

# 1、Android Studio配置环境问题

## 2、SVN搭建

## 3、anroid重点代码

### 1).自定义View问题

#### 1.自定义侧滑栏不回弹

```
1 //MotionEvent.ACTION_UP 分支加上 return true;
2 if (ev.getAction() == MotionEvent.ACTION_UP) {
3     int width = linearLayout.getWidth();
4     if (getScrollX() > width / 2) {
5         • smoothScrollTo(width, 0);
6     } else {
7         • smoothScrollTo(0, 0);
8     }
9     return true;
10 }
```

#### 2.Fragment传值必须使用setArguments,禁止使用构造方法传递

```
1 //错误：
2 //Fragment fragment = new Fragment("传递的值");
3 //正确：
4 Fragment fragment = new Fragment();
5 Bundle bundle = new Bundle();
6 bundle.putString("param", "传递的值");
7 fragment.setArguments(bundle);
```

### 3.服务和广播闪退问题

```
builder.setPositiveButton(text: "确定", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        try {
            //对输入的数值要判断
            if (Integer.valueOf(editText.getText().toString()) > 1000 || editText.getText().toString().equals("") || Integer.valueOf(editText.getText().toString()) < 0) {
                Toast.makeText(getApplicationContext(), text: "数值错误, 请重新输入", Toast.LENGTH_LONG).show();
            } else {
                update(i, Integer.valueOf(editText.getText().toString()));
            }
        } catch (Exception e) {
            Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_LONG).show();
        }
    }
});
```

在dao层,ui层,服务和广播的地方一定要try catch包裹起来捕获异常,以防程序崩溃闪退.

#### 4.获取网络数据顺序问题

- 递归法

```

1      private void setJson(final int i) {           //获取数据的方法
2          if (i > 4) return;                        //调用到4时结束
3          HashMap<String, String> hashMap = new HashMap<>(); //准备json
参数
4          hashMap.put("CarId", "" + i);
5          hashMap.put("UserName", "user1");
6          //调用voller(当前这个是封装好的)访问网络接口
7          httpJson.Post("GetCarAccountBalance.do ", hashMap, new
Response.Listener<JSONObject>() {
8              @Override
9              public void onResponse(JSONObject jsonObject) {
10                 //获取相应数据
11                 try {
12                     f23BaseDatas.add(new F23BaseData("第" + (i) + "号小
车的余额", string.valueOf(jsonObject.getInt("Balance"))));
13                     f23Adapter.notifyDataSetChanged();
14                 } catch (JSONException e) {
15                     e.printStackTrace();
16                 }
17                 //获取数据后自调，再次访问网络。
18                 setJson(i + 1);
19             }
20         });
21     }
22

```

## 5.VideoView播放视频靠左的。无法居中问题

- 布局包容法

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    >
    <VideoView
        android:id="@+id/f8_video"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="center"
    />
</LinearLayout>

```

在外层的线性布局视图中，设置一个gravity="center" 即可让视频居中播放(可以满足需求)

- 自定义法

```

1 package com.example.recorder;
2
3 import android.content.Context;
4 import android.util.AttributeSet;
5 import android.widget.VideoView;
6
7 public class MyVideoView extends VideoView {
8     public MyVideoView(Context context, AttributeSet attrs, int defStyle) {
9         super(context, attrs, defStyle);
10    }
11
12    public MyVideoView(Context context, AttributeSet attrs) {
13        super(context, attrs);
14    }
15
16    public MyVideoView(Context context) {
17        super(context);
18    }
19
20    @Override
21    protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
22        // TODO Auto-generated method stub
23
24        int width = getDefaultSize(0, widthMeasureSpec);
25        int height = getDefaultSize(0, heightMeasureSpec);
26        setMeasuredDimension(width, height);
27    }
28 }
29

```

复制

## 6.VideoView能播放，但弹出“无法播放此视频”



视频有时候会出现异常弹出,但能正常播放的时候,这个时候就需要给视频播放做一定的处理来拦截他的弹窗,以免造成体验不良

