中山大学移动信息工程学院本科生实验报告

（2017 年秋季学期）

课程名称：移动应用开发　　任课教师：郑贵锋

年级	15 级	专业（方向）	软件工程(移动)
学号	15352133	姓名	黄少豪
电话	13727024545	Email	328730316@qq.com
开始日期	2017.12.22	完成日期	2017.1.19

一、 实验题目

莫比乌斯——Mobius，一款资讯类应用

二、 实现内容

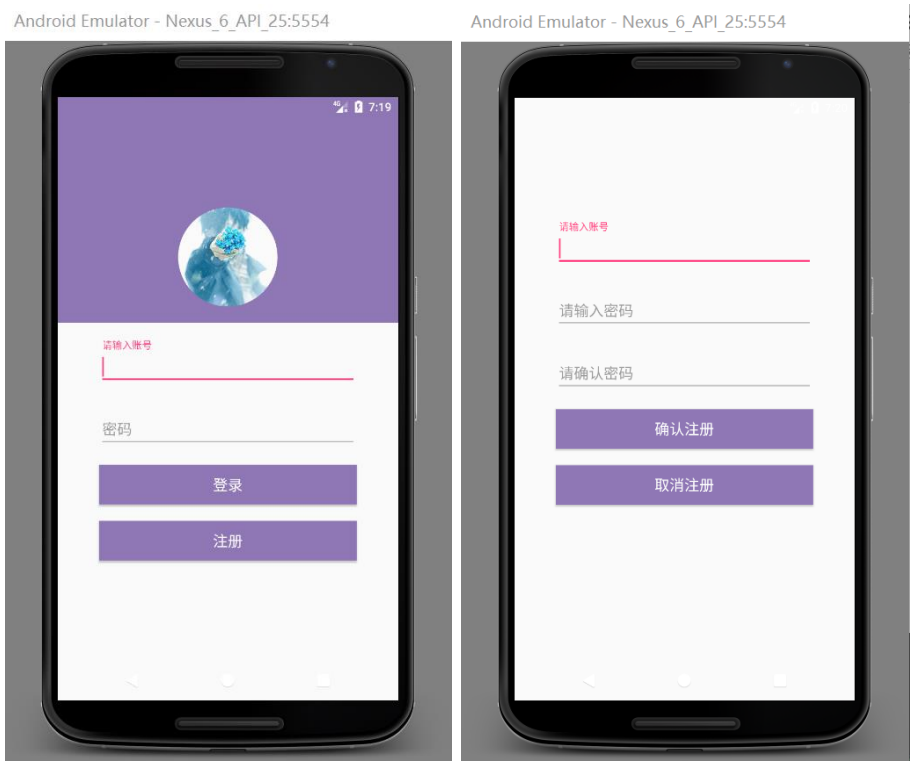
Mobius 是一款资讯类应用，主要是方便用户的资讯获取与资讯发表，通过资讯交换拉进人与人之间的距离。

我们小组的初衷是做一个属于中大人的资讯交互类应用，校内一切资讯都能上传到这个 app 上分享。诸如公益时活动，体育章活动，寻物启示，二手商品交易，社团招新，中大新闻社报道等都可以在这个平台上注册一个官方账号或是个人账号发布文章与招聘。个人文章也是可以发布到这个平台的，我们会有专门的分类通道将资讯分好类，也有专门的后台审核审阅上传的资讯，努力为用户营造一个干净清爽的资讯环境。

