

Report Of Electronic Engineering Intriduction B

Final Project

GROUP 6

GROUP LEADER : Wang Xinyu (516030910557);

GROUP MEMBERS: Fang Yihang (516030910546)

Zhang Hao (516030910559)

Long Quanyu (516030910551)

Content

1. Intrduction.....	2
1.1. Project Backgroud.....	2
1.2. About Android Programming.....	2
1.3. Our Idea – Intrduction of the Pomodoro Technique.....	3
2. User Interface Designing.....	5
2.1. About MainActivity	5
2.2. About AddProjectActivity.....	6
2.3. About EditActivity	8
2.4. About EvaluationActivity	8
2.5. About Working Activity	9
2.6. About Help Activity	13
3. Logic Programming	15
3.1. About AddActivity	15
3.2. About EditActivity	17
3.3. About ProjectData.....	18
3.4. About SysApplication	20
3.5. About the Adapter (MyAdapter.class).....	22
3.6. About Statistics	23
4. Inspiration	25
4.1. Team Leader: Wang Xinyu.....	25
4.2. Team member: Fang Yihang	25
4.3. Team member: Zhang Hao	26
4.4. Team member: Long Quanyu.....	26

1. Introduction

1.1. Project Background

According to the instruction of Professor Wang, our team is required to finish an Android application by our own. This is a challenging assignment, because we are all beginners for Android. As a result, to accomplish this assignment, we need to rely on broad reading and online searching. And in the following two parts, I will introduce the general development process of our project.

1.2. About Android Programming

Firstly, we should learn some knowledge about our target operating system.

Android is a mobile operating system developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets. Android's user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a virtual keyboard for text input. In addition to touchscreen devices, Google has further developed Android TV for televisions, Android Auto for cars, and Android Wear for wrist watches, each with a specialized user interface. Variants of Android are also used on notebooks, game consoles, digital cameras, and other electronics.

Initially developed by Android Inc., which Google bought in 2005, Android was unveiled in 2007, along with the founding of the Open Handset Alliance – a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices. Beginning with the first commercial Android device in September 2008, the operating system has gone through multiple major releases, with the current version being 7.0 "Nougat", released in August 2016. Android applications ("apps") can be downloaded from the Google Play store, which features over 2.7 million apps as of February 2017. Android has been the best-selling OS on tablets since 2013, and runs on the vast majority[a] of smartphones. As of May 2017, Android has two billion monthly active users, and it has the largest installed base of any operating system.

Android's source code is released by Google under an open source license, although most Android devices ultimately ship with a combination of free and open source and proprietary software, including proprietary software required for accessing Google services. Android is popular with technology companies that require a ready-made, low-cost and customizable operating system for high-tech devices. Its open nature has encouraged a large community of developers and

enthusiasts to use the open-source code as a foundation for community-driven projects, which deliver updates to older devices, add new features for advanced users or bring Android to devices originally shipped with other operating systems. The extensive variation of hardware in Android devices causes significant delays for software upgrades, with new versions of the operating system and security patches typically taking months before reaching consumers, or sometimes not at all. The success of Android has made it a target for patent and copyright litigation as part of the so-called "smartphone wars" between technology companies.

Secondly, we should learn how to program.

And according to our summary, the learning process of Android programming should include the following steps:

- a) Get familiar with the basic grammar of Java, including encapsulation, inheritance, and package, etc;
- b) Learn how to use Android Studio;
- c) Use online resources to learn about Android User Interface widgets and basic Android activity life periods.
- d) By applying our knowledge to the project, we can gradually learn more knowledge as we program.

And the next part is about the idea about the application of our project.

1.3. Our Idea – Introduction of the Pomodoro Technique

The root idea of our application is the Pomodoro Technique. And the following information is based on Wikipedia:

The Pomodoro Technique is a time management method developed by Francesco Cirillo in the late 1980s. The technique uses a timer to break down work into intervals, traditionally 25 minutes in length, separated by short breaks. These intervals are named pomodoros, the plural in English of the Italian word pomodoro (tomato), after the tomato-shaped kitchen timer that Cirillo used as a university student.

There are six steps in the technique:

- 1) Decide on the task to be done.

- 2) Set the pomodoro timer (traditionally to 25 minutes).
- 3) Work on the task until the timer rings.
- 4) After the timer rings, put a checkmark on a piece of paper.
- 5) If you have fewer than four checkmarks, take a short break (3–5 minutes), then go to step 2.
- 6) After four pomodoros, take a longer break (15–30 minutes), reset your checkmark count to zero, then go to step 1.

The stages of planning, tracking, recording, processing and visualizing are fundamental to the technique. In the planning phase tasks are prioritized by recording them in a "To Do Today" list. This enables users to estimate the effort tasks require. As pomodoros are completed, they are recorded, adding to a sense of accomplishment and providing raw data for subsequent self-observation and improvement.

For the purposes of the technique, a pomodoro is the interval of time spent working. After task completion, any time remaining in the pomodoro is devoted to overlearning. Regular breaks are taken, aiding assimilation. A short (3–5 minutes) rest separates consecutive pomodoros. Four pomodoros form a set. A longer (15–30 minute) rest is taken between sets.