```

*/
video_show.setOnErrorListener(new MediaPlayer.OnErrorListener() {
    @Override
    public boolean onError(MediaPlayer mp, int what, int extra) {
        return true;
    }
});

```

在VideoView播放视频时可能会出现无法播放的错误情况，但视频还是能正常的播放的。

但是这个弹窗很影响审题（可能会认为你这个代码有问题）。

Return 返回的默认为false 这个要把他改为true 拦截这个事件，让他自己消费。

## 7.list高级排序

//使用Collections.sort函数,参数1:集合， 参数2:接口

```

1 Collections.sort(Datas, new Comparator<F11BaseData>() {
2     @Override
3     public int compare(F11BaseData o1, F11BaseData o2) {
4         //o1集合1, o2集合2
5         int i = o1.load - o2.load; //o1-o2升序, o2-o1降序
6         return i; //返回
7     }
8 });
9 //修改后没有返回值，Collections自动改值，可以之间使用Datas集合，
10 //给listView排序就能更快
11

```

## 8.图片手势放大

```

1      //截取触摸事件
2      mItemf5OpenimageImage.setOnTouchListener(new View.OnTouchListener() {
3          @Override
4          public boolean onTouch(View v, MotionEvent event) {
5              switch (event.getAction()){
6                  //获取点击事件
7                  case MotionEvent.ACTION_DOWN:
8                      //判断是否是双击: getTime:用与储存时间戳long型
9                      if (System.currentTimeMillis()-getTime<=200){
10                         //开启属性动画
11                         ObjectAnimator objectAnimatorX;
12                         ObjectAnimator objectAnimatorY;
13                         AnimatorSet animatorSet = new AnimatorSet();
14                         //isScale: 用户存储状态
15                         if (!isScale){
16                             //双击
17                             objectAnimatorX =
18                             ObjectAnimator.ofFloat(mItemf5OpenimageImage,"ScaleX",1f,2f);
19                             objectAnimatorY =
20                             ObjectAnimator.ofFloat(mItemf5OpenimageImage,"ScaleY",1f,2f);
21                             }else {
22                                 objectAnimatorX =
23                                 ObjectAnimator.ofFloat(mItemf5OpenimageImage,"ScaleX",1f);
24                                 objectAnimatorY =
25                                 ObjectAnimator.ofFloat(mItemf5OpenimageImage,"ScaleY",1f);
26                             }
27                             animatorSet.play(objectAnimatorX).with(objectAnimatorY);
28                             //开启动画
29                             animatorSet.start();
30                             isScale = !isScale;
31                         }else {
32                             getTime = System.currentTimeMillis();
33                         }
34                     break;
35                 }
36             return false;
37         }
38     });
39 }

```

## 9.Fragment中TextView的跑马灯(滚动)模式

```

1  <TextView
2      android:id="@+id/f44_tv_text2"
3      android:layout_width="350dp"
4      android:layout_height="wrap_content"
5      android:textSize="30sp"
6      android:text="沈海高速胶州、莱西服务器封闭施工通知"
7      android:singleLine="true"
8      android:ellipsize="marquee"

```

```

9      android:marqueeRepeatLimit="marquee_forever"
10     android:focusable="true"
11     android:focusableInTouchMode="true"
12     android:textColor="#000"
13     android:gravity="center"/>
14

```

注意: **singleLine**:限制一横显示

**Ellipsize**:样式设置为跑马灯

**marqueeRepeatLimit**:无限循环

**focusableInTouchMode**给与触摸焦点

**focusable**:给与显示焦点

## 10.MpChartAndroid的使用

关于使用MpAndroidChart的过程中遇到的问题:

这里先给上一个官方链接吧

<https://weeklycoding.com/mpandroidchart-documentation/>

这里边有官方示例，有集成的MpAndroidChart APP，也有官方的教程视频，不过好像有点少，并且全英文，画图过程中可能会出些小毛病。数据显示不全其中，有个方法：  
`chart.setExtraOffsets (left,top,right,bottom);` //用于与边界有边距

### 1.LineChart

#### 初始化

```

1  List<Entry> lineEntryList;
2  LineDataSet lineDataSet;
3  LineData lineData;
4  private LineChart mChart;
5  private void initView() {
6      //初始化图表，数据绑定
7      mChart = (LineChart) findViewById (R.id.chart);
8      lineEntryList = new ArrayList<> ();
9      //第二个参数是那个描述，以及那个Legend，好像都是被他默认一样了
10     lineDataSet = new LineDataSet (lineEntryList,"线形图");
11     // 添加数据
12     lineData = new LineData ();
13     lineData.addDataSet (lineDataSet);
14
15     // 把数据交给Chart控件
16     mChart.setData (lineData);
17
18     //记得重绘
19     mChart.invalidate ();
20 }

```

## 这里有三种添加数据的方式