三、 课堂实验结果

（1） 实验截图

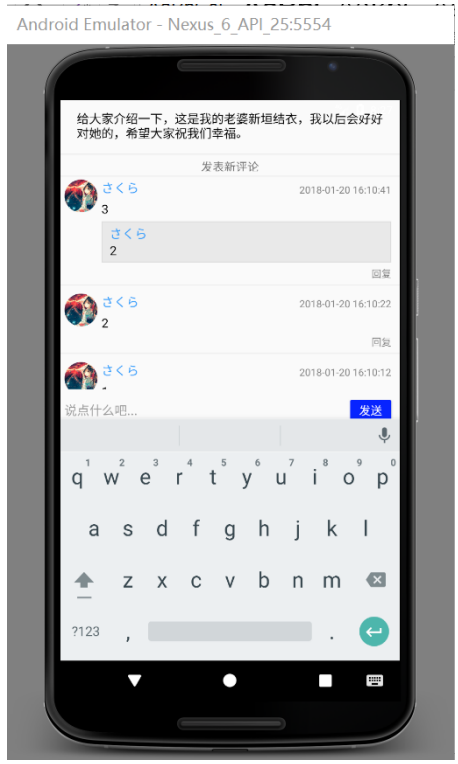
1. 登录/注册界面


Android Emulator - Nexus_6_API_25:5554
7:19
请输入账号
密码
登录
注册
Android Emulator - Nexus_6_API_25:5554
请输入账号
请输入密码
请确认密码
确认注册
取消注册

2. 主界面


Android Emulator - Nexus_6_API_25:5554
8:18
Mobius
测试
浏览量: 9 评论数: 0
【Slay】安室奈美惠Finally销量突破200w
浏览量: 9 评论数: 2
【居然没完】last idol第二季制作人加盟
浏览量: 2 评论数: 0
震惊，新垣结衣男友居然是?
浏览量: 8 评论数: 3
【跳票预定？】雷亚宣布Cytus2今年1月发布
浏览量: 1 评论数: 0
Android Emulator - Nexus_6_API_25:5554
8:25
Mobius
正在刷新...
上次更新 1-20 07:37
测试
浏览量: 9 评论数: 0
【Slay】安室奈美惠Finally销量突破200w
浏览量: 9 评论数: 2
【居然没完】last idol第二季制作人加盟
浏览量: 2 评论数: 0
震惊，新垣结衣男友居然是?
浏览量: 8 评论数: 3
【跳票预定？】雷亚宣布Cytus2今年1月发布
浏览量: 1 评论数: 0

3. 文章界面



4. 搜索界面

5. 主界面侧栏/个人中心

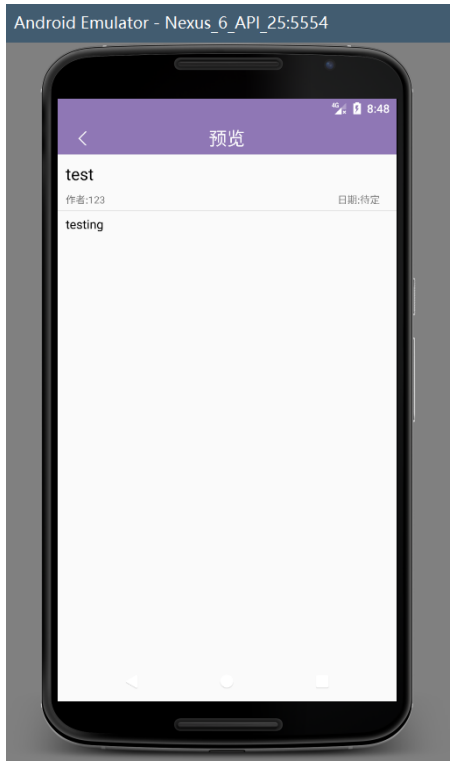

Android Emulator - Nexus_6_API_25:5554
8:33
さくら
为了能做想做的事情，不想做的也应该全力以赴
发布帖子
我的帖子
我的评论
历史足迹
个人资料
配色方案


Android Emulator - Nexus_6_API_25:5554
8:33
账号资料
さくら
为了能做想做的事情，不想做的也应该全力以赴
个人中心
发布帖子
我的帖子
我的评论
历史记录
配色方案
返回主页
退出账号

