

实验报告

课程名称：智能终端软件开发技术

APP名称：扯淡聊天室

专业：计算机科学与技术

成员1姓名：刘非凡 学号：3170102894

成员2姓名：章雨婷 学号：3170102582

电子邮件地址：3170102894@zju.edu.cn, 3170102582@zju.edu.cn

手机：17326085029, 18888922655

任课老师：张寅

目录

实验报告

目录

Chapter 1: 应用简介
Part 1.1 背景
Part 1.2 与同类APP的对比
Part 1.3 运行环境
Chapter 2: 总体架构及功能设计
Part 2.1 总体架构
Part 2.2 聊天功能设计
整体框架
Part 2.3 游戏功能设计
Chapter 3: 关键数据结构/算法
Part 3.2 手机电量交互
Part 3.3 Hardware.Sensor
Chapter 4: 开发困难及解决方案
困难
Chapter 5: 小组分工
Chapter 6: 总结

Chapter 1: 应用简介

Part 1.1 背景

本应用参照了一个小众的搞怪app“Die With me”的格局，总体是一个以多人聊天室为基础的社交软件，并添加了趣味的游戏作为拓展功能。拓展的游戏参考了另一个奇葩的app“Send me to heaven”。

Part 1.2 与同类APP的对比

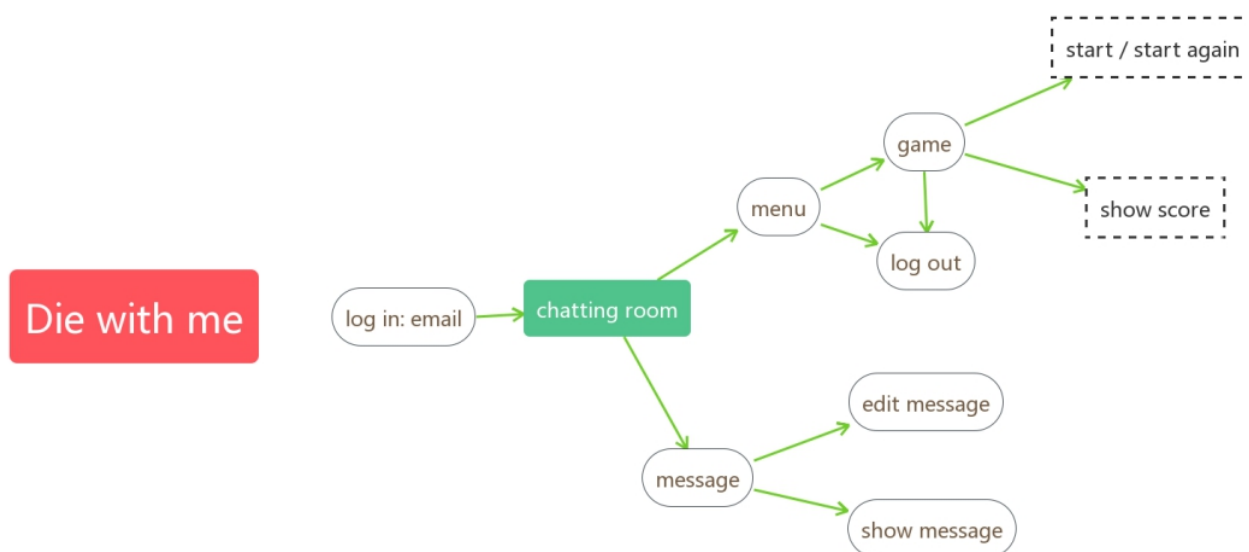
我们的应用总体参照“Die with me”的框架设计，但改变其对电量低于15%才能使用的要求，加入一些趣味的元素，比如进入聊天室是以不确定的电量作为标准的，并将电量不同阶段以不同颜色显示。总体上设计理念更偏向一种社交聊天的软件，支持多用户在线聊天，大家可以畅所欲言。无聊时还有趣味搞怪的游戏可以享受。