```
1 // 一
2 List<ILineDataSet> lineDataSetss =new ArrayList<> ();
3 lineData = new LineData (lineDataSetss);
4
5 // 二 这里可以放多个LineDataSet ，也可以只放一个
6 lineData = new LineData(lineDataSet,new LineDataSet (null,""));
7
8 // 三
9 lineData = new LineData ();
10 lineData.addDataSet (lineDataSet);
```

## 添加数据，下面添加的是一个随机数

```
1 private void getData() {
2     for (int i = 0; i < 10; i++) {
3         //添加一个Entry 一个坐标点
4         lineEntryList.add (
5             new Entry (lineEntryList.size (),new Random
6             ().nextInt (100))
7         );
8     }
9     //记得刷新，不然这憨憨他不知道是咋回事
10    lineDataSet.notifyDataSetChanged ();
11    lineData.notifyDataChanged ();
12    mChart.notifyDataSetChanged ();
13    //这里不必重新绘制，当然也允许你重新绘制喽
14 }
```

## 一些Chart的属性，也就那么点，将就一下吧

```
1 private void initchartstyle() {
2     //获得Chart 的 X, 左Y, 右Y 轴
3     XAxis xAxis = mChart.getXAxis ();
4     YAxis left = mChart.getAxisLeft ();
5     YAxis axisRight = mChart.getAxisRight ();
6
7     /**
8      * 以下的属性 ， 因为不同的轴他们之间是有联系的，属性差不多都一致，
9      * 都有，所以我就只拿X轴做示例了，
10     * 举一反三啊！
11     */
12     //设置轴的间隙，另外的轴相似，基本上属性差不多值，应为他们的基类都
13     是一家族的
14     xAxis.setGranularity (1f);
15
16     //设置轴的位置,默认在顶层，Top
17     xAxis.setPosition (XAxis.XAxisPosition.BOTTOM);
18
19     //设置轴的最小，最大值
```

```

18     xAxis.setAxisMinimum (0f);
19     xAxis.setAxisMaximum (10);
20
21     //设置轴的标签格式
22     xAxis.setValueFormatter (new IAxisValueFormatter () {
23         @Override
24         public String getFormattedValue(float value, AxisBase
axis) {
25             return value + "日";
26         }
27     });
28
29     //这个值用于设置它是否绘制，显示
30     xAxis.setEnabled (true);
31
32     //网格 色,以及是否绘制
33     xAxis.setGridColor (Color.YELLOW);
34     xAxis.setEnabled (true);
35
36
37     //设置是否绘制标签 ,标签居中吗 ? 居中就true
38     xAxis.setDrawLabels (true);
39     xAxis.setCenterAxisLabels (false);
40
41     //这里设置轴显示的标签数量，force 这个值没有详细了解，但是选true会
    出一点小问题，x轴的位置，可能不会对应
42     xAxis.setLabelCount (8,true);
43
44
45     //这里是一些LineDataSet 的属性
46     //设置它的颜色，可以是当个的颜色，也能是多个，可以 int[]
    list<Integer>
47     lineDataSet.setColors (Color.RED,Color.GRAY);
48     //是否绘制中心的圆洞
49     lineDataSet.setDrawCircleHole (false);
50     //是不是需要填充
51     lineDataSet.setDrawFilled (true);
52     //这里设置完它的属性啊，必须的告诉它更新了，得更新了，重绘
53
54     //mChart的一些属性
55     mChart.setNoDataText ("未得数据的提示");
56
57     //描述
58     Description description = new Description ();
59     description.setText ("Hello Chart");
60     mChart.setDescription (description);
61     //设置是否显示描述
62     mChart.getDescription ().setEnabled (true);
63
64
65     //legend 图表旁边的图示
66     Legend legend = mChart.getLegend ();

```