6. 发布帖子

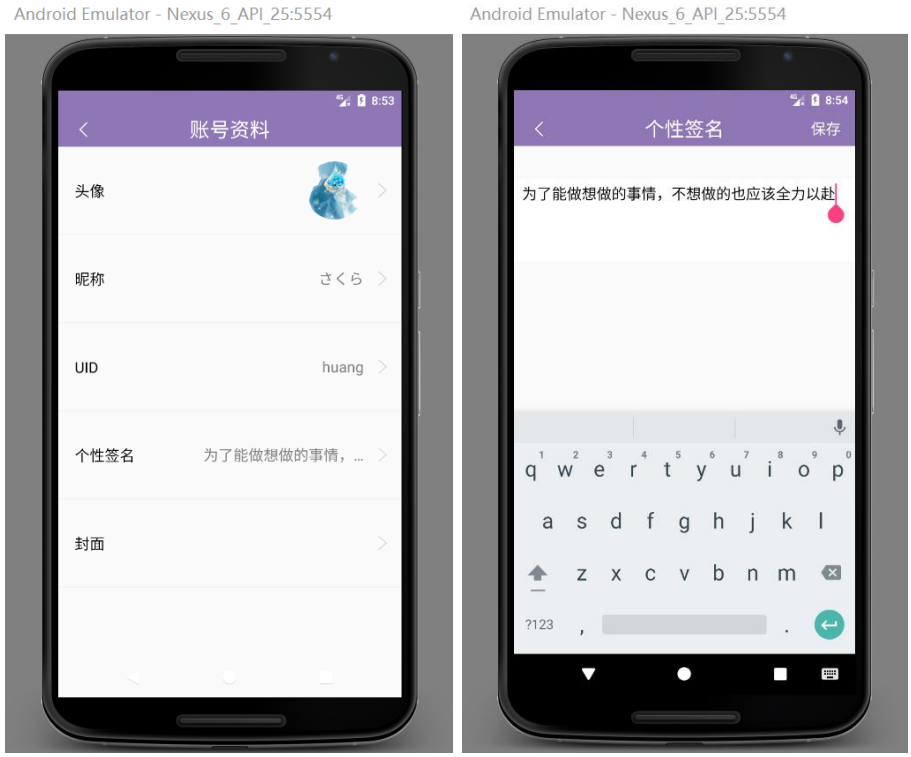

Android Emulator - Nexus_6_API_25:5554
8:46
编辑
请输入标题
标签
预览
发布
编辑区
资源区
添加资源


Android Emulator - Nexus_6_API_25:5554
8:47
编辑
test
标签
预览
发布
编辑区
testing
资源区
添加资源



