

# Coursework Report

Oriane Blanc-Dougados  
40332138@live.napier.ac.uk

Edinburgh Napier University - Mobile Applications Development (SET08114)

## Abstract

**HappyMood** is an application which answers a need of entertainment to people who are in a bad mood, or just boring. The idea is to bring people somewhere else than in their own life with all the issues they can have, by using their imagination to do a pause.

In order to choose which kind of application I wanted create, I asked myself : what do you need that is not exist, or something less famous than most of the others application ? At the moment of the year, I'm feel often a bit sad, because of the weather, because the holidays are fare, because I have a lot of work and plenty others reasons not always very logical.

That's why I have choose to design an application which make smile people, or just distract them during few minutes and bring them in a place more relax.

## 1 Introduction

### HappyMood, how works the application ?

At the beginning of the application, the user have to complete his profile with his tastes. Then, he clicks to continue and a nice landscape appears, with a description of the landscape strong incentive the user to imagine himself be in this landscape. Then, after a click on the next button at the bottom of the page, the user can click on a link to play a special motivation music on Youtube. Finally, after click on the last next button the user can click on a link to access a recipe of food he likes. Then the user can return to the menu by clicking the button menu.

## 2 Software design

The design of the application is very light because the application is already very heavy, there is a lot of pictures and it's detrimental to the good working. That's why I have choose to don't use nice background like I have planed, or heavy animations. For this coursework I have privileged the operation.

To specify that the data is erased when the button "New" is pressed, a TAG appears with the mention "Data erased".

I have define a specific theme for this application, which is handy if for some reason I want change it, I don't need to change all the background of activities manually, I just have to change the style of the background and thanks to the theme it's apply for all the application. It's the same thing for the color and the size of the writing.

For the background of the application I have choose the color light pink, and for some writing the color light blue. Both colors are known as soothing, and we can easily imagine that the user which use HappyMood need to feel better. And more, light color are less aggressive for the eyes.

Listing 1: HappyMood styles.xml (XML)

```

1 <resources>
2
3 <style name="HappyMoodTheme" parent="Theme.<
  AppCompat.Light.DarkActionBar">
4   <item name="android:windowNoTitle">true</item>
5   <item name="android:windowBackground">@color/<
    light_pink</item>
6 </style>
7
8 <style name="text_blue">
9   <item name="android:textColor">@color/light.blue</<
    item>
10  <item name="android:textSize">15dp</item>
11 </style>
12
13 <style name="text_pink">
14   <item name="android:textColor">@color/soft_pink</<
    item>
15   <item name="android:textSize">20dp</item>
16 </style>
17
18 </resources>
```

## 3 Application

### 3.1 Store Data

In order to preserve user's choice between each use, the SharedPreferences class is used to save the data in the device. It's more pleasant to the user, once he enter his taste he isn't oblige to do it at each use.

The code below show how the data (user's choice) is stored.

Listing 2: HappyMood (Java)

```

1 #import android.content.SharedPreferences;
2 #import android.preference.PreferenceManager
3
4 SharedPreferences preferences = PreferenceManager.<←
  getDefaultSharedPreferences(Context.this);
5
6 SharedPreferences.Editor editor = preferences.edit();
7
8 editor.putString(Key_Name, Value);
9
10 editor.commit();

```

The code below show how the data store is recovered.

Listing 3: HappyMood (Java)

```

1 #import android.content.SharedPreferences;
2
3 String choice;
4
5 SharedPreferences preference = PreferenceManager.<←
  getDefaultSharedPreferences(getApplicationContext());
6
7 choice = preference.getString("Key_Name", "Default.Value");

```

However if the user wants change his choices, there is a button "New" to erase the data, then the user can enter the new choice.

Listing 4: Erase the SharedPreference

```

1 SharedPreferences PREFS = PreferenceManager.<←
  getDefaultSharedPreferences(getApplicationContext());
2
3 PREFS.edit().clear().commit();

```

## 3.2 Manage Activities

Except the Main Activity and the MainActivity2 which start the application, the three others activities are build the same way. Switch the user choice done in MainActivity2, each activity react by using a case of the switch written inside.

```

switch chocie do
|   case choice1
|       instruction();
|       break;
|   case choice2
|       ... etc
end

```

**Algorithm 1:** Main structure code Activity 3, 4 and 5

## 3.3 Manage User Choice

The user choice is taken in MainActivity2 thanks to three RadioGroup, each representing a topic : the kind of place the user would like travel, the kind of music he likes the most and the kind of food he enjoy eat. In each RadioGroup there are 4 possibilities, by example for the Travel : Beach, Mountain, Campaign, and Home.