Part 1.3 运行环境

Android Studio

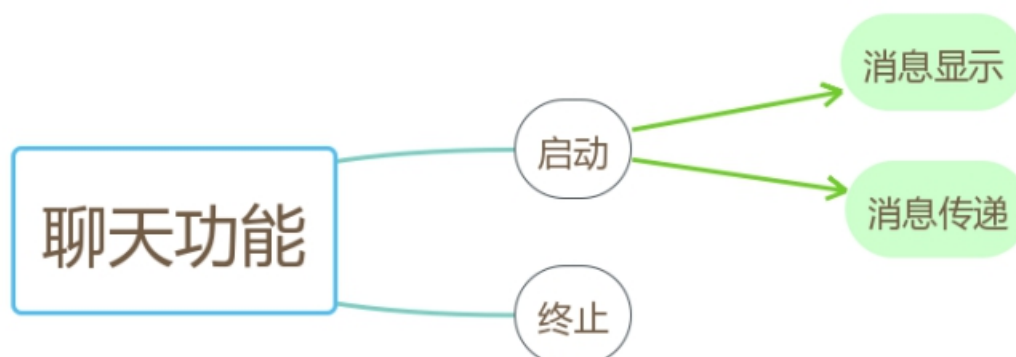
Chapter 2: 总体架构及功能设计

Part 2.1 总体架构



Part 2.2 聊天功能设计

整体框架



- 启动

生成一个0-100的随机数作为电量的上限，只有电量低于这个值的人才能进入聊天室，因此可以说这个设计在原来的基础上增加了不确定性，让体验者感到趣味。整体而言电量越低，进去的概率越大，因此整体设计理念上，将电量和聊天功能更加有趣的结合。

启动后将自动跳转到登陆界面：



- 终止

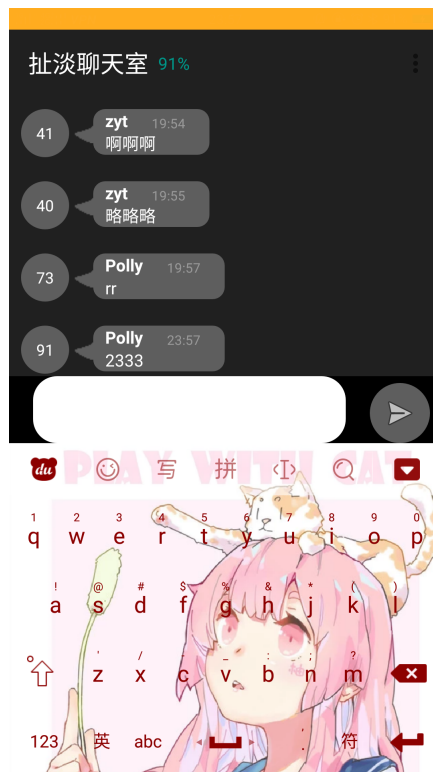
当不满足电量要求时终止聊天程序。

显示界面如图：



- 消息显示

由输入的文本框，发送按钮和显示聊天内容的列表构成，聊天列表只显示昵称时间和消息内容，头像采用自动生成，以电量值为内容。如下图所示：



- 消息传递

采用了Google的Firebase数据库工具，本APP中的登陆框架以及数据库均由其提供。

在用户登陆上我们采用了邮件登陆的形式，第一次登陆的用户将会以邮箱注册。我们后台管理：

- 用户界面
- 这是目前测试已经注册的邮箱。

Authentication

用户

登录方法

模板

用量

🔍 按电子邮件地址、电话号码或用户 UID 搜索

添加用户

🔄

⋮

ID	提供方	创建时间	上次登录日期	用户的 UID ↑
823465056@qq.com	✉	2019年12月2...	2019年12月2...	IHYJIUPDymbzKZnZEAIC0vfpF2
200962818@qq.com	✉	2019年12月2...	2020年1月1日	PA31BRJGjEgEB6KuXkGuPU5wR...
feifanliu29@gmail.com	✉	2019年12月2...	2019年12月2...	SWE1ScHdzyggzhCg3ApCJhXCU...
897842242@qq.com	✉	2019年12月2...	2019年12月2...	WdaE7MruDIQKmQOq94RKJ2ETO...
zjuzyting@gmail.com	✉	2019年12月2...	2019年12月2...	jkQtsToPe1TbGTuNOjmvzHsn3Eh1