A goal of the technique is to reduce the impact of internal and external interruptions on focus and flow. A pomodoro is indivisible. When interrupted during a pomodoro, either the other activity must be recorded and postponed (inform – negotiate – schedule – call back) or the pomodoro must be abandoned

2. User Interface Designing

2.1. About MainActivity

For MainActivity, it is a little complicated, because it is actually a construction of two layout files. The first one use widgets as shown below:

<file: activity_main_test.xml>

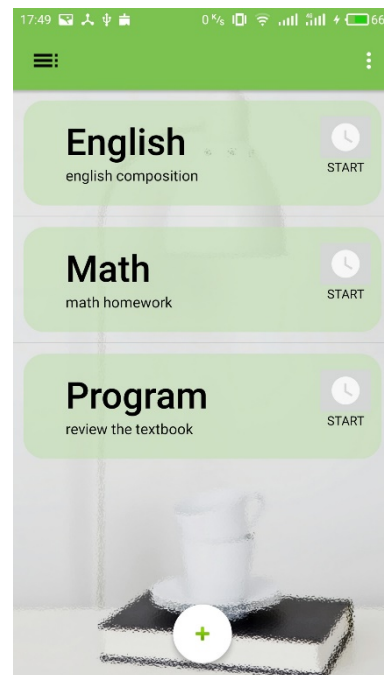
```
<android.support.design.widget.AppBarLayout
```

```
<android.support.v7.widget.Toolbar
```

```
<ImageButton
```

```
</android.support.design.widget.AppBarLayout>
```

```
<android.support.design.widget.FloatingActionButton
```



Of course, this is not the whole part of the MainActivity, actually, in this xml file, another layout file is included:

```
<include layout="@layout/content_main" />
```

So, in content_main.xml:

<file:content_main.xml>

```
<ListView
```

This layout file uses an important widget for the MainActivity: ListView. The reason why it is important, is that this listview implements all operations to the project list.

2.2. About AddProjectActivity

This is a rather clear activity, and it's layout could be easily seen.

And to simplify the understanding of this activity, the widgets are put below:

```

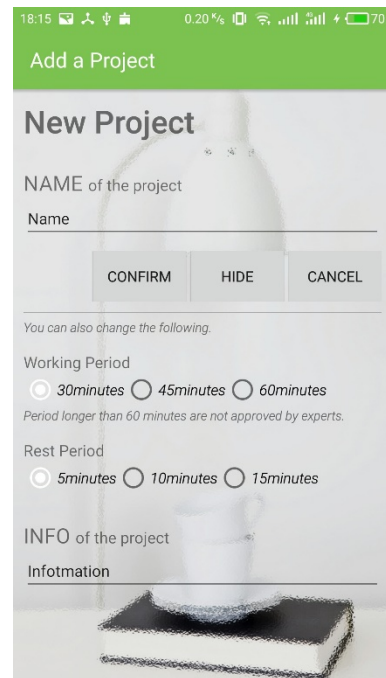
17  <TextView...>
23
24  <LinearLayout
25      android:layout_width="wrap_content"
26      android:layout_height="wrap_content"
27      android:orientation="horizontal"
28      android:layout_marginTop="25dp">
29
30      <TextView...>
35
36      <TextView...>
41
42  </LinearLayout>
43
44  <EditText...>
50
51  <RelativeLayout

```

```

52      android:layout_width="match_parent"
53      android:layout_height="wrap_content"
54      android:layout_marginTop="10dp">
55
56      <Button...>
63      <Button...>
71
72      <Button...>
80
81  </RelativeLayout>
82
83  <LinearLayout
84      android:layout_width="match_parent"
85      android:layout_height="wrap_content"
86      android:orientation="vertical"
87      android:id="@+id/linear_layout_more_changes"
88      android:visibility="gone">
89

```



```
90 <View android:layout_height="1dp"...>
94
95 <TextView...>
102
103 <LinearLayout...>
159
160 <LinearLayout
161     android:layout_width="match_parent"
162     android:layout_height="wrap_content"
163     android:orientation="vertical"
164     android:layout_marginTop="15dp">
165
166 <TextView...>
171
172 <LinearLayout
173     android:layout_width="wrap_content"
174     android:layout_height="wrap_content"
175     android:orientation="vertical">
```

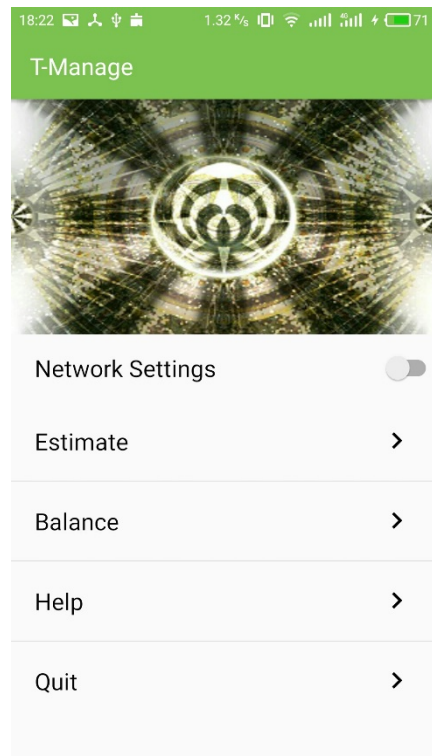
```
176
177 <RadioGroup
178     android:layout_width="wrap_content"
179     android:layout_height="wrap_content"
180     android:orientation="horizontal"
181     android:id="@+id/breakOptions">
182
183 <RadioButton...>
190
191 <RadioButton...>
197
198 <RadioButton...>
204
205 </RadioGroup>
206
207 </LinearLayout>
208 </LinearLayout>
209
```

```
210 <LinearLayout
211     android:layout_width="wrap_content"
212     android:layout_height="wrap_content"
213     android:orientation="horizontal"
214     android:layout_marginTop="25dp">
215
216 <TextView...>
221
222 <TextView...>
227
228 </LinearLayout>
229
230 <EditText...>
236
237 </LinearLayout>
238
239
240 </LinearLayout>
```

2.3. About EditActivity

This is an activity connecting the MainActivity with EvaluationActivity and Help Activity. In a word, this is a very useful transition activity, and it's convenient for users to switch activities on this activity. Also, it could be easily seen that, the layout file contains four widgets:

- 1) ImageView
- 2) RelativeLayout
 - i. TextView
 - ii. SwitchButton
- 3) Listview



For the reason that this layout is easily seen, the codes are omitted.

