|  |  |
| --- | --- |
| **TRƯỜNG ĐHKTCN** | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM** |
| **KHOA ĐIỆN TỬ** | ***Độc lập - Tự do - Hạnh phúc*** |

**BÁO CÁO CUỐI KỲ**

Môn học : Lập trình trên thiết bị di động

Bộ môn : Công nghệ thông tin

Sinh viên : Hà Huy Nam

MSSV : K215480106063

Lớp : 57KMT.01

Ngành : Kỹ thuật máy tính

Giáo viên hướng dẫn : Đỗ Duy Cốp

Ngày giao đề

Ngày hoàn thành

Yêu cầu :

bài tập 1: app đơn giản

bài tập 2: app sử dụng thông báo

bài tập 3: mit app inventor

|  |
| --- |
| **GIÁO VIÊN HƯỚNG DẪN** |
| *(Ký và ghi rõ họ tên)* |

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Thái Nguyên, ngày 6 tháng 3 năm 2025*

**GIÁO VIÊN HƯỚNG DẪN**

*(Ký ghi rõ họ tên)*

# **I. MỤC LỤC**

[**I. MỤC LỤC** 3](#_Toc191641363)

[**II. DANH MỤC HÌNH ẢNH** 5](#_Toc191641364)

[II**I. LỜI CẢM ƠN** 6](#_Toc191641365)

[**IV. LỜI NÓI ĐẦU** 7](#_Toc191641366)

[**V. NỘI DUNG BÀI TẬP LỚN** 8](#_Toc191641367)

[**1. Giới thiệu về lập trình trên thiết bị di động** 8](#_Toc191641368)

[**2. Lập trình Android** 9](#_Toc191641369)

[**3. Bài tập 1: App đơn giản** 11](#_Toc191641370)

[**3.1 Giới thiệu** 11](#_Toc191641371)

[**3.2 Phân tích yêu cầu** 11](#_Toc191641372)

[**3.3 Hướng xây dựng phần mềm** 11](#_Toc191641373)

[**3.4 Quá trình xây dựng phần mềm** 11](#_Toc191641374)

[**3.5 File Manifest (xml)** 14](#_Toc191641375)

[**3.6 File Main Activity (java)** 16](#_Toc191641376)

[**3.7 File Tiền phòng (java)** 19](#_Toc191641377)

[**3.8 File Web (java)** 20](#_Toc191641378)

[**3.9 Kết quả** 24](#_Toc191641379)

[**3.10 Đánh giá** 27](#_Toc191641380)

[**3.11 Link Github** 27](#_Toc191641381)

[**4. Bài tập 2: App sử dụng thông báo** 28](#_Toc191641382)

[**4.1 Giới thiệu** 28](#_Toc191641383)

[**4.2 Phân tích yêu cầu** 28](#_Toc191641384)

[**4.3 Hướng xây dựng phần mềm** 28](#_Toc191641385)

[**4.4 File Manifest (xml)** 28](#_Toc191641386)

[**4.5 File Main Activity (java)** 29](#_Toc191641387)

[**4.7 File ApiService (java)** 31](#_Toc191641388)

[**4.8 File Post (java)** 32](#_Toc191641389)

[**4.9 File RetrofitInsctance (java)** 32](#_Toc191641390)

[**4.10 Kết quả** 34](#_Toc191641391)

[**4.11 Đánh giá** 35](#_Toc191641392)

[**4.12 Link Github** 35](#_Toc191641393)

[**5. Bài tập 3: MIT App Inventor** 36](#_Toc191641394)

[**5.1 Giới thiệu** 36](#_Toc191641395)

[**5.2 Phân tích yêu cầu** 36](#_Toc191641396)

[**5.3 Hướng xây dựng phần mềm** 37](#_Toc191641397)

[**5.4 Tạo giao diện** 37](#_Toc191641398)

[**5.5 Thực hiện logic** 39](#_Toc191641399)

[**5.6 Kết quả** 41](#_Toc191641400)

[**5.7 Đánh giá** 42](#_Toc191641401)

[**5.8 Link Github** 42](#_Toc191641402)

[**VI. TÀI LIỆU THAM KHẢO** 43](#_Toc191641403)

# **II. DANH MỤC HÌNH ẢNH**

Hình 1.1 Sử dụng phần mềm trên thiết bị di động

Hình 1.2 Android với mọi lĩnh vực đời sống

Hình 2.1 Hệ điều hành Android

Hình 2.2 Ngôn ngữ Java

Hình 2.3 Android IDE

Hình 2.4 IntelliJ IDE

Hình 3.1 Chọn “Empty Activity”

Hình 3.2 Chọn Groovy DSL – build.gradle, API 24 – Android 7.0

Hình 3.3 Màn hình làm việc

Hình 3.4 File Manifest

Hình 3.5 Giao diện khi tính tiền phòng

Hình 3.6 Kết quả khi chọn phòng và số giờ

Hình 3.7 Giao diện “xem phim”

Hình 4.1 Kết quả khi chọn “Thông Báo 1”

Hình 4.2 Kết quả khi chọn “Thông Báo 2”

Hình 5.1 MIT App Inventor

Hình 5.2 Menu

Hình 5.3 Giải toán

Hình 5.4 Thông tin cá nhân và nghe nhạc

Hình 5.5 Logic Menu

Hình 5.6 Logic Nghe nhạc

Hình 5.7 Logic tính toán

Hình 5.8 Trang Menu

Hình 5.9 Trang Thông tin cá nhân và nghe nhạc

Hình 5.10 Trang Giải toán

# II**I. LỜI CẢM ƠN**

Trước hết, tôi xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy giáo Đỗ Duy Cốp, người đã hướng dẫn và hỗ trợ tận tình trong suốt quá trình học tập và nghiên cứu môn Lập trình trên thiết bị di động. Nhờ sự chỉ dẫn chu đáo và những lời khuyên quý báu của thầy, tôi đã có được những kiến thức nền tảng vững chắc và sự tự tin để thực hiện đề tài này