```

67         legend.setEnabled (true);
68         //多个图示时，可以设置它的排列方式，垂直 || 水平，一般是多条线才会
        用到
69         legend.setOrientation (Legend.LegendOrientation.VERTICAL);
70         //设置图示位置 设置它的位置只能垂直的，水平的分开设，
71         legend.setVerticalAlignment
        (Legend.LegendVerticalAlignment.CENTER);
72         legend.setHorizontalAlignment
        (Legend.LegendHorizontalAlignment.RIGHT);
73
74         mChart.invalidate ();
75     }

```

## 2.BarChart

**BarChart的一些属性,因为大多数属性都是一样的，所以底下就简单讲述一下吧**

```

1  //初始化数据
2      List<BarEntry> barEntryList;
3      BarDataSet barDataSet;
4      BarData barData;
5      private void initView() {
6          mF7BarChart = view.findViewById (R.id.f7_barChart);
7          barEntryList = new ArrayList<> ();
8          //设置数据，差不多也就这里有点区别了
9          barDataSet = new BarDataSet (barEntryList,"");
10         //这里和LineChart 属性差不多
11         barData = new BarData (barDataSet);
12         mF7BarChart.setData (barData);
13         mF7BarChart.invalidate ();
14     }

```

**这里简述一下他的独有属性吧，它其实有好几种类型，有堆栈 (Stack) ,有成组的 (Group) , , , , 多着呢，想要研究就去官方文档吧！**

```

1  堆栈的与其他的图吧，这里呢，就不详细写了
2  //成组图主要就是一个方法的调用 group
3  //官方说法： 7 * ( barSpace + barwidth ) + groupSpace = 1
4  //需要显示柱状图的类别 数量
5  int barAmount = barData.getDataSetCount ();
6  // 组与组之间的间隙 groupSpace
7  float groupSpace = 0.06f;
8  //这里我为了简单就分开了，组内柱之间的间隙为barSpace，柱宽为barwidth
9  float barSpace = ((1f- groupSpace) / barAmount ) * 0.01f;
10 float barwidth = ((1f- groupSpace) / barAmount ) * 0.99f;
11 //设置柱宽
12 barData.setBarwidth (barwidth);
13 //这个方法是必须的，不能少，少了就完了
14 mBarChart.groupBars (0,groupSpace,barSpace);
15 //设置完记得刷新，重新绘制
16 mBarChart.invalidate();

```



## HorizontalBarChart

这是一个单独的控件，使用方法和BarChart的写法几乎一模一样，只不过X轴的位置变成了左右，左右变成了上下而已，差不多就是将BarChart 翻转了一下而已，可以理解为向右旋转了90度而已，这里就不做解释了

## 3.PieChart

饼图也是常见的一种图表，它的使用方法，和前边的图示有点类似，只不过它有一些地方有点坑

我花费了大半天撸它，难受的一批呀！

```
1      //首先初始化
2      PieDataSet pieDataSet ;
3      PieData pieData;
4      List<PieEntry> pieEntries;
5      private void initView() {
6          pieEntries = new ArrayList<> ();
7          mF7PieChart = view.findViewById (R.id.f7_pieChart);
8          //这里我长教训了，PieChart这个图表好像需要给他数据，不然它死活不搭理你呀，getData();方法用于获取数据，
9          getData();
10         pieDataSet = new PieDataSet (pieEntries,"");
11         pieData = new PieData (pieDataSet);
12         mF7PieChart.setData (pieData);
13         mF7PieChart.invalidate ();
14     }
```

这里稍微写了点PieChart的属性

```
1      //设置是否绘制中间的那个洞
2      mF7PieChart.setDrawHoleEnabled (false);
3      //是否绘制值的那个标签
4      pieDataSet.setDrawValues (true);
5      //pieData特有的，就是那个值引出的线，两条
6      pieDataSet.setValueLinePart1Length (0.5f);
7      pieDataSet.setValueLinePart2Length (1.05f);
8      //设置这个部分之间的间隙，给点白边间隙
9      pieDataSet.setSliceSpace (6f);
10     //设置是否绘制饼图 Entry的标签
11     mF7PieChart.setDrawEntryLabels (false);
12     mF7PieChart.setDrawCenterText (false);
13     //这个是为了避免太靠边，导致值显示不全，这个方法
14     mF7PieChart.setExtraOffsets (20,50,20,10);
```

## 4、android与WebView互通

## 1.解决WebView跨域 问题