2.4. About EvaluationActivity

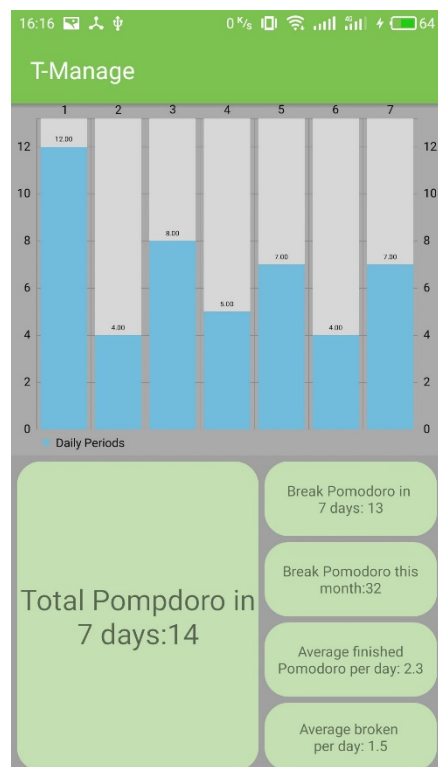
In this Activity, the layout file contains a widget that is not original. Actually, the bar chart is a widget from an open source project on Github.

And the website of the open source project is :

<https://github.com/PhilJay/MPAndroidChart>

MPAndroidChart
created by Philipp Jahoda

Then, the rest part is simple, and it is just a combination of textviews.



2.5. About Working Activity

Since our initial purpose is to improve users' work efficiency, our working activity needs to be as simple as possible. In my UI, there is only 3 compositions, one circle progress bar and two buttons: "Complete" and "Break".

The two buttons is easy to set, and I only need to add another XML file to turn them into circle buttons:

```
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="oval"
    android:layout_height="10dp"
    android:layout_width="10dp">

    <solid android:color="#AAFF1111"/>
    <corners android:radius="360dip"/>
    <padding
        android:left="10dp"
        android:top="10dp"
        android:right="10dp"
        android:bottom="10dp"
    />
</shape>
```

And the progress bar is much more difficult than those two buttons. First, we need a new View. There is an XML called attrs.xml, in which we can see some properties of the new View:

```
<declare-styleable name="CountDownProgress">

    <!-- 默认圆实心的颜色 -->
    <attr name="default_circle_solide_color" format="color"/>
    <!-- 默认圆边框的颜色 -->
    <attr name="default_circle_stroke_color" format="color"/>
    <!-- 默认圆边框的宽度 -->
    <attr name="default_circle_stroke_width" format="dimension"/>
    <!-- 默认圆的半径 -->
    <attr name="default_circle_radius" format="dimension"/>
    <!-- 进度条的颜色 -->
    <attr name="progress_color" format="color"/>
    <!-- 进度条的宽度 -->
    <attr name="progress_width" format="dimension"/>
    <!-- 小圆的实心颜色 -->
    <attr name="small_circle_solide_color" format="color"/>
    <!-- 小圆的边框颜色 -->
    <attr name="small_circle_stroke_color" format="color"/>
    <!-- 小圆的边框宽度 -->
    <attr name="small_circle_stroke_width" format="dimension"/>
    <!-- 小圆的半径 -->
    <attr name="small_circle_radius" format="dimension"/>
    <!-- 文字的颜色 -->
    <attr name="text_color" format="color"/>
    <!-- 文字的字体大小 -->
    <attr name="text_size" format="dimension"/>
</declare-styleable>
```

Then we declare the new View, CountdownProgress, in our layout:

```
<com.example.a73936.tomatotimemanagement.CountDownProgress
    android:id="@+id/countdownProgress"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="20dp"
    android:layout_centerHorizontal="true"
/>
```

And in the Java file "CountDownProgress.java", we create the new View(We may not be able to show all of the code due to the space limitation):

```
public class CountdownProgress extends View {

    private static int defaultCircleSolideColor = Color.BLUE;
    private static int defaultCircleStrokeColor = Color.WHITE;
    private static int defaultCircleStrokeWidth = 15;
    private static int defaultCircleRadius = 300;
    private static int progressColor = Color.RED;
    private static int progressWidth = 10;
    private static int smallCircleSolideColor = Color.YELLOW;
    private static int smallCircleStrokeColor = Color.RED;
    private static float smallCircleStrokeWidth = 0;
    private static float smallCircleRadius = 0;
    private static int textColor = Color.BLACK;
    private static float textSize = 80;
    private static Paint defaultCriclePaint;
    private static Paint progressPaint;
    private static Paint smallCirclePaint;
    private static Paint smallCircleSolidePaint;
    private static Paint textPaint;
    private static float currentAngle;
    private static String textDesc;
    private static long countdownTime;
    private static int mStartSweepValue = -90;

    public CountdownProgress(Context context) { super(context); }

    public CountdownProgress(Context context, AttributeSet attrs) { super(context, attrs); }

    public CountdownProgress(Context context, AttributeSet attrs, int defStyleAttr) {...}

    private void setPaint() {...}
```