Thầy không chỉ giúp chúng tôi hiểu rõ hơn về các khái niệm phức tạp trong lĩnh vực lập trình android mà còn truyền cảm hứng để tôi khám phá những ứng dụng thực tiễn của công nghệ này trong cuộc sống. Sự tận tâm và nhiệt huyết của thầy chính là động lực lớn lao giúp tôi yêu thích hơn công việc lập trình phần mềm

Tôi cũng xin bày tỏ lòng biết ơn đến toàn thể giảng viên và các đồng môn đã luôn sẵn sàng chia sẻ kiến thức và kinh nghiệm trong suốt quá trình học tập và nghiên cứu

Tôi xin cam đoan rằng toàn bộ nội dung trong báo cáo này được thực hiện dựa trên sự tìm hiểu, nghiên cứu từ các nguồn tài liệu uy tín, bao gồm sách, bài báo khoa học, tài liệu học thuật, các website chính thống

Ký tên sinh viên

Hà Huy Nam

# **IV. LỜI NÓI ĐẦU**

Trong thời đại công nghệ số phát triển mạnh mẽ, các thiết bị di động đã trở thành một phần không thể thiếu trong cuộc sống con người. Với sự phổ biến rộng rãi của điện thoại thông minh và máy tính bảng, nhu cầu phát triển các ứng dụng di động ngày càng gia tăng, phục vụ nhiều lĩnh vực như giải trí, giáo dục, thương mại điện tử, y tế và tự động hóa. Trong đó, nền tảng Android chiếm lĩnh thị phần lớn nhờ tính linh hoạt, mã nguồn mở và khả năng tương thích cao với nhiều dòng thiết bị khác nhau

Việc xây dựng một ứng dụng trên Android không chỉ yêu cầu kiến thức về ngôn ngữ lập trình như Java, Kotlin mà còn đòi hỏi hiểu biết sâu về giao diện người dùng, cơ sở dữ liệu, quản lý tài nguyên và tối ưu hóa hiệu suất ứng dụng

Báo cáo này sẽ trình bày về các bài tập đã học trong chương trình môn học Lập trình trên thiết bị di động (Android). Mặc dù đã cố gắng hoàn thiện, nhưng không thể tránh khỏi những thiếu sót, tôi rất mong nhận được sự góp ý để có thể cải thiện hơn trong các nghiên cứu tiếp theo

# **V. NỘI DUNG BÀI TẬP LỚN**

# **1. Giới thiệu về lập trình trên thiết bị di động**

Lập trình di động là quá trình tạo ra các ứng dụng hoặc phần mềm chạy trên các thiết bị di động, chẳng hạn như điện thoại thông minh và máy tính bảng, thông qua nền tảng hoặc hệ điều hành hỗ trợ (Android hoặc iOS)



Hình 1.1 Sử dụng phần mềm trên thiết bị di động (ảnh minh họa)

Quá trình này bao gồm mã hóa, thiết kế giao diện người dùng (UI), thử nghiệm ứng dụng và triển khai ứng dụng lên các cửa hàng ứng dụng như App Store và Google Play. Với sự tiến bộ liên tục của công nghệ, lập trình di động đã trở thành một lĩnh vực quan trọng, mang đến nhiều cơ hội nghề nghiệp hấp dẫn. Đây là một phần thiết yếu của công nghệ thông tin hiện đại, nơi kết nối và sự tiện lợi của thiết bị di động là điều cần thiết trong cuộc sống hàng ngày



Hình 1.2 Android với mọi lĩnh vực đời sống

# **2. Lập trình Android**

Lập trình Android là quá trình phát triển các ứng dụng di động chạy trên hệ điều hành Android. Được Google ra mắt vào năm 2008, Android đã trở thành hệ điều hành phổ biến nhất cho các thiết bị di động như điện thoại thông minh và máy tính bảng. Phát triển ứng dụng Android bao gồm viết mã, thiết kế giao diện người dùng và tích hợp nhiều tính năng khác nhau để tạo ra các ứng dụng hữu ích cho người dùng



Hình 2.1 Hệ điều hành Android

Mặc dù Android hỗ trợ nhiều ngôn ngữ lập trình nhưng ngôn ngữ lập trình chủ yếu phổ biến để phát triển ứng dụng là Java



Hình 2.2 Ngôn ngữ Java

Java là ngôn ngữ lâp trình đầu tiên trên nền tảng Android. Nhờ tính ổn định và hệ sinh thái phong phú, đến nay Java vẫn là sự lựa chọn phổ biến nhất đối với nhiều nhà phát triển

Để lập trình ứng dụng Android, chúng ta cần IDE – môi trường tích hợp dùng để viết code để phát triển ứng dụng

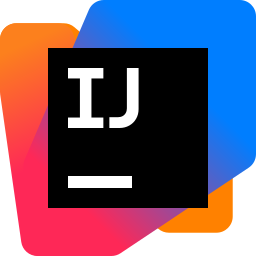
A blue and black logo

AI-generated content may be incorrect.