7. 我的帖子界面/配色界面

8. 账号资料界面/修改资料界面



（2） 实验步骤以及关键代码

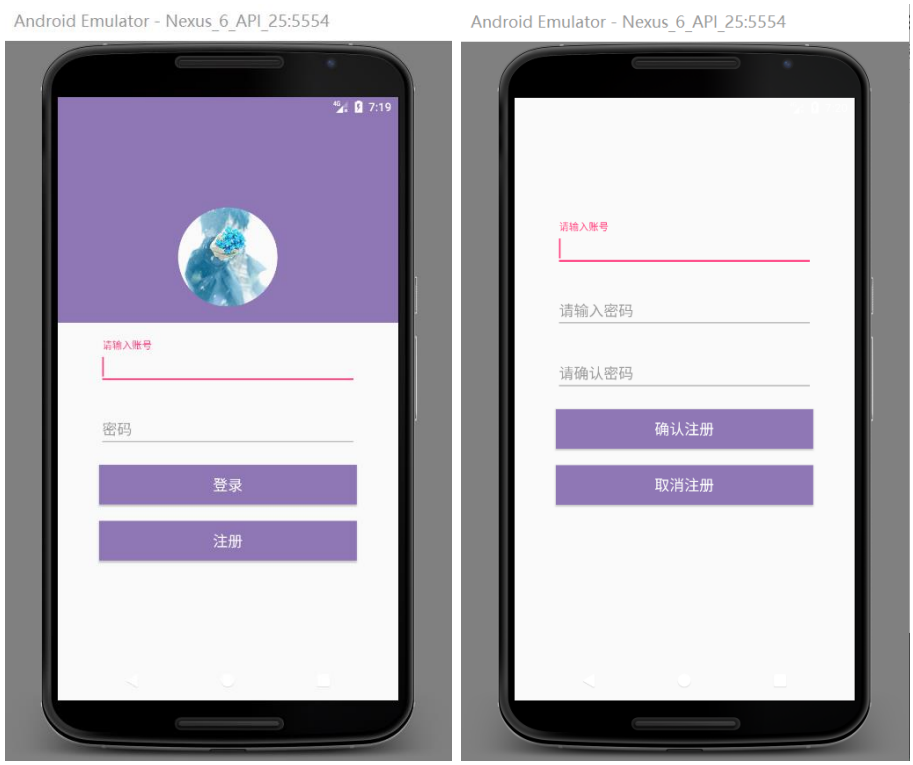
1. 后台服务器的搭建

后台服务器使用的框架是 tomcat+mysql。由于后台 API 较多，这里简单列举几项：

API 名称	输入参数	输出数据	作用
getNews	offset——开始位置 tot——数据数量 key——筛选字段名 value——筛选字段值	JSON 列表——满足筛选规则的一定数量的帖子列表	选出一定数量的符合条件的帖子
getComments	offset——开始位置 tot——数据数量 key——筛选字段名 value——筛选	JSON 列表——满足条件的一定数量的评论列表	选出一定数量的符合条件的评论

	字段值		
getUsers	key——筛选字段名 value——筛选字段值	JSON 列表——满足条件的一定数量的用户列表	选出一定数量的符合条件的用户
saveNews	uid——用户 ID title——文章标题 content——文章内容 images——文章图片列表 tags——文章标签列表	上传状态	新建帖子
saveComment	uid——用户 ID nid——文章 ID content——评论内容 reply——回复的评论 ID	上传状态	发表评论或回复
registerUser	uid——用户 ID pass——账户密码	注册状态	注册用户
login	uid——用户 ID pass——账户密码	登录状态	登录

2. 登录/注册界面的搭建



点击注册按钮触发 Intent 跳转进入注册界面

```
public void register(View view) {
    Intent intent = new Intent(this, RegisterActivity.class);
    startActivityForResult(intent, 1);
}
```

在注册界面输入账号，密码，确认密码完成账号注册

```
service.registerUser(uid, pass)
        .subscribeOn(Schedulers.newThread())
        .observeOn(AndroidSchedulers.mainThread())
        .subscribe(new Subscriber<User>() {
            @Override
            public void onCompleted() {
            }
            @Override
            public void onError(Throwable e) {
                Toast.makeText(RegisterActivity.this, "出现错误", Toast.LENGTH_SHORT).show();
                Log.e("ERROR", e.getMessage());
            }
            @Override
            public void onNext(User user) {
                String s = user.getName();
                if (s.equals("Success")) {
                    Intent intent = new Intent();
                    setResult(RESULT_OK, intent);
                    finish();
                }
                else {
                    Toast.makeText(RegisterActivity.this, s, Toast.LENGTH_SHORT).show();
                }
            }
        });
```

如果返回 Success 表示完成注册，否则就会显示注册失败原因比如账号已被注册等

点击确认注册后且注册成功后，注册界面会 finish 回到登录界面，登录界面显示注册成功

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (resultCode == RESULT_OK) {
        Toast.makeText(LoginActivity.this, "注册成功", Toast.LENGTH_SHORT).show();
    }
}
```

在登录界面输入相应的账号密码完成登录。伪代码如下：

(1) 通过 retrofit 发送网络请求到服务器，传递账号密码参数。

(2) 服务器收到请求发送给对应的 api，api 检查数据库是否存在该账户且账号密码是否匹配

(3) 若匹配则反馈一个登录成功消息，反馈用户资料

(4) 安卓应用收到登录成功消息后，首先将登录状态与返回的用户资料保存到一个以用户 ID 命名的 SharedPreferences 文件中，方便账号的登录状态与资料修改保存。然后进入主界面

3. 主界面/侧栏搭建

使用一个 DrawerLayout 实现主界面的侧栏功能