每页行数：

50

第 1-5 位，共 5 位

<

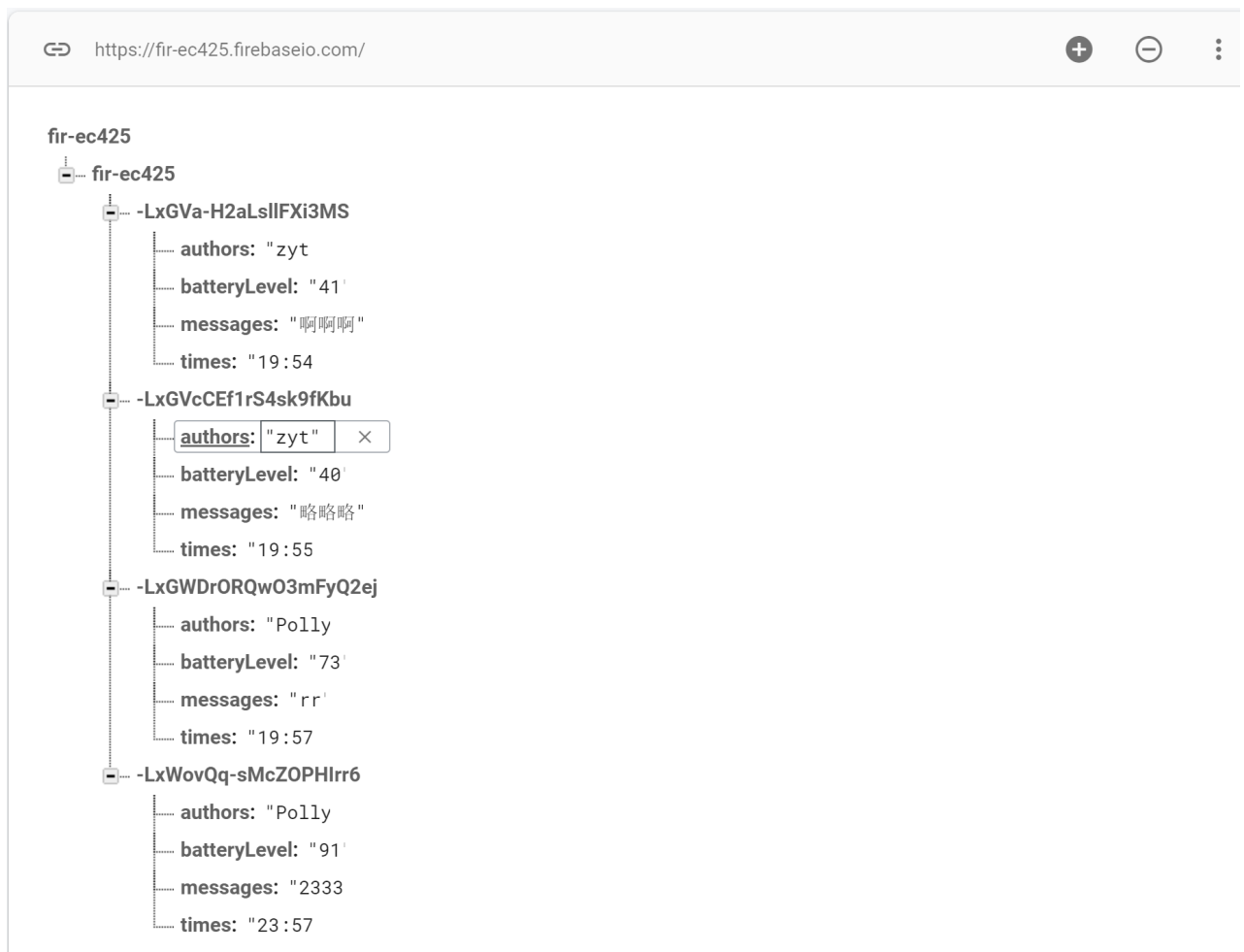
>

- 登陆方式
- 采用游戏密码形式登陆以及注册。登陆注册需要连外网。

登录提供方

提供方	状态
✉ 电子邮件地址/密码	已启用

- 数据库（实时数据库）
- 下图显示了当前数据库内的用户数据和聊天内容。



Part 2.3 游戏功能设计

我们添加的趣味游戏，记录了根据感应器判断手机从某一点开始回到该点的时间。游戏设计上，我们以扔手机为主题，以幽默的方式，达到恶搞和抒博体验者一笑的目的。





飞行时长记录手机从开始位置移动后回到原点的时间，最高飞行记录记录了最短的时间，游戏简单却充满趣味。

Chapter 3 : 关键数据结构/算法

Part 3.1 Firebase数据库

Firebase是一个移动平台，可协助快速开发高品质的应用程序，同时Firebase也提供了庞大数据库供安卓开发使用。

首先纳入 google-services 插件和 Google 的 Maven 代码库：

```
1 buildscript {
2     repositories {
3         google()
4         jcenter()
5     }
6 }
7 dependencies {
8     classpath 'com.android.tools.build:gradle:3.4.1'
9     classpath 'com.google.gms:google-services:4.3.2'
10 }
11 }
```

在应用的 Gradle 文件中添加Firebase SDK 的依赖项和apply plugin：

```
1 implementation 'com.google.firebase:firebase-core:17.2.1'
2 implementation 'com.google.firebase:firebase-auth:17.0.0'
3 implementation 'com.google.firebase:firebase-database:17.0.0'
4 implementation 'com.firebaseui:firebase-ui:0.6.2'
5 implementation 'com.firebaseui:firebase-ui-auth:4.0.0'
```

给定的google-service.json文件:

```
1  {
2    "project_info": {
3      "project_number": "457222238227",
4      "firebase_url": "https://fir-ec425.firebaseio.com",
5      "project_id": "fir-ec425",
6      "storage_bucket": "fir-ec425.appspot.com"
7    },
8    "client": [
9      {