Hình 2.3 Android IDE

Android studio là môi trường phát triển tích hợp IDE chính thức cho hệ điều hành Android của Google, được xây dựng nền của IntelliJ IDE của JetBrains và được thiết kế dành riêng cho phát triển Android

Trong khuôn khổ bài báo cáo này sẽ sử dụng phiên bản Android studio Ladybug | 2024.2.1, là phiên bản kế tiếp Koala, ra mắt tháng 9/2024. Phiên bản Ladybug nâng cấp giao diện mặc định mới, với thiết kế UI hợp lý và các tính năng nâng cao, giúp cả nhà phát triển mới hay dày dặn kinh nghiệm điều hướng quy trình làm việc của họ hiệu quả hơn



Hình 2.4 IntelliJ IDE

Qua phần giới thiệu ngắn gọn các công cụ dành cho phát triển Android trên đã giúp người đọc có sự chuẩn bị làm quen với các công cụ cần thiết để bước chân vào giới lập trình phần mềm Android phục vụ cho nhu cầu người dùng hay giải quyết các bài toán thực tế cho doanh nghiệp

# **3. Bài tập 1: App đơn giản**

## **3.1 Giới thiệu**

Đây là bài tập đầu tiên để khởi động với môn Lập trình trên thiết bị di động. Với bài này, mục đích chính là giúp cho sinh viên có những cái nhìn đầu tiên, “tiếp xúc” với ngôn ngữ Java/ Kotlin – ngôn ngữ cho xây dựng ứng dụng trên thiết bị di động hiện nay

## **3.2 Phân tích yêu cầu**

Với yêu cầu: xây dựng phần mềm đơn giản – Simple App cần các yêu cầu cơ bản của một ứng dụng app thường thấy, phục vụ cho mục đích nhất định, có giao diện thân thiện phù hợp người dùng, dễ sử dụng, tính năng vừa đủ, ổn định

## **3.3 Hướng xây dựng phần mềm**

Phần mềm sẽ có các chức năng sau: Tính tiền phòng trọ, hiển thị video

## **3.4 Quá trình xây dựng phần mềm**

Dưới đây là mô tả chi tiết quá trình xây dựng phần mềm Android “Simple App” từ đầu

Vẫn như mọi project Android khác, đầu tiên chúng ta vào Android Studio, thực hiện các thao tác như mở project mới, lựa chọn java/kotlin, API level, empty activity

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.1 Chọn “Empty Activiy”

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.2 Chọn Groovy DSL – build.gradle, API 24 – Android 7.0

A screenshot of a computer program

AI-generated content may be incorrect.

Hình 3.3 Màn hình làm việc

Sau khi bấm “Finish” sẽ thấy giao diện của Android studio, sau đó đợi hệ thống load các file build-in sẵn có cần thiết, cài đặt emulator để run test ngay trong android studio để đến bắt đầu cuộc xây dựng “Simple App”

A screen shot of a computer program

AI-generated content may be incorrect.

Hình 3.4 File Manifest

Cần vào file manifest để cấu hình chung cho project. File manifest trong Android studio là một tệp quan trọng chứa thông tin cơ bản của ứng dụng, định nghĩa cấu trúc và hành vi của ứng dụng, gồm: tên package để xác định ứng dụng như “project.hahuynam.simpleapp”, các activity như màn hình chính, màn hình các chức năng. Quyền: Xác định ứng dụng cần những quyền gì, cơ bản có những quyền sau: truy cập bộ nhớ thiết bị, internet

Tổng quan ứng dụng: ứng dụng “Simple App” cho phép người dùng chọn loại phòng: cao cấp(200k/h), thường(100k/h). Người dùng nhập số giờ thuê phòng(tăng/ giảm bằng nút bấm), nhấn nút “OK” để tính tiền dựa trên loại phòng và số giờ. Có các nút “Reset” để người dùng đặt lại các lựa chọn, nút “Quay lại”, nút mở “Web” xem phim

## **3.5 File Manifest (xml)**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.SimpleApp"  
 tools:targetApi="35">  
  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 <activity  
 android:name=".Web"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

\*Những điểm cần chú ý:

* Cấu trúc tổng quát: xác định phiên bản XML version=1.0, encoding=utf-8. Namespace “android”: để dùng các thuộc tính có sẵn cho android, “tools”: cung cấp công cụ hỗ trợ trong quá trình phát triển ứng dụng
* Các khai báo trong block application: android:allowBackup="true": back-up dữ liệu khi khôi phục hoặc cài đặt lại
* Định nghĩa các quy tắc trích xuất dữ liệu khi backup

“android:dataExtractionRules="@xml/data\_extraction\_rules"”

* Đặt icon chính của ứng dụng

android:icon="@mipmap/ic\_launcher"

* Đặt tên ứng dụng theo chuỗi được định nghĩa trong res/values/strings.xml

android:label="@string/app\_name":

* Đặt icon tròn cho ứng dụng

android:roundIcon="@mipmap/ic\_launcher\_round"

* áp dụng theme Theme.SimpleApp từ file res/values/styles.xml

android:theme="@style/Theme.SimpleApp":

* Chỉ định phiên bản API tối đa mà ứng dụng hướng tới

tools:targetApi="35"

* khai báo MainActivity là một màn hình trong ứng dụng

android:name=".MainActivity"

* Cho phép MainActivity được truy cập từ các thành phần khác bên ngoài ứng dụng

android:exported="true"

* <intent-filter>: Xác định rằng MainActivity là màn hình chính của ứng dụng. <action android:name="android.intent.action.MAIN" />: Xác định đây là Activity khởi động chính của ứng dụng. Biến MainActivity thành entry point (điểm vào) của ứng dụng, hiển thị trên màn hình chính của điện thoại. android:name=".Web": Khai báo Web.java là một Activity khác trong ứng dụng. android:exported="true":cho phép Web được truy cập từ bên ngoài ứng dụng