Then rewrite onDraw function:

```

@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    setPaint();
    canvas.save();
    canvas.translate(getPaddingLeft(), getPaddingTop());
    //画默认圆
    canvas.drawCircle(defaultCircleRadius, defaultCircleRadius, defaultCircleRadius, defaultCirclePaint);

    //...
    canvas.drawArc(new RectF(0, 0, defaultCircleRadius*2, defaultCircleRadius*2), mStartSweepValue, 360*currentAngle, false, progressPaint);
    //...
    float textWidth = textPaint.measureText(textDesc);
    float textHeight = (textPaint.descent() + textPaint.ascent()) / 2;
    canvas.drawText(textDesc, defaultCircleRadius - textWidth/2, defaultCircleRadius - textHeight, textPaint);

    //画小圆
    float currentDegreeFlag = 360*currentAngle;
    float smallCircleX = 0, smallCircleY = 0;

    float hudu = (float) Math.abs(Math.PI * currentDegreeFlag / 180); //Math.abs: 绝对值, Math.PI: 表示π, 弧度 = 度*π / 180
    smallCircleX = (float) Math.abs(Math.sin(hudu) * defaultCircleRadius + defaultCircleRadius);
    smallCircleY = (float) Math.abs(defaultCircleRadius - Math.cos(hudu) * defaultCircleRadius);
    canvas.drawCircle(smallCircleX, smallCircleY, smallCircleRadius, smallCirclePaint);
    canvas.drawCircle(smallCircleX, smallCircleY, smallCircleRadius - smallCircleStrokeWidth, smallCircleSolidePaint); //画小圆的实心

    canvas.restore();
}

```

Rewrite onMeasure function:

```

@Override
protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
    int widthMode = MeasureSpec.getMode(widthMeasureSpec);
    int heightMode = MeasureSpec.getMode(heightMeasureSpec);
    int widthSize;
    int heightSize;
    int strokeWidth = Math.max(defaultCircleStrokeWidth, progressWidth);
    if(widthMode != MeasureSpec.EXACTLY) {
        widthSize = getPaddingLeft() + defaultCircleRadius*2 + strokeWidth + getPaddingRight();
        widthMeasureSpec = MeasureSpec.makeMeasureSpec(widthSize, MeasureSpec.EXACTLY);
    }
    if(heightMode != MeasureSpec.EXACTLY) {
        heightSize = getPaddingTop() + defaultCircleRadius*2 + strokeWidth + getPaddingBottom();
        heightMeasureSpec = MeasureSpec.makeMeasureSpec(heightSize, MeasureSpec.EXACTLY);
    }

    super.onMeasure(widthMeasureSpec, heightMeasureSpec);
}

```

Then are some other functions: startCountDownTime, countdownMethod, setCountdownTime, etc.

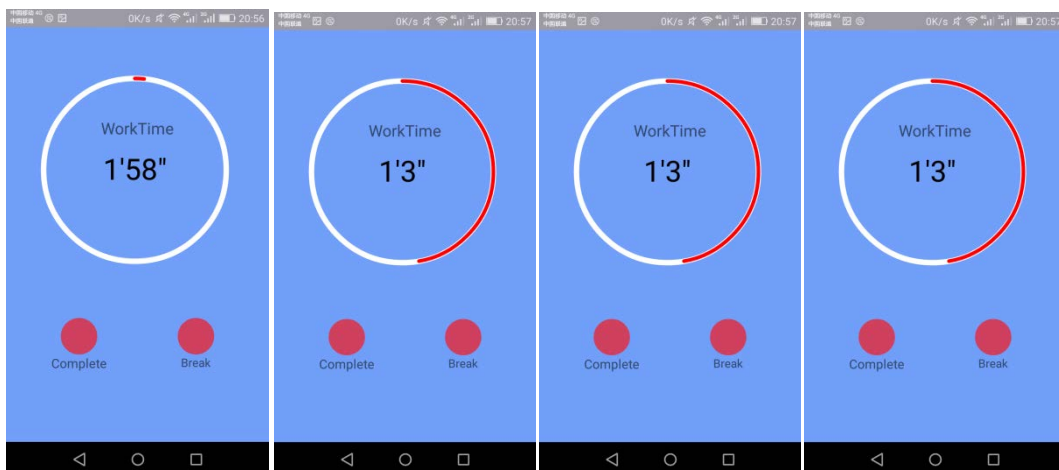
After finishing CountdownProgress.java, we can use it in WorkActivity.java:

```
private CountdownProgress countdownProgress;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_work);

    countdownProgress = (CountdownProgress) findViewById(R.id.countdownProgress);
    countdownProgress.setCountdownTime(2*60*1000);
    countdownProgress.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            countdownProgress.startCountdownTime(new CountdownProgress.OnCountdownFinishListener() {
                @Override
                public void countdownFinished() {
                    Toast.makeText(WorkActivity.this, "工作结束, 该休息了", Toast.LENGTH_LONG).show();
                }
            });
        }
    });
});
```

Here we set 2 minutes as an example. When we really use this app, we can set any work time we want.

Then we can get the WorkActivity as follows:



2.6. About Help Activity

It's relatively easy to set the help activity because we don't need to create a new View. And we can use a ListView to solve this activity.