10       "client_info": {
11         "mobilesdk_app_id": "1:457222238227:android:e70ec5da35277f26314fe6",
12         "android_client_info": {
13           "package_name": "com.example.diewithme"
14         }
15       },
16       "oauth_client": [
17         {
18           "client_id": "457222238227-
m5m0uuv83rdop1blg9bku733rl18eh8q.apps.googleusercontent.com",
19           "client_type": 1,
20           "android_info": {
21             "package_name": "com.example.diewithme",
22             "certificate_hash": "d7add74413794ad8e9428902feb2e080272d0dfb"
23           }
24         },
25         {
26           "client_id": "457222238227-
ms50qpevb9v0g14bbsf5cgoeomkk7qvs.apps.googleusercontent.com",
27           "client_type": 3
28         }
29       ],
30       "api_key": [
31         {
32           "current_key": "AIzaSyDlWP8seTrEuXHazS8pGczlMkpXtYYL-Vk"
33         }
34       ],
35       "services": {
36         "appinvite_service": {
37           "other_platform_oauth_client": [
38             {
39               "client_id": "457222238227-
ms50qpevb9v0g14bbsf5cgoeomkk7qvs.apps.googleusercontent.com",
40               "client_type": 3
41             }
42           ]
43         }
44       }
45     ],
46     {
47       "client_info": {
48         "mobilesdk_app_id": "1:457222238227:android:1c28a4cdc0e289b5314fe6",
```