## **3.6 File Main Activity (java)**

package project.hahuynam.simpleapp;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.RadioButton;  
import android.widget.RadioGroup;  
import android.widget.TextView;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class MainActivity extends AppCompatActivity {  
 private Button buttonQuayLai, buttonWeb, buttonIncrease, buttonDecrease, buttonOK, buttonReset;  
 private RadioGroup radioGroup;  
 private RadioButton radioButtonCaoCap, radioButtonThuong;  
 private TextView textSoGio, textHienThiSoGio, textGiaPhong, textSoPhong;  
 private int soGio = 0;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 // Ánh xạ các thành phần giao diện  
 buttonQuayLai = findViewById(R.id.*Button\_QuayLai*);  
 buttonWeb = findViewById(R.id.*Button\_Web*);  
 buttonIncrease = findViewById(R.id.*Button\_Increase*);  
 buttonDecrease = findViewById(R.id.*Button\_Decrease*);  
 buttonOK = findViewById(R.id.*Button\_OK*);  
 buttonReset = findViewById(R.id.*button11*);  
 radioGroup = findViewById(R.id.*RadioGroup*);  
 radioButtonCaoCap = findViewById(R.id.*RadioButton\_CaoCap*);  
 radioButtonThuong = findViewById(R.id.*RadioButton\_Thuong*);  
 textSoGio = findViewById(R.id.*Text\_SoGio*);  
 textHienThiSoGio = findViewById(R.id.*Text\_HienThiSoGio*);  
 textGiaPhong = findViewById(R.id.*textView29*);  
 textSoPhong = findViewById(R.id.*textView28*);  
  
 // Thiết lập sự kiện bấm nút tăng số giờ  
 buttonIncrease.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 soGio++;  
 textHienThiSoGio.setText(String.*valueOf*(soGio));  
 }  
 });  
 // Thiết lập sự kiện bấm nút giảm số giờ  
 buttonDecrease.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if (soGio > 0) {  
 soGio--;  
 }  
 textHienThiSoGio.setText(String.*valueOf*(soGio));  
 }  
 });  
 // Thiết lập sự kiện bấm nút OK  
 buttonOK.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 String loaiPhong = "";  
 int giaPhong = 0;  
  
 if (radioButtonCaoCap.isChecked()) {  
 loaiPhong = "Cao cấp";  
 giaPhong = 200; // Giá phòng cao cấp mỗi giờ  
 } else if (radioButtonThuong.isChecked()) {  
 loaiPhong = "Thường";  
 giaPhong = 100; // Giá phòng thường mỗi giờ  
 }  
 int tongTien = giaPhong \* soGio;  
  
 textGiaPhong.setText("Tổng giá: " + tongTien + " VND");  
 textSoPhong.setText("Loại phòng: " + loaiPhong);  
  
 Toast.makeText(MainActivity.this, "Đã tính toán!", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 // Thiết lập sự kiện bấm nút quay lại  
 buttonQuayLai.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Toast.makeText(MainActivity.this, "Quay lại được chọn", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 // Thiết lập sự kiện bấm nút web  
 buttonWeb.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(MainActivity.this, Web.class);  
 startActivity(intent);  
 }  
 });  
  
 // Thiết lập sự kiện bấm nút reset  
 buttonReset.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 // Reset dữ liệu  
 soGio = 0;  
 radioGroup.clearCheck();  
 textHienThiSoGio.setText(String.*valueOf*(soGio));  
 textGiaPhong.setText("");  
 textSoPhong.setText("");  
 Toast.makeText(MainActivity.this, "Dữ liệu đã được reset", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
}

\*Những điểm cần chú ý:

* Các nút bấm: buttonQuayLai, buttonWeb, buttonIncrease, buttonDecrease, buttonOK, buttonReset
* Nhóm chọn loại phòng: radioGroup chứa hai radio button radioButtonCaoCap và radioButtonThuong
* TextView hiển thị thông tin: textSoGio (số giờ), textHienThiSoGio (số giờ hiển thị), textGiaPhong (tổng giá), textSoPhong (loại phòng)
* Biến số giờ: soGio = 0 (mặc định ban đầu là 0)
* Khi ứng dụng chạy, giao diện activity\_main.xml sẽ được hiển thị
* Ánh xạ các thành phần trong giao diện. Liên kết các biến trong Java với các thành phần trong “activity\_main.xml” để có thể thao tác với chúng
* Số giờ: Khi nhấn nút Increase, biến soGio tăng 1, Cập nhật giá trị mới lên “textHienThiSoGio”. Giảm số giờ: Khi nhấn Decrease, nếu soGio > 0 thì giảm 1, Khi nhấn Decrease, nếu soGio > 0 thì giảm 1
* Tính tổng tiền(Nút “OK”): Kiểm tra người dùng chọn loại phòng nào. Nếu chọn: “cao cấp” – giá mỗi giờ = 200VND, “thường” – giá mỗi giờ = 100VND (mức giá minh họa). Cập nhật giá phòng và loại phòng lên layout giao diện. Hiển thị thông báo “Quay lại được chọn” dạng android pop-up (giống dạng thông báo mặc định từ hệ thống)

## **3.7 File Tiền phòng (java)**

package project.hahuynam.simpleapp;  
  
public class TienPhong {  
  
 private float soGio;  
 private String loaiPhong;  
  
 // Phương thức tính toán tiền phòng dựa trên loại phòng và số giờ thuê  
 public double tinhTienPhong() {  
 if (loaiPhong == null || loaiPhong.isEmpty()) {  
 throw new IllegalArgumentException("Loại phòng không được để trống");  
 }  
  
 if (loaiPhong.equalsIgnoreCase("thuong")) {  
 return soGio \* 100000;  
 } else if (loaiPhong.equalsIgnoreCase("vip")) {  
 return soGio \* 500000;  
 }  
  
 // Trường hợp loại phòng không hợp lệ  
 throw new IllegalArgumentException("Loại phòng không hợp lệ: " + loaiPhong);  
 }  
  
 // Getter và Setter cho số giờ  
 public float getSoGio() {  
 return soGio;  
 }  
  
 public void setSoGio(float soGio) {  
 this.soGio = soGio;  
 }  
  
 // Getter và Setter cho loại phòng  
 public String getLoaiPhong() {  
 return loaiPhong;  
 }  
  
 public void setLoaiPhong(String loaiPhong) {  
 this.loaiPhong = loaiPhong;  
 }  
}