```
<android.support.v4.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/dl"
android:layout_width="match_parent"
android:layout_height="match_parent">
```

DrawerLayout 内部需要有两个布局，一个是主界面的布局，另一个是左侧栏的布局。

主界面使用一个 RecycleView 来存储从服务器获得的帖子列表，帖子的发送默认是按照发布顺序排序的，越靠上说明越晚发布。另外为了实现上拉刷新与下拉加载更多的效果，使用一个来自 github 的名叫 SmartRefreshLayout 的布局。

```
<com.scwang.smartrefresh.layout.SmartRefreshLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/refreshLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_marginTop="68dp"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp">
    <com.scwang.smartrefresh.layout.header.ClassicsHeader
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
    <android.support.v7.widget.RecyclerView
        android:id="@+id/activity_main_list"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent">
    </android.support.v7.widget.RecyclerView>
    <com.scwang.smartrefresh.layout.footer.ClassicsFooter
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
</com.scwang.smartrefresh.layout.SmartRefreshLayout>
```

上拉刷新与下拉加载更多的代码

```
refreshLayout = (RefreshLayout) findViewById(R.id.refreshLayout);
refreshLayout.setOnRefreshListener((refreshlayout) -> {
        getDatas("", "", "20");
        refreshlayout.finishRefresh();
});
refreshLayout.setOnLoadmoreListener((refreshlayout) -> {
        getDatas("", maxNid, "20");
        refreshlayout.finishLoadmore();
});
```

getDatas 函数会使用 retrofit 调用 api 从服务器获取数据。其中有三个参数，一个是 key 表示搜索关键字，这里默认为空；一个是 offset 表示帖子的开始 ID；一个是 tot，表示获取的帖子数目，这里默认一次请求为 20 个。

左边侧栏的头像与背景封面，主界面帖子右边的图片都是使用 github 上一个叫做 Glide 的图片加载工具从服务器下载，代码如下：

```
public void getImage(String url, ImageView imageView) {
    Glide.with(MainActivity.this)
            .load(baseUrl + url)
            .crossFade()
            .error(R.color._main)
            .into(imageView);
}
```

传入两个参数，一个是 url 表示图片在服务器的 url，另一个是 imageView 表示需要加载图片的 view 控件。

4. 个人中心界面搭建

该界面主要是点击各个按钮进入各个 Activity 以及个人中心部分的线性布局。

头像与背景封面的加载同第三步。

5. 账户资料界面搭建

账号资料界面使用一个 RecycleView 存储各行数据，其中当是头像行和封面行时，隐藏右边的 TextView。当不是头像行时，隐藏右边的 ImageView。

个性签名只显示一行，当超出时使用省略号在行末代替剩余内容。

点击头像行和封面行会调用系统相册选择相片，而点击其他行时会进入修改资料界面。

6. 我的帖子界面搭建

这个界面的布局主要就是一个 RecycleView，里面存储着通过服务器 API 获取到的关于我的帖子。

7. 配色方案界面搭建

在这个界面可以选择想要的主题色。本篇实验报告主要使用紫色，想要看其他颜色的效果可以移步到另外三个组员的实验报告。

修改皮肤的原理就是将选中的颜色保存到一个名为 Color 的 SharedPreferences 文件中，然后在每个 Activity 的初始进入与回调函数中都调用一个 changeTheme 函数从文件中取出颜色并修改 Activity 中的控件颜色。

8. 搜索界面搭建

搜索框使用一个叫做 AutoCompleteTextView 的控件。