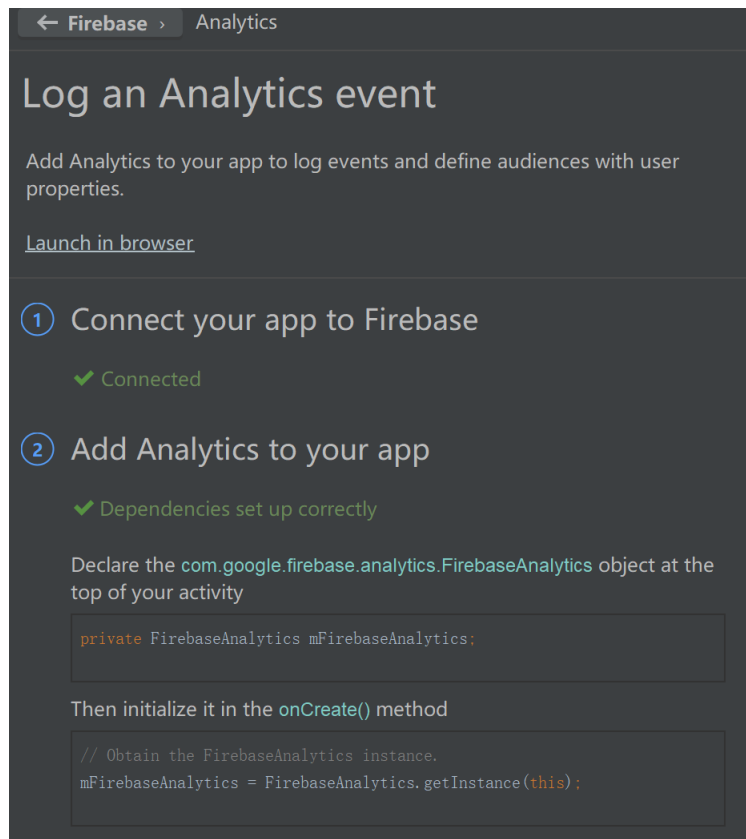


```

49     "android_client_info": {
50         "package_name": "ru.macroid.chat"
51     }
52 },
53     "oauth_client": [
54         {
55             "client_id": "457222238227-
ms50qpevb9v0g14bbsf5cgoeomkk7qvs.apps.googleusercontent.com",
56             "client_type": 3
57         }
58     ],
59     "api_key": [
60         {
61             "current_key": "AIzaSyDlWP8seTrEuXHazS8pGczlMkpXtYYL-vk"
62         }
63     ],
64     "services": {
65         "appinvite_service": {
66             "other_platform_oauth_client": [
67                 {
68                     "client_id": "457222238227-
ms50qpevb9v0g14bbsf5cgoeomkk7qvs.apps.googleusercontent.com",
69                     "client_type": 3
70                 }
71             ]
72         }
73     }
74 },
75 ],
76 "configuration_version": "1"
77 }

```

Android Studio连接Firebase:



Firebase数据库权限设置:

```
1 {  
2   /* visit https://firebase.google.com/docs/database/security to learn more about  
   security rules. */  
3   "rules": {  
4     ".read": true,  
5     ".write": true  
6   }  
7 }
```

代码中Firebase相关变量的定义和调用代码:

数据库:

```
1 private FirebaseDatabase database;  
2 private DatabaseReference myRef;  
3  
4 database = FirebaseDatabase.getInstance();  
5 myRef = database.getReference("fir-ec425");//指定数据库下的参考条目
```

身份验证:

```

1  if (FirebaseAuth.getInstance().getCurrentUser() == null) {
2
3      startActivityForResult(
4          AuthUI.getInstance()
5              .createSignInIntentBuilder()
6              .build(),
7          SIGN_IN_REQUEST_CODE
8      );
9  } else {
10
11      Toast.makeText(this, getString(R.string.welcome_sign) +
12          FirebaseAuth.getInstance().getCurrentUser().getDisplayName(),
13          Toast.LENGTH_LONG).show();
14  }