## **3.8 File Web (java)**

package project.hahuynam.simpleapp;  
  
import android.content.Intent;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.net.Uri;  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.VideoView;  
import androidx.appcompat.app.AppCompatActivity;  
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.IOException;  
import java.io.InputStream;  
  
public class Web extends AppCompatActivity {  
  
 private VideoView videoView;  
 private Button buttonPlay, buttonStop, buttonResume, buttonReplay, buttonReturn;  
 private TextView textLuotXem;  
 private int luotXem = 0;  
 private ImageView imageView;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_web);  
  
 // Ánh xạ các thành phần giao diện  
 videoView = findViewById(R.id.videoView);  
 buttonPlay = findViewById(R.id.playMusicButton);  
 buttonStop = findViewById(R.id.button\_stop2);  
 buttonResume = findViewById(R.id.button\_resume);  
 buttonReplay = findViewById(R.id.button\_replay);  
 buttonReturn = findViewById(R.id.Button\_Return);  
 textLuotXem = findViewById(R.id.LuotXem);  
 imageView = findViewById(R.id.imageView);  
  
 // Hiển thị hình ảnh từ thư mục assets  
 try {  
 InputStream imageStream = getAssets().open("image/netflix\_logo.png"); // Đường dẫn tới hình ảnh trong thư mục assets  
 Bitmap bitmap = BitmapFactory.decodeStream(imageStream);  
 imageView.setImageBitmap(bitmap);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 // Hiển thị video từ thư mục assets  
 try {  
 InputStream videoStream = getAssets().open("video/alice\_inborderland\_3\_trailer.mp4");  
 File tempFile = File.createTempFile("tempVideo", ".mp4", getCacheDir());  
 try (FileOutputStream outStream = new FileOutputStream(tempFile)) {  
 byte[] buffer = new byte[1024];  
 int length;  
 while ((length = videoStream.read(buffer)) != -1) {  
 outStream.write(buffer, 0, length);  
 }  
 }  
 videoStream.close();  
 Uri videoUri = Uri.fromFile(tempFile);  
 videoView.setVideoURI(videoUri);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 // Thiết lập sự kiện bấm nút "Phát"  
 buttonPlay.setOnClickListener(v -> videoView.start());  
  
 // Thiết lập sự kiện bấm nút "Dừng"  
 buttonStop.setOnClickListener(v -> videoView.pause());  
  
 // Thiết lập sự kiện bấm nút "Tiếp tục"  
 buttonResume.setOnClickListener(v -> videoView.start());  
  
 // Thiết lập sự kiện bấm nút "Phát lại"  
 buttonReplay.setOnClickListener(v -> {  
 videoView.seekTo(0); // Đặt video về đầu  
 videoView.start(); // Bắt đầu phát lại  
 luotXem++; // Tăng số lượt xem  
 textLuotXem.setText(String.valueOf(luotXem)); // Cập nhật TextView hiển thị số lượt xem  
 });  
  
 // Thiết lập sự kiện bấm nút "Back"  
 buttonReturn.setOnClickListener(v -> {  
 Intent intent = new Intent(Web.this, MainActivity.class);  
 startActivity(intent);  
 });  
 }  
}

\*Những điểm cần chú ý:

* Các thành phần giao diện chính của “Web.java”: “VideoView videoView”: Dùng để phát video. “Button buttonPlay”, “buttonStop”, “buttonResume”, “buttonReplay”, “buttonReturn”: Các nút điều khiển video và quay lại màn hình chính. “TextView textLuotXem”: Hiển thị số lượt xem video. “ImageView imageView”: Dùng để hiển thị hình ảnh. “int luotXem”: Biến lưu số lượt xem video
* Hình ảnh được hiển thị từ thư mục “asset”: “assets/image/netflix\_logo.png”. “getAssets().open(...)”: Mở file ảnh trong thư mục “assets”. “BitmapFactory.decodeStream(imageStream)”: Chuyển đổi dữ liệu ảnh thành đối tượng Bitmap. “imageView.setImageBitmap(bitmap)”: Gán ảnh vào ImageView
* Hiển thị video từ thư mục “assets”: vì Android không cho phép VideoView phát trực tiếp từ “assets,” nên cần copy video ra file tạm (tempFile) trong bộ nhớ cache rồi phát video. Cụ thể: Lấy video từ thư mục “assets/video” bằng “getAssets().open(...)”. Tạo một file tạm trong bộ nhớ cache (getCacheDir()) để lưu video. Đọc video từ “assets” và ghi vào file tạm. Tạo Uri của file tạm và phát video với “VideoView”
* Thiết lập các nút điều khiển video: Phát video khi bấm nút "Phát". Tạm dừng video khi bấm nút "Dừng". Tiếp tục video khi bấm nút "Tiếp tục". Phát lại video từ đầu và tăng số lượt xem khi bấm "Phát lại"
* Quay lại MainActivity khi bấm "Back": Tạo Intent để chuyển từ Web về MainActivity. Gọi “startActivity(intent)” để thực hiện chuyển màn hình

## **3.9 Kết quả**

A screen shot of a cell phone

AI-generated content may be incorrect.

Hình 3.5 Giao diện khi vào tính tiền phòng

Kết quả khi chọn phòng và số giờ. Kết quả hiển thị như ảnh dưới

A screenshot of a cell phone

AI-generated content may be incorrect.

Hình 3.6 Kết quả khi chọn phòng và số giờ

Giao diện “xem phim”, các video được lưu trong thư mục asset: “video/alice\_inborderland\_3\_trailer.mp4”

A screen shot of a cell phone

AI-generated content may be incorrect.

