các loại sensor hiện được Android OS hỗ trợ:
 - TYPE_ACCELEROMETER: cảm biến gia tốc, đơn vị m/s^2
 - TYPE_AMBIENT_TEMPERATURE: cảm biến nhiệt độ môi trường, đơn vị đo là $0C$
 - TYPE_GRAVITY: cảm biến trọng lực
 - TYPE_GYROSCOPE: cảm biến con quay hồi chuyển, đơn vị rad/s
 - TYPE_LIGHT: cảm biến ánh sáng, đơn vị đo là lx

Sensor trong Android

- Các loại sensor hiện được Android OS hỗ trợ:
 - TYPE_HEART_RATE: cảm biến nhịp tim (mỗi phút)
 - TYPE_LINEAR_ACCELERATION: cảm biến gia tốc tuyến tính
 - TYPE_MAGNETIC_FIELD: cảm biến từ tính, đơn vị là μT
 - TYPE_PRESSURE: cảm biến áp suất, đơn vị đo là mbar
 - TYPE_PROXIMITY: cảm biến khoảng cách gần (tiệm cận), đơn vị đo là cm
 - TYPE_RELATIVE_HUMIDITY: cảm biến độ ẩm, đơn vị là %
 - TYPE_ROTATION_VECTOR: cảm biến xoay
 - TYPE_GAME_ROTATION_VECTOR: cảm biến xoay 2D
 - TYPE_SIGNIFICANT_MOTION: cảm biến chuyển động

Sensor trong Android

- Trong android, các sensor được quản lý chung bởi **SensorManager**, một dịch vụ hệ thống

SensorManager sm = (SensorManager) getSystemService(SENSOR_SERVICE);

- Thông qua SensorManager lập trình viên có thể:
 - Lấy danh sách các sensor có trong hệ thống hiện tại
 - Lấy đối tượng để làm việc trực tiếp với từng sensor
 - Đăng kí listener để xử lý sự kiện do các sensor báo về
- Muốn lấy một sensor cụ thể, sử dụng phương thức **getDefaultSensor(TYPE)**, tham số **TYPE** sẽ quy định kiểu đối tượng Sensor muốn lấy ra

Sensor trong Android

- Các bước làm việc với sensor:
 - 1. Lấy **SensorManager** từ các dịch vụ hệ thống, thông qua **getSystemService(Context.SENSOR_SERVICE)**
 - 2. Từ **SensorManager** lấy đối tượng Sensor điều khiển cảm biến cần sử dụng **sensor = sensorService.getDefaultSensor(TYPE);**
 - 3. Cài đặt các bộ nghe (listener) phù hợp để xử lý các số liệu do cảm biến trả về
 - 4. Tắt sensor trong những tình huống không cần thiết để tránh thiết bị tiêu tốn năng lượng
 - 5. Xử lý lỗi hoặc thay đổi độ nhạy của thiết bị

iOS Sensors

- Không phải mọi kiểu máy iPhone, iPad hoặc iPod touch đều có cùng các bộ cảm biến.
- Các tính năng được cung cấp bởi các cảm biến cũng dựa trên phiên bản hệ điều hành được cài đặt trên thiết bị.
- Các cảm biến phổ biến nhất được sử dụng trên iPhone:
 - **Cảm biến tiệm cận:** Cảm biến này xác định mức độ gần của iPhone với khuôn mặt. Nó giúp iPhone tự động tắt màn hình khi đưa điện thoại lên gần tai để gọi điện. Điều này là cần thiết để ngăn chặn tình trạng vô tình chạm vào nút trên mặt khi nói chuyện.
 - **Gia tốc kế / Cảm biến chuyển động:** Cảm biến này giúp màn hình tự động chuyển từ chế độ ngang sang dọc và quay lại một lần nữa dựa trên việc đang cầm điện thoại theo chiều dọc hay chiều ngang.