```

读数据库数据:

```

1  myRef.addChildEventListener(new ChildEventListener()
2  {
3
4      @Override
5      public void onChildAdded(DataSnapshot dataSnapshot, String s) {
6
7          i = dataSnapshot.getChildren().iterator();
8
9          while (i.hasNext()) {
10              nameReceivedFromFirebase = ((DataSnapshot)
11              i.next()).getValue().toString();
12              messageReceiverFromFirebase = ((DataSnapshot)
13              i.next()).getValue().toString();
14              timeReceivedFromFirebase = ((DataSnapshot)
15              i.next()).getValue().toString();
16              batterReceivedFromFirebase = ((DataSnapshot)
17              i.next()).getValue().toString();
18          }
19
20          receivedMessagesFromFirebase = new
21          ConstructorMessages(nameReceivedFromFirebase, messageReceiverFromFirebase,
22          timeReceivedFromFirebase, batterReceivedFromFirebase);
23          constructorMessages.add(receivedMessagesFromFirebase);
24          dataAdapter.notifyDataSetChanged();
25          messagesRecycler.smoothScrollToPosition(constructorMessages.size());
26      }
27  }

```

写入数据库:

```

1 myRef.push()
2     .setValue(new ConstructorMessages(
3         constructorPushedMessage.getAuthors(),
4         constructorPushedMessage.getMessages(),
5         constructorPushedMessage.getTimes(),
6         constructorPushedMessage.getBatteryLevel()
7     ));

```

Part 3.2 手机电量交互

该款APP主要设计理念在于获取手机电量并随机设置许可进入电量以形成随机控制人员进入的功能。

当手机电量发生改变时，系统会对外发送Intent的Action为ACTION_BATTERY_CHANGED常量广播。当手机电量过低时，系统会发送Intent的Action为ACTION_BATTERY_LOW常量的广播。所以可以通过开发监听对应的Intent的BroadcastReceiver：

```

1 intentFilter = new IntentFilter(Intent.ACTION_BATTERY_CHANGED);
2 batteryStatus = registerReceiver(null, intentFilter);
3 battery = batteryStatus.getIntExtra("level", -1);

```

同时，根据手机电量变化也设计了相关电量颜色变化显示：

```

1 private void setBatteryInToolbar() {
2     runOnUiThread(new Runnable() {
3         @Override
4         public void run() {
5             batteryInToolbar.setText(battery + "%");
6             if(battery <= 100 && battery > 90)
7                 batteryInToolbar.setTextColor(Color.parseColor("#009688"));
8             if(battery <= 90 && battery > 80)
9                 batteryInToolbar.setTextColor(Color.parseColor("#4CAF50"));
10            if(battery <= 80 && battery > 70)
11                batteryInToolbar.setTextColor(Color.parseColor("#8BC34A"));
12            if(battery <= 70 && battery > 60)
13                batteryInToolbar.setTextColor(Color.parseColor("#CDDC39"));
14            if(battery <= 60 && battery > 50)
15                batteryInToolbar.setTextColor(Color.parseColor("#FFEB3B"));
16            if(battery <= 50 && battery > 40)
17                batteryInToolbar.setTextColor(Color.parseColor("#FFC107"));
18            if(battery <= 40 && battery > 30)
19                batteryInToolbar.setTextColor(Color.parseColor("#FF9800"));
20            if(battery <= 30 && battery > 20)
21                batteryInToolbar.setTextColor(Color.parseColor("#FF5722"));
22            if(battery <= 20 && battery > 10)
23                batteryInToolbar.setTextColor(Color.parseColor("#FF0000"));
24            if(battery == 10)
25                batteryInToolbar.setTextColor(Color.parseColor("#FF0000"));
26            if(battery == 9)
27                batteryInToolbar.setTextColor(Color.parseColor("#EBEBEB"));
28        }
29    });
30 }