Hình 3.7 Giao diện “xem phim”

## **3.10 Đánh giá**

Nhìn chung ứng dụng đã hoàn thành đầy đủ các yêu cầu đề ra, nhưng còn chưa hợp lý ở UI/UX (giao diện và trải nghiệm người dùng) do đây mới chỉ là bài tập thực hành lập trình android, bài tập có thể được nâng cấp về UI/UX và tối ưu về hiệu suất hệ thống để ổn định hơn

## **3.11 Link Github**

[Hhuynam/SimpleApp: Bài tập Lập trình Android](https://github.com/Hhuynam/SimpleApp)

https://github.com/Hhuynam/SimpleApp

A qr code on a white background

AI-generated content may be incorrect.

# **4. Bài tập 2: App sử dụng thông báo**

## **4.1 Giới thiệu**

Với mỗi ứng dụng hiện nay đều có chức năng cơ bản là thông báo về cho người dùng, với các nội dung như thông báo lỗi trong hệ thống, thông báo thông tin công việc, thông báo nhắc nhở đến hạn công việc,...

## **4.2 Phân tích yêu cầu**

Yêu cầu: Xây dựng ứng dụng với khả năng lấy dữ liệu – fetch data từ server hay một API chuyên fetch data

## **4.3 Hướng xây dựng phần mềm**

Để fetch dữ liệu thông báo từ server hoặc API trong Android, bạn có thể sử dụng “Retrofit”

Retrofit là một thư viện HTTP client dành cho Android và Java, được phát triển bởi Square, giúp dễ dàng gọi API RESTful và xử lý dữ liệu trả về một cách hiệu quả

## **4.4 File Manifest (xml)**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <uses-permission android:name="android.permission.INTERNET" />  
  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.LapTrinhAndroid2025"  
 tools:targetApi="31">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

\*Những điểm cần chú ý:

* Ứng dụng yêu cầu quyền “INTERNET**”** để có thể truy cập mạng, ví dụ như gọi API hoặc tải dữ liệu từ server. Nếu không khai báo, ứng dụng sẽ không thể kết nối internet
* Các khai báo trong block application trong bài này tương tự như trong bài tập 1

## **4.5 File Main Activity (java)**

package project.hahuynam.laptrinhandroid2025;  
  
import android.os.Bundle;  
  
import androidx.activity.EdgeToEdge;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.graphics.Insets;  
import androidx.core.view.ViewCompat;  
import androidx.core.view.WindowInsetsCompat;  
  
import android.os.Bundle;  
import android.widget.TextView;  
import androidx.appcompat.app.AppCompatActivity;  
import retrofit2.Call;  
import retrofit2.Callback;  
import retrofit2.Response;  
  
public class MainActivity extends AppCompatActivity {  
  
 private TextView textView;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 textView = findViewById(R.id.textView);  
  
 // Gọi API để lấy last\_id  
 fetchLastId();  
 }  
  
 private void fetchLastId() {  
 ApiService apiService = RetrofitInstance.*getApiService*();  
 Call<Post> call = apiService.getLastId("last\_id");  
  
 call.enqueue(new Callback<Post>() {  
 @Override  
 public void onResponse(Call<Post> call, Response<Post> response) {  
 if (response.isSuccessful() && response.body() != null) {  
 // Hiển thị last\_id lên giao diện người dùng  
 int lastId = response.body().getLast\_id();  
 textView.setText("Last ID: " + lastId);  
 } else {  
 textView.setText("Error: " + response.message());  
 }  
 }  
  
 @Override  
 public void onFailure(Call<Post> call, Throwable t) {  
 textView.setText("Failure: " + t.getMessage());  
 }  
 });  
 }  
}

\*Những điểm cần chú ý:

* Packge của ứng dụng: package “project.hahuynam.laptrinhandroid2025”
* import retrofit2.Call;, retrofit2.Callback;, retrofit2.Response;: Dùng để xử lý yêu cầu và phản hồi từ API
* Kế thừa AppCompatActivity để có thể hiển thị giao diện người dùng (UI)
* Phương thức onCreate(): Gán giao diện cho Activity từ file “activity\_main.xml”: “setContentView(R.layout.activity\_main)”. textView = findViewById(R.id.textView);: Kết nối biến textView với TextView trong giao diện. “fetchLastId();”: Gọi hàm để lấy dữ liệu từ API
* Phương thức fetchLastId(): Lấy API service từ Retrofit: RetrofitInstance.getApiService(); trả về một instance của ApiService, giúp gọi API. Gọi API getLastId("last\_id"): Gửi yêu cầu API để lấy dữ liệu “last\_id”. Xử lý phản hồi (onResponse): Nếu API trả về dữ liệu hợp lệ (response.isSuccessful()), lấy giá trị last\_id và hiển thị lên “TextView”, nếu có lỗi, hiển thị thông báo lỗi: xử lý lỗi kết nối (onFailure): nếu lỗi xảy ra (mất mạng, server lỗi, timeout, v.v.), hiển thị thông báo lỗi

## **4.7 File ApiService (java)**

package project.hahuynam.laptrinhandroid2025;  
  
import retrofit2.Call;  
import retrofit2.http.GET;  
import retrofit2.http.Query;  
  
public interface ApiService {  
 @GET("android/api.aspx")  
 Call<Post> getLastId(@Query("action") String action);  
}

\*Những điểm cần chú ý:

* ApiService.java, một interface được sử dụng với Retrofit để định nghĩa API của ứng dụng Android. Interface này chỉ chứa các phương thức để gửi yêu cầu HTTP đến server
* “@GET("android/api.aspx")”: phương thức này sẽ gửi một yêu cầu GET đến đường dẫn "android/api.aspx". nếu API có URL https://example.com/android/api.aspx thì Retrofit sẽ gửi yêu cầu đến đường dẫn đó
* “Call<Post> getLastId(@Query("action") String action)”: Yêu cầu API sẽ trả về một đối tượng Post (chứa dữ liệu JSON phản hồi)
* “@Query("action") String action”: Thêm tham số action vào URL của request. Nếu gọi getLastId("last\_id"), Retrofit sẽ gửi request đến “https://example.com/android/api.aspx?action=last\_id”. Tham số action=last\_id giúp API biết người dùng đang yêu cầu gì