```
1 | webSetting.setAllowUniversalAccessFromFileURLs(true);
```

# 5、文档

## 1. 文档题要写的地方

文档题只要写项目背景和标有“填写”的部分:

项目背景、项目功能模块、总结.

## 2. 项目背景

这个部分一般会在题目中给出,找到合适的部分复制粘贴即可.

## 3. 项目功能模块

题目会给出题号,找到对应题号的题目写就可以了,项目功能模块部分有功能需求(3.1)、流程图(3.2)、模块设计(3.3)、详细设计 (3.4)、数据库设计(3.5)、实现效果图(3.6).

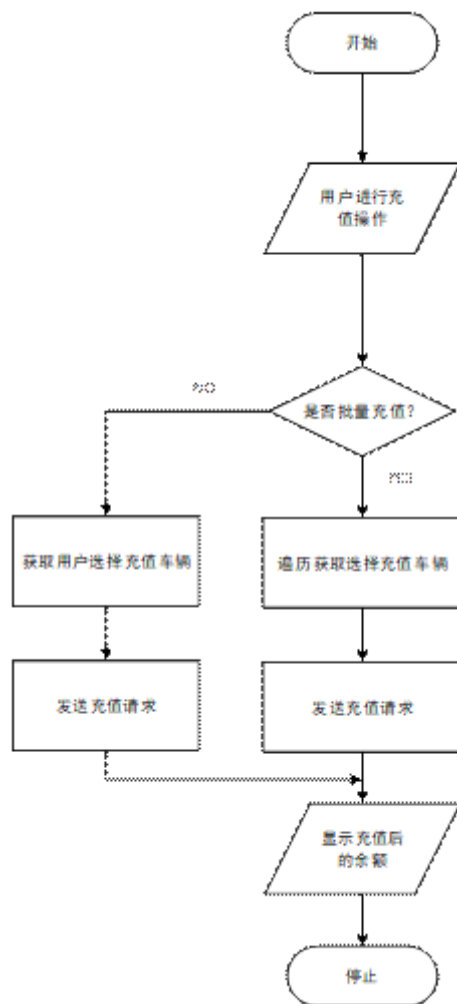
### 3.1 功能需求:

题目里也有,直接复制粘贴,但要注意不要把“注意”部分粘贴过来了稍微处理一下.

### 3.2 流程图:

流程图对应项目功能模块所给的题号对应画图,流程图要有“开始”和“结束”流程图只要画出主要功能就可以了.

参考样例:



### 3.2.1流程图

## 3.3 模块设计:

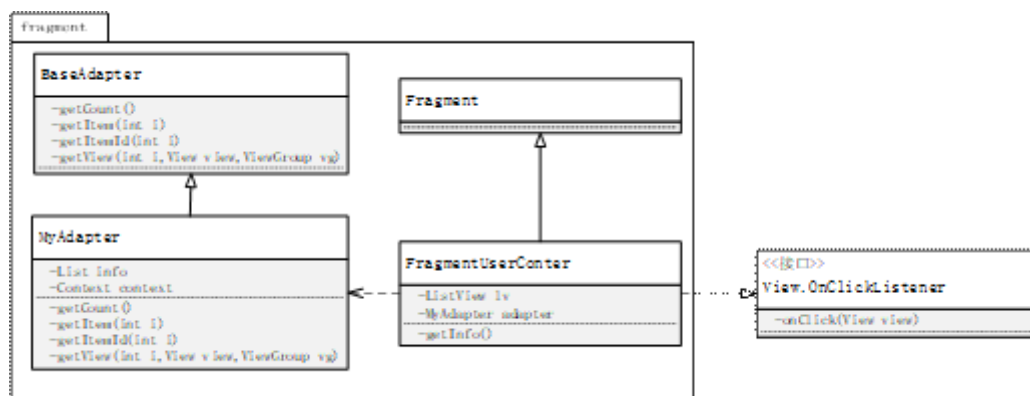
模块设计分别有类图和时序图

### 3.3.1 类图:

要有包、类、接口、继承、实现,这个4个都必须画哪怕代码里没有接口继承什么的也要画个出来,因为这是给分点.

参考样例:

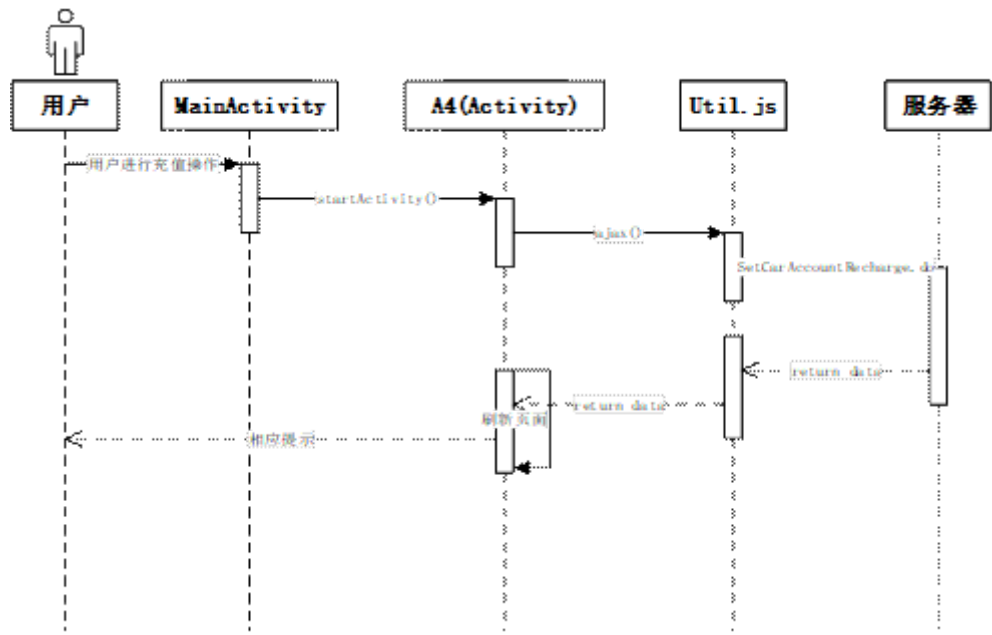
#### 3.3.1.1类图



### 3.3.2 时序图:

时序图要注意阶梯关系(就是从上到下)参考样例:

3.3.2.1时序图



#####

3.4 详细设计:

详细设计要写3个方法和3个对应的代码截图,代码注意要打注释命,名格式正确参考样例:

| 方法

项目	说明
方法名称及类型	getData()
方法功能描述	获取1~4号小车的用户信息与车辆信息
Input输入参数	无
Return返回值	无
Exception抛出异常	无

| 实现描述

3.4.1实现描述