Listing 5: Record User Chocie (Java)

```

1 rgTravel = (RadioGroup) findViewById(R.id.rgTravel); //<←
  rgTravel is the id of the RadioGroup define in the XML file <←
  corresponding
2 int selectedId = rgTravel.getCheckedRadioButtonId();
3 radioButton = (RadioButton) findViewById(selectedId);
4 travel = radioButton.getText().toString();

```

Listing 6: Record User Chocie (XML)

```

1 <RadioGroup
2   android:checkedButton="@+id/rbMountain" //the choice <←
   by default
3   android:id="@+id/rgTravel">
4
5   <RadioButton
6     android:text="@string/mountain"
7     android:id="@+id/rbMountain"/>
8
9   <RadioButton
10    android:text="@string/beach"
11    android:id="@+id/rbBeach" />
12
13   etc..

```

In order to display a picture, a music and a recipe, between the panel of possibilities, I have created a way to use randomly one of the possible option.

This is the code to create a random number called r. Each number corresponds to a possibility for a user choice.

Listing 7: Random function (Java)

```

1 switch (choice){
2   case "Beach" :
3     Random r = new Random();
4     int value = minRandom+r.nextInt(maxRandom - <←
  minRandom);
5     switch (value){
6       case 1:
7         ImageView beach = (ImageView) findViewById(R.id.<←
  .picture);
8         beach.setImageResource(R.drawable.beach);
9         break;
10      case 2:
11        ImageView beach2 = (ImageView) findViewById(R.<←
  id.picture);
12        beach2.setImageResource(R.drawable.beach2);
13        break;
14      etc..

```

## 3.4 Manage Data

All the data concerning the possible choice of the user are stored in the file xml "Strings" and in the folder "Drawable". We can easily imagine the evolution of the application by re-new regularly this two folders, which permit to don't touch to the rest of the application. Thanks to this maintenance, the application don't become boring, the user see new pictures, and new links.

## 4 Conclusion

### 4.1 Application evaluation

By comparison with the original concept, the application is much more simple. I had imagined an application with a menu, and the user will be able to chose what kind of things he wants see. By example in this menu with items, we could find the item motivation music, or the item jokes, or the item positive citations. But it will ask a lot of activities more.

### 4.2 Improvements

I assume that in the application that I have build there are four possibilities of landscape beach and only one for all the others landscape (home, campaign, mountain). It's because a picture is very heavy and the more the application is heavy

the less it's easy to run and the user could be upset by waiting.

Concerning the choice of music display and the choice of recipe, there are four choices for each kind of music and food. It could be practicable to add more choice for each kind, but in the same that will increase the weight of data. However the main reason of my choice of possibility is that in matter of music especially it will be much more appreciable to have last hits.

In this view, this application could be very cool if there is regular update. If the application wants stay attractive, it have to be often renew. That necessitate a lot of maintenance, so a person to pay to do that work which mean a loose of money. To counter this we can imagine create a web application that compare on Google the pictures of beach the most seen, the pop songs the most heard, etc. From this point of view, the application HappyMood could be more expanded if it was a website. I didn't do web programming but I imagine that there is more liberty and less restriction to do a website (I think mostly to the weight of pictures for example).

A drawback of this application is that you have to be connect to internet to access to the music and the recipe because there are both linked to a website. A way to improve this could be to have the songs and the recipes in the application, which means add a lot of heavy data. So it solves the problem of the internet connection but the application will be very slow to run. Nowadays, even a phone without 4G can easily find a way to access internet, internet connection are anyway, at university, at work, at home, in restaurants, in pubs, in shops... So after thinking I don't think the internet access is really a drawback or in case it's negligible.

### 4.3 Personal evaluation

This module is a bit difficult to me because I'm not used to so much independence, and even if I have learned Java during the first semester of this year, I'm always struggle with the code. But by practicing I will probably quickly improve my Java level. Concerning the Android Studio software, that takes me ages to understand how it works. I have surely loose a lot of time there. For the next time I will keep in mind that it's not a loose of time to work on the software, how to use it properly, instead of jump on the coursework.

I should also ask more help instead of stay blocked during days on a point, I loose time and often I don't find the solution by myself.

I have created a StackOverflow account, where I have written a question with a part of code I don't understand where was the error, and finally 10min after the problem was solved and I wasn't able to see it be myself. It's a very handy website and I should watch the questions of other users to continue to improve myself.

## References

How define a background for an activity :  
[www.experience-developpement.fr/android-definir-un-fond-decran/](http://www.experience-developpement.fr/android-definir-un-fond-decran/)  
How use the drawable folder, the uses of drawables :  
[www.developpez.com/tutoriels/android/utilisation-drawable](http://www.developpez.com/tutoriels/android/utilisation-drawable)

How launch a webpage :

[www.developpez.net/forums/d112920/java/java-mobile/android](http://www.developpez.net/forums/d112920/java/java-mobile/android)

A lot of things about how create an android application :

<https://openclassrooms.com/courses/creez-des-applications-pour-android>

How define a theme :

<https://developer.android.com/studio/write/theme-editor.html>