## **4.8 File Post (java)**

package project.hahuynam.laptrinhandroid2025;  
  
public class Post {  
 private int last\_id;  
  
 public int getLast\_id() {  
 return last\_id;  
 }  
  
 public void setLast\_id(int last\_id) {  
 this.last\_id = last\_id;  
 }  
}

\*Những điểm cần chú ý:

* Post.java là một class model trong Android để đại diện cho dữ liệu trả về từ API. Đây là một Java class có một thuộc tính duy nhất last\_id. last\_id lưu trữ dữ liệu nhận được từ API
* Getter method “getLast\_id()”: Dùng để lấy giá trị của last\_id. Khi cần truy cập last\_id chỉ cần gọi: “int id = post.getLast\_id();”
* Setter method: “setLast\_id”: Dùng để cập nhật giá trị của last\_id. Cách gọi: “post.setLast\_id(100);”

## **4.9 File RetrofitInsctance (java)**

package project.hahuynam.laptrinhandroid2025;  
  
import retrofit2.Retrofit;  
import retrofit2.converter.gson.GsonConverterFactory;  
  
public class RetrofitInstance {  
 private static final String *BASE\_URL* = "https://57kmt.duckdns.org/";  
  
 private static Retrofit *retrofit*;  
  
 public static ApiService getApiService() {  
 if (*retrofit* == null) {  
 *retrofit* = new Retrofit.Builder()  
 .baseUrl(*BASE\_URL*)  
 .addConverterFactory(GsonConverterFactory.*create*())  
 .build();  
 }  
 return *retrofit*.create(ApiService.class);  
 }  
}

\*Những điểm cần chú ý:

* RetrofitInstance.java, class chịu trách nhiệm thiết lập Retrofit để gọi API trong ứng dụng Android
* Khai báo URL gốc của API: "https://57kmt.duckdns.org/": Đây là địa chỉ server mà ứng dụng sẽ gửi request đến. Nếu API có endpoint /android/api.aspx, request sẽ gửi đến

“ https://57kmt.duckdns.org/android/api.aspx”

* Khai báo Retrofit instance: Đối tượng retrofit được sử dụng để quản lý kết nối API
* Phương thức tạo Retrofit instance: phương thức “static ApiService” giúp lấy một instance của ApiService
* Kiểm tra và khởi tạo Retrofit: Chỉ khởi tạo Retrofit một lần duy nhất để tránh lãng phí tài nguyên. baseUrl(BASE\_URL): Cấu hình URL gốc của API. “addConverterFactory(GsonConverterFactory.create())”: Dùng Gson để tự động chuyển đổi dữ liệu JSON sang đối tượng Java (Post.java)
* Trả về API Service: ApiService.class là interface chứa các phương thức gọi API. Khi gọi getLastId("last\_id"), Retrofit sẽ gửi request đến

“https://57kmt.duckdns.org/android/api.aspx?action=last\_id”

## **4.10 Kết quả**

A black cell phone with a white screen

AI-generated content may be incorrect.

Hình 4.1 Kết quả khi chọn “Thông Báo 1”



Hình 4.2 Kết quả khi chọn “Thông Báo 2”

## **4.11 Đánh giá**

Nhìn chung ứng dụng đã hoàn thành đầy đủ các yêu cầu đề ra, nhưng còn chưa hợp lý ở UI/UX (giao diện và trải nghiệm người dùng) do đây mới chỉ là bài tập thực hành lập trình android, bài tập có thể được nâng cấp về UI/UX và tối ưu về hiệu suất hệ thống để ổn định hơn

## **4.12 Link Github**

[Hhuynam/LapTrinhAndroid2025: Môn Lập trình Android 2025](https://github.com/Hhuynam/LapTrinhAndroid2025)

https://github.com/Hhuynam/LapTrinhAndroid2025

A qr code on a white background

AI-generated content may be incorrect.

# **5. Bài tập 3: MIT App Inventor**

## **5.1 Giới thiệu**

MIT App Inventor – MIT AI2 là ngôn ngữ lập trình trực quan dựa vào các khối – các thành phần kéo thả mà không lập trình kiểu viết mã truyền thống. Là sản phẩm của Google xậy dựng và đang được Viện Công nghệ Massachusetts (MIT) duy trì



Hình 5.1 MIT App Inventor

“Ngôn ngữ” này giúp tạo ứng dụng trên hai hệ điều hành Android và iOS, tính đến ngày 25/9/2023 (theo Wikipedia), MIT App vẫn là ứng dụng đang trong giai đoạn thử nghiệm – beta, ngoài ra, phiên bản cải tiến có tên là “Kodular” ra mắt vào 2018 được quảng cáo là đẹp và hiện đại hơn. Là một ngôn ngữ phát hành mở và MIT App hướng tới mục đích giáo dục là dạy lập trình cho học sinh, sinh viên. App chạy trên nền web với giao diện đồ họa GUI thân thiện và dễ sử dụng tương tự các phần mềm như “Scratch”,”StarLogo” (cũng lập trình kiểu kéo thả)

MIT App rất phù hợp với những người không chuyên về lập trình hoặc các bạn học sinh muốn tiếp cận với nghề lập trình phần mềm. Giúp họ hiểu được hướng để làm một ứng dung, ứng dụng chứa những thành phần gì, làm ra làm sao, logic để phát nhạc, hay bấm nút để chuyển trang,...