iOS Sensors

- Các cảm biến phổ biến nhất được sử dụng trên iPhone:
 - **Cảm biến ánh sáng xung quanh:** Điều này xác định lượng ánh sáng có sẵn trong khu vực xung quanh thiết bị. Với thông tin đó, thiết bị sẽ tự động điều chỉnh độ sáng màn hình để tiết kiệm pin và giảm mỏi mắt.
 - **Cảm biến độ ẩm:** Cảm biến này không hiển thị trong giao diện trên màn hình hoặc không thể truy cập thông qua iOS. Cảm biến độ ẩm (hoặc nước) là một chỉ báo nhỏ màu đỏ xuất hiện sau khi điện thoại bị ngập trong nước. Tùy thuộc vào kiểu máy, nó có thể xuất hiện trong Đầu nối Dock, cổng Lightning hoặc trong giắc cắm tai nghe.
 - **Con quay hồi chuyển:** Con quay hồi chuyển ba trục đã được đưa vào bắt đầu với iPhone 4. Bằng cách kết hợp con quay hồi chuyển với gia tốc kế, cảm biến này có thể hoạt động trên sáu trục. Điều này làm cho các thiết bị nhạy hơn và mạnh mẽ hơn để chơi game, vì chúng có thể phản ứng dựa trên cách chúng được cầm và di chuyển.

iOS Sensors

- Các cảm biến phổ biến nhất được sử dụng trên iPhone:
 - **La bàn:** Tất cả các mẫu iPhone bắt đầu với iPhone 3GS cũng có la bàn được tích hợp sẵn. Cảm biến này được sử dụng với tính năng GPS của thiết bị và công cụ nhận biết vị trí khác để giúp xác định vị trí iPhone.
 - **Phong vũ biểu (Barometer):** Khí áp kế đánh giá áp suất không khí để giúp xác định độ cao một phần. Cảm biến này được sử dụng cho các tính năng định vị và chỉ đường.
 - **Touch ID:** Cảm biến vân tay nhúng trong nút Trang chủ cho phép bạn truy cập an toàn vào thiết bị bằng dữ liệu sinh trắc học của mình.
 - **Face ID:** Được giới thiệu cùng với iPhone X, Face ID sử dụng một hệ thống phức tạp để nhận dạng khuôn mặt của một người nhằm cung cấp xác thực an toàn cho cả thiết bị và các dịch vụ

12

iOS Sensors

- **CoreMotion** kiểm soát quyền truy cập vào gia tốc kế và con quay hồi chuyển
- **CoreLocation** kiểm soát quyền truy cập vào GPS

Sensor trong React Native

- React Native cho phép kết nối giữa code native với code Javascript. React Native có thể can thiệp sâu vào phần cứng thông qua API native của nền tảng.
 - Tuy nhiên, điều này lại khiến developer cần phải chọn nhiều thư viện để thực hiện cùng một công việc cho các platform khác nhau. Nó làm mất đi tính chất cross-platform
- Thư viện **react-native-sensors** cung cấp interface cho cảm biến Accelerometer, Gyroscope và Magnetometer trên cả Android và iOS

• Ví dụ:

```
import React, { Component } from 'react';
import {
  StyleSheet,
  Text,
  View
} from 'react-native';
import { Accelerometer } from "react-native-sensors";
const Value = ({name, value}) => (
  <View style={styles.valueContainer}>
    <Text style={styles.valueName}>{name}</Text>
    <Text style={styles.valueValue}>{new String(value).substr(0, 8)}</Text>
  </View>
)
export default class App extends Component {
  constructor(props) {
    super(props);
    // Dòng 20
    new Accelerometer({
      updateInterval: 400 // defaults to 100ms
    })
    .then(observable => {
      observable.subscribe(({x,y,z}) => this.setState({x,y,z}));
    })
    .catch(error => {
      console.log("The sensor is not available");
```

```
this.state = {x: 0, y: 0, z: 0};
```

Accelerometer values

```
x: 1.057373
y: 7.747573
z: 5.562713
```

```
// Dòng 33
render() {
  return (
    <View style={styles.container}>
      <Text style={styles.headline}>
        Accelerometer values
      </Text>
      <Value name="x" value={this.state.x} />
      <Value name="y" value={this.state.y} />
      <Value name="z" value={this.state.z} />
    </View>
  );
}
```