```

private void getData() {
    //清空原有列表
    al.clear();
    //创建一个游标对象
    Cursor c = null;
    //查询1~4号小车的用户信息和车辆信息
    for (int i = 1; i < 5; i++) {
        c = MainActivity.db
            .rawQuery("select c.carbrand," +
                "c.carnumber,u.pname from user_info as u join car_info as c " +
                "on u.pcardid=c.pcardid where number = ?", new String[]{i+""});

        //判断游标中是否含有数据
        if(c.moveToNext()){
            ArrayList<String> a = new ArrayList<>();
            a.add(i+"");
            a.add(c.getString(0));
            a.add(c.getString(1));
            a.add(c.getString(2));
            //获得数据后刷新
            refer(a,i);
        }
        //关闭游标
        c.close();
    }
}

```

### 3.5 数据库设计:

数据库设计要写两个到三个的表图,参考样例:

user\_info 表

序号	字段名	类型	是否是主键	是否允许为空
1	username	text	否	否
2	pname	text	否	否
3	pcardid	text	否	否
4	psex	text	否	否
5	ptel	text	否	否
6	pregisterdate	text	否	否

### 3.6 实现效果图:

字面意思,把该项目功能模块的成品图截下来,截2~3张图就可以了

## 4. 总结

总结要背,提前背好就可以了, 参考样例:

在此次项目中我们完成xxx模块、xxx模块功能、xxx模块、xxx功能、xxx模块、xxx功能、xxx功能、xxx功能等功能模块的编写。

在当前这个先进生产线应用成为现实, 身为学生, 我们更应该努力学习, 在将来走出大学校园走向社会为社会做出贡献。

这次比赛中, 我们知道了在项目中团队合作的重要性, 我们需要遇到难题和困难时沉着冷静, 并能积极思考解决并攻克难题。