## **5.2 Phân tích yêu cầu**

MIT App (sau đây chỉ gọi là MIT) giúp làm được ứng dụng trên thiết bị di động với đa dạng các chức năng – hiển thị nội dung text, ảnh, âm thanh, tính toán, tuy nhiên cần chú ý, MIT giới hạn 10 màn hình nên cần phải xác định rõ hướng xây dựng chức năng của ứng dụng để cho phù hợp. Thêm nữa, MIT chỉ là ứng dụng dùng để học tập nên cần xác định rõ mục đích: xây dựng app “nhanh, gọn, nhẹ” hay theo thuật ngữ là mô hình phát triển ứng dụng nhanh RAD (là phương pháp phát triển phần mềm nhấn mạnh vào chu kỳ phát hành nhanh và lặp đi lặp lại, chủ yếu tập trung vào việc cung cấp phần mềm hoạt động trong thời gian ngắn).

Chính vì thế nên xây dựng ứng dụng có chức năng cơ bản là: hiển thị thông tin cá nhân, nghe nhạc, tính toán là cực kỳ phù hợp vì đây đều là các tính năng cơ bản trên mỗi thiết bị di động nhưng rất hữu dụng

## **5.3 Hướng xây dựng phần mềm**

Như đã nói ở trước, ứng dụng có 3 chức năng sau: giới thiệu thông tin cá nhân, nghe nhạc, tính toán. Nhìn chung cần hiển thị giao diện ở mức đủ về mặt nội dung thông tin và giao diện tránh đơn điệu nhưng cũng tránh màu mè, cỡ chữ đủ nhìn. Với chức năng giới thiệu thông tin cá nhân: hiển thị họ tên, năm sinh, đơn vị làm việc (học tập), mô tả (không bắt buộc), ảnh chân dung. Tiếp đó là chức năng nghe nhạc: upload file nhạc .mp3, các nút phát nhạc, dừng nhạc, tắt nhạc (play/pause/stop). Chức năng tính toán: các nút tính toán cộng/trừ/nhân/chia, hiển thị kết quả, nút reset để đặt lại cho người dùng thực hiện phép tính khác

## **5.4 Tạo giao diện**

Tạo Menu như sau: 2 nút: “Tính toán” để vào chức năng tính toán và “Nhạc” để vào chức năng giới thiệu thông tin cá nhân và nghe nhạc

A screenshot of a cell phone

AI-generated content may be incorrect.

Hình 5.2 Menu

Ở menu bar MIT, tạo các trang chức năng bằng nút “+”

A screenshot of a phone

AI-generated content may be incorrect.

Hình 5.3 Giải toán

A screenshot of a phone

AI-generated content may be incorrect.

Hình 5.4 Thông tin cá nhân và nghe nhạc

## **5.5 Thực hiện logic**

\* Menu lựa chọn: Dùng khối when-do. Đặt sự kiện mở another screen để mở màn hình MusicPlayer khi nhấn nút Nut\_NgheNhac

A screenshot of a computer

AI-generated content may be incorrect.

Hình 5.5 Logic Menu

\* Trang hiển thị thông tin và nghe nhạc: tương tự như khối trên nhưng hành động khi click vào nút là thuộc Player. Chọn trong Player các chức năng là Start/Pause/Stop. Nút quay lại lấy sự kiện open another screen của Screen1 – quay lại Menu

A screenshot of a chat

AI-generated content may be incorrect.

Hình 5.6 Logic Nghe nhạc

\* Trang tính toán: Initialize global – khởi tạo và gán các tham số num\_1/num\_2/kq = 0. Lập trình để khi click Nut\_TinhToan thì thực hiện các phép toán, mỗi phép toán dùng điều kiện để rẽ nhánh khi được chọn và khi không được chọn. Các label là các text để hiển thị UI phản hồi lại người dùng khi chọn, mặc định đặt ẩn – false. Logic các phép toán cơ bản: “+,-,\*/”. Cuối cùng, nút đặt lại – Nut\_Reset để đặt các textbox nhập phép toán và biến kiểm tra – checked về khởi đầu, như vậy cho phép người dùng thực hiện nhiều phép tính

A screenshot of a computer

AI-generated content may be incorrect.

Hình 5.7 Logic Tính toán

## **5.6 Kết quả**

Kết quả khi chạy trên smartphone

A person holding a cellphone

AI-generated content may be incorrect.

Hình 5.8 Trang Menu

A hand holding a cellphone

AI-generated content may be incorrect.

Hình 5.9 Trang Thông tin cá nhân và nghe nhạc

A person holding a cell phone

AI-generated content may be incorrect.

Hình 5.10 Trang Giải toán

## **5.7 Đánh giá**

Ứng dụng được tự sinh viên xây dựng và kiểm thử, về hình thức đã hoàn toàn đáp ứng đúng, đủ yêu cầu đề ra. Tuy nhiên về mặt giao diện cần phải nghiên cứu và cải thiện nhiều vì trông rất đơn sơ, thiếu tính bố cục, có thể làm theo dạng grid (ô lưới) hoặc linear (chiều dọc) để làm sao trông tổng thể giao diện theo môt form (tiêu chuẩn hóa) giúp cho ứng dụng chuyên nghiệp hơn, hiển thị thích nghi trên các dạng màn hình khác nhau sao cho không bị hỏng layout

## **5.8 Link Github**

[Hhuynam/LapTrinhAndroid2025 at MIT\_APP\_Inventor](https://github.com/Hhuynam/LapTrinhAndroid2025/tree/MIT_APP_Inventor)

https://github.com/Hhuynam/LapTrinhAndroid2025/tree/MIT\_APP\_Inventor

A qr code on a white background

AI-generated content may be incorrect.

# **VI. TÀI LIỆU THAM KHẢO**

**What is mobile programming? Common mobile programming languages**

https://tokyotechlab.com/blogs/lap-trinh-mobile-la-gi

**What is Android programming? Popular programming languages for Android**

https://tokyotechlab.com/blogs/lap-trinh-android-la-gi

[**MIT App Inventor - Wikipedia**](https://en.wikipedia.org/wiki/MIT_App_Inventor)

https://en.wikipedia.org/wiki/MIT\_App\_Inventor