15

Sensor trong Flutter

- Package: `sensors_plus` 1.2.1
- Một plugin Flutter để truy cập cảm biến gia tốc kế, con quay hồi chuyển và từ kế.

```
import 'package:sensors_plus/sensors_plus.dart';  
accelerometerEvents.listen((AccelerometerEvent event) { print(event); });  
// [AccelerometerEvent (x: 0.0, y: 9.8, z: 0.0)]  
userAccelerometerEvents.listen((UserAccelerometerEvent event) { print(event); });  
// [UserAccelerometerEvent (x: 0.0, y: 0.0, z: 0.0)]  
gyroscopeEvents.listen((GyroscopeEvent event) { print(event); });  
// [GyroscopeEvent (x: 0.0, y: 0.0, z: 0.0)]  
magnetometerEvents.listen((MagnetometerEvent event) { print(event); });  
// [MagnetometerEvent (x: -23.6, y: 6.2, z: -34.9)]
```


Mục lục

1. Cảm biến (sensor) trong thiết bị di động
2. **Các vấn đề về bảo mật cho ứng dụng**
3. Chính sách về quyền riêng tư

Cân nhắc về Bảo mật cho Thiết bị Di động

- Thiết bị lưu trữ thông tin cá nhân có giá trị
- Thiết bị giờ có dung lượng lớn hơn, nhiều ứng dụng để cài hơn
⇒ nhiều lỗ hổng hơn
 - Các mối đe dọa từ xưa nay vẫn còn (ví dụ: bảo mật kém trên các trình duyệt, đặc biệt là các trình duyệt trên điện thoại di động)
 - Ứng dụng đã cài đặt từ nguồn gây mất an toàn (nhiều ứng dụng hơn, khó tin cậy tác giả trong thị trường mở)
 - Chia sẻ giữa các ứng dụng.
 - Dữ liệu cá nhân bị bỏ lại trên hệ thống tệp.

Cần nhắc về Bảo mật cho Thiết bị Di động

- Thiết bị vốn đã kém an toàn hơn
 - Di động, dễ bị đánh cắp
 - Nhiều người dùng khác biết mật khẩu
 - Thường sử dụng mật khẩu yếu hơn (do khó nhập dữ liệu)
 - Kích thước màn hình hạn chế, môi trường xung quanh gây phiền nhiễu
⇒ người dùng thậm chí bỏ qua bảo mật
- Bài học: Các lập trình viên ứng dụng có vai trò lớn hơn trong việc bảo mật cho người dùng

Các bước có hệ thống để bảo mật ứng dụng

- Không triển khai ngẫu nhiên “security stuff”. Thay vào đó, hãy xác định mô hình mối đe dọa:
 - Tài nguyên (dữ liệu) của bạn là gì? Giá trị của chúng là gì?
 - Những cuộc tấn công nào có thể xảy ra (trộm cắp, DoS)? Chúng có thể bắt nguồn từ đâu (mạng, ứng dụng)?
- Xác định các chiến thuật bảo mật:
 - Phát hiện: Xác định rằng cuộc tấn công đang diễn ra (hoặc đã xảy ra mất mát)
 - Kháng cự: Làm cho việc mất mát khó xảy ra hơn.
 - Giảm thiểu: Giới hạn mức độ mất mát/vi phạm.
 - Khôi phục: Khôi phục ứng dụng/hệ điều hành về “trạng thái tốt”

Các bước có hệ thống để bảo mật ứng dụng (tiếp)

- Thực hiện các chiến thuật sử dụng các kỹ thuật bảo mật:
 - Xác thực (ví dụ: hai yếu tố, chứng chỉ)
 - Kiểm soát truy cập (ví dụ: quyền sở hữu tệp, mã hóa, chứng chỉ)
 - Dấu vết kiểm soát (ví dụ: nhật ký)
 - Tính toàn vẹn của dữ liệu (ví dụ: checksum, mã hóa)
 - Không giấu diếm các sai phạm (một khi phát hiện ra các thiếu sót bảo mật thì phải sửa chữa ngay)

Cần nhắc về bảo mật Android (tiếp)