```
<AutoCompleteTextView
    android:id="@+id/activity_search_topbar_searchbar"
    android:layout_width="0px"
    android:layout_height="30dp"
    android:completionThreshold="1"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="70dp"
    android:layout_marginBottom="5dp"
    android:background="@drawable/search_bar_background"
    android:textColor="@color/black"
    android:textSize="18sp"
    android:hint="搜索您感兴趣的消息"/>
```

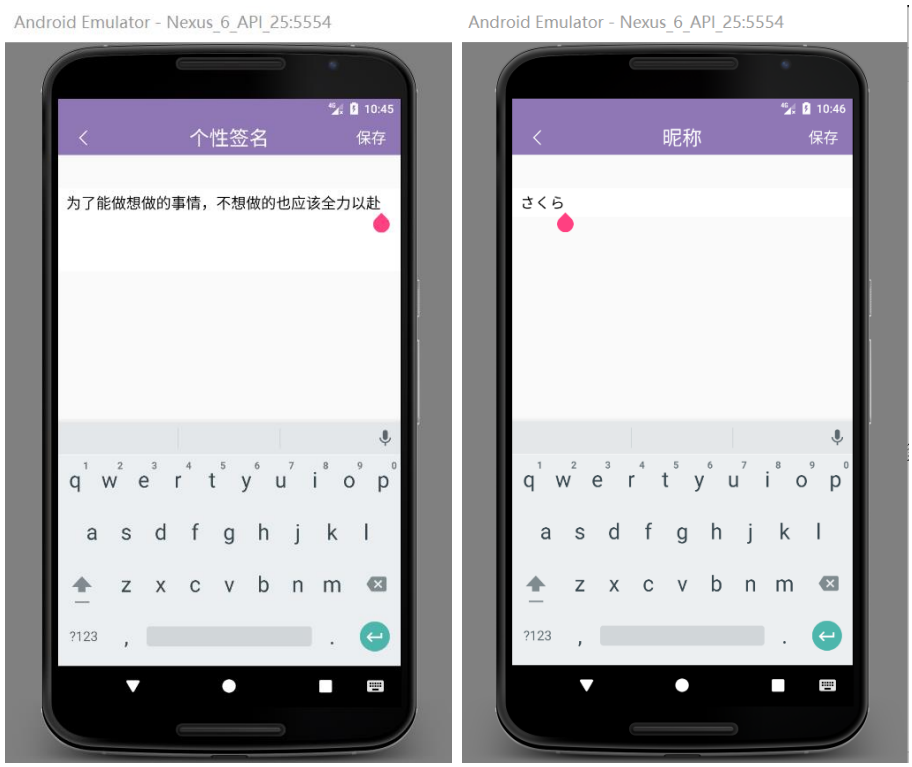
使用 AutoCompleteTextView 的监听事件，监听输入事件，并将输入结果通过下拉框显示出来以供用户选择。

```
atv = (AutoCompleteTextView) findViewById(R.id.activity_search_topbar_searchbar);
arrayAdapter = new ArrayAdapter<String>(SearchActivity.this, android.R.layout.simple_list_item_1, new ArrayList<String>());
atv.setAdapter(arrayAdapter);
atv.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {
    }
    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {
        arrayAdapter.clear();
        arrayAdapter.add(s.toString());
        arrayAdapter.notifyDataSetChanged();
    }
    @Override
    public void afterTextChanged(Editable s) {
    }
});
```

点击某个选项，发送请求到服务器，获取与搜索内容相关的帖子

```
atv.setOnItemClickListener((parent, view, position, id) -> {
        getDatas(arrayAdapter.getItem(position), "", "10000000");
});
```

9. 修改资料界面搭建



在账号资料界面点击昵称或个性签名会进入修改资料界面。在修改界面中点击保存按钮后，会发生如下步骤：

(1) 上传修改资料到服务器数据库

(2) 将修改的账户资料保存到本地

(3) 返回到账号资料界面

代码如下：