```

```

17         if(battery == 8)
batteryInToolbar.setTextColor(Color.parseColor("#D5D5D5"));
18         if(battery == 7)
batteryInToolbar.setTextColor(Color.parseColor("#BEBEBE"));
19         if(battery == 6)
batteryInToolbar.setTextColor(Color.parseColor("#A5A5A5"));
20         if(battery == 5)
batteryInToolbar.setTextColor(Color.parseColor("#8B8B8B"));
21         if(battery == 4)
batteryInToolbar.setTextColor(Color.parseColor("#727272"));
22         if(battery == 3)
batteryInToolbar.setTextColor(Color.parseColor("#535353"));
23         if(battery == 2)
batteryInToolbar.setTextColor(Color.parseColor("#383838"));
24         if(battery == 1)
batteryInToolbar.setTextColor(Color.parseColor("#202020"));
25         if(battery == 0)
batteryInToolbar.setTextColor(Color.parseColor("#000000"));
26     }
27 });
28 }

```

Part 3.3 Hardware.Sensor

Android平台支持一些用于监视设备动作的传感器以便于更好的安卓开发设计，该APP游戏界面设计中主要使用了加速传感器（TYPE_ACCELEROMETER）来计算手机加速度的变化以得到手机在空中飞行的时长：

```

1 //Retrieve SensorManager, Android's tool for manipulating phone sensors
2 sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
3
4 //Set SensorManager to retrieve accelerometer data
5 accel = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
6
7 //Register this Activity as a listener for changes in phone acceleration detected by
the accelerometer
8 sensorManager.registerListener(this, accel, SensorManager.SENSOR_DELAY_NORMAL);

```

Chapter 4 : 开发困难及解决方案

困难

- 困难

连接Firebase时对于采用的功能以及对应的gradle依赖和依赖的版本都不是很了解，采用不同版本的依赖会出现错误。

- 解决方案

将firebase相关的依赖都与tool内与firebase连接后自带配置的依赖版本一致。

```
1 //implementation 'com.google.firebase:firebase-analytics:17.0.0'
2 implementation 'com.google.firebase:firebase-auth:17.0.0'
3 implementation 'com.google.firebase:firebase-database:17.0.0'
```

部分依赖采用Firebase给定Json配置时给出的依赖项:

```
1 implementation 'com.google.firebase:firebase-core:17.2.1'
```

一些依赖参考网络上类似Firebase代码的相关配置,探索版本:

```
1 implementation 'com.firebaseui:firebase-ui:0.6.2'
2 implementation 'com.firebaseui:firebase-ui-auth:4.0.0'
```

- 未解决的问题

数据库达到一定容量, 及消息达到一定数量时, 会导致程序闪退问题, 需要清理数据库才能解决。但实际监听数据库的流量, 并未达到Firebase提供的数据库容量上限, 因此尚未明确该问题的根本原因。

Chapter 5 : 小组分工

刘非凡: Firebase相关连接建立和聊天功能代码

章雨婷: 游戏相关代码以及界面设计

Chapter 6 : 总结

该款APP小程序主要设计理念是概念性轻聊天软件, 聊天室以手机电量为主要用户信息, 通过邮箱注册登入聊天室, 实现不同安卓终端设备之间的同步聊天和聊天记录更新。同时在界面设计上融入了电量颜色变化警示。在功能上实现了基本的用户登入登出, 发送消息和同步接收, 以及聊天室配备的小游戏。基本上身为一个聊天软件, 功能齐全, 富有新意和创意。

这款APP可以改进的功能还有很多, 例如:

- 聊天室仅限于文字聊天和相关emoji表情包, 由于Firebase上申请的数据库容量过小, 难以实现语音和图片的传输, 后续改进可以考虑扩充数据库的大小, 增添语音和图片的传输功能
- 在推送提醒上, 概念型的简易版聊天室缺少像QQ、微信等社交软件的消息推送功能, 实用性不是特别强
- 聊天室小游戏设计方面, 由于需要获取手机运行加速度, 只有在手机回归源抛出起始点时才可计算抛出的时间, 满足测试的手机轨迹必须为垂直自由落体, 不是很人性化

总之, 作为聊天APP总体基础功能基本上实现了, 界面美观大方, UI交互非常边界流畅, 加载速度较快, 总体来说实现的APP功能比较完善, 也学到了很多知识, 收获颇丰。