- Những điều cần tránh:
 - Để lại dữ liệu riêng tư trong các tệp trên thiết bị thẻ SD (bộ nhớ ngoài)
 - Xây dựng ứng dụng có cơ sở dữ liệu nhưng không nghĩ đến SQL injection
 - Ứng dụng của bạn bị kẻ khác làm nhái và trở thành Trojan horse, một loại phần mềm ác tính
 - Sử dụng các thuật toán mã hóa có thể đảo ngược

* <https://jon.oberheide.org/files/summercon12-bouncer.pdf>

Ví dụ của Permission Requests

- `<uses-permission android:name="android.permission.READ_CONTACTS"/>`
- `<uses-permission android:name="android.permission.INTERNET"/>`
- `<uses-permission
 android:name="android.permission.ACCESS_NETWORK_STATE"/>`
- `<uses-permission
 android:name="android.permission.ACCESS_COARSE_LOCATION"/>`
- `<uses-permission
 android:name="android.permission.ACCESS_FINE_LOCATION"/>`
- `<uses-permission android:name=
 "com.wiley.fordummies.androidsdk.tictactoe.LAUNCHACTIVITY"/>`

Example of a custom permission

Lưu ý: Các phần tử phân quyền phải nằm ngoài khối `<application>` và bên trong khối `<manifest>` của `AndroidManifest.xml`

Quyền tùy chỉnh: Định nghĩa và Vị trí

➤ Quyền phải được khai báo:

```
<permission  
  android:name =  
    "com.wiley.fordummies.androidsdk.tictactoe.LAUNCHACTIVITY"  
  android:label="Launch Tic-Tac-Toe Activity"  
  android:description="@string/permission_launch_activity"  
  android:protectionLevel="normal" />
```

Đặt trong tệp AndroidManifest.xml bên ngoài khối <application>, bên trong khối <manifest> (giống như các phần tử <uses-permission>).

Quyền tùy chỉnh: Khai báo Cần, Yêu cầu

- Khai báo cần thông qua thuộc tính android: thuộc tính cho phép trong định nghĩa hoạt động trong tệp kê khai:

```
<activity  
    android:name=".Login"  
    android:label="@string/app_name"  
    android:launchMode="standard"  
    android:screenOrientation="portrait"  
    android:permission="...LAUNCHACTIVITY"/>
```

- Yêu cầu: <uses-permission android:name="...LAUNCHACTIVITY"/>
- Được yêu cầu trong bất kỳ gói riêng biệt hay gói chứa nào

Kiểm tra quyền trong Android

- Khi một cuộc gọi được thực hiện đến một chức năng hệ thống: Để ngăn chặn một lệnh gọi trái phép
- Khi bắt đầu một Activity: Để ngăn một ứng dụng trái phép khởi chạy Activity của các ứng dụng khác
- Khi gửi hoặc nhận chương trình Broadcast: Để xác định ai có thể nhận hoặc gửi Broadcast của bạn
- Khi truy cập và hoạt động trên Content Provider: Để ngăn một ứng dụng trái phép truy cập vào dữ liệu trong Content Provider
- Khi ràng buộc hoặc bắt đầu, một Service: Để ngăn một ứng dụng trái phép sử dụng Service.

Ví dụ về mục nhập *logcat* trong khi cấp quyền không thành công

```
02-28 12:48:00.864: ERROR/AndroidRuntime(378) :  
    java.lang.SecurityException: Permission Denial: starting Intent {  
        act=com.wiley.fordummies.androidsdk.tictactoe.Login  
        cmp=com.wiley.fordummies.androidsdk.tictactoe/.Login }  
    from ProcessRecord{407740c0  
        378:com.wiley.fordummies.androidsdk.tictactoe/10033} (pid=378,  
    uid=10033)  
    requires  
  
    com.wiley.fordummies.androidsdk.tictactoe.permission.LAUNCHACTIVITY  
02-28 21:04:39.758: ERROR/AndroidRuntime(914): at  
    com.wiley.fordummies.androidsdk.tictactoe.SplashScreen$1.run  
        (SplashScreen.java:36)
```