First of all, we need a ListView to set the main activity of Help:

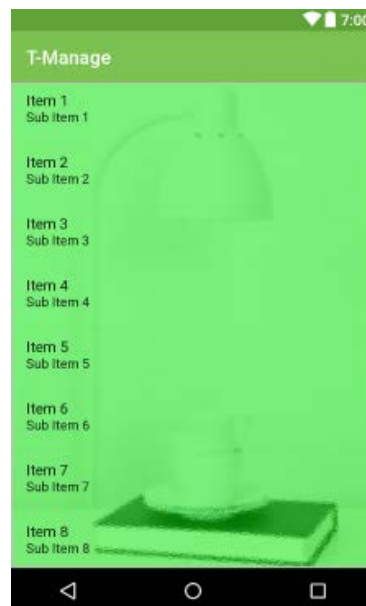
```
public class main_help extends Activity {
    private ListView help_list;
    private ArrayList<String> list = new ArrayList<>();
    @Override
    public void onCreate (Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_help);
        help_list = (ListView)findViewById(R.id.HelpMainList);
        list.add("Help");
        list.add("Q&A");
        list.add("Contact Us");
        ArrayAdapter<String> helpArrayAdapter = new ArrayAdapter<>
            (this, android.R.layout.simple_list_item_1, list);
        help_list.setAdapter(helpArrayAdapter);
        help_list.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                if(list.get(position).equals("Help")){
                    Intent intent = new Intent();
                    intent.setClass(main_help.this, HelpActivity.class);
                    startActivity(intent);
                }
                if (list.get(position).equals("Q&A")){
                    Intent intent = new Intent();
                    intent.setClass(main_help.this, QuestionAndAnswer.class);
                    startActivity(intent);
                }
                if (list.get(position).equals("Contact Us")){
                    Intent intent = new Intent();
                    intent.setClass(main_help.this, ContactUsActivity.class);
                    startActivity(intent);
                }
            }
        });
    }
}
```

We divide the help message into three types: “Q&A(which gives some solution to the possible problems)”, “Help”(which teaches the new users how to use this app) and “Contact Us”(if users need to contact us). When users click one of them, he will reach the corresponding activity. The other three activities are quite easy because there is only one text in any one of them.

Then we need to program the XML files. They also quite easy so I only show the XML file of main help activity:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    android:background="@drawable/add_project_background4"
    android:orientation="vertical">
    <ListView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/HelpMainList"
        android:background="#7700FF00"
    />
</LinearLayout>
```

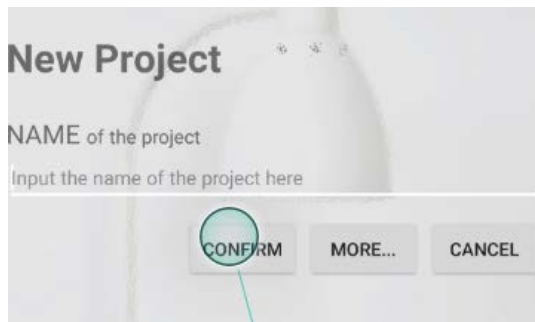
And the main help activity is just like this:



3. Logic Programming

3.1. About AddActivity

This class includes several functions to give response to the click of certain buttons when applied to create new projects.



Here are the buttons and their related functions:

CONFIRM: Creating() (This is also mentioned in class ProjectData.)

```
public void Creating(View v) {
    thirty = (RadioButton) findViewById(R.id.radio_button_30_min);
    fortyfive = (RadioButton) findViewById(R.id.radio_button_45_min);
    sixty = (RadioButton) findViewById(R.id.radio_button_60_min);
    five = (RadioButton) findViewById(R.id.radio_button_5_min);
    ten = (RadioButton) findViewById(R.id.radio_button_10_min);
    fifteen = (RadioButton) findViewById(R.id.radio_button_15_min);
    workingGroup = (RadioGroup) findViewById(R.id.workOptions);
    workingGroup.setOnCheckedChangeListener(new AddProjectActivity.workingOnCheckedChangeListener());
    breakGroup = (RadioGroup) findViewById(R.id.breakOptions);
    breakGroup.setOnCheckedChangeListener(new AddProjectActivity.breakOnCheckedChangeListener());
    Name = (EditText) findViewById(R.id.name);
    ProjectName=Name.getText().toString();
    Text = (EditText) findViewById(R.id.text);
    ProjectText=Text.getText().toString();
    if (ProjectName.equals(""))
        Toast.makeText(AddProjectActivity.this, "Please input the name of the project.", Toast.LENGTH_LONG).show();
    else{
        ProjectData d = new ProjectData();
        MainActivity.All_projects.add(d);
        Intent intent = new Intent();
        intent.setClass(AddProjectActivity.this, MainActivity.class);
        startActivity(intent);
        AddProjectActivity.this.finish();
    }
}
```

MORE...: callMoreChanges()

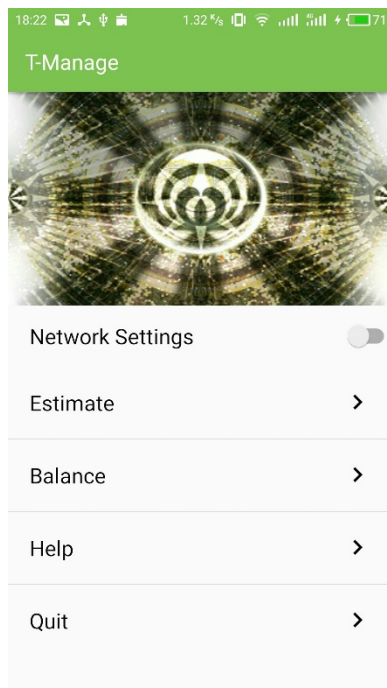
```
public void callMoreChanges(View v) {  
    LinearLayout linear = (LinearLayout) findViewById(R.id.linear_layout_more_changes);  
    Button btn = (Button) findViewById(R.id.btn_more_changes);  
  
    if (btn.getText().equals("More...")) {  
        linear.setVisibility(View.VISIBLE);  
        btn.setText("Hide");  
    }  
    else {  
        linear.setVisibility(View.GONE);  
        btn.setText("More...");  
    }  
}
```

CANCEL: cancelCreating()

