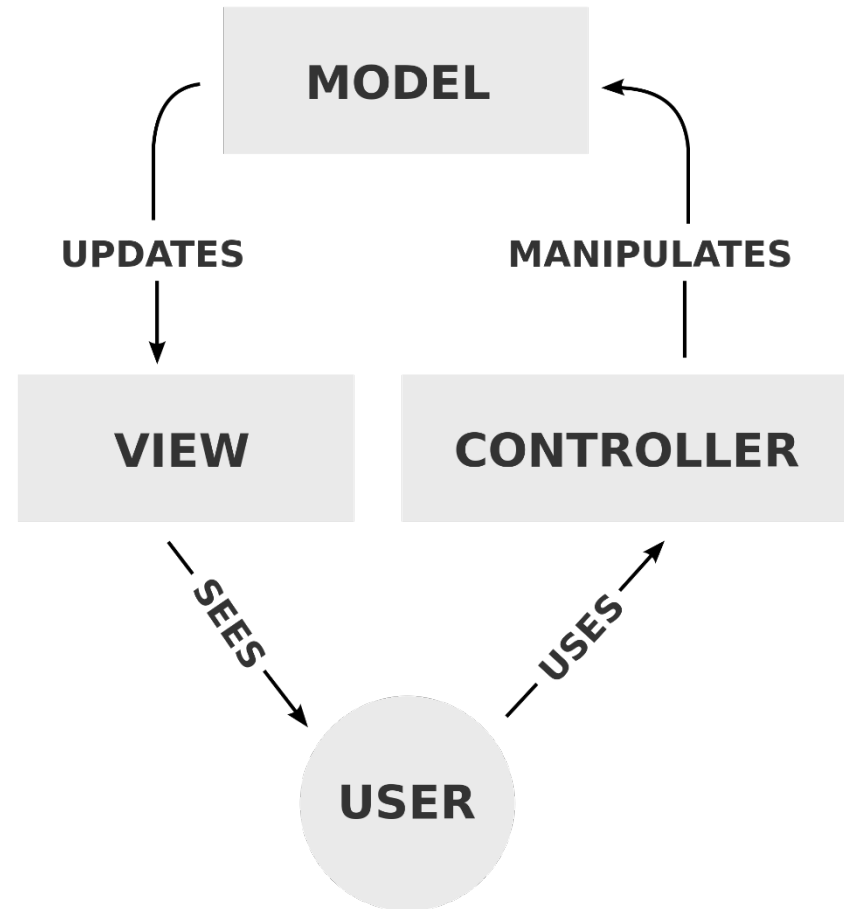Android Application Development
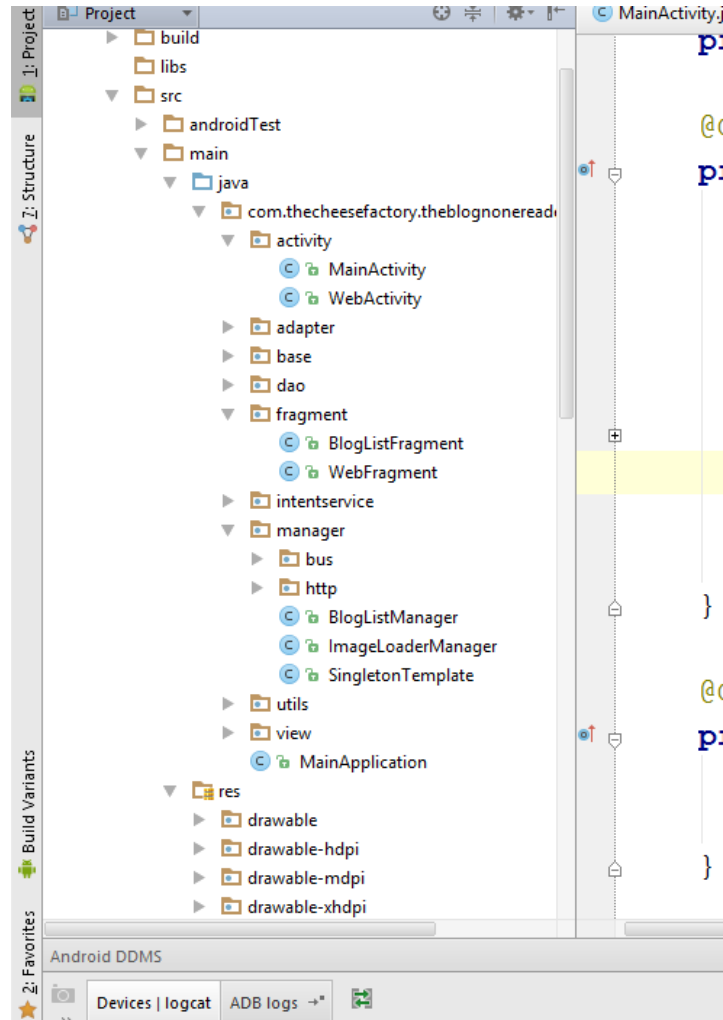
MVC

Model View Controller (MVC)

Best Practices in Android

- Controller
    - Activity
- View
    - XML
- Model
    - Singleton

MVC in Action

- Classify classes with packages

- Override the Application class

- Declares in `AndroidManifest.xml`

- Model as Singleton

- Initialization

Classify classes with packages

Live Demo

Custom Application

```java
package com.thecheesefactory.courses6.dessertmaker;

import ...

/**
 * Created by nuuneoi on 1/15/15 AD.
 */
public class MainApplication extends Application {

    @Override
    public void onCreate() {
        super.onCreate();

        Contextor.getInstance().init(getApplicationContext(

    }

}
```

```xml
package= com.thecheesefactory.theblognonere

<uses-permission android:name="android.perm

<application
    android:name=".MainApplication"
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="TheBlognoneReader"
    android:theme="@style/AppTheme" >
    <activity
```

Singleton Pattern Best Practices

```java
public class HTTPEngine {

    private static HTTPEngine instance;

    public static HTTPEngine getInstance() {

        if (instance == null)

            instance = new HTTPEngine();

        return instance;

    }


    private HTTPEngine() {

    }

}
```

Singleton Pattern Best Practices

```java
public class HTTPEngine {

    private static HTTPEngine instance;

    public static HTTPEngine getInstance() {

        if (instance == null)

            instance = new HTTPEngine();

        return instance;

    }


    private Context mContext;

    private HTTPEngine() {

        mContext = Contextor.getInstance().getContext();

    }

}
```