Các mục nhập *logcat* mẫu cho Định nghĩa Quyền hoặc Lỗi Vị trí

```
02-28 16:53:09.838: DEBUG/PackageManager(77): Permissions:  
com.wiley.fordummies.androidsdk.tictactoe.LAUNCHACTIVITY
```

```
02-28 17:04:18.888: WARN/PackageParser(77): Unknown element under  
<application>:
```

```
permission at /data/app/vmdl1654102309.tmp Binary XML file line #11
```

```
02-28 17:04:20.438: WARN/PackageManager(77): Unknown permission  
com.wiley.fordummies.androidsdk.tictactoe.LAUNCHACTIVITY in  
package  
com.wiley.fordummies.androidsdk.tictactoe
```

Kiểm tra quyền khi chạy ứng dụng (Android 6+)

- Một số quyền nhất định yêu cầu ủy quyền rõ ràng của người dùng trong thời gian chạy :

Permission Group	Permission
<u>CALENDAR</u>	<u>READ CALENDAR</u> , <u>WRITE CALENDAR</u>
<u>CAMERA</u>	<u>CAMERA</u>
<u>CONTACTS</u>	<u>READ CONTACTS</u> , <u>WRITE CONTACTS</u> , <u>GET ACCOUNTS</u>
<u>LOCATION</u>	<u>ACCESS_FINE_LOCATION</u> , <u>ACCESS_COARSE_LOCATION</u>
<u>MICROPHONE</u>	<u>RECORD_AUDIO</u>
<u>PHONE</u>	<u>READ_PHONE_STATE</u> , <u>CALL_PHONE</u> , <u>READ_CALL_LOG</u> , <u>WRITE_CALL_LOG</u> , <u>ADD_VOICEMAIL</u> , <u>USE_SIP</u> , <u>PROCESS_OUTGOING_CALLS</u>
<u>SENSORS</u>	<u>BODY_SENSORS</u>
<u>SMS</u>	<u>SEND_SMS</u> , <u>RECEIVE_SMS</u> , <u>READ_SMS</u> , <u>RECEIVE_WAP_PUSH</u> , <u>RECEIVE_MMS</u>
<u>STORAGE</u>	<u>READ_EXTERNAL_STORAGE</u> , <u>WRITE_EXTERNAL_STORAGE</u>

Nguồn: <https://developer.android.com/guide/topics/permissions/requesting.html>

Cần nhắc về bảo mật iOS

- Minh bạch về dữ liệu và tài nguyên liên quan đến quyền riêng tư:
 - Dữ liệu cá nhân, bao gồm vị trí, sức khỏe, tài chính, liên hệ và thông tin nhận dạng cá nhân khác
 - Nội dung do người dùng tạo như email, tin nhắn, dữ liệu lịch, danh bạ, thông tin trò chơi, hoạt động Apple Music, dữ liệu HomeKit và nội dung âm thanh, video và ảnh
 - Các tài nguyên được bảo vệ như thiết bị ngoại vi Bluetooth, tính năng tự động hóa gia đình, kết nối Wi-Fi và mạng cục bộ
 - Các khả năng của thiết bị như máy ảnh và micrô

* <https://support.apple.com/vi-vn/guide/security/welcome/web>

Cần nhắc về bảo mật iOS

- permission in Info.plist:

Limited Photos

```
<key>PHPhotoLibraryPreventAutomaticLimitedAccessAlert</key> <true/>
```

Camera

```
<key>NSCameraUsageDescription</key> <string>$(PRODUCT_NAME) camera description.</string>
```

Photos

```
<key>NSPhotoLibraryUsageDescription</key> <string>$(PRODUCT_NAME)photos description.</string>
```

Save Photos

```
<key>NSPhotoLibraryAddUsageDescription</key> <string>$(PRODUCT_NAME) photos add description.</string>
```

Location

```
<key> NSLocationWhenInUseUsageDescription</key> <string>$(PRODUCT_NAME) location description.</string>
```

Apple Music

```
<key>NSAppleMusicUsageDescription</key> <string>$(PRODUCT_NAME) My description about why I need this capability</string>
```

Calendar

```
<key>NSCalendarsUsageDescription</key> <string>$(PRODUCT_NAME) My description about why I need this capability</string>
```