```
public void cancelCreating(View v) {  
    /*Intent intent = new Intent();  
    intent.setClass(AddProjectActivity.this, MainActivity.class);  
    startActivity(intent);*/  
    AddProjectActivity.this.finish();  
}
```


3.2. About EditActivity

This class is related to the functions on this interface:



There are four buttons, thus four related branches:

```
listview.setOnItemClickListener((parent, view, position, id) -> {  
    if(listview_string.get(position).equals("Estimate")) {  
        Intent intent = new Intent();  
        intent.setClass(EditActivity.this, EvaluationActivity.class);  
        startActivity(intent);  
    }  
  
    if(listview_string.get(position).equals("Balance")) {  
        Toast toast=Toast.makeText(getApplicationContext(), "This is under development.", Toast.LENGTH_SHORT);  
        toast.show();  
    }  
  
    if(listview_string.get(position).equals("Help")) {  
        Intent intent = new Intent();  
        intent.setClass(EditActivity.this, main_help.class);  
        startActivity(intent);  
    }  
  
    if(listview_string.get(position).equals("Quit")) {  
        SysApplication.getInstance().exit();  
    }  
});
```

3.3. About ProjectData

In our time management application, we designed several classes to implement a wide range of functions.

The most basic one is called “ProjectData”, which includes four variables named `working_time`, `break_time`, `Name` and `Text` respectively.

```
public class ProjectData {
    public int working_time, break_time;
    public String Name, Text;
    public ProjectData() {
        this.working_time=AddProjectActivity.workingTime;
        this.break_time=AddProjectActivity.breakTime;
        this.Name=AddProjectActivity.ProjectName;
        this.Text=AddProjectActivity.ProjectText;
    }
}
```

These variables get their values from these widgets:

```
private RadioGroup workingGroup = null;
private RadioGroup breakGroup = null;
private EditText Name = null;
private EditText Text = null;
```

For instance, in order to get the time of working:

```
private class workingOnCheckedChangeListener implements RadioGroup.OnCheckedChangeListener {
    @Override
    public void onCheckedChanged(RadioGroup group, int checkedId) {
        if (thirty.getId() == checkedId) {
            workingTime = 30;
        }
        if (fortyfive.getId() == checkedId) {
            workingTime = 45;
        }
        if (sixty.getId() == checkedId) {
            workingTime = 60;
        }
    }
}
```

After get these values, the user is able to use a button to create a project. The function is like this:

```
public void Creating(View v) {
    thirty = (RadioButton) findViewById(R.id.radio_button_30_min);
    fortyfive = (RadioButton) findViewById(R.id.radio_button_45_min);
    sixty = (RadioButton) findViewById(R.id.radio_button_60_min);
    five = (RadioButton) findViewById(R.id.radio_button_5_min);
    ten = (RadioButton) findViewById(R.id.radio_button_10_min);
    fifteen = (RadioButton) findViewById(R.id.radio_button_15_min);
    workingGroup = (RadioGroup) findViewById(R.id.workOptions);
    workingGroup.setOnCheckedChangeListener(new AddProjectActivity.workingOnCheckedChangeListener());
    breakGroup = (RadioGroup) findViewById(R.id.breakOptions);
    breakGroup.setOnCheckedChangeListener(new AddProjectActivity.breakOnCheckedChangeListener());
    Name = (EditText) findViewById(R.id.name);
    ProjectName=Name.getText().toString();
    Text = (EditText) findViewById(R.id.text);
    ProjectText=Text.getText().toString();
    if (ProjectName.equals(""))
        Toast.makeText(AddProjectActivity.this, "Please input the name of the project.", Toast.LENGTH_LONG).show();
    else {
        ProjectData d = new ProjectData();
        MainActivity.All_projects.add(d);
        Intent intent = new Intent();
        intent.setClass(AddProjectActivity.this, MainActivity.class);
        startActivity(intent);
        AddProjectActivity.this.finish();
    }
}
```

At the same time, the data will be stored in a static array of the class “ProjectData”.

```
public static ArrayList<ProjectData> All_projects = new ArrayList<>();
```

And in the function Creating(), after every ProjectData is created, it will be added to the end of the array.

3.4. About SysApplication

This class is used to implement the function of quitting the whole application after a click on a certain button.

```

import android.app.Activity;
import android.app.Application;

import java.util.LinkedList;
import java.util.List;

public class SysApplication extends Application {
    private List<Activity> mList = new LinkedList<>();
    private static SysApplication instance;

    private SysApplication() {
    }

    public synchronized static SysApplication getInstance() {
        if (null == instance) {
            instance = new SysApplication();
        }
        return instance;
    }

    public void addActivity(Activity activity) { mList.add(activity); }

    public void exit() {
        try {
            for (Activity activity : mList) {
                if (activity != null)
                    activity.finish();
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            System.exit(0);
        }
    }
}

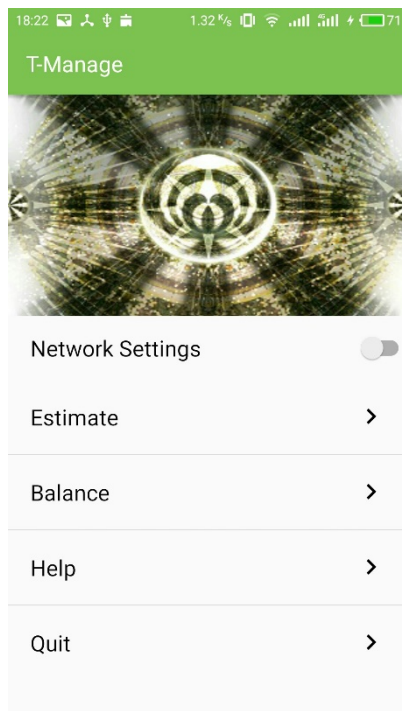
```

After being applied to class MainActivity and class EditActivity like this:

```
SysApplication.getInstance().addActivity(this);
```

```
listview.setOnItemClickListener((parent, view, position, id) -> {  
    if(listview_string.get(position).equals("Estimate")) {  
        Intent intent = new Intent();  
        intent.setClass(EditActivity.this, EvaluationActivity.class);  
        startActivity(intent);  
    }  
  
    if(listview_string.get(position).equals("Balance")) {  
        Toast toast=Toast.makeText(getApplicationContext(), "This is under development.", Toast.LENGTH_SHORT);  
        toast.show();  
    }  
  
    if(listview_string.get(position).equals("Help")) {  
        Intent intent = new Intent();  
        intent.setClass(EditActivity.this, main_help.class);  
        startActivity(intent);  
    }  
  
    if(listview_string.get(position).equals("Quit")) {  
        SysApplication.getInstance().exit();  
    }  
});
```

If you click “Quit”, you will directly quit the application rather than return to the main interface.



3.5. About the Adapter (MyAdapter.class)

After creating a project and showing its information on the main interface, if the user wants to start one of these projects, the application must know exactly which project he/she wants to begin. And for this we need an adapter.

One of the preparations is to define a class implementing HashMap:

```

}import java.io.Serializable;
}import java.util.HashMap;

public class SerMap implements Serializable {
    public HashMap map;
} public SerMap() {

} }

} public HashMap getMap() { return map; }

} public void setMap(HashMap map) {
    this.map = map;
} }
}

```

```

listview_main = (ListView) findViewById(R.id.listview_todos);
MyAdapter myAdapter = new MyAdapter(this);
listview_main.setAdapter(myAdapter);

```

```

public View getView(final int position, View convertView, ViewGroup parent) {
    ViewHolder holder;
    if (convertView==null) {
        convertView=mInflater.inflate(R.layout.single_todo_layout, null);
        holder= new ViewHolder();
        holder.title=(TextView) convertView.findViewById(R.id.text_project_name);
        holder.info= (TextView) convertView.findViewById(R.id.text_project_note);
        holder.bt= (Button) convertView.findViewById(R.id.button_project_start);
        convertView.setTag(holder);
    }
    else {
        holder= (ViewHolder) convertView.getTag();
    }
    holder.title.setText(All_projects.get(position).Name);
    holder.info.setText(All_projects.get(position).Text);
    holder.bt.setOnClickListener((v) -> {
        Toast.makeText(MainActivity.this, "kkkkk"+All_projects.get(position).working_time, Toast.LENGTH_SHORT).show();
    });
    return convertView;
}

```

After these works, the application will exactly know which project to start.

3.6. About Statistics

Our app contains a function called Statistics. When you create, finish, and break a Pomodoro, the app will write the information into the file which contains in the app. Here I use the class called “FileOutputStream”, when the activity changes, it will simulate the function called “write”.

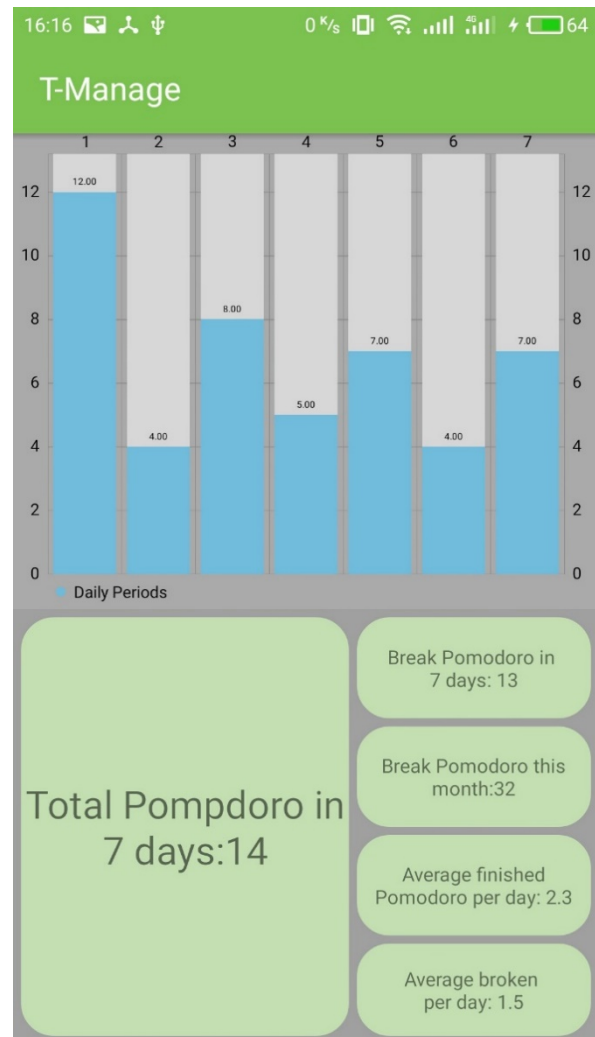
The file named “date” has the form like this:

```
2017/05/29 | c | b | b | f | c | f | c | c | b | c | f | f |
```

```
2017/05/30 | c | f | c | c | b | b | f | f |
```

The “c” means create a new Pomodoro, “b” means the user break the Pomodoro when it’s proceeding, and “f” means uses finish a Pomodoro. It’s necessary to get the those records, because in terms of functions of our app, we dedicate to help users to get rid of the symptom of procrastination, So we should accumulate the data since the user start to use the app, then feed the data back.

This is the Statistic part. It can counts the times when user create, finish, and break the Pomodoro, storing in the “date” file. When the user open “Eliminate”, the function called “read” will open the file, and count the time that “c”, ”f” and ”b” occurs. Then store them in totalPomodoro, lateSevenDaysPomodoro, monthPomodoro, perDayPomodoro, totalBrokenPomodoro, monthBrokenPomodoro, lateSevenDaysBrokenPomodoro, perDayBrokenPomodoro. Then it will be illustrated like this.