```
service.saveInfo(uid, trans.get(title), content)
        .subscribeOn(Schedulers.newThread())
        .observeOn(AndroidSchedulers.mainThread())
        .subscribe(new Subscriber<ResponseBody>() {
            @Override
            public void onCompleted() {
            }
            @Override
            public void onError(Throwable e) {
                Toast.makeText(TextActivity.this, "保存失败", Toast.LENGTH_SHORT).show();
            }
            @Override
            public void onNext(ResponseBody responseBody) {
                Toast.makeText(TextActivity.this, "保存成功", Toast.LENGTH_SHORT).show();
                SharedPreferences sharedPref;
                Context context;
                context = getApplicationContext();
                sharedPref = context.getSharedPreferences(uid, Context.MODE_PRIVATE);
                SharedPreferences.Editor editor = sharedPref.edit();
                editor.putString(trans.get(title), content);
                editor.commit();
            }
        });
```

这是上传修改数据的代码。第一个参数 uid 表示用户 ID。第二个参数是字段名用以区分比如昵称和个性签名。第三个参数是 content，表示修改的内容。

数据在服务器数据库修改成功后会将修改的数据保存在本地的 SharePreferences 文件中将用户资料保存到本地无需通过服务器获得。

（3） 实验遇到困难以及解决思路

1. 上拉刷新/下拉加载更多的功能实现

   使用 github 某大神提供的 SmartRefreshLayout 实现

2. 搜索框的下拉框实现

   使用 AutoCompleteTextView 实现

3. 皮肤切换的实现

   因为不能动态修改 xml 文件的内容，所以只能暴力修改了。为每一个 Activity 配一个 changeTheme 函数，进入 Activity 或通过上层 activity 返回到该层 activity 时都调用一下这个函数。在这个函数中将在这个 activity 中与主题色相关的控件的颜色设置成保存在 SharePreferences 文件中的颜色。在配色方案界面中可以将选中的颜色保存到对应的 SharePreferences 文件中从而完成换肤的工作。

4. 保存用户登录状态

   同样的，在本地使用 SharePreferences 文件保存账户的登录状态与一

些配置信息实现用户的状态保存。

5. 主界面加载慢

   主界面之所以加载慢是因为从服务器加载了很多数据。所以限制一次请求的数据量很重要。这里主要是将每次请求返回的数据数量限制为 20 条。当需要加载更多时，移到界面最下端下拉加载更多，此时又会向服务器申请多 20 条数据。

四、 课后实验结果

五、 实验思考及感想

本次实验我主要负责服务器与数据库的实现与维护，UI 界面的设计，一些简单界面的实现。由于在码这篇报告的时候时间已经所剩无几，所以我就简单讲一下。

1. 我们小组秉承着一定要做有用户间互动的理念，将原来要做的通讯类仿微信应用成功转型为咨询类应用。之所以转型，一来还不知道如何实现数据在应用的实时更新(聊天需要数据实时更新)，二来感觉很少有人做资讯类应用，三来则是我们的应用初衷(见实验内容)。
2. 通过期中的教训，本次我们小组很早就开始期末 project，同时也很注重团队交流。我们小组都有一个共同理念，就是宁可功能还不齐全，也要脸长得好，所以我们花了很多时间在 UI 的设计上。
3. 编辑模式的创新。一般情况下，在移动端编辑图文并茂的文章都是不容易的。我们小组注意到 markdown 的编辑模式对于图文并茂的文章设计非常友好，所以就采纳了 markdown 的设计模式。设计一个资源区，将图片资源导入资源区，然后当需要使用图片时，直接点击相应的图片就可以在光标位置后面添加图片的 ID 了(ID 用以区分图片)。然后我们在发布文章前还可以先预览看看效果再决定是否发布。最后图片资源只有在点击发布时才会上传服务器。

本次实验我的工作量其实蛮大的，所以在写这篇报告的时候，总觉得要写很多东西但又不知道还要写什么东西，再加上时间所剩无几所以也有些语无伦次。所以最后就在这里感谢一下 TAs 和老师在这个学期的辛勤指教吧。老师 TA 们，您们辛苦了！