Siri

```
<key>NSSiriUsageDescription</key> <string>$(PRODUCT_NAME) My description about why I need this capability</string>
```

31

Bảo mật SQLite: SQL Injection

- Trường nhập: Name: <Enter Name>

- Truy vấn dự định:

```
SELECT e-mail FROM user_information WHERE NAME = 'Bob'
```

- Kẻ tấn công nhập chuỗi:

```
'Bob';SELECT table_names FROM user_tables
```

- Truy vấn trở thành:

```
SELECT e-mail FROM user_information WHERE name = 'Bob'; SELECT  
table_names FROM user_tables
```

- Kẻ tấn công biết tất cả các bảng.

Giải pháp chống SQL Injection :

```
// AccountDbSchema.java

public class AccountDbSchema {
    public static final class AccountsTable {
        public static final String NAME = "accounts";
        public static final class Cols { /* Name and password columns */ }
    } }

// AccountSingleton.java

private static final String INSERT_STMT = "INSERT INTO " + AccountsTable.NAME +
    " (name, password) VALUES (?, ?)" ;
// ...

// Account model object includes name, password fields to insert into DB

public void addAccount(Account account) {
    ContentValues contentValues = getContentValues(account);
    mDatabase.beginTransaction();
    try {
        SQLiteStatement statement = mDatabase.compileStatement(INSERT_STMT);
        statement.bindString(1, account.getName());
        statement.bindString(2, account.getPassword());
        statement.executeInsert(); mDatabase.setTransactionSuccessful();
    } finally { mDatabase.endTransaction(); }
}
// . . .
```

Quy tắc chung: Giảm thiểu lỗ hổng ứng dụng

- Đừng vội công khai mã nguồn
- Che dấu dữ liệu nhạy cảm (ví dụ: mật khẩu)
- Mã hóa các tệp nhạy cảm
- Không ghi các tệp tạm thời không cần thiết
- Yêu cầu ít quyền nhất
- Tạo điểm kiểm tra dữ liệu ứng dụng
- Ghi nhật ký dữ liệu (cũng mã hóa nhật ký của bạn!)
- Giữ các Intent Filter để Activity không phản hồi với các Intent chung chung
- Nhắc người dùng cấp quyền truy cập dữ liệu nhạy cảm

Xử lý các yêu cầu quyền trong Flutter

- Package: **permission_handler**

```
import 'package:permission_handler/permission_handler.dart';
```

```
final PermissionHandler _permissionHandler = PermissionHandler();  
var result = await  
_permissionHandler.requestPermissions([PermissionGroup.contacts]);
```

- Các trạng thái quyền có thể có là:

Status	Why
Granted	Your app has been given permission to use the feature in question
Denied	Your app has been given permission to use the feature in question
Disabled	Feature in question is disabled i.e. when the Location feature is disabled.
Restricted (iOS only)	OS has restricted access to a specific feature due to something like parental controls.
Unknown	Unknown state of permissions.

35

Xử lý các yêu cầu quyền trong ReactNative

- Package: **react-native-permissions**

```
import React, { useState, useEffect } from 'react'
import { PERMISSIONS, check, request } from 'react-native-permissions'
const App = (props) => {
  const [cameraGranted, setCameraGranted] = useState(false);
  const handleCameraPermission = async () => {
    const res = await check(PERMISSIONS.IOS.CAMERA);

    if (res === RESULTS.GRANTED) {
      setCameraGranted(true);
    } else if (res === RESULTS.DENIED) {
      const res2 = await request(PERMISSIONS.IOS.CAMERA);
      res2 === RESULTS.GRANTED
        ? setCameraGranted(true);
        : setCameraGranted(false);
    }
  };

  useEffect(() => {
    handleCameraPermission();
  }, []);
  return (
    {cameraGranted
      ? <Text>Camera Granted! Display in-app camera...</Text>
      : <Text>Camera Disallowed</Text>
    }
  );
};
```