```
public class FileDBManager {
    private File mFile;

    private Context mContext;

    private String mFileName = "myfile";

    public FileDBManager(Context context) {
        mFile = context.getFileStreamPath(mFileName);
    }

    public void write(String info){
        try {
            FileOutputStream fos = mContext.openFileOutput(mFileName,Context.MODE_APPEND);
            fos.write(info.getBytes());
            fos.close();
        }catch (Exception e){
            e.printStackTrace();
        }
    }

    public String read(){
        try {
            FileInputStream fis = mContext.openFileInput(mFileName);
            BufferedReader reader = new BufferedReader(new InputStreamReader(fis));
            String info = reader.readLine();
            fis.close();
            return info;
        }catch (Exception e){
            e.printStackTrace();
        }
        return "";
    }
}
```

While user is using this app and try to manage his time efficiently, and users can refer to the data simultaneously. User will refer to those data, then will have the knowledge that his working hours, whether easy to be interrupted and distracted or not, whether he is diligent, whether he is proficient to use the app to improve his time management, whether he has the ability to multitask.

4. Inspiration

4.1. Team Leader: Wang Xinyu

By working on this project, I find it is amazingly inspiring.

As a team leader, I think this project is not easy, because it needs much effort of the entire group, and none of us could be lazy. As a result, the team leader is supposed to help each team member with their motivation of the project, and in this process, I developed my ability to connect with the team members.

As for the professional knowledge, this project teaches me a lot. Firstly, a good Android requires enough knowledge about the understanding of program designing, especially on the understanding of object oriented programming. So, we all get a further understanding on such widely used programming method.

I have to admit I made a lot of mistakes, because this project is hard, and my teammates are all new learners. However, I believe my teammates have all tried their best, and this project is perfectly completed. So from this experience, I learned the power of cooperation.

More importantly, our team intimately worked together. When one of us encounter a difficulty, we all help with him, and at most cases, the seemingly tough obstacle is quickly overcome. From this, I learned the strength of team spirit.

Finally, thank Xinbing Wang and all teaching assistants for giving us the opportunity to take a step further into the programming world.

4.2. Team member: Fang Yihang

During the establishment of these classes in this application, I was deeply impressed by all the efforts it takes to develop a relatively small application. And I was grateful to have a chance to learn something about android application development and coding in Java. Also, I would like to thank all my group members, especially our leader. He has done much in coding as well as making the whole group cooperate.

The most difficult point was to start from the ABC of Java and understand all those abstract conceptions related to the application, for instance, the conception of adapter. And I did wish we could have had a professional to help us with our work, for I often needed to search for ways of implementations.

The problem of cooperation was also delaying our progress. When you get other group members' coding, you usually do not understand what they mean or how to apply them. Also, we often needed to transfer information like pieces of coding, which led to inefficiency and higher risks of making mistakes.

Although it was a difficult task, we finally made it. And we also learnt the necessity of self-study and the importance of cooperation.

4.3. Team member: Zhang Hao

Honestly, my knowledge about Android development is next to nothing before I select this class, so is my understanding to UI. And I'm happy to see that after nearly a semester, my group can finish the development of an Android App.

During the procedure of developing, I realize that it's impossible to finish a project on one's own because there are too many things to do and there are also too many things to learn. So it's a must to work together and divide the task into small ones.

What's more, this course cultivates my ability of independent learning. In future, we will leave school, but we will still need to learn because the world is changing so quickly. And the capacity of self-learning won't appear from nowhere, so I'm glad to have an opportunity to practice self-learning when I'm still in school.

All in all, I obtained much knowledge and experience in this class. And the experiences shared by our upperclassmen is quite valuable for me.

4.4. Team member: Long Quanyu

As far as I am concerned, Android App development experience is so impressive that we even want to continue to improve our project if were not for time limitation.

Firstly, the experiment simulates our passion of being as android developer. Since we had the first android system smart phone, we have been users, never concerning how those wonderful apps are developed. Now it's time for us created a new app, and experienced the sense of achievement when adding a new function to the app. Although there were amounts of obstacles laid in our path of development, we never got depressed, and tried our best to figure out the solutions and debug the program.

Secondly, the experiment enhances our capacity of self-study. Java, xml language, and android studio, none of us have studied one of these before, so the volume of knowledge that we are going to study is sizable. During the experiment, we invoke all the methods such as official files, books in library, online videos to qualify ourselves with necessary knowledge. Moreover, we attach more importance to the self-study after the experiment, because several years later, knowledge may fade away, so learning how to learn is of vital importance, it's also critical to enhance the efficiency, and to learn where and how to find useful information.

Thirdly, we have experienced a totally-engaged development, and the spirit of cooperating. Every member in our team has allocated a specific task, which contributes to the efficiency. We helped each other, and convey what we have learned and thought in mind to others, which will also

improve the ability of coordinating, communication and teamwork spirit.

To conclude, our teammates speak highly of this experiment, and we will treasure this experience, and stir ourselves up to get prepared for successively upcoming challenges.