Mục lục

1. Cảm biến (sensor) trong thiết bị di động
2. Các vấn đề về bảo mật cho ứng dụng
3. **Chính sách về quyền riêng tư**

Khái niệm

- Chính sách quyền riêng tư là tài liệu được tạo để đi kèm với sản phẩm (ứng dụng, trang web, v.v.) mô tả sản phẩm và công ty đứng sau sản phẩm đó sẽ làm thế nào với khách hàng hoặc dữ liệu của khách hàng:
 - Thu thập
 - Sử dụng
 - Tiết lộ = công khai
 - Quản lý

Khái niệm (2)

Cách tạo ra chính sách quyền riêng tư

- Hãy tự đặt cho mình một vài câu hỏi sau nhé:
 - Bạn cần thu thập những dữ liệu nào?
 - Thu thập thế nào?
 - Bạn sẽ làm gì/có thể làm gì với những dữ liệu đó ?
 - Điều gì sẽ xảy ra với những dữ liệu đó sau khoảng thời gian X?
 - Đây có phải là những dữ liệu ẩn danh?
 - Có quảng cáo không?
 - Dữ liệu này có được chia sẻ với tổ chức nào khác không?

Khái niệm (3)

Nhìn chung chính sách quyền riêng tư phải giải quyết được các vấn đề sau:

- Thông tin
- Lựa chọn
- Truy cập
- Bảo mật
- Đền bù/bồi thường
- Cập nhật

Ví dụ:

- Google: <https://www.google.com/policies/privacy/>
- Apple: <http://www.apple.com/legal/privacy/en-ww/>
- Facebook: <https://www.facebook.com/policy.php>
- Twitter: <https://twitter.com/privacy?lang=en>

Ví dụ (2)

- Lưu ý các ví dụ về chính sách quyền riêng tư chủ yếu bằng "tiếng Anh thông thường, đơn giản dễ hiểu!"
- Không có thuật ngữ chuyên sâu hoặc liên quan đến luật
- Một số chính sách quyền riêng tư đã được tự động xử lý/cài đặt

Nguyên nhân

Chúng ta cần chính sách quyền riêng tư vì:

- Bạn đang thu thập dữ liệu cá nhân
- Bạn đang sử dụng dịch vụ của bên thứ ba
- Quy định của chính phủ
- Quy định của kho ứng dụng
- Nguy cơ/Rủi ro chuyển nhượng khách hàng
- Có thể bị kiện tụng
- Tham khảo thêm tại <http://www.iubenda.com/en/mobile>

Nguyên nhân (2)

Có gì trong chính sách quyền riêng tư:

- **Thông tin** - thông tin cá nhân nào đang được thu thập trên trang web
- **Lựa chọn** - khách hàng có những lựa chọn nào/dữ liệu của họ có được thu thập và sử dụng hay không/nếu có thì như thế nào
- **Quyền truy cập** - khách hàng có thể xem dữ liệu nào đã được thu thập và thay đổi như thế nào/chỉnh sửa nếu cần

Nguyên nhân (3)

Có gì trong chính sách quyền riêng tư (tiếp):

- **Bảo mật** – nêu rõ mọi dữ liệu được thu thập được lưu trữ/bảo vệ như thế nào.
- **Đền bù/bồi thường** - khách hàng có thể làm gì nếu chính sách quyền riêng tư không được đáp ứng
- **Cập nhật** - thông báo những thay đổi về chính sách như thế nào

Nguyên nhân (4)

Chính sách quyền riêng tư mang đến cho bạn điều gì?

- Tiết lộ những gì đang xảy ra
- Mức độ tin cậy với nhà phát triển
- Đáp ứng các yêu cầu từ nhà xuất bản/cơ quan chính phủ
- Yêu cầu phân tích của Google :

<https://support.google.com/analytics/answer/2700409?hl=en>

Nguyên nhân (5)

- Hãy nhớ rằng: quyền riêng tư không HẢN là bảo mật
- Chính sách quyền riêng tư chỉ ra những gì bạn đang thu thập và những gì bạn dự định làm
- Và bạn PHẢI bảo vệ dữ liệu bạn thu thập! Nếu không bảo vệ được tốt thì tốt nhất đừng thu thập



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you
for your